ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH

TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN

BÁO CÁO PHÂN TÍCH BỘ DỮ LIỆU

GVHD: ThS. Nguyễn Thị Anh Thư

Nhóm thực hiện: Nhóm 8

Trương Hoàng Khiêm	23520730
Nguyễn Nghĩa Trung Kiên	23520801
Trần Anh Kiệt	23520820
Nguyễn Duy	24520384
Huỳnh Lê Minh Thành	22521346
Võ Phan Kiều My	24521095

Thành phố Hồ Chí Minh, 12/2025

MỤC LỤC

TÌM HIỂU DỮ LIỆU **4**
1. Giới thiệu bộ dữ liệu 4
2. Tìm hiểu dữ liệu tài nguyên khóa học (Course resource type) 5
2.1. Course Info (course.json) 5
2.2. Video (video.json) 6
2.3. Problem (problem.json) 7
2.4. School (school.json) 8
2.5. Teacher (teacher.json) 8
2.6. Field/Discipline (course-field.json) 9
3. Tìm hiểu dữ liệu hành vi người học (Student behaviour type) 10
3.1. Student profile (user.json) 10
3.2. Video watching (user-video.json) 10
3.3. Exercising (user-problem.json) 11
3.4. Comment (comment.json) 12
3.5. Reply (reply.json) 13
3.6. Xiaomu (user-xiaomu.json) 13
4. Tìm hiểu dữ liệu khái niệm và các mối quan hệ (Concepts and links) 14
4.1. Concept (concept.json) 14
4.2. Concept-prerequisite (cs/math/psy.json) 14
4.3. Concept (Concept-course, Concept-video, Concept-problem, Concept-comment, Concept-paper, Concept-others) 15
4.4. Paper (paper.json) 16
4.5. Other (other.json) 17
CHUẨN BỊ DỮ LIỆU **19**
1. Khám phá dữ liệu 19
1.1. Course resource (course, problem, teacher) 19
1.1.1. Thống kê mô tả 19
1.1.2. Trực quan hóa dữ liệu 22
1.1.3. Phân tích thống kê 28
1.1.3.1. Phân tích ANOVA với cột “type” và “score” trong bảng problem 28

1.1.3.2. Phân tích t-test với “score” theo “language” (English/Chinese) 29
1.1.3.3. Phân tích hệ số tương quan pearson correlation 30
1.1.4. Khai phá tri thức 30
1.2. Student behaviour (user, user-video, user-problem, comment, reply) 34
1.2.1. Thống kê mô tả 34
1.2.2. Trực quan hóa dữ liệu 43
2. Làm sạch dữ liệu và chuyển đổi dữ liệu 49
2.1. Dữ liệu tĩnh về học viên và khóa học 49
2.2. Dữ liệu về kết quả học tập 50
2.3. Dữ liệu về các tương tác video 54
2.4. Dữ liệu về các tương tác comment 54
3. Gán nhãn dữ liệu 54
3.1. Cấu trúc dữ liệu đầu vào 54
3.2. Chiến lược gán nhãn 57
3.2.1. Tổng quan 57
3.2.2. Thành phần học tập (L - Learning) 57
3.2.3. Thành phần bình luận (S - Sentiment) 59
3.2.4. Tài nguyên khóa học (C - Course resource) 60
3.2.5. Thành phần thời gian (T - Time) 60
3.2.6. Trọng số các yếu tố 61
4. Chia tập dữ liệu 62
4.1. Xây dựng bộ dữ liệu time series 62
4.1.1. Bộ dữ liệu gốc 64
4.1.2. Bộ dữ liệu làm giàu bằng Node2Vec 65
4.1.3. Bộ dữ liệu xử lý bằng SMOTE 65
4.1.4. Bộ dữ liệu kết hợp giữa SMOTE và Node2Vec 66
4.2. Chia dữ liệu train/test/dev 70
PHÂN TÍCH VẤN ĐỀ **72**
1. Bối cảnh vấn đề và nhu cầu kinh doanh 72
2. Câu hỏi nghiên cứu và mục tiêu đề tài 72
3. Kết quả đề tài và khả năng ứng dụng 73

TÌM HIỂU DỮ LIỆU

1. Giới thiệu bộ dữ liệu

Bộ dữ liệu MOOC-CubeX được duy trì bởi Nhóm Knowledge Engineering thuộc Đại học Tsinghua và được hỗ trợ bởi XuetangX, một trong những trang web MOOC lớn nhất ở Trung Quốc. Đây là một kho dữ liệu khổng lồ và đa dạng, được xây dựng để hỗ trợ các nghiên cứu về học tập thích ứng trong môi trường MOOC.

Bộ dữ liệu này bao gồm:

- 4.216 khóa học
- 230.263 video
- 358.265 bài tập
- 637.572 khái niệm chi tiết (fine-grained concepts)
- Hơn 296 triệu bản ghi hành vi thô của 3.330.294 sinh viên

Với quy mô và độ bao phủ cao, MOOC-CubeX cung cấp một nền tảng vững chắc cho các nghiên cứu chuyên sâu. Bộ dữ liệu này nổi bật nhờ các đặc điểm chính sau:

- **Tính bao phủ cao:** MOOC-CubeX tổng hợp nhiều nguồn tài nguyên học thuật đa dạng, từ tài nguyên khóa học đến các bản ghi về hành vi học tập, làm bài tập và thảo luận của sinh viên.
- **Quy mô lớn:** So với các kho dữ liệu giáo dục mở khác, MOOC-CubeX có quy mô lớn hơn đáng kể, phù hợp cho việc khám phá và xây dựng các mô hình học sâu với yêu cầu dữ liệu cao.
- **Tập trung vào khái niệm**: Dữ liệu không đồng nhất được tổ chức theo các khái niệm chi tiết, giúp các tài nguyên trở nên liên quan, dễ dàng được biểu diễn, tìm kiếm và mô hình hóa hơn

Dữ liệu nội bộ (các thông tin thu thập trực tiếp trên nền tảng**)**	**Dữ liệu bên ngoài (**các thông tin thu thập thêm từ bên ngoài**)**
Course info, video, problem, teacher, teacher, student profile, video watching, exercising, comment, reply, xiaomu, concept	School, field/discipline, paper, other

2. Tìm hiểu dữ liệu tài nguyên khóa học (Course resource type)

2.1. Course Info (course.json)

- **Số dòng dữ liệu thực tế:** 3781
- **Kích thước/định dạng:** 44MB/json file

Mỗi khóa học (course) bao gồm nhiều tài nguyên (resources) bao gồm:

ID	Tên cột	Mô tả	Kiểu dữ liệu	Miền giá trị
1	about	Giới thiệu chi tiết về nội dung khóa học.	string	
2	id	Định danh duy nhất của khóa học.	string	
3	field	Danh sách các lĩnh vực mà khóa học thuộc về.	array<string>	
4	name	Tên đầy đủ của khóa học.	string	
5	prerequisites	Mô tả về các kiến thức tiên quyết cần có trước khi bắt đầu khóa học.	string	
6	resource	Danh sách các tài nguyên học tập của khóa học, được sắp	array<struct<string>>	

ID	Tên cột	Mô tả	Kiểu dữ liệu	Miền giá trị
1	about	Giới thiệu chi tiết về nội dung khóa học.	string	
2	id	Định danh duy nhất của khóa học.	string	
		xếp theo thứ tự.		

2.2. Video (video.json)

- **Số dòng dữ liệu thực tế:** 59581
- **Kích thước/định dạng:** 608MB/json file

Video lưu giữ thông tin về khóa học bao gồm:

ID	Tên cột	Mô tả	Kiểu dữ liệu	Miền giá trị
1	ccid	Định danh duy nhất của video.	string	
2	name	Tiêu đề của video. Tên của chương mà video đó thuộc về cũng được ghi lại.	string	
3	start	Thời gian bắt đầu của từng câu trong phụ đề video (đến cấp độ mili giây)	array<double>	
4	end	Thời gian kết thúc của từng câu trong phụ đề video (đến cấp độ mili giây)	aarray<double>	
5	text	Lưu phụ đề của video	array<string>	

2.3. Problem (problem.json)

- **Số dòng dữ liệu thực tế:** 2454422
- **Kích thước/định dạng:** 1.29GB/json file

Các khóa học đã bao gồm những nội dung của mỗi bài tập. Chú ý rằng mỗi nhóm bài tập tương đương với nhiều loại loại câu hỏi (problem), bao gồm:

ID	Tên cột	Mô tả	Kiểu dữ liệu	Miền giá trị
1	problem_id	Định danh duy nhất của câu hỏi (problem)	bigint	
2	exercise_id	Định danh của nhóm bài tập (exercise) chứa câu hỏi này, bắt đầu bằng Ex_. Giúp liên kết ngược lại với resource trong khóa học	string	.
3	language	Ngôn ngữ được sử dụng trong đề bài.	string	Chinese / English.
4	title	Tiêu đề của nhóm bài tập.	string	
5	content	Nội dung chi tiết của đề bài.	string	
6	option	Các lựa chọn cho câu hỏi (nếu là trắc nghiệm).	struct/string	
7	answer	Đáp án đúng của câu hỏi.	string	
8	score	Điểm số của câu hỏi.	double	[0;100]
9	type	ID và tên của dạng câu hỏi.	bigint	[1;9]
10	location	Vị trí của câu hỏi trong	string	

		cấu trúc chương của khóa học.		
11	context_id	Mảng chứa các ID liên quan đến khái niệm mà câu hỏi này kiểm tra.	array<bigint>	

2.4. School (school.json)

- **Số dòng dữ liệu thực tế:** 429
- **Kích thước/định dạng:** 627KB/json file

Lưu các thông tin về trường học của course, bao gồm:

ID	Tên cột	Mô tả	Kiểu dữ liệu	Miền giá trị
1	id	Mã định danh trường, bắt đầu bằng S_.	string	
2	name	Tên trường bằng tiếng Trung.	string	
3	name_en	Tên trường bằng tiếng Anh.	string	
4	sign	Tên viết tắt của trường bằng tiếng Anh.	string	
5	about	văn bản giới thiệu về trường.	string	
6	motto	khẩu hiệu của trường.	string	

2.5. Teacher (teacher.json)

- **Số dòng dữ liệu thực tế:** 17018
- **Kích thước/định dạng:** 9MB/json file

Lưu các thông tin về các giáo viên dạy các khóa học, bao gồm:

ID	Tên cột	Mô tả	Kiểu dữ liệu	Miền giá trị

1	id	Mã định giáo viên, bắt đầu bằng T_.	string	
2	name	Tên giáo viên bằng tiếng Trung.	string	
3	name_en	Tên giáo viên bằng tiếng Anh.	string	
4	job_title	Chức danh của giáo viên.	string	
5	about	Hồ sơ của giáo viên	string	
6	org_name	Tên đơn vị, tổ chức mà giáo viên đang công tác	string	

2.6. Field/Discipline (course-field.json)

- **Số dòng dữ liệu thực tế:** 632
- **Kích thước/định dạng:** 62KB/json file

Lĩnh vực của các khóa học:

ID	Tên cột	Mô tả	Kiểu dữ liệu	Miền giá trị
1	course_id	Mã định danh khóa học	bigint	
2	course_name	Tên khóa học	string	
3	field	Danh sách lĩnh vực của khóa học được gán nhãn thủ công	array<string>	

3. Tìm hiểu dữ liệu hành vi người học (Student behaviour type)

3.1. Student profile (user.json)

- **Số dòng dữ liệu thực tế:** 3330294
- **Kích thước/định dạng:** 806MB/json file

Tệp chứa thông tin về học viên

ID	Tên cột	Mô tả	Kiểu dữ liệu	Miền giá trị
1	id	Mã định danh học viên, bắt đầu bằng U_	string	
2	name	Tên học viên	string	
3	gender	Giới tính học viên	bigint	[0;232]
4	school	Trường của học viên	string	
5	year_of_birth	Năm sinh học viên	bigint	[1111;9989]
6	course_order	Danh sách khóa học đã chọn	array<bigint>	
7	enroll_time	Thời gian chọn khóa học tương ứng với course_order	array<string>	

3.2. Video watching (user-video.json)

- **Số dòng dữ liệu thực tế:** 310360
- **Kích thước/định dạng:** 3.18GB/json file

ID	Tên cột	Mô tả	Kiểu dữ liệu	Miền giá trị
1	user_id	Mã định danh học viên, bắt đầu bằng	string	

		U_		
2	seq	Mảng, chuỗi thời gian người dùng xem video, mỗi đối tượng trong mảng bao gồm thời gian xem video, thời gian bắt đầu và kết thúc của video, tốc độ xem video, v.v.	object	

3.3. Exercising (user-problem.json)

- **Số dòng dữ liệu thực tế:** 133384333
- **Kích thước/định dạng:** 22.45GB/json file

Tệp chứa thông tin làm bài tập của học viên.

ID	Tên cột	Mô tả	Kiểu dữ liệu	Miền giá trị
1	log_id	ID bản ghi câu hỏi của người dùng, kết hợp khóa duy nhất của user_id và problem_id	string	
2	user_id	ID người dùng, bắt đầu với U_	string	
3	problem_id	ID vấn đề, bắt đầu	string	

		với Pm_		
4	is_correct	Câu trả lời có đúng hay không	bigint	[0;1]
5	attempts	Số lần thử trả lời câu hỏi	bigint	[1;458]
6	score	Điểm số	double	[-1;100]
7	submit_time	Thời gian trả lời câu hỏi	string	

3.4. Comment (comment.json)

- **Số dòng dữ liệu thực tế:** 8395141
- **Kích thước/định dạng:** 2.36GB/json file

Tệp chứa bình luận của người dùng về video hoặc khóa học, mỗi tài nguyên có thể có nhiều bình luận, mỗi bình luận là bình luận của người dùng về tài nguyên đó.

ID	Tên cột	Mô tả	Kiểu dữ liệu	Miền giá trị
1	id	ID bình luận, bắt đầu bằng Cm_	string	
2	user_id	ID người dùng đã bình luận	bigint	
3	text	Nội dung bình luận	string	
4	create_time	Thời gian bình luận	string	

3.5. Reply (reply.json)

- **Số dòng dữ liệu thực tế:** 331011
- **Kích thước/định dạng:** 52.3MB/json file

Tệp chứa phản hồi của học viên đối với bình luận của học viên khác.

ID	Tên cột	Mô tả	Kiểu dữ liệu	Miền giá trị
1	id	ID phản hồi, bắt đầu bằng Rp_	string	
2	user_id	ID người dùng đã phản hồi, bắt đầu bằng U_	string	
3	text	Nội dung phản hồi	string	
4	create_time	Thời gian phản hồi	string	

3.6. Xiaomu (user-xiaomu.json)

- **Số dòng dữ liệu thực tế:** 108351
- **Kích thước/định dạng:** 10.14MB/json file

Tệp chứa thông tin về câu hỏi của người dùng, bao gồm ID người dùng, loại câu hỏi và nội dung câu hỏi.

ID	Tên cột	Mô tả	Kiểu dữ liệu	Miền giá trị
1	user_id	Mã định danh học viên, bắt đầu bằng U_	string	

2	question_type	Thể loại câu hỏi	string	
3	question	Nội dung câu hỏi	string	

4. Tìm hiểu dữ liệu khái niệm và các mối quan hệ (Concepts and links)

4.1. Concept (concept.json)

- **Số dòng dữ liệu thực tế:** 637572
- **Kích thước/định dạng:** 162MB/json file

Khái niệm được lấy từ tiêu đề video

ID	Tên cột	Mô tả	Kiểu dữ liệu	Miền giá trị
1	id	Id của concept theo format k_(tên concept)_(ngành)	string	
2	name	Tên của khái niệm đảm bảo giống trong id	string	
3	context	Ngữ cảnh mà khái niệm xuất hiện từ Wiki/Baidu Encyclopedia, Zhihu Q&A (50 từ trước và sau khái niệm). Chuỗi này đảm bảo có tên khái niệm	array<string>	

4.2. Concept-prerequisite (cs/math/psy.json)

Bao gồm bản chú thích và dự đoán các khái niệm trong lĩnh vực máy tính/toán học/tâm lý học.

ID	Tên cột	Mô tả	Kiểu dữ liệu	Miền giá trị

1	c1	Khái niệm tiên quyết	string	
2	c2	Khái niệm mục tiêu	string	
3	ground_truth	Nhãn thực tế (biểu diễn xem thực tế c1 có phải là tiên quyết của c2 không)	bigint	[0;1]
4	text_predict	Dự đoán từ mô hình NLP	double	[0.0; 1.0]
5	graph_predict	Dự đoán từ mô hình đồ thị	double	[0.0; 1.0]

4.3. Concept (Concept-course, Concept-video, Concept-problem, Concept-comment, Concept-paper, Concept-others)

Mỗi loại liên kết được lưu trong một file txt, mỗi dòng trong file thể hiện liên kết giữa một khái niệm và một tài nguyên tương ứng

ID	Liên kết	Mô tả	Format dòng
1	Concept-course	Liên kết giữa khái niệm và khóa học	{concept ID}\t{course ID}
2	Concept-video	Khái niệm liên quan đến video	{concept ID}\t{ccid}
3	Concept-problem	Khái niệm liên quan đến vấn đề/câu hỏi trong các khóa	{concept ID}\t{question ID}
4	Concept-comment	Khái niệm liên quan đến bình luận	{concept ID}\t{review ID}

5	Concept-paper	Khái niệm liên quan đến các luận văn/bài báo	{concept ID}\t{paper ID}
6	Concept-others	Khái niệm liên hệ với những tài nguyên bên ngoài	{concept ID}\t{resource ID}
7	Concept-reply	Liên kết giữa khái niệm và các phản hồi bình luận.	{concept ID}\t{resource ID}

4.4. Paper (paper.json)

- **Số dòng dữ liệu thực tế:** 5410742
- **Kích thước/định dạng:** 6.86GB/json file

Những bài báo liên quan đến khái niệm

ID	Tên cột	Mô tả	Kiểu dữ liệu	Miền giá trị
1	id	ID bài báo	string	
2	concept	ID khái niệm liên quan	string	
3	abstract	Tóm tắt của bài báo	string	
4	authors	Thông tin tác giả, bao gồm ID và tên	string	
5	doi	Định danh DOI	string	
6	lang	Ngôn ngữ (en cho tiếng Anh, zh cho tiếng Trung)	string	
7	pages	Số trang của bài báo	struct<string>	
8	num_citation	Số lần trích dẫn (tính đến năm 2020)	bigint	[0;248303]

9	score	Điểm mức độ liên quan giữa bài báo và khái niệm (điểm cao hơn nghĩa là liên quan hơn)	double	[3.0;91919.43]
10	sourcetype	Loại nguồn (hiện tại là publication)	string	
11	title	Tiêu đề bài báo	string	
12	venue	Hội nghị hoặc tạp chí công bố	object	
13	urls	Danh sách URL của bài báo	array<string>	
14	year	Năm xuất bản	bigint	[0;2021]

4.5. Other (other.json)

- **Số dòng dữ liệu thực tế:** 210349
- **Kích thước/định dạng:** 792MB/json file

Tệp này chứa các tài liệu liên quan được thu thập từ các nguồn bên ngoài như Wikipedia, Baidu Baike và Zhihu Q&A.

ID	Tên cột	Mô tả	Kiểu dữ liệu	Miền giá trị
1	id	ID dữ liệu	string	
2	concept	Khái niệm liên quan đến thông tin này	string	

3	type	Nguồn dữ liệu (zhihu, baike, wiki)	string	
4	context	Nội dung dữ liệu	string	

CHUẨN BỊ DỮ LIỆU

1. Khám phá dữ liệu

1.1. Course resource (course, problem, teacher)

1.1.1. Thống kê mô tả

a. Course

Bảng **course** gồm 3781 khóa học unique với 6 cột bao gồm: id, name, field, prerequisites, about, và resource. Trong có các cột trên thì chỉ có cột about và prerequisites chứa 2 giá trị null.

Có 3240 tên khóa học khác nhau và giá trị xuất hiện nhiều nhất với 9 lần. Cột field có 147 lĩnh vực khóa học khác nhau phổ biến nhất là "[]" (khóa học không có thông tin về lĩnh vực) xuất hiện 3234 lần. Cột prerequisites có 793 giá trị unique cho khóa học khác nhau và phổ biến nhất xuất hiện 68 lần.

```
course.describe(include="object")
```

	id	name	field	prerequisites	about	resource
count	3781	3781	3781	3779	3779	3781
unique	3781	3240	147	793	3227	3781
top	C_584313	线性代数	[]		高校教师教学&科研能力提升全周期培养计划	[{'titles': ['第一课 导论与三家分晋', '导论', '导论'], 'reso...
freq	1	9	3234	2578	68	1

b. Problem

Bảng **problem** chứa 2454422 dòng, hầu hết các trường quan trọng như answer, content, exercise_id, language, và typetext đều không có giá trị null nào. Có 149564 giá trị null ở cột option, có vẻ là hợp lý, vì chỉ các câu hỏi trắc nghiệm mới có dữ liệu ở trường này. Các dạng câu hỏi khác như tự luận hay đúng/sai sẽ để trống.

Cột score có 574372 giá trị null. (khoảng 23%) các câu hỏi trong hệ thống không được gán điểm số cụ thể.

```
Đang khởi tạo Spark Session...
Đang đọc file: /content/mooccubex_data/entities/problem.json...
Đọc file thành công.
Đang tính toán số lượng giá trị NULL...

--- Kết quả số lượng giá trị NULL cho mỗi trường ---
+------+-------+----------+-----------+--------+--------+------+----------+------+-----+----+--------+
|answer|content|context_id|exercise_id|language|location|option|problem_id| score|title|type|typetext|
+------+-------+----------+-----------+--------+--------+------+----------+------+-----+----+--------+
|     0|      0|         0|          0|       0|       0|149564|         0|574372|    0|   0|       0|
+------+-------+----------+-----------+--------+--------+------+----------+------+-----+----+--------+
```

Tứ phân vị của cột score trong problem đều là 1, cho thấy đa số dữ liệu score mang giá trị là 1

```
problem = spark.read.json(basePath + 'problem.json')

level = [0.25, 0.5, 0.75]
error = 0.01

quartiles = problem.approxQuantile("score", [0.25, 0.5, 0.75], 0.0)
```

```
quartiles
```

```
[1.0, 1.0, 1.0]
```

```
+-----+-------+
|score|  count|
+-----+-------+
|  1.0|1453055|
| NULL| 574372|
|  2.0| 307644|
| 10.0|  37579|
|  5.0|  27567|
|  3.0|  14992|
|  4.0|  13517|
|  0.5|  10034|
|  6.0|   2705|
|  1.5|   2344|
|  8.0|   2106|
| 20.0|   1800|
|  0.0|   1796|
|  7.0|    839|
| 15.0|    681|
|  2.5|    539|
|  9.0|    371|
| 12.5|    310|
| 12.0|    292|
| 30.0|    231|
+-----+-------+
only showing top 20 rows
```

Các câu hỏi được chia theo ngôn ngữ tiếng Trung (2210105) và tiếng Anh (244317)

```
# --- PHÂN TÍCH THEO NGÔN NGỮ ---
print("--- Thống kê số lượng theo ngôn ngữ ---")
df_lang_counts = df_problems.groupBy("language").count() \
                            .withColumnRenamed("count", "problem_count")

df_lang_counts.show(truncate=False)
```

```
--- Thống kê số lượng theo ngôn ngữ ---
+--------+-------------+
|language|problem_count|
+--------+-------------+
|Chinese |2210105      |
|English |244317       |
+--------+-------------+
```

- **Các câu hỏi phổ biến nhất:** Đều thuộc các khóa học đại cương có quy mô lớn. Tỷ lệ trả lời đúng rất cao cho thấy đây là những câu hỏi kiểm tra kiến thức cơ bản mà phần lớn sinh viên đều nắm vững.
- **Các câu hỏi "khó" nhất:** Có tỷ lệ đúng là **0.0** một cách đáng ngờ. Vì phần lớn trong số này là **câu hỏi chủ quan**, khả năng cao là chúng chưa được chấm điểm hoặc hệ thống ghi nhận điểm mặc định là 0, chứ không phản ánh độ khó thực sự về mặt kiến thức.

```
--- Top 10 problem được giải nhiều nhất ---
+-------+---------+--------+-------------+------------+
|id     |title    |typetext|attempt_count|correct_rate|
+-------+---------+--------+-------------+------------+
|3296033|第35讲作业|多选题  |151080       |0.93        |
|3296034|第36讲作业|多选题  |150954       |0.72        |
|3296031|第33讲作业|判断题  |150910       |0.99        |
|3296029|第31讲作业|判断题  |150897       |0.89        |
|3296032|第34讲作业|判断题  |150881       |0.98        |
|3295999|第27讲作业|多选题  |150835       |0.88        |
|3296030|第32讲作业|判断题  |150810       |0.99        |
|3296002|第30讲作业|判断题  |150804       |0.96        |
|3296001|第29讲作业|多选题  |150757       |0.87        |
|3295997|第25讲作业|判断题  |150735       |0.89        |
+-------+---------+--------+-------------+------------+
only showing top 10 rows


--- Top 10 problem khó nhất (tỷ lệ đúng thấp nhất, với ít nhất 1000 lượt giải) ---
+-------+-----------------------------------------------------------------+--------+-------------+------------+
|id     |title                                                            |typetext|attempt_count|correct_rate|
+-------+-----------------------------------------------------------------+--------+-------------+------------+
|1484499|Exercises                                                        |主观题  |1560         |0.0         |
|1610376|第13讲 Unwritten etiquette in office 测试题                      |主观题  |1160         |0.0         |
|1610484|第14讲 科技英语语言特征 测试题                                   |主观题  |1590         |0.0         |
|1647391|第五章 商品与商品经济--课程习题5.3                               |多选题  |4822         |0.0         |
|1647360|第三章 人类社会及其发展规律--思考题                              |主观题  |4116         |0.0         |
|1610423|第22讲  Five Ws in a job application letter 测试题               |主观题  |1447         |0.0         |
|5943624|第21讲  CV and Résumé 测试题                                     |主观题  |2070         |0.0         |
|1484535|Exercises                                                        |主观题  |1646         |0.0         |
|1647378|第四章 资本主义生产关系的产生和资本主义生产方式的形成--阶段测验1|多选题  |4557         |0.0         |
|1647366|第四章 资本主义生产关系的产生和资本主义生产方式的形成--思考题   |主观题  |4117         |0.0         |
+-------+-----------------------------------------------------------------+--------+-------------+------------+
only showing top 10 rows
```

c. Teacher

Bảng **teacher** chứa 17018 giá trị, cột name_en (tên tiếng Latin của giáo viên) bị thiếu khá nhiều, nhưng không ảnh hưởng đến giá trị của bộ dữ liệu. Về cột name, có 13967 giá trị unique, cho thấy có vẻ 1 giáo viên có thể thuộc nhiều tổ chức ở những thời gian khác nhau.

Ở phần job_title, giá trị 副教授 (phó giáo sư) xuất hiện giá trị nhiều nhất, có thể dùng các giá trị job_title để tính mức độ điểm cho giáo viên.

- Các chức danh phổ biến nhất là "副教授" (Phó Giáo sư), "讲师" (Giảng viên), và "教授" (Giáo sư). Điều này khẳng định nền tảng có sự tham gia giảng dạy chủ yếu từ các học giả và chuyên gia có trình độ cao từ các cơ sở giáo dục chính quy.
- Chức danh "助教" (Trợ giảng) cũng xuất hiện với số lượng đáng kể, cho thấy có một đội ngũ hỗ trợ tham gia vào việc xây dựng và vận hành khóa học.
- **Dữ liệu nhiễu:** Có những giá trị không rõ ràng như một khoảng trắng hoặc ký tự "x", cần được làm sạch trong các bước xử lý dữ liệu tiếp theo.
- Ngoài các chức danh học thuật truyền thống, còn có các vai trò khác như "高级工程师" (Kỹ sư cao cấp) và "研究员" (Nghiên cứu viên), cho thấy sự đa dạng trong chuyên môn của đội ngũ giảng dạy, không chỉ giới hạn trong môi trường học thuật.

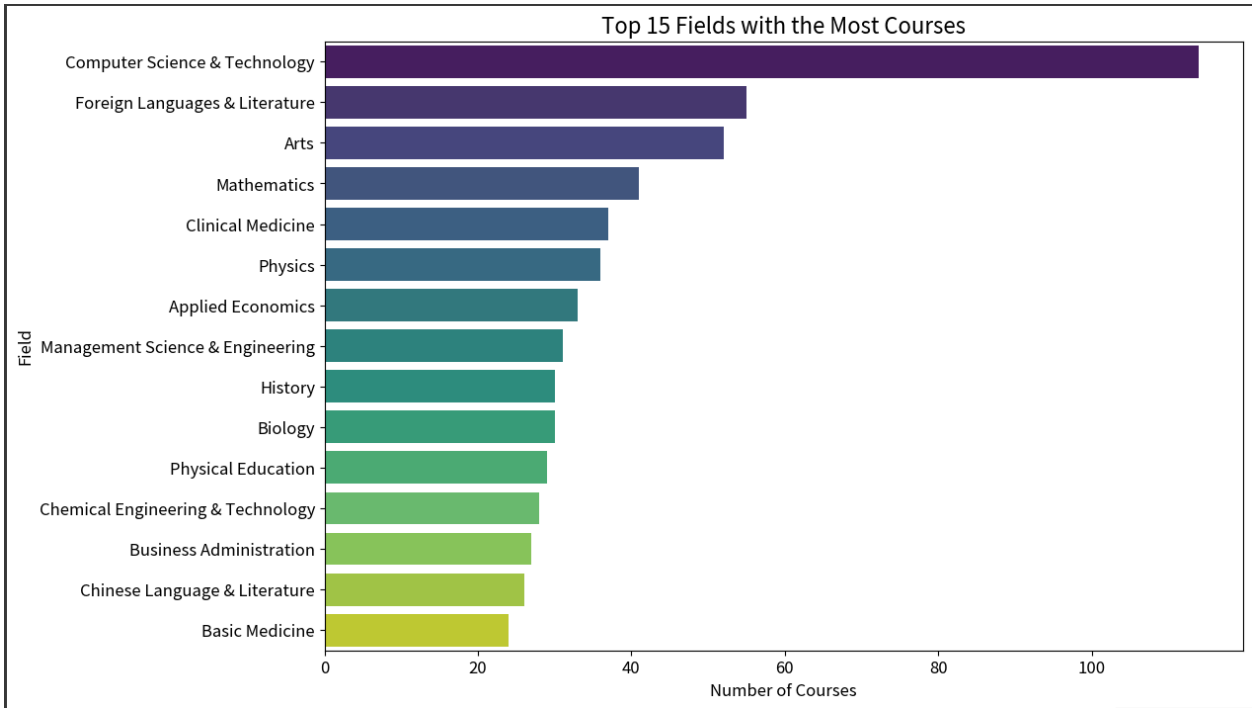
```
teacher.describe()
```

	id	name	name_en	about	job_title	org_name
count	17018	17018	9525	17018	17018	17018
unique	17018	13967	5062	12606	1406	1005
top	T_1	顾礼平			副教授	清华大学
freq	1	20	4142	3125	4288	1140

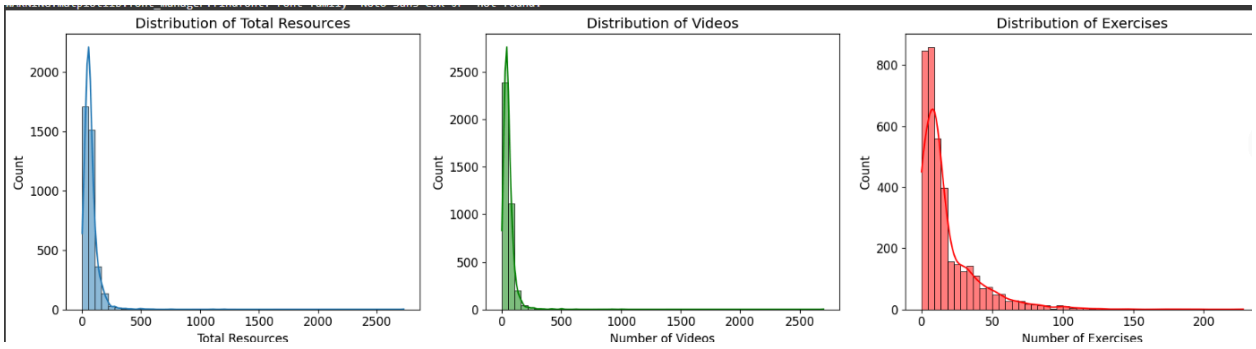
1.1.2. Trực quan hóa dữ liệu

a. Course

Từ bảng **course,** cột **field** cho biết lĩnh vực chuyên môn của từng khóa học. Phân tích số lượng khóa học theo từng lĩnh vực cho thấy sự phân bổ và các lĩnh vực thế mạnh của nền tảng. Dữ liệu thực tế từ biểu đồ cho thấy Computer Science & Technology là lĩnh vực chiếm ưu thế tuyệt đối với hơn 110 khóa học, nhiều gần gấp đôi so với lĩnh vực đứng thứ hai.

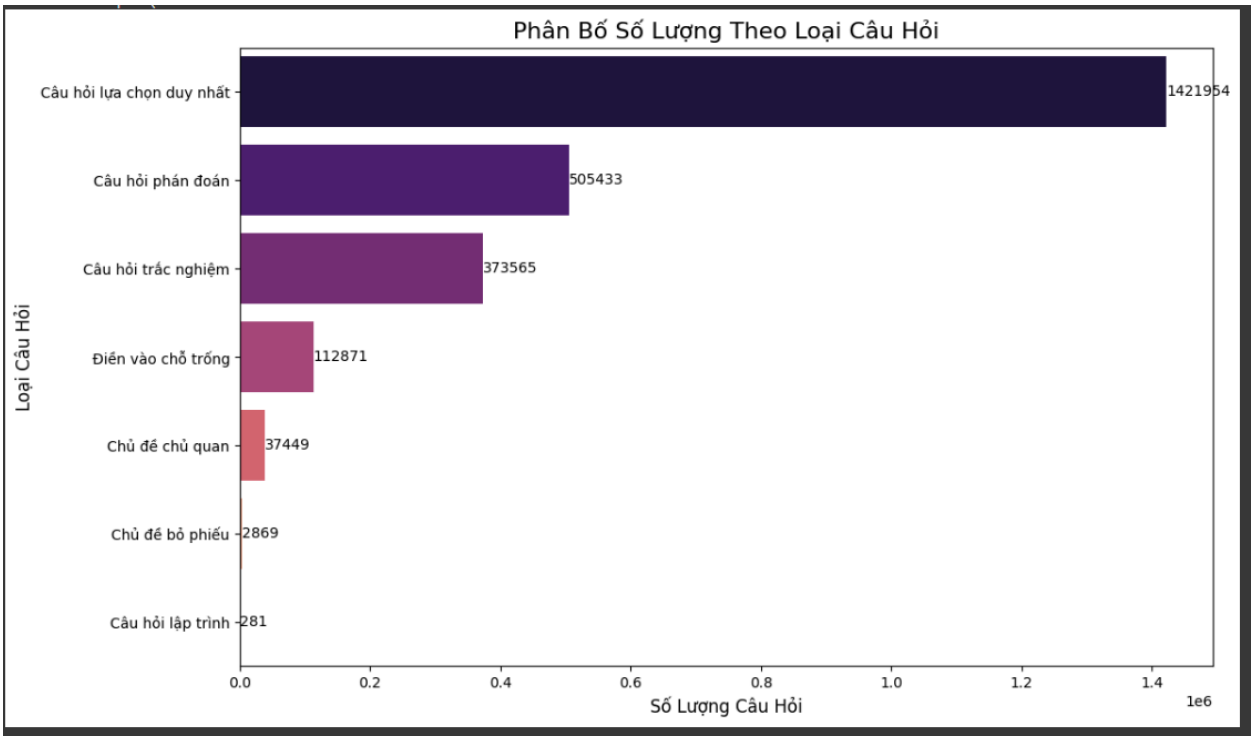


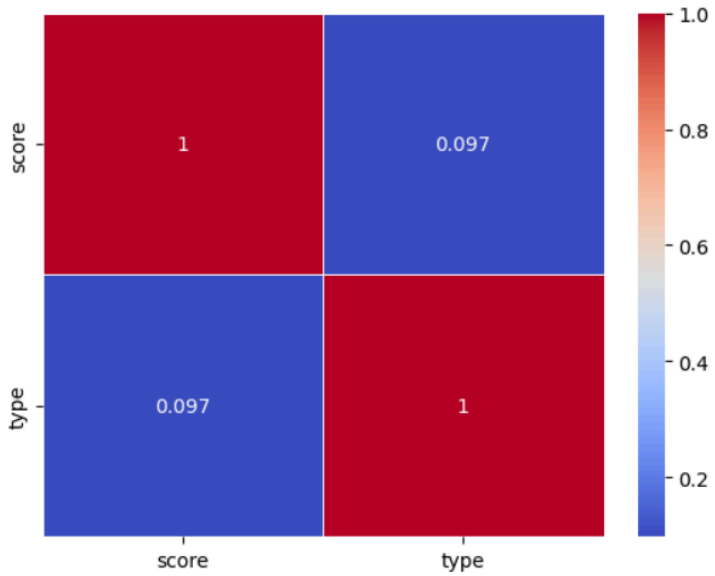
- **Số lượng tài nguyên đa dạng:** Các khóa học có số lượng tài nguyên rất khác nhau, từ tối thiểu là **1** cho đến tối đa là **2728**.
- **Video chiếm ưu thế:** Trung bình, mỗi khóa học có khoảng **53 video** nhưng chỉ có khoảng **18 bài tập**. Điều này cho thấy video là tài nguyên học tập chính trong các khóa học này.
- **Phân bổ không đều:** Độ lệch chuẩn (std) khá lớn so với giá trị trung bình (mean), cho thấy có sự chênh lệch lớn về quy mô giữa các khóa học. Có những khóa học rất lớn với hàng trăm tài nguyên, trong khi phần lớn các khóa học có quy mô vừa phải (75% các khóa học có ít hơn 87 tài nguyên).



b. Problem

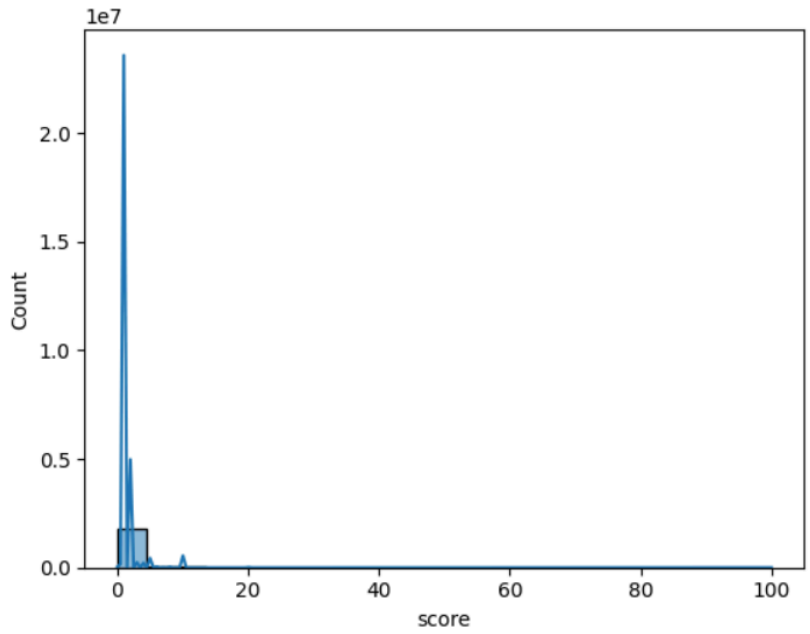
Bảng **problem** chứa trắc nghiệm là chủ đạo, các dạng câu hỏi như "Lựa chọn duy nhất" và "Phán đoán" chiếm tỷ lệ cao. Các dạng bài tập đòi hỏi kỹ năng phân tích sâu hơn như "Chủ đề" hay "Câu hỏi lập trình" thì rất hiếm.



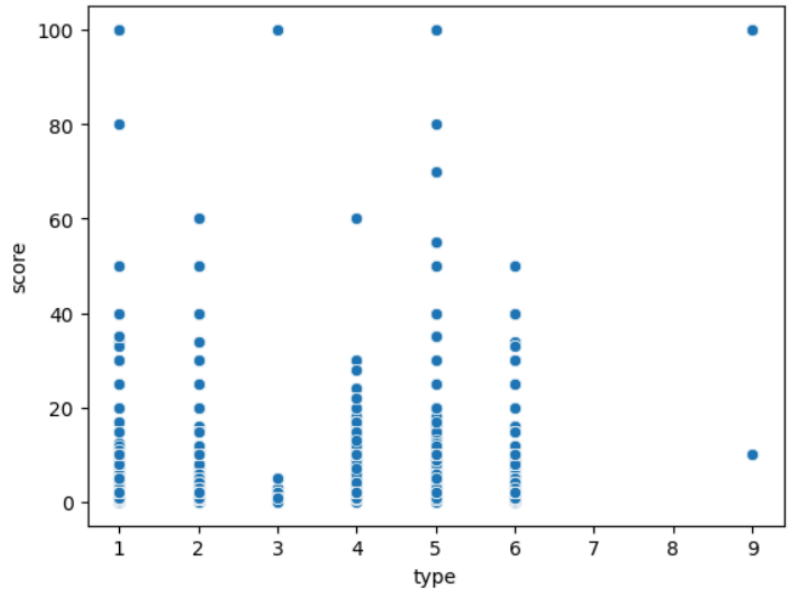


Mối quan hệ giữa score và type rất yếu gần như không có tương quan tuyến tính. Hai biến này gần như độc lập với nhau theo khía cạnh tuyến tính.

<Axes: xlabel='score', ylabel='Count'>



<Axes: xlabel='type', ylabel='score'>



Qua phân phối giữa score và type có thể thấy loại câu hỏi không ảnh hưởng quá nhiều đến score. Score dường như được chuẩn hóa về 1 số giá trị nhất định ở các khóa học và chủ yếu là 1 (làm đúng).

c. Teacher

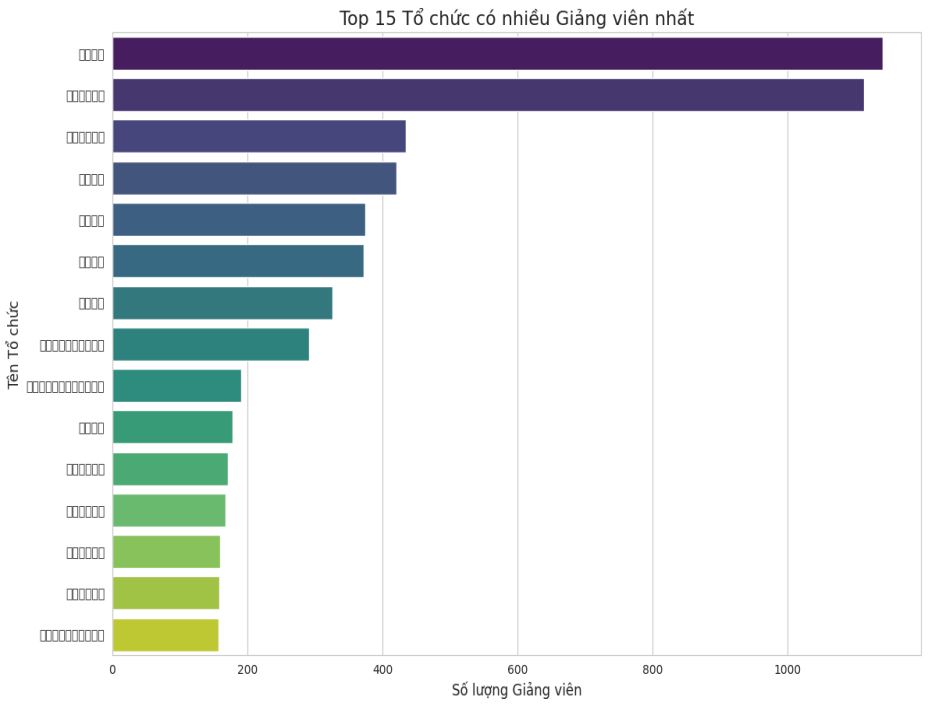
Phân tích tên tổ chức (org_name) cho biết những đơn vị nào đóng góp nhiều giảng viên nhất cho nền tảng, phản ánh các mối quan hệ đối tác và nguồn gốc của các khóa học.

- **Các trường đại học hàng đầu:** "清华大学" (Đại học Thanh Hoa) và "西安交通大学" (Đại học Giao thông Tây An) là hai tổ chức có số lượng giảng viên đông đảo nhất, vượt trội so với các đơn vị khác. Điều này hoàn toàn phù hợp với thông tin rằng bộ dữ liệu MOOCCubeX được duy trì bởi Đại học Thanh Hoa, khẳng định vai trò trung tâm của trường và các đối tác lớn trong việc xây dựng nội dung.
- **Sự xuất hiện của "学堂在线"** (XuetangX) trong top đầu cho thấy nền tảng này cũng có một đội ngũ giảng viên riêng hoặc đóng vai trò là một đơn vị hợp tác chính.

Bảng dữ liệu: Top 15 Tổ chức có nhiều Giảng viên nhất

	org_name	count
0	清华大学	1140
1	西安交通大学	1113
2	安徽新华学院	435
3	西京学院	421
4	南开大学	374
5	重庆大学	372
6	学堂在线	326
7	深圳信息职业技术学院	291
8	重庆三峡医药高等专科学校	191
9	山东大学	178
10	北京师范大学	171
11	南通职业大学	168
12	桂林理工大学	160
13	华南理工大学	159
14	陕西工业职业技术学院	157

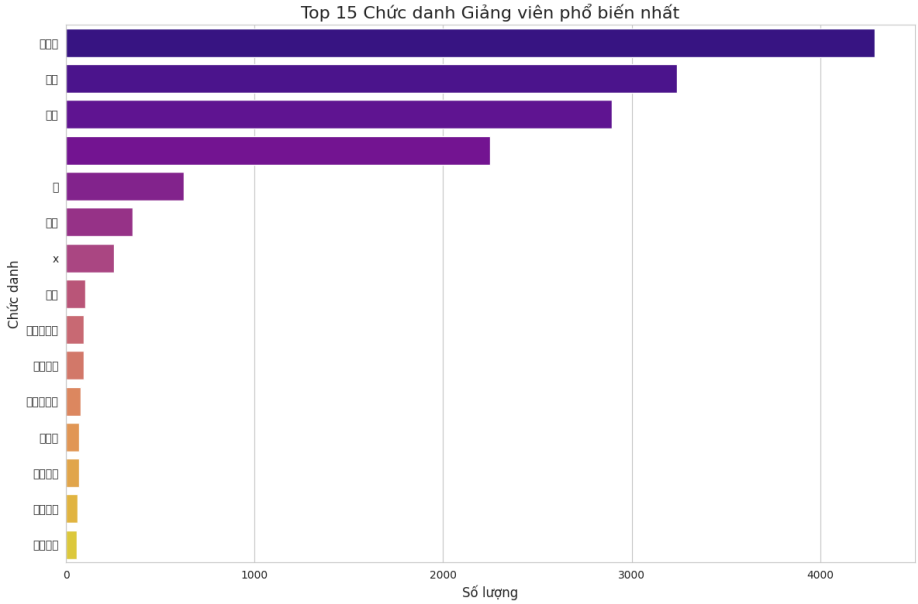
/tmp/ipython-input-2385648930.py:80: FutureWarning:



Bảng dữ liệu: Top 15 Chức danh Giảng viên phổ biến nhất

	job_title	count
0	副教授	4288
1	讲师	3239
2	教授	2895
3		2250
4	略	625
5	助教	351
6	x	256
7	教师	101
8	高级工程师	96
9	副研究员	96
10	副主任医师	76
11	研究员	71
12	主任医师	70
13	助理教授	63
14	主治医师	57

/tmp/ipython-input-2385648930.py:100: FutureWarning:

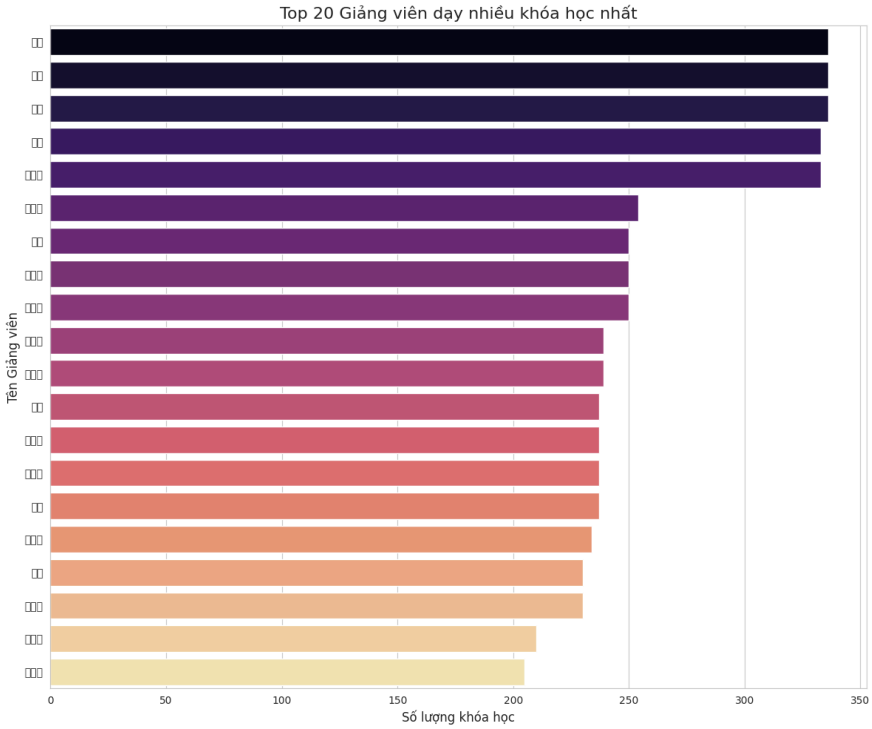


***Phân tích giảng viên có nhiều khóa học nhất:**

- **Những giảng viên nòng cốt:** Một nhóm nhỏ các giảng viên như "李焰", "王焰", "刘丹" chịu trách nhiệm cho một số lượng khóa học rất lớn (hơn 330 khóa học mỗi người). Đây có thể là những giảng viên chủ chốt, trưởng bộ môn hoặc những người có vai trò điều phối nội dung quan trọng trên nền tảng.
- **Sự phân bố đồng đều ở top đầu:** Nhiều giảng viên trong top 20 có số lượng khóa học tương đương nhau, cho thấy có thể có sự phân công hoặc hợp tác theo nhóm trong việc xây dựng các chuỗi khóa học.
- **Tiềm năng khai thác:** Dữ liệu về các giảng viên tích cực nhất này có thể được sử dụng để đề xuất các khóa học liên quan cho người học hoặc để làm nổi bật các chuyên gia hàng đầu trong từng lĩnh vực.

Bảng dữ liệu: Top 20 Giảng viên dạy nhiều khóa học nhất

	name	course_count
0	李焰	336
1	王旭	336
2	刘丹	336
3	闾博	333
4	沈雨瞳	333
5	燕连福	254
6	顾帆	250
7	崔海浪	250
8	聂元昆	250
9	李正雄	239
10	朱南丽	239
11	杨华	237
12	伊景冰	237
13	何志敏	237
14	李勤	237
15	胡鞍钢	234
16	韩锐	230
17	李家胜	230
18	李正风	210
19	刘洪玉	205



1.1.3. Phân tích thống kê

1.1.3.1. Phân tích ANOVA với cột “type” và “score” trong bảng problem

Phân tích phương sai (ANOVA – Analysis of Variance) được sử dụng để kiểm định xem có sự khác biệt có ý nghĩa thống kê giữa các nhóm hay không. Trong trường hợp này:

- **Biến phụ thuộc:** score
- **Biến độc lập:** type
- **Mục tiêu:** kiểm tra xem điểm số trung bình của các loại bài tập khác nhau có khác biệt đáng kể hay không

```
from statsmodels.formula.api import ols
from statsmodels.stats.anova import anova_lm

formula_anova = 'score ~ C(type)'

model_anova = ols(formula_anova, data=problem).fit()
anova_results = anova_lm(model_anova)


print(anova_results)
```

```
                df        sum_sq        mean_sq             F  PR(>F)
C(type)        6.0  2.978584e+06  496430.708387  203175.55588     0.0
Residual  1880043.0  4.593619e+06       2.443358           NaN     NaN
```

Nhận xét: Kết quả cho thấy điểm số trung bình của các nhóm loại bài tập khác nhau có sự khác biệt rất rõ rệt.

- F-value cực lớn ($\approx 2\times10^5$)
- p-value gần bằng 0

=> Các chỉ số trên cho thấy type có ảnh hưởng đáng kể đến score của người học.

1.1.3.2. Phân tích t-test với “score” theo “language”

Kiểm định t-test được dùng để xem score của hai nhóm ngôn ngữ có khác biệt có ý nghĩa thống kê hay không.

- **Nhóm 1:** score_EN
- **Nhóm 2**: score_CN
- **Mục tiêu:** kiểm tra xem ngôn ngữ bài tập có ảnh hưởng đến điểm số hay không.

```
from scipy import stats

score_EN = problem[problem['language'] == 'English']['score'].dropna()
score_CN = problem[problem['language'] != 'English']['score'].dropna()


t_stat, p_value = stats.ttest_ind(score_EN, score_CN, equal_var=False)

print(f"T-Statistic: {t_stat:.4f}")
print(f"P-Value: {p_value:.4f}")
```

```
T-Statistic: -55.0193
P-Value: 0.0000
```

Nhận xét:

- Giá trị t âm (-55.02) nghĩa là nhóm EN có điểm trung bình thấp hơn nhómCN
- Mức chênh lệch rất lớn (vì |t| ≈ 55) cho thấy sự khác biệt không chỉ có ý nghĩa thống kê mà còn rất mạnh về mặt thực tế.
- Điểm số trung bình của các bài tập bằng tiếng Anh thấp hơn đáng kể so với các bài tập không phải tiếng Anh.

1.1.3.3. Phân tích hệ số tương quan pearson correlation

```
correlation_value = problem['score'].corr(problem['type'], method='pearson')

print(f"{correlation_value:.4f}")
correlation_matrix = problem[['score', 'type']].corr(method='pearson')
```

```
0.0974
```

Nhận xét: phân tích tương quan Pearson giữa biến score và type cho thấy hệ số r khoảng 0.09, cho thấy mối quan hệ tuyến tính gần như không tồn tại giữa hai biến. Mối liên hệ giữa score và type có thể không phải tuyến tính, mà phụ thuộc vào các yếu tố khác..

1.1.4. Khai phá tri thức

Phần này nhằm khai phá các mối quan hệ tiềm ẩn giữa các khóa học mà học viên đã đăng ký, để phát hiện ra những tổ hợp khóa học thường đi cùng nhau. Từ đó, ta có thể:

- Đề xuất gợi ý khóa học tiếp theo cho học viên.
- Phân tích xu hướng học tập (ví dụ: nhóm học viên hay học chung nhóm khóa học về một lĩnh vực nhất định).

Trong phần này nhóm sử dụng bộ dữ liệu user.json, với các course_order là các transactions để làm đầu vào cho bài toán khai phá tri thức.

***Mô tả phương pháp:** Áp dụng thuật toán Apriori để tìm ra các tập mục thường xuyên (frequent itemsets) và sinh ra các luật kết hợp (association rules). Apriori hoạt động dựa trên nguyên lý: "Nếu một tập mục phổ biến, thì mọi tập con của nó cũng phổ biến.

- Các tham số:
    - min_support: ngưỡng tần suất xuất hiện tối thiểu (ví dụ 0.2 tương ứng 20% giao dịch).
    - min_confidence: độ tin cậy tối thiểu của luật.
    - min_lift: độ nâng, thể hiện mức độ độc lập giữa các mục.

***Quy trình thực hiện:**

- Chuyển course_order thành dạng danh sách (list).
- Tạo ma trận nhị phân (0/1) – mỗi cột là một khóa học.
- Do bộ dữ liệu khá lớn, nhiều học viên và nhiều khóa học dẫn đến không đủ bộ nhớ khi nên nhóm đã giới hạn dữ liệu lại để có thể thực hiện khai phá tri thức (demo)

```
transactions = user["course_order"].tolist()

te = TransactionEncoder()
te_ary = te.fit(transactions).transform(transactions)
trans_df = pd.DataFrame(te_ary, columns=te.columns_)

trans_df = trans_df.loc[:, trans_df.sum(axis=0) / len(trans_df) >= 0.02]
```

+ Code + Markdown

```
trans_df
```

	629559	677049	679390	696994	697791	697821	707081	707373	799931	840084	883345	890220	916828	936971
0	False	False	False	False	False	False	False	False	False	False	False	False	False	False
1	False	False	False	True	False	False	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False	False	False	False	False	False	False
4	False	False	False	False	True	False	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
999995	False	False	False	False	False	False	False	False	False	False	False	False	False	False
999996	False	False	False	False	False	False	False	False	False	False	False	False	False	False
999997	False	False	False	False	False	False	False	False	False	False	False	False	False	False
999998	False	False	False	False	False	False	False	False	False	False	False	False	False	False

- Sử dụng apriori từ thư viện mlxtend để tiến hành tìm luật kết hợp với tham số min_support là 0.005 (do sự đa dạng của bộ dữ liệu nên để min_support cao sẽ không có kết quả)

```
frequent_itemsets = apriori(trans_df, min_support=0.005, use_colnames=True)
frequent_itemsets["length"] = frequent_itemsets["itemsets"].apply(len)
print("\nFrequent itemsets:")
print(frequent_itemsets.sort_values("support", ascending=False))
```

```
Frequent itemsets:
     support          itemsets  length
13  0.060455          (936971)       1
10  0.035261          (883345)       1
3   0.034807          (696994)       1
4   0.033698          (697791)       1
2   0.028041          (679390)       1
0   0.027384          (629559)       1
12  0.026636          (916828)       1
7   0.024229          (707373)       1
9   0.023941          (840084)       1
1   0.023799          (677049)       1
5   0.022455          (697821)       1
11  0.020988          (890220)       1
6   0.020663          (707081)       1
8   0.020600          (799931)       1
27  0.008819  (916828, 679390)       2
28  0.008315  (883345, 697791)       2
22  0.007615  (707373, 679390)       2
25  0.007393  (883345, 679390)       2
23  0.007360  (799931, 679390)       2
26  0.007244  (890220, 679390)       2
46  0.007196  (883345, 916828)       2
38  0.007146  (916828, 707373)       2
14  0.007042  (679390, 629559)       2
42  0.006909  (799931, 916828)       2
47  0.006897  (890220, 916828)       2
30  0.006831  (883345, 697821)       2
19  0.006799  (916828, 629559)       2
```

Nhận xét:

- Với tập phổ biến đơn (length = 1), khóa học có mã 936971 có support cao nhất (khoảng 6%), là khóa học phổ biến nhất, kế đó là mã 883345 với khoảng 3.5%, cho thấy không có khóa nào quá áp đảo, cho thấy sự đa dạng lựa chọn khóa học của học viên.
- Với các cặp khóa học đi cùng nhau (length > 1), (916828, 679390) là 1 cặp khóa học có khoảng gần 1% học viên sẽ học cả 2 khóa học này, có thể là do môn học tiên quyết.

Sau khi xác định các tập phổ biến, nhóm tiến hành khai phá các luật kết hợp giữa các khóa học bằng hàm association_rules() trong thư viện mlxtend, với ngưỡng:

- Confidence tối thiểu: 0.3

- Lift > 1 để lọc các luật có mối quan hệ thực sự ý nghĩa (không ngẫu nhiên)

```
from mlxtend.frequent_patterns import association_rules

rules = association_rules(frequent_itemsets, metric='confidence', min_threshold=0.3)
rules[rules['lift'] > 1].sort_values(['lift','confidence'], ascending=False)
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	representativity	leverage	conviction	zhangs_metric	jaccard
2	(799931)	(679390)	0.020600	0.028041	0.007360	0.357282	12.741398	1.0	0.006782	1.512262	0.940898	0.178290
7	(799931)	(916828)	0.020600	0.026636	0.006909	0.335388	12.591543	1.0	0.006360	1.464561	0.939944	0.171324
8	(890220)	(916828)	0.020988	0.026636	0.006897	0.328616	12.337301	1.0	0.006338	1.449788	0.938645	0.169347
3	(890220)	(679390)	0.020988	0.028041	0.007244	0.345150	12.308748	1.0	0.006655	1.484246	0.938453	0.173364
4	(916828)	(679390)	0.026636	0.028041	0.008819	0.331093	11.807470	1.0	0.008072	1.453056	0.940355	0.192311
5	(679390)	(916828)	0.028041	0.026636	0.008819	0.314504	11.807470	1.0	0.008072	1.419941	0.941714	0.192311
1	(707373)	(679390)	0.024229	0.028041	0.007615	0.314293	11.208330	1.0	0.006936	1.417455	0.933396	0.170530
0	(707081)	(679390)	0.020663	0.028041	0.006237	0.301844	10.764376	1.0	0.005658	1.392180	0.926240	0.146867
6	(697821)	(883345)	0.022455	0.035261	0.006831	0.304208	8.627334	1.0	0.006039	1.386534	0.904398	0.134244

Nhận xét:

- Các luật có lift cao nhất như (799931) → (679390), (799931) → (916828) và (890220) → (916828) cho thấy rằng học viên đăng ký khóa 799931 hoặc 890220 có khả năng rất cao sẽ học thêm các khóa 679390 hoặc 916828. Lift > 12 chứng tỏ mối quan hệ khá mạnh (xác suất đồng thời không phải ngẫu nhiên mà có tính gắn kết nội dung)
- **Khóa học trọng tâm:** 679390 và 916828 liên tục xuất hiện ở vế phải (consequent) của nhiều luật khác nhau, có thể là các khóa học cốt lõi hoặc bắt buộc, thường được đăng ký sau khi học viên hoàn thành các khóa khác
- Những luật có lift từ 8–10 (như (697821 → 883345) tuy thấp hơn nhưng vẫn mang ý nghĩa gợi ý rõ ràng (các khóa này có thể thuộc cùng chủ đề hoặc cấp độ học tương tự)

***Ứng dụng luật kết hợp vào hệ gợi ý khóa học:** Dựa trên các luật kết hợp có độ tin cậy cao (confidence > 0.3, lift > 1), nhóm xây dựng mô hình gợi ý khóa học đơn giản: nếu học viên đã học khóa A (antecedent), hệ thống gợi ý thêm khóa B (consequent).

```python
def recommend_courses(user_courses, rules_df, top_n=3):
    recs = []
    for course in user_courses:
        subset = rules_df[rules_df["antecedents"] == course]
        subset = subset.sort_values(by=["lift", "confidence"], ascending=False)
        recs.extend(subset["consequents"].tolist())

    recs = [r for r in recs if r not in user_courses]
    return list(dict.fromkeys(recs))[:top_n]
```

```python
user_courses = [799931, 890220]

recommendations = recommend_courses(user_courses, rules_filtered, top_n=5)
print(recommendations)
```

```
[679390, 916828, 707373, 840084, 629559]
```

Nhận xét: [679390, 916828, 707373, 840084, 629559] các khóa học này xuất hiện thường xuyên cùng với những khóa mà học viên đã đăng ký trước đó, thể hiện mối quan hệ học tập rõ ràng. Các khóa 679390 và 916828 nổi bật nhất khi xuất hiện ở nhiều luật có lift trên 10, chứng tỏ chúng là những khóa học trọng tâm, thường được chọn sau khi học viên hoàn thành các khóa cơ sở khác. Các khóa 707373, 840084 và 629559 có mức hỗ trợ thấp hơn nhưng vẫn thể hiện xu hướng học cùng nhóm chủ đề với hai khóa chính trên.

1.2. Student behaviour (user, user-video, user-problem, comment, reply)

1.2.1. Thống kê mô tả

a. user-problem

File dữ liệu user-problem.json chứa 7 cột: attempts, is_correct, log_id, score, submit_time, user_id và có 133384333 dòng

```
user_problem = spark.read.json(basePath + 'user-problem.json')
user_problem.show(10)
```

```
+--------+----------+----------------+----------+-----+-------------------+----------+
|attempts|is_correct|          log_id|problem_id|score|        submit_time|   user_id|
+--------+----------+----------------+----------+-----+-------------------+----------+
|       1|         0|   10000_6906522|Pm_6906522| NULL|2020-10-27 10:11:56|   U_10000|
|       1|         0|   10000_6906523|Pm_6906523| NULL|2020-10-27 10:12:13|   U_10000|
|       1|         1|   10000_6906524|Pm_6906524| NULL|2020-10-27 10:12:28|   U_10000|
|       1|         0|   10000_6906525|Pm_6906525| NULL|2020-10-27 10:14:56|   U_10000|
|       1|         0|   10000_6906526|Pm_6906526| NULL|2020-10-27 10:15:18|   U_10000|
|       1|         0|   10000_6906527|Pm_6906527| NULL|2020-10-27 10:15:41|   U_10000|
|       1|         0|   10000_6906528|Pm_6906528| NULL|2020-10-27 10:16:21|   U_10000|
|       3|         0|10000130_3624759|Pm_3624759| -1.0|2020-05-19 16:57:44|U_10000130|
|       2|         0|10000130_3624760|Pm_3624760| -1.0|2020-05-24 16:33:08|U_10000130|
|      11|         0|10000130_3624762|Pm_3624762| -1.0|2020-05-18 18:26:12|U_10000130|
+--------+----------+----------------+----------+-----+-------------------+----------+
only showing top 10 rows
```

Các cột đều đầy đủ giá trị trừ cột score bị thiếu 70836262 dòng (khoảng hơn 50%), vì trong file problem.json, phần score không có điểm (NaN) nên bảng user-problem cũng bị thiếu tương ứng

```
miss = user_problem.select([
    F.sum(F.col(c).isNull().cast('int')).alias(c)
    for c in user_problem.columns
])

miss.show()
```

```
[Stage 8:=============================================================>(166 + 2) / 168]
+--------+----------+------+----------+--------+-----------+-------+
|attempts|is_correct|log_id|problem_id|   score|submit_time|user_id|
+--------+----------+------+----------+--------+-----------+-------+
|       0|         0|     0|         0|70836262|          0|      0|
+--------+----------+------+----------+--------+-----------+-------+
```

Phân tích số lần thử (attempts) của học viên trong từng problems, có thể thấy đa số mọi problems thì các học viên đều có xu hướng chỉ thực hiện 1 lần, và trong số các lần thử đó thì số lần thử và thực hiện đúng chiếm tỉ lệ cũng rất cao

[34]:
```
c_a_pd
```

[34]:

	attempts	correct_submit
0	1	110730557
1	2	2839807
2	3	877817
3	4	318111
4	5	59950
...	...	...
119	190	1
120	191	1
121	202	2
122	299	1
123	458	1

124 rows × 2 columns

Thống kê cột score và attempts, có thể thấy miền giá trị của cột score từ -1 đến 100, trong đó -1 có thể là giá trị lỗi do số lần xuất hiện rất ít, cột attempts với số lần thử nhiều nhất lên đến 458 lần cho 1 problem

```
+-------+------------------+------------------+
|summary|             score|          attempts|
+-------+------------------+------------------+
|  count|          62548071|         133384333|
|   mean|1.1597642475656842|1.0587057776867992|
| stddev|1.5961156676355879|0.4070407094843258|
|    min|              -1.0|                 1|
|    max|             100.0|               458|
+-------+------------------+------------------+
```

b. user

File dữ liệu học viên (user.json) chứa 3330294 dòng và 7 cột: course_order, enroll_time, gender, id, name, school và year_of_birth

```
data_user.show(truncate=True)
```

```
+--------------------+--------------------+------+-----+------+--------+-------------+
|        course_order|         enroll_time|gender|   id|  name|  school|year_of_birth|
+--------------------+--------------------+------+-----+------+--------+-------------+
|   [682129, 2294668]|[2019-10-12 10:28...|     0| U_22|    我|        |         2015|
|[597214, 605512, ...|[2019-05-20 16:06...|     1| U_24|王帅国|清华大学|         6558|
|           [1903985]|[2020-08-07 18:59...|     0| U_25|王帅国|清华大学|         NULL|
|[696679, 1704639,...|[2020-03-01 21:24...|     1| U_53|于歆杰|清华大学|         1973|
|[682442, 682164, ...|[2019-10-09 02:17...|     2| U_54|马昱春|清华大学|         NULL|
|            [696679]|[2019-12-20 12:06...|     1| U_67|李小马|学堂在线|         6798|
|    [696692, 948431]|[2020-01-21 10:15...|     2| U_68|    秋|清华大学|         NULL|
|[375775, 375778, ...|[2019-02-26 19:21...|     1| U_69| 培源_|清华大学|         NULL|
|[676664, 707135, ...|[2020-04-22 18:23...|     0| U_90|刘俊洋| qinghua|         NULL|
|    [677018, 881485]|[2020-07-12 11:10...|     1|U_104|周凯华|  雨课堂|         NULL|
|[609639, 597334, ...|[2019-06-04 17:03...|     1|U_105|吕秋亮|        |         NULL|
|    [375775, 797938]|[2019-04-02 15:05...|     0|U_108|刘典典|清华大学|         NULL|
|[696994, 735338, ...|[2019-09-20 16:34...|     1|U_112|姚保峰|学堂在线|         NULL|
|[696740, 676664, ...|[2020-02-12 11:00...|     1|U_118|张一嵩|清华大学|         NULL|
|[801420, 676905, ...|[2020-02-11 16:29...|     1|U_119|  刘震|清华大学|         1976|
```

Kiểm tra schema, nhận thấy các trường dữ liệu đều có thể chứa giá trị null. Thống kê số lượng giá trị null trong các cột dữ liệu, cột **gender, name** và **school** chứa 54 giá trị null, cột **year_of_birth** chứa 3281764 giá trị null (khoảng hơn 98% dữ liệu là null), có thể là do khi đăng kí không bắt buộc nhập năm sinh

```
miss = data_user.select([
    F.sum(F.col(c).isNull().cast("int")).alias(c)
    for c in data_user.columns
])

miss.show()
```

```
[Stage 5:================================================>        (6 + 1) / 7]
+------------+-----------+------+---+----+------+-------------+
|course_order|enroll_time|gender| id|name|school|year_of_birth|
+------------+-----------+------+---+----+------+-------------+
|           0|          0|    54|  0|  54|    54|      3281764|
+------------+-----------+------+---+----+------+-------------+
```

Có 4701 khóa học khác nhau với số học viên đăng kí từ 1 đến 231674 học viên đăng kí

```
c_u_count.sort_values(by='count', ascending=True)
```

	course_id	count
4700	1890074	1
3442	735249	1
3443	1771162	1
3447	1908916	1
1141	947689	1
...	...	...
4295	883345	78374
1166	697791	96210
413	676932	125789
2257	696994	181697
2935	936971	231674

4701 rows × 2 columns

Có 1198696 học viên tham gia từ 2 khóa học trở lên (khoảng 30%), cho thấy xu hướng tham gia nhiều khóa học không cao, học viên không có nhu cầu học nhiều khóa học, ngoài ra có một số học viên đăng kí rất nhiều khóa học (>1000 khóa học)

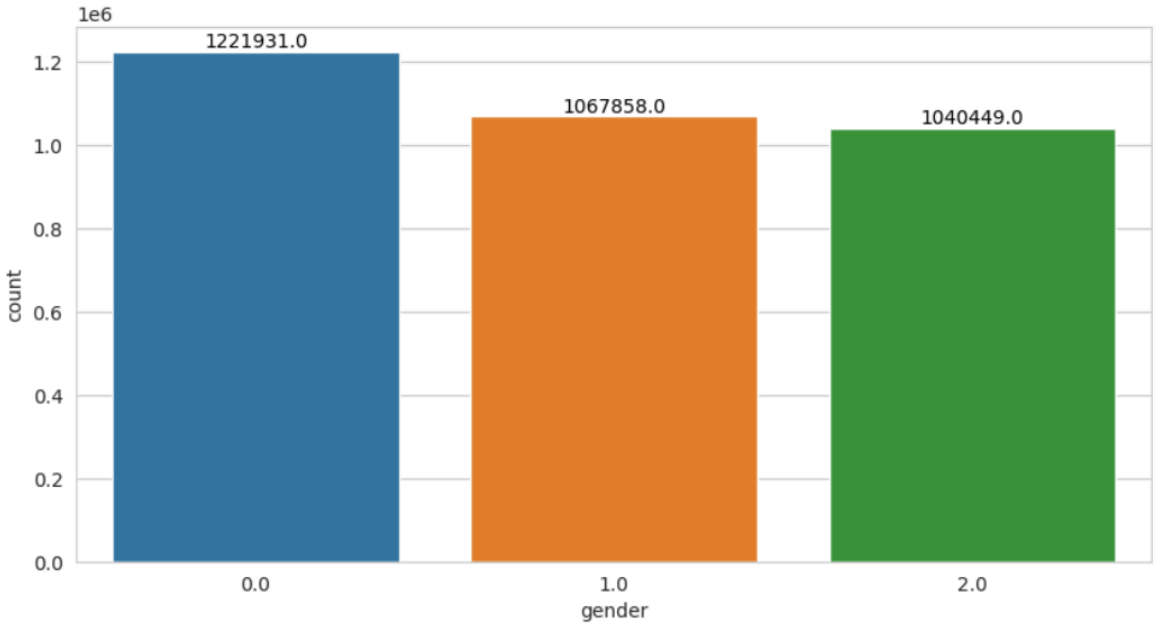
[99]:
```
user_pd[user_pd['num_course'] >= 2]
```

[99]:

	id	num_course
0	U_9137692	3715
1	U_30963921	2000
2	U_25254297	1976
3	U_13956967	1945
4	U_13344314	1887
...	...	...
1198691	U_34712057	2
1198692	U_34712058	2
1198693	U_34712061	2
1198694	U_34712092	2
1198695	U_34712099	2

1198696 rows × 2 columns

Thống kê giá trị Gender có 5 giá trị **0, 1, 2, 3, 232**, và **null,** trong đó 3, 232 và null chiếm số lượng rất ít nên có thể xem là giá trị ngoại lai, xem như giá trị [0;2] là hợp lệ.



Thống kê giá trị School cho thấy số lượng trường là 24974 trường, ngoài 54 giá trị null còn có 2201841 học viên để trống trường (khoảng hơn 60%)

```
count_school.sort_values(by='count', ascending=False)
```

	school	count
12547		2201841
9382	清华大学	18412
14397	昆明理工大学	15351
5982	湖南大学	13388
1406	河南工业大学	10198
...	...	...
10704	黄文鹏	1
10703	标榜	1
10701	广州中医药大学第二临床医学院	1
10699	....	1
24973	智能	1

24974 rows × 2 columns

Lọc ra số lượng đăng kí khóa học của học viên thuộc các trường không rỗng và tính trung bình, kết quả là trung bình học viên thuộc các trường sẽ học khoảng 2.11 khóa học

```
data_school_nonull = data_user.filter(col("school") != "")


avg_all = data_school_nonull.select(
    F.avg("num_course").alias("avg_overral")
)

avg_all.collect()[0]["avg_overral"]
```

```
2.1165465407183097
```

Nhận xét: Kết quả này cho thấy học viên thuộc các trường có nhiều học viên đăng kí thường sẽ học nhiều khóa học hơn so với học viên thuộc các trường có ít học viên đăng kí

```
[Stage 32:=================================================>       (6 + 1) / 7]
+--------------------+-----------------------+
|              school|avg_courses_per_student|
+--------------------+-----------------------+
|              清华大学|     4.1979144036497935|
|陕西工业职业技术学院|     3.8400726942298955|
|              重庆大学|     3.0058884655351576|
|          华南理工大学|     2.2674235611510793|
|          河北工业大学|      2.235670082175104|
|          昆明理工大学|     2.0757605367728487|
|              湖南大学|     1.7518673438900507|
|          湖北科技学院|     1.6933225321143386|
|          广西科技大学|      1.255706783838817|
|          河南工业大学|     1.2158266326730731|
+--------------------+-----------------------+
```

Thống kê giá trị cột year_of_birth chỉ chiếm 48530/3330294 dòng dữ liệu, giá trị trải dài từ 1111 đến 9989, có thể thấy dữ liệu thiếu rất nhiều và không chính xác, không mang giá trị cụ thể

```
yob = data_user.select('year_of_birth')
yob.describe().show()
```

```
[Stage 1:=========================================>              (5 + 2) / 7]
+-------+------------------+
|summary|     year_of_birth|
+-------+------------------+
|  count|             48530|
|   mean|2039.0162991963734|
| stddev| 358.6743025264895|
|    min|              1111|
|    max|              9989|
+-------+------------------+
```

Trong 48530 dòng dữ liệu thì có đến hơn 46000 học viên sinh năm 2020, chỉ khoảng 1-5 tuổi, dữ liệu không hợp lệ

```
valid = yob_pd[(yob_pd['year_of_birth'] > 1950) & (yob_pd['year_of_birth'] <= 2025) ]
valid.value_counts()
```

```
year_of_birth
2020.0          46524
2016.0            443
1997.0            258
1996.0            187
1998.0            141
1995.0             97
1994.0             64
1992.0             53
1993.0             51
1987.0             34
1982.0             30
```

c. comment

Phân tích ban đầu được thực hiện để hiểu rõ cấu trúc và chất lượng của tập dữ liệu bình luận. Tệp comment.json là một tập dữ liệu lớn, chứa **8,395,141 bản ghi (dòng)** và bao gồm 5 trường thông tin (cột): create_time, id, resource_id, text.

create_time	id	resource_id	text	user_id
2019-08-05 12:55:27	Cm_1	NULL	测试评论	10030806
2019-08-05 16:56:43	Cm_4	NULL	嗯嗯	1705400
2019-08-07 21:05:38	Cm_5	NULL	是的，我也看不到	10031537
2019-08-09 13:06:06	Cm_7	NULL	大师傅as	10031502
2019-08-09 16:38:56	Cm_12	NULL	点赞	10031397
2019-08-09 17:29:01	Cm_13	NULL	好滴	10031397
2019-08-09 17:41:54	Cm_14	NULL	很好，赞一个	10031528
2019-08-09 17:44:31	Cm_16	NULL	老师好	10031531
2019-08-09 18:20:05	Cm_19	NULL	好的	10031356
2019-08-09 18:20:07	Cm_20	NULL	好的	10031356
2019-08-10 15:22:06	Cm_23	NULL	讨论区无直接粘贴功能，无@老师或学生提醒指定人员回答功能	10031509
2019-08-10 22:39:23	Cm_24	NULL	收到	10031531
2019-08-12 12:20:04	Cm_29	NULL	用手机4G热点看视频，速度相对还可以	10031666
2019-08-12 12:35:17	Cm_31	NULL	图片显示全幅速度比较慢	10031666
2019-08-12 13:08:27	Cm_32	NULL	好好学习，天天向上。	10031546
2019-08-12 13:11:48	Cm_33	NULL	优秀！	10031546
2019-08-12 14:08:32	Cm_36	NULL	7、讨论区自己发布的话题无法进行重新编辑 \n8、作答习题时，每个题目需要分别提交答案，不能一次性提交\n9、章节看完之后，只能点击返回，没有进入到下一章的按钮	10031572
2019-08-12 14:31:21	Cm_39	NULL	看得到吗？	10031607
2019-08-12 14:35:54	Cm_40	NULL	有道理	10031511
2019-08-12 14:42:54	Cm_43	NULL	你真棒	10031508

- **Cấu trúc dữ liệu (Schema):** Các trường dữ liệu chính bao gồm create_time (thời gian bình luận), id (mã bình luận), resource_id (mã tài nguyên được bình luận), text (nội dung bình luận), và user_id (mã người dùng).
- **Khả năng chứa giá trị null:** Tất cả các trường đều được thiết lập là nullable = true, cho thấy khả năng thiếu sót dữ liệu.

Để đánh giá chất lượng dữ liệu, một thống kê về các giá trị rỗng (null) đã được thực hiện.

```
-> File dữ liệu 'comment.json' chứa 8395141 dòng và 5 cột.
   Các cột bao gồm: ['create_time', 'id', 'resource_id', 'text', 'user_id']

--------------------------------------------------

--- Cấú trúc (Schema) của dữ liệu: ---
root
 |-- create_time: string (nullable = true)
 |-- id: string (nullable = true)
 |-- resource_id: string (nullable = true)
 |-- text: string (nullable = true)
 |-- user_id: long (nullable = true)
```

```
--- Bảng thống kê số lượng giá trị null cho mỗi cột: ---
+-----------+---+-----------+----+-------+
|create_time| id|resource_id|text|user_id|
+-----------+---+-----------+----+-------+
|          0|  0|    6108653|   0|      0|
+-----------+---+-----------+----+-------+
```

- Dữ liệu về create_time, id, và user_id rất đầy đủ và không có giá trị nào bị thiếu.
- Tuy nhiên, cột resource_id thiếu tới **6,108,653** giá trị. Điều này cho thấy một phần lớn các bình luận không được liên kết trực tiếp với một tài nguyên cụ thể (video, bài tập) trong tệp này, có thể chúng là bình luận chung cho cả khóa học hoặc dữ liệu liên kết nằm ở một tệp khác.
- Cột text cũng thiếu một số lượng đáng kể dữ liệu, cho thấy có những bình luận được ghi lại nhưng không có nội dung.

d. reply

File dữ liệu phản hồi (reply.json) chứa 331011 dòng và 4 cột gồm id, user_id, text, create_time

```
df_reply.show(30, truncate=True)
```

```
+-------------------+-----+-------------------------------------+----------+
|        create_time|   id|                                 text|   user_id|
+-------------------+-----+-------------------------------------+----------+
|2019-08-05 12:55:54| Rp_1|                             测试回复|U_10030806|
|2019-08-09 16:39:06| Rp_2|                                 赞|U_10031397|
|2019-08-10 22:39:35| Rp_3|                              好喜欢|U_10031531|
|2019-08-12 14:43:34| Rp_4|                             你也好棒|U_10031508|
|2019-08-12 14:44:51| Rp_5|                               嗯对|U_10031508|
|2019-08-12 14:47:58| Rp_6|                           人工智能是|U_10031508|
|2019-08-13 09:41:32| Rp_7|          我的观点就是，你说啥就时啥|U_10031536|
|2019-08-13 09:41:42| Rp_8|          我的观点就是，你说啥就时啥|U_10031536|
|2019-08-13 09:41:53| Rp_9|          我的观点就是，你说啥就时啥|U_10031536|
|2019-08-13 09:41:59|Rp_10|          我的观点就是，你说啥就时啥|U_10031536|
|2019-08-19 18:08:53|Rp_11|                            11111|U_10057014|
|2019-09-05 14:50:14|Rp_13|                              发达|   U_11731|
|2019-09-05 14:50:18|Rp_14|                              打发|   U_11731|
|2019-09-05 14:50:26|Rp_15|                           打发大水|   U_11731|
|2019-09-06 18:40:35|Rp_16|                               嗯| U_6875014|
|2019-09-09 14:39:01|Rp_17|      不好意思，电脑出错了，重复发了|U_10745511|
|2019-09-09 19:00:15|Rp_18|                       我说的是如果|U_11031591|
|2019-09-09 19:00:43|Rp_19|                 好吧，可能得问问老师了|U_11035492|
|2019-09-10 21:09:31|Rp_20|嘻嘻，大家快来讲讲自己的看法呀！会...|U_11086632|
|2019-09-11 08:49:26|Rp_21|      非等压过程是有焓变，H=U+PV可...| U_9734721|
|2019-09-11 08:55:00|Rp_22|为什么老说我注水，打广告，不能评论...|U_10745584|
|2019-09-11 11:57:45|Rp_23|                  我也被灌水了哈哈哈|U_10745556|
|2019-09-12 08:43:34|Rp_24|                             不错哦| U_4622505|
|2019-09-12 09:08:46|Rp_25|                             你也不错| U_4622603|
|2019-09-12 10:45:03|Rp_26|同学的答案基本意思对了，有一点可以...|U_10926748|
|2019-09-12 11:27:59|Rp_27|                               优秀| U_4622588|
|2019-09-12 15:33:32|Rp_28|水分解成氧气和氢气，温度一般要求达...| U_9734771|
|2019-09-12 15:39:27|Rp_29|既然是乳化燃料，水和油一起混合，那...| U_9734771|
|2019-09-12 21:29:41|Rp_30|                           百度？呵呵| U_9734741|
|2019-09-12 21:30:23|Rp_31|                           少烧燃料| U_9734741|
+-------------------+-----+-------------------------------------+----------+
only showing top 30 rows
```

Kiểm tra schema, ta nhận thấy rằng toàn bộ các trường dữ liệu đều có thể chứa giá trị null

```
df_reply.printSchema()
```

```
root
 |-- create_time: string (nullable = true)
 |-- id: string (nullable = true)
 |-- text: string (nullable = true)
 |-- user_id: string (nullable = true)
```

Tuy nhiên khi thống kê số lượng null trong bảng df_reply thì cả 4 cột đều không có giá trị NULL do các id là các trường duy nhất khác null, cột text dành cho việc phản hồi nên hầu như đều phải có ký tự và thời gian tạo phải được khởi tạo khi phản hồi

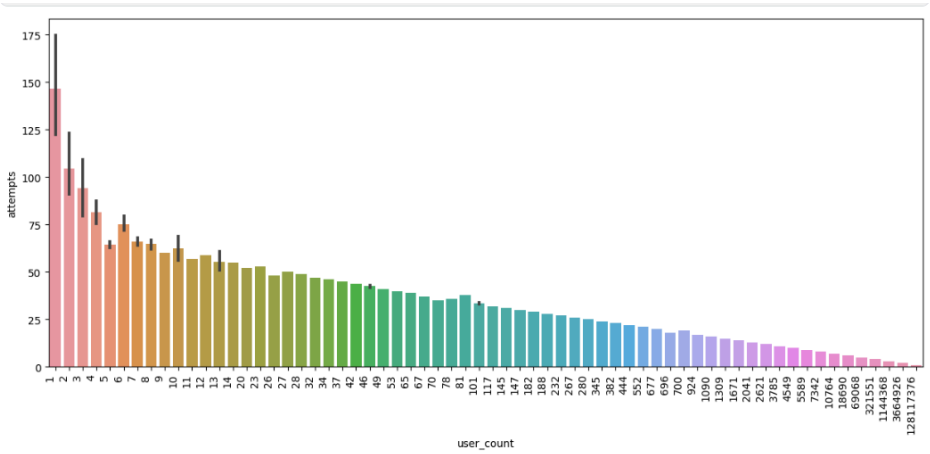
```
from pyspark.sql import functions as F
miss = df_reply.select([
    F.sum(F.col(c).isNull().cast("int")).alias(c)
    for c in df_reply.columns
])

miss.show()
```

```
+-----------+---+----+-------+
|create_time| id|text|user_id|
+-----------+---+----+-------+
|          0|  0|   0|      0|
+-----------+---+----+-------+
```

1.2.2. Trực quan hóa dữ liệu

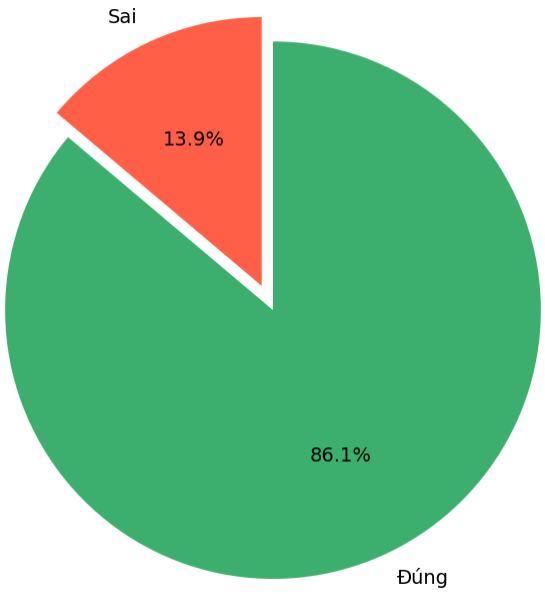
a. user-problem

Để trực quan hoá các giá trị phổ biến nhất của mẫu user-problem.json, nhóm vẽ biểu đồ cho các trường attempts, correct và user_id như sau:



Nhận xét: Biểu đồ cột này mô tả tần suất nộp bài của học viên, phản ánh một mô hình tương tác điển hình trong môi trường học tập trực tuyến (LMS), nơi đa số người dùng chỉ thực hiện số lượng hoạt động tối thiểu.

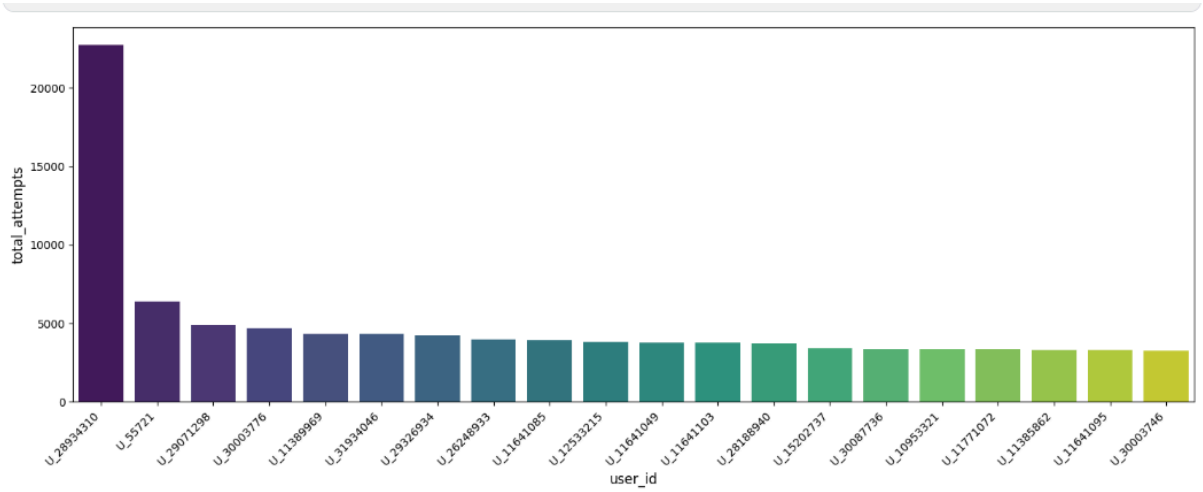
- **Đỉnh cao nhất:** Biểu đồ đạt đỉnh cao nhất ở cột 1 lần nộp bài và 2 lần nộp bài. Nhóm này bao gồm những học viên hoàn thành bài tập ngay lần thử đầu tiên, hoặc những người đã xem bài tập nhưng nhanh chóng từ bỏ sau một hoặc hai lần tương tác ban đầu.
- **Phân bố lệch dương:** Phân bố tần suất cho thấy sự mất cân bằng rõ rệt: số lượng người dùng giảm mạnh theo quy tắc phân bố mũ (Exponential Decay) khi tần suất nộp bài tăng lên.



Nhận xét: Biểu đồ tròn này thể hiện sự phân bố về tỷ lệ các lần nộp bài tập được đánh dấu là đúng (is_correct = 1) so với tỷ lệ các lần nộp bài bị đánh dấu là sai (is_correct = 0) trong toàn bộ dữ liệu tương tác của học viên.

- Tỷ lệ sai hơn 86% cho thấy bài tập có thể rất khó, buộc học viên phải thử đi thử lại nhiều lần để đạt được kết quả đúng.

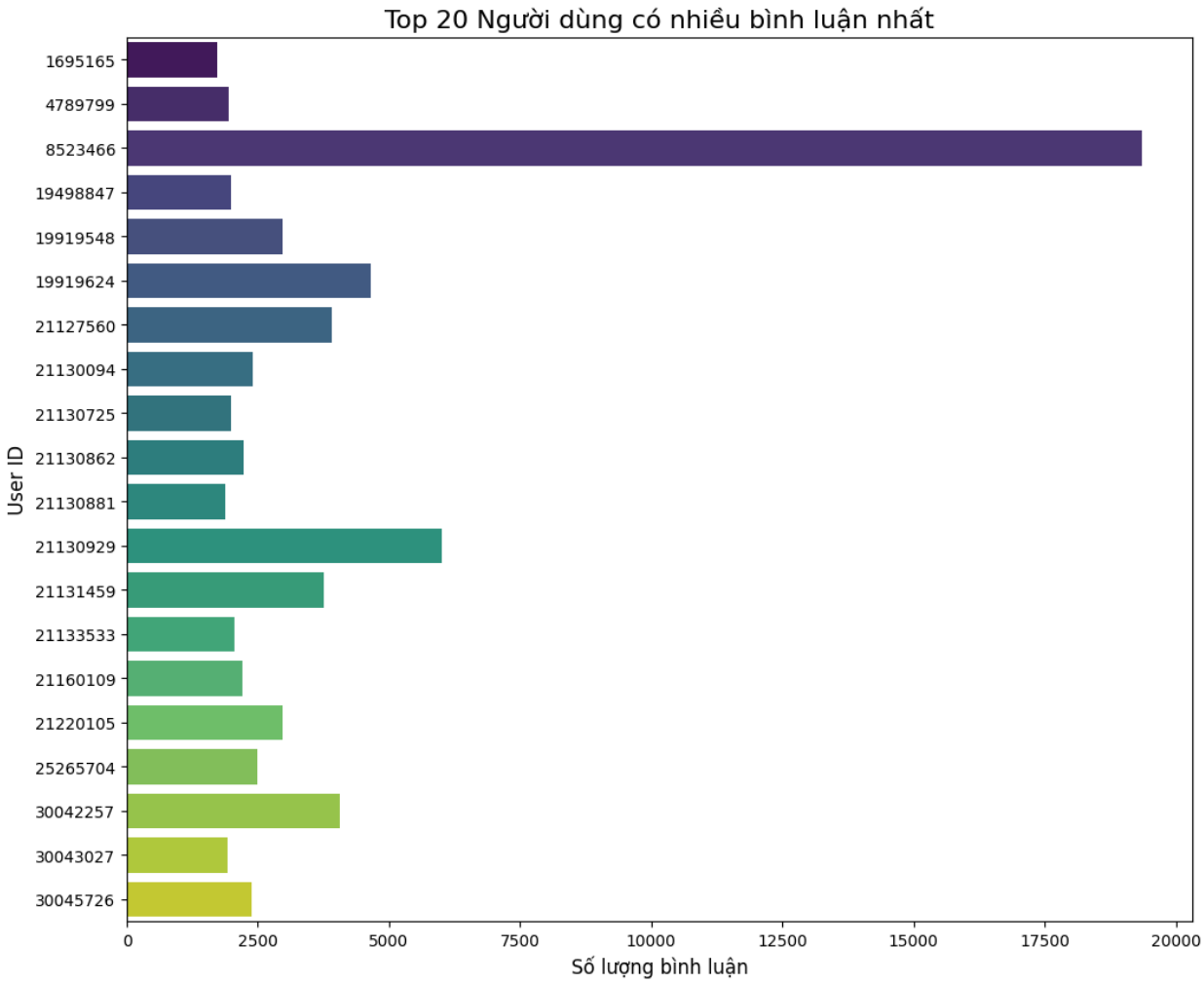
- Tỷ lệ 13.9% nộp bài đúng phản ánh rằng chỉ có một tỷ lệ nhỏ học viên hoàn thành nhiệm vụ ngay từ những lần thử ban đầu. Điều này củng cố tầm quan trọng của việc phân tích số lần nộp bài (attempts), vì nó là đặc trưng dẫn đến việc đạt được kết quả đúng.



Nhận xét: Biểu đồ cột này làm nổi bật nhóm người dùng có tương tác chuyên sâu nhất với hệ thống bài tập. Đây là một phân tích quan trọng để xác định các trường hợp ngoại lệ (outliers) và hành vi học tập cực đoan.

- Học viên đứng vị trí số 1 có số lần nộp bài vượt trội (> 20000 attempts), cao hơn đáng kể so với người dùng ở vị trí thứ 2. Điều này cho thấy đây là một trường hợp ngoại lệ rõ ràng.
- Từ vị trí thứ 2 trở đi, số lần nộp bài giảm đi nhanh chóng, cho thấy sự khác biệt về mức độ tương tác giữa người dùng top 1-2 và phần còn lại.

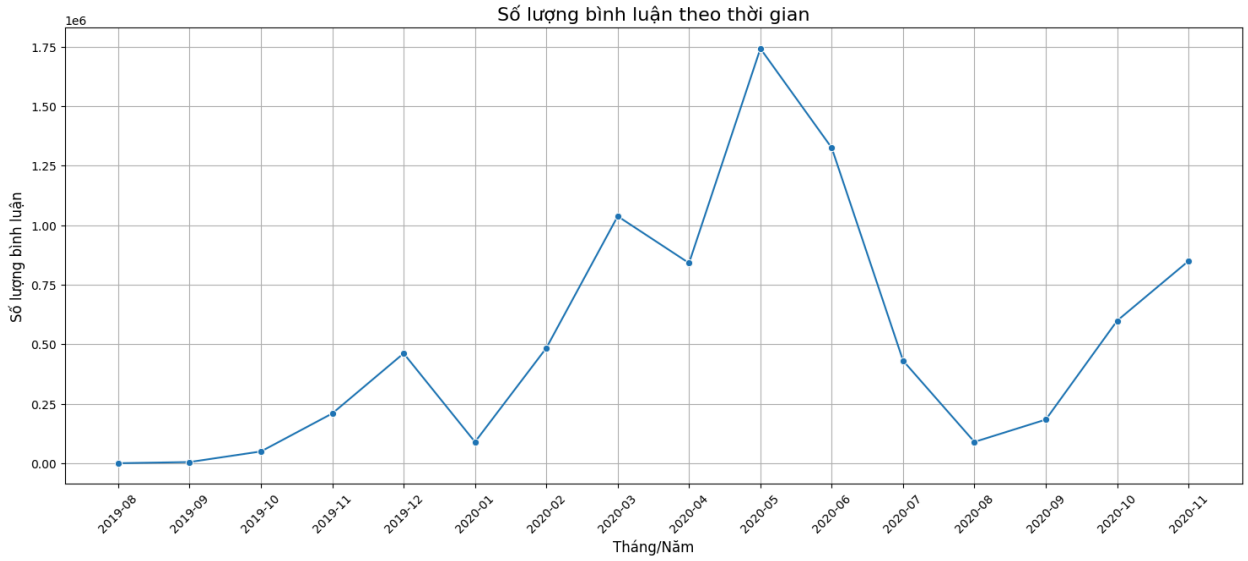
b. comment



Nhận xét:

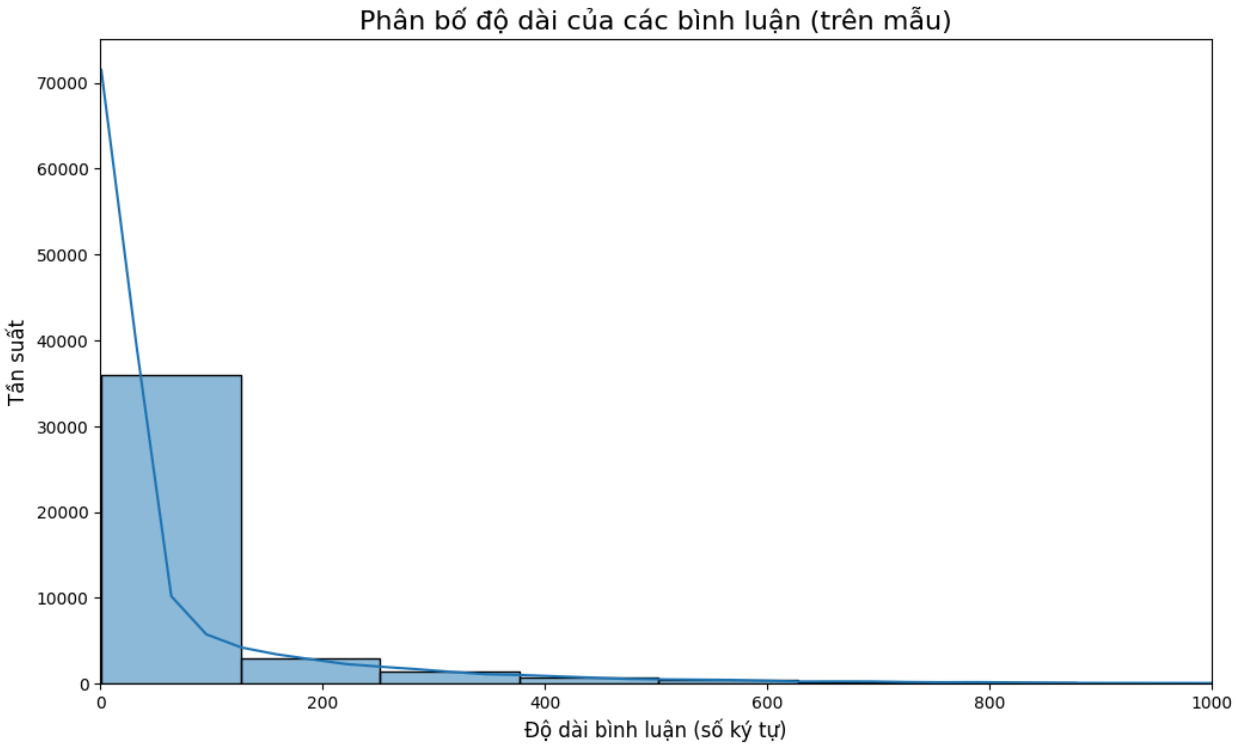
- **Mức độ tương tác rất chênh lệch:** Biểu đồ cho thấy một sự chênh lệch rõ rệt trong hoạt động của người dùng. Một nhóm nhỏ người dùng ("superusers") tạo ra một số lượng bình luận cực kỳ lớn.
- **Người dùng nổi bật:** Người dùng có ID 8523466 là người tích cực nhất với gần 20,000 bình luận, cao hơn đáng kể so với những người dùng khác. Những người dùng top đầu này có thể là trợ giảng (TA), những học viên đặc biệt tâm huyết, hoặc quản trị viên của nền tảng.

- **Phân phối "Long-tail":** Dữ liệu thể hiện rõ quy luật "đuôi dài" (long-tail), nơi phần lớn người dùng chỉ có một vài bình luận, trong khi một số ít người dùng lại có tần suất tương tác rất cao. Việc xác định và khuyến khích những "superusers" này có thể là chìa khóa để thúc đẩy sự tương tác trong cộng đồng.



Nhận xét:

- **Hoạt động theo chu kỳ:** Biểu đồ cho thấy hoạt động bình luận có tính chu kỳ rõ rệt, tăng và giảm theo các giai đoạn trong năm.
- **Đỉnh điểm hoạt động:** Lượng tương tác tăng mạnh từ cuối năm 2019 và đạt đỉnh vào khoảng **tháng 5 năm 2020** với hơn 1.75 triệu bình luận. Giai đoạn này trùng khớp với các học kỳ chính và cũng có thể được khuếch đại bởi sự bùng nổ của việc học trực tuyến toàn cầu do đại dịch COVID-19.
- **Giai đoạn thấp điểm:** Lượng bình luận giảm mạnh vào các tháng 7-8, có thể tương ứng với kỳ nghỉ hè hoặc thời gian giữa các khóa học.
- **Tính ứng dụng:** Dữ liệu này rất quý giá cho việc lập kế hoạch. Nền tảng có thể đẩy mạnh việc ra mắt các khóa học mới hoặc tổ chức các sự kiện tương tác vào những tháng cao điểm (ví dụ: tháng 3-5 và tháng 9-11) để tối đa hóa sự tham gia của người học.



Nhận xét:

- **Bình luận ngắn là chủ đạo:** Biểu đồ phân bố lệch phải rất mạnh, cho thấy **đại đa số các bình luận đều rất ngắn**, tập trung chủ yếu trong khoảng dưới 150 ký tự.
- **Bản chất tương tác:** Điều này cho thấy phần lớn tương tác trên nền tảng là các phản hồi nhanh, câu hỏi ngắn, hoặc các bình luận đơn giản như "cảm ơn", "hay quá", "em hiểu rồi".
- **Vẫn có các thảo luận sâu:** Mặc dù số lượng ít, phần "đuôi dài" của biểu đồ cho thấy vẫn có những người dùng viết các bình luận dài hơn. Đây có thể là những thảo luận chi tiết, các câu hỏi phức tạp hoặc những chia sẻ kiến thức có giá trị, đóng góp chiều sâu cho các cuộc hội thoại học thuật.

2. Làm sạch dữ liệu và chuyển đổi dữ liệu

2.1. Dữ liệu tĩnh về học viên và khóa học

Bộ dữ liệu tĩnh được tạo ra với các trường chứa thông tin không thay đổi về học viên cùng với các khóa học như mã học viên (user_id), mã khóa học mà học viên đăng ký (course_id).

- **Bổ sung các thông tin về khóa học:** sử dụng bảng course để tính tổng số tài nguyên mà khóa học cung cấp (số video và số exercise) bằng cách duyệt qua các resource_id trong danh sách các tài nguyên của khóa học, lấy ra các id bắt đầu bằng **V_** và **Ex_** và đánh dấu resource_id đó là is_video hoặc is_exercise, sau đó tính tổng is_video và is_exercise cho khóa học.

```
# Tính video count và ex count cho course

course_exploded = course.select(
    col("id").alias("course_id"),
    explode(col("resource")).alias("resource_data")
)

course_categorized = course_exploded.select(
    col("course_id"),
    col("resource_data.resource_id").alias("resource_id"),
    when(col("resource_id").rlike("^V_.*"), 1).otherwise(0).cast('integer').alias("is_video"),
    when(col("resource_id").rlike("^Ex_.*"), 1).otherwise(0).cast('integer').alias("is_exercise")
)


course_agg = (
    course_categorized.groupBy("course_id")
      .agg(
          F.sum("is_video").alias("total_videos"),
          F.sum("is_exercise").alias("total_exercises")
      )
)
```

- Nối bảng chứa các thông tin vừa tạo cùng với bảng user_course ban đầu và chuẩn hóa cột thời gian (enroll_time) từ datetime **yyyy-mm-dd hh:mm:ss** sang **yyyy-mm-dd** để tiện lợi cho việc tính toán và chia dữ liệu về sau.

Sau đó, nhóm trích xuất thêm đặc trưng từ khóa học là “is_presiquites” mang giá trị 0/1 tượng trưng cho khóa học đó có cần khóa học tiên quyết hay không, “num_fields” bằng cách đếm số lượng fields của khóa học đề cập đến, và cuối cùng là

“total_students_enrolled” đại diện cho tổng số học viên đã tham gia khóa học vào thời điểm hiện tại.

```
is_not_nan_series = course['prerequisites'].notna()
course['is_prerequisites'] = is_not_nan_series.astype(int)

course = course.drop(['about', 'field', 'name', 'resource', 'prerequisites'], axis=1)
course
```

:

	course_id	total_students_enrolled	total_videos	total_exercises	num_fields	is_prerequisites
0	C_680777	14843	19	9	1	0
1	C_682515	21069	133	44	0	0
2	C_696855	14863	161	12	0	0
3	C_680958	14268	107	70	1	1
4	C_680808	10363	79	15	1	0

Ở dữ liệu về học viên sẽ tiến hình trích xuất số khóa học từ cột course_order của học viên để lấy ra tổng số course_id mà học viên đã đăng kí và tạo thành cột mới “total_courses_enrolled”, ngoài ra nhóm cũng trích xuất thêm giá trị gender bằng cách drop các giá trị ngoại lai (giá trị thiểu số khác với 0/1/2). Dữ liệu tĩnh về học viên và khóa học sau cùng được nối lại với nhau như sau:

	course_id	user_id	gender	total_courses_enrolled	total_students_enrolled	total_videos	total_exercises	num_fields	is_prerequisites
0	C_883345	U_10384705	1.0	8	78374	1006	194	0	0
1	C_676937	U_11161839	0.0	4	21331	84	30	1	0
2	C_883345	U_11403604	1.0	1	78374	1006	194	0	0
3	C_883345	U_11509911	1.0	3	78374	1006	194	0	0
4	C_676937	U_11646006	1.0	6	21331	84	30	1	0

Bộ dữ liệu tĩnh sau khi xử lí

2.2. Dữ liệu về kết quả học tập

Bộ dữ liệu ghi lại hành vi làm bài tập của học viên trong khóa học, tính từ lúc bắt đầu đăng kí khóa học (enroll_time), bao gồm số lượng problem và exercise đã làm (problems_done, exercisse_touch), số lần thử (total_attempts), số lần làm đúng (total_submissions), số điểm nhận được (total_earned_score).

- **Tạo bảng chứa thông tin các exercise và problems của 1 course:** sử dụng bảng exercise_problem có sẵn, kết hợp với bảng course_exercise tự tạo bằng cách explode các resource_id và lấy ra các id bắt đầu bằng **Ex_.** (mỗi course có nhiều exercise, mỗi exercise có nhiều problem)
- **Hợp nhất dữ liệu:** dùng bảng user_problem nối với bảng problem để bổ sung điểm của bài tập (problem_score)
    - **earned_score:** được tính bằng cách lấy điểm đã được ghi nhận trực tiếp trong user_problem, nếu không có sẽ dựa vào is_correct, nếu is_correct là 1 thì sẽ lấy điểm tối đa từ problem score, ngược lại là 0.

```
fact = (
    user_problem.alias("up")
    .join(
        problem.select("problem_id", F.col("score").alias("problem_score")),
        on="problem_id", how="left"
    )
    .withColumn(
        "earned_score",
        F.when(F.col("up.score").isNotNull(), F.col("up.score"))
         .otherwise(
            F.when(F.col("up.is_correct")==1, F.coalesce(F.col("problem_score"), F.lit(0.0)))
             .otherwise(F.lit(0.0))
         )
    )
    .select(
        "up.user_id", "up.problem_id", "up.is_correct", "up.attempts",
        "up.submit_time", "problem_score", "earned_score"
    )
)
```

    - Nối ngược lại với bảng user_course tĩnh và course_exercise_problem để lấy ra chi tiết chứa đầy đủ các khóa ngoại cần thiết (user_id, course_id, exercise_id, problem_id) và các đặc trưng tương tác (is_correct, attempts, earned_score), bảng sau khi nối có dạng như sau:

```
[Stage 54:>                                                         (0 + 1) / 1]
+-------+---------+-----------+----------+----------+--------+-------------------+-------------+------------+
|user_id|course_id|exercise_id|problem_id|is_correct|attempts|        submit_time|problem_score|earned_score|
+-------+---------+-----------+----------+----------+--------+-------------------+-------------+------------+
|U_10000|C_2033958| Ex_7007033|Pm_6906523|         0|       1|2020-10-27 10:12:13|         NULL|         0.0|
|U_10000|C_2033958| Ex_7007033|Pm_6906524|         1|       1|2020-10-27 10:12:28|         NULL|         0.0|
|U_10000|C_2033958| Ex_7007033|Pm_6906522|         0|       1|2020-10-27 10:11:56|         NULL|         0.0|
|U_10000|C_2033958| Ex_7007033|Pm_6906526|         0|       1|2020-10-27 10:15:18|         NULL|         0.0|
|U_10000|C_2033958| Ex_7007033|Pm_6906527|         0|       1|2020-10-27 10:15:41|         NULL|         0.0|
+-------+---------+-----------+----------+----------+--------+-------------------+-------------+------------+
only showing top 5 rows
```

- **Thống kê và trích xuất các đặc trưng mới:** mục tiêu của bước này là trích xuất ra các thông tin về việc làm bài tập của học viên trong một khóa học, chia theo từng mốc thời gian được định nghĩa thành các 4 giai đoạn, mỗi giai đoạn là 2 tuần, được tính từ thời điểm học viên bắt đầu đăng kí khóa học (enroll_time).
    - **Tính số ngày nộp bài**: Tính toán days_since_enroll (số ngày kể từ khi học viên đăng ký đến ngày nộp bài)

```
time_difference = test['submit_date'] - test['enroll_date_']

test['days_since_enroll'] = time_difference.dt.days
```

    - **Giữ lại các thông tin tĩnh:** được trích xuất bằng cách drop_duplicates dựa trên cặp khóa (user_id, course_id), bao gồm: course_exercises, course_videos, duration_days, và remaining_time.

```
static_cols = ["user_id", "course_id", "total_exercises", "total_videos", "duration_days", "remain_day"]

static_info = (
    df[static_cols]
    .drop_duplicates(subset=["user_id", "course_id"])
    .rename(columns={
        "total_exercises": "course_exercises",
        "total_videos": "course_videos",
        "remain_day": "remaining_time"
    })
)
```

    - **Tạo các đặc trưng về mức độ làm bài tập của học viên:** thực hiện tính tổng hợp (groupby và agg) cho các chỉ số hoạt động theo từng phase (problems_done, exercises_touched, total_attempts, correct_submissions, và total_earned_score), các giá trị bị thiếu (học viên không hoạt động trong phase đó) được điền bằng 0.

```
agg_phase = (
    df
    .groupby(["user_id", "course_id", "phase"], as_index=False)
    .agg(
        problems_done       = ("problem_id", "nunique"),
        exercises_touched   = ("exercise_id", "nunique"),
        total_attempts      = ("attempts", "sum"),
        correct_submissions = ("is_correct", "sum"),
        total_earned_score  = ("earned_score", "sum"),
        avg_earned_score    = ("earned_score", "mean"),
        avg_problem_score   = ("score", "mean")
    )
)
```

    - **Tính thêm các thông số phụ:**

- **accuracy:** tỷ lệ nộp bài đúng (correct_submissions/total_attempts)
- **Exercise_coverage:** phần trăm bài tập đã làm (exercises_touched/course_exercises).
- **problems_per_day:** tần suất làm bài tập (problems_done/14 ngày).
- **Earned_per_attempt:** mức độ hiệu quả khi làm bài (total_earned_score/total_attempts).

Dữ liệu sau khi thống kê sẽ tiếp tục được xử lí chia thành các giai đoạn theo hướng tích lũy, cụ thể được xử lý theo 4 bước chính:

- **Phân nhóm dữ liệu:** Nhóm theo từng cặp (user_id, course_id) để phân tích riêng biệt từng người học trong từng khóa học.
- **Chia giai đoạn thời gian:** Xác định 4 phase học tập liên tiếp dựa trên enroll_date và các mốc kết thúc phase. Mỗi phase được xác định rõ khoảng thời gian [start, end).
- **Tính toán chỉ số:** Với mỗi phase, tính 12 chỉ số quan trọng:
  - **Phạm vi học tập:** exercises_touched, problems_done, exercise_coverage
  - **Hiệu suất:** accuracy, correct_submissions, total_attempts
  - **Điểm số:** total_earned_score, avg_earned_score, avg_problem_score, avg_earned_ratio
  - **Nhịp độ**: problems_per_day, earned_per_attempt
- **Xử lý trường hợp đặc biệt:**
  - Phase không có dữ liệu trả về toàn bộ chỉ số bằng 0
  - Tránh chia cho 0 trong các phép tính tỉ lệ
  - Đảm bảo tính duy nhất khi đếm (sử dụng nunique)

Kết quả: Mỗi bản ghi đầu ra chứa đầy đủ thông tin người dùng, khóa học, phase và các chỉ số thống kê, sẵn sàng cho phân tích.

```
final_df[final_df.user_id == "U_1000982"]
```

ouched	problems_done	total_attempts	correct_submissions	total_earned_score	avg_earned_score	avg_earned_ratio	avg_problem_score	accuracy	problems_per_day	earned_per_attempt	user_id	course_id	phase	phase_start	phase_end
9	42	42	7	5.6	0.133333	0.1875	0.8	0.166667	4.666667	0.133333	U_1000982	C_947149	phase_1	2020-06-30	2020-07-08
0	0	0	0	0.0	0.000000	0.0000	0.0	0.000000	0.000000	0.000000	U_1000982	C_947149	phase_2	2020-07-08	2020-07-16
0	0	0	0	0.0	0.000000	0.0000	0.0	0.000000	0.000000	0.000000	U_1000982	C_947149	phase_3	2020-07-16	2020-07-24
0	0	0	0	0.0	0.000000	0.0000	0.0	0.000000	0.000000	0.000000	U_1000982	C_947149	phase_4	2020-07-24	2020-07-31

2.3. Dữ liệu về các tương tác video

Dữ liệu tương tác video ghi lại chi tiết hành trình tiêu thụ nội dung chính bài giảng của khóa học của người học.Đây là nguồn dữ liệu quan trọng phản ánh hành vi học tập thụ động (passive learning), cho phép phân tích sâu hơn về mức độ tập trung, thói quen tua lại (review) các nội dung khó hoặc tua nhanh (skip) các nội dung đã biết. Quy trình xử lý dữ liệu này được chia thành các bước chính sau:

a/ Làm sạch và chuẩn hóa cấu trúc dữ liệu

Dữ liệu gốc được lưu dưới dạng JSON với cấu trúc mảng lồng nhau: mỗi bản ghi chứa một chuỗi (seq) các hành động xem, và mỗi hành động lại chứa chi tiết các đoạn (segment) với thời gian bắt đầu, kết thúc và tốc độ xem.

Để đưa dữ liệu vào mô hình, nhóm thực hiện **Flattening/Explode** bằng PySpark.

```
from pyspark.sql.functions import col, explode, from_json, expr
from pyspark.sql.types import StructType, StructField, StringType, ArrayType
df_flatten = df_user_video.withColumn("seq", explode(col("seq")))

df_flatten = df_flatten.withColumn("segment", explode(col("seq.segment")))

df_flatten = df_flatten.select(
    col("user_id"),
    col("seq.video_id").alias("video_id"),
    col("segment.start_point"),
    col("segment.end_point"),
    col("segment.speed"),
    col("segment.local_start_time")
)
from pyspark.sql.functions import from_unixtime

df_flatten = df_flatten.withColumn("local_start_time", from_unixtime("local_start_time", "yyyy-MM-dd HH:mm:ss"))
```

b/ Tính toán các chỉ số hành vi nâng cao

Thay vì chỉ tính tổng thời gian, nhóm đã sử dụng **Window Functions** để so sánh hành động hiện tại với hành động trước đó của người dùng. Điều này cho phép trích xuất các hành vi học tập tinh vi như tua lại (review), tua nhanh (skip) và thay đổi tốc độ.

Các chỉ số được tính toán bao gồm:

- **rewind_time**: Tổng thời gian tua lại (khi điểm bắt đầu hiện tại nhỏ hơn điểm kết thúc trước đó).
- **fast_forward_time**: Tổng thời gian tua nhanh (bỏ qua nội dung).
- **interaction_count**: Tổng số thao tác tương tác với video.
- **session_count**: Số phiên học riêng biệt (ngắt quãng > 30 phút).
- **speed_change_count**: Số lần thay đổi tốc độ xem.

```
from pyspark.sql.functions import col, min, max, count, sum, lag, first, when, countDistinct

from pyspark.sql.window import Window
window_spec = Window.partitionBy("user_id", "video_id").orderBy("local_start_time")
df_lagged = df_flatten.withColumn("prev_end_point", lag("end_point").over(window_spec)) \
                      .withColumn("prev_speed", lag("speed").over(window_spec)) \
                      .withColumn("prev_start_time", lag("local_start_time").over(window_spec))
df_user_video_final = df_lagged.groupBy("user_id", "video_id").agg(
    round(max("end_point"), 2).alias("max_watch_point"),
    sum(
        when(col("prev_end_point") > col("start_point"),
             round(col("prev_end_point") - col("start_point"), 2))
        .otherwise(0)
    ).alias("rewind_time"),
    sum(
        when(col("start_point") > col("prev_end_point"),
             round(col("start_point") - col("prev_end_point"), 2))
        .otherwise(0)
    ).alias("fast_forward_time"),
    first("local_start_time").alias("first_local_start_time"),
    round(sum(col("end_point") - col("start_point")), 2).alias("actual_watch_time"),
    round(sum((col("end_point") - col("start_point")) / col("speed")), 2).alias("weighted_watch_time"),
    count("*").alias("interaction_count"),
    (sum(when(col("local_start_time").cast("long") - col("prev_start_time").cast("long") > 1800, 1).otherwise(0)) + 1).alias("session_count"),
    sum(when(col("speed") != col("prev_speed"), 1).otherwise(0)).alias("speed_change_count"),
    countDistinct("speed").alias("distinct_speeds_count"),
    hour(first("local_start_time")).alias("start_hour"),
    dayofweek(first("local_start_time")).alias("start_dayofweek")
)
df_user_video_final = df_user_video_final.withColumn(
    "average_speed",
    round(col("actual_watch_time") / col("weighted_watch_time"), 2)
)
df_user_video_final.show(truncate=True)
```

c/ Tích hợp và tính tích lũy theo giai đoạn

Dữ liệu được phân chia và tích lũy theo từng giai đoạn (Phase). Nhóm thực hiện kỹ thuật **Join tích lũy** giữa bảng dữ liệu khóa học đã phân chia phase (uc_expanded) và dữ liệu hành vi video (df_user_video_final).

Quy trình thực hiện bao gồm:

- **Chuẩn bị:** Chuyển đổi thời gian xem video thành ngày (event_date).
- **Ghép nối (Join):** Liên kết bảng Phase và bảng Video với điều kiện event_date <= phase_deadline. Điều này đảm bảo tính tích lũy: dữ liệu của Phase 2 sẽ bao gồm tổng hợp hành vi của cả Phase 1 và Phase 2.
- **Tổng hợp (Aggregation):** Tính tổng các chỉ số hành vi (thời gian xem, tua lại, tua nhanh, tương tác...) cho từng cặp (user_id, course_id, phase).
- **Xử lý dữ liệu thiếu:** Điền giá trị 0 cho các cột hành vi nếu người dùng không có hoạt động trong phase đó.

```
uv_prep = df_user_video_final.withColumn("event_date", F.to_date("first_local_start_time"))
joined_df = uc_expanded.alias("uc").join(
    uv_prep.alias("uv"),
    on=[
        F.col("uc.user_id") == F.col("uv.user_id"),
        F.array_contains(F.col("uc.video_list"), F.col("uv.video_id")),
        F.col("uv.event_date") <= F.col("uc.phase_deadline"),
        F.col("uv.event_date") >= F.col("uc.enroll_date")
    ],
    how="left"
)
cumulative_agg = joined_df.groupBy(
    "uc.user_id", "uc.course_id", "uc.phase"
).agg(
    F.first("uc.num_videos").alias("num_videos"),
    F.first("uc.total_video_duration").alias("total_video_duration"),
    F.countDistinct("uv.video_id").alias("num_watched_in_course"),
    F.sum("uv.actual_watch_time").alias("sum_actual_watch_time"),
    F.sum("uv.weighted_watch_time").alias("sum_weighted_watch_time"),
    F.sum("uv.interaction_count").alias("sum_interactions"),
    F.sum("uv.session_count").alias("sum_sessions"),
    F.sum("uv.speed_change_count").alias("sum_speed_changes"),
    F.sum("uv.fast_forward_time").alias("sum_fast_forward"),
    F.sum("uv.rewind_time").alias("sum_rewind"),
    F.max("uv.max_watch_point").alias("max_watch_point_agg"),
    F.min("uv.first_local_start_time").alias("first_watch_time"),
    F.first("uc.enroll_date").alias("enroll_date_ref")
)
fill_zeros_cols = [
    "num_watched_in_course", "sum_actual_watch_time", "sum_weighted_watch_time",
    "sum_interactions", "sum_sessions", "sum_speed_changes",
    "sum_fast_forward", "sum_rewind", "max_watch_point_agg"
]
cumulative_agg = cumulative_agg.fillna(0, subset=fill_zeros_cols)
```

e/ Xử lý giá trị ngoại lệ và Hoàn thiện bộ dữ liệu (Handling Edge Cases & Data Finalization)

Để đảm bảo bộ dữ liệu đầu ra sạch và sẵn sàng cho mô hình huấn luyện, nhóm thực hiện các bước xử lý kỹ thuật cuối cùng nhằm giải quyết các trường hợp ngoại lệ (như chia cho 0) và định dạng lại dữ liệu.

- **Phép chia an toàn (Safe Division):** Sử dụng hàm lambda nz (Not Zero) để xử lý các trường hợp mẫu số bằng 0 hoặc Null. Nếu học viên chưa xem video nào (mẫu số = 0), giá trị tỷ lệ sẽ được trả về là Null thay vì gây lỗi chương trình, sau đó sẽ được điền 0 ở bước tiếp theo.
- **Tính toán các chỉ số chi tiết:** Bổ sung thêm các chỉ số về tần suất tương tác (interactions_per_video), số phiên xem trung bình (sessions_per_video), và độ trễ khi bắt đầu học (days_until_first_watch).
- **Làm sạch và Định dạng:**
    - Sử dụng .fillna(0) cho các cột metric để đảm bảo không còn giá trị Null nào đi vào mô hình.
    - Sử dụng hàm F.round(col, 2) để làm tròn các chỉ số thập phân, giúp giảm kích thước lưu trữ và chuẩn hóa độ chính xác.
- **Sắp xếp dữ liệu:** Kết quả cuối cùng được sắp xếp theo user_id, course_id và phase để đảm bảo tính thứ tự của chuỗi thời gian.

```
nz = lambda c: F.when((F.col(c).isNull()) | (F.col(c) == 0), F.lit(None)).otherwise(F.col(c))

result_df = cumulative_agg.withColumn("coverage_ratio", F.col("num_watched_in_course") / nz("num_videos")) \
    .withColumn("course_watch_ratio", F.col("sum_actual_watch_time") / nz("total_video_duration")) \
    .withColumn("weighted_watch_ratio", F.col("sum_weighted_watch_time") / nz("total_video_duration")) \
    .withColumn("interactions_per_video", F.col("sum_interactions") / nz("num_watched_in_course")) \
    .withColumn("sessions_per_video", F.col("sum_sessions") / nz("num_watched_in_course")) \
    .withColumn("speed_changes_per_session", F.col("sum_speed_changes") / nz("sum_sessions")) \
    .withColumn("fast_forward_per_watch", F.col("sum_fast_forward") / nz("sum_actual_watch_time")) \
    .withColumn("rewind_per_watch", F.col("sum_rewind") / nz("sum_actual_watch_time")) \
    .withColumn("max_watch_point_ratio", F.col("max_watch_point_agg") / nz("total_video_duration")) \
    .withColumn("days_until_first_watch", F.datediff(F.col("first_watch_time"), F.col("enroll_date_ref")))
metric_cols = [
    "coverage_ratio", "course_watch_ratio", "weighted_watch_ratio",
    "interactions_per_video", "sessions_per_video", "speed_changes_per_session",
    "fast_forward_per_watch", "rewind_per_watch", "max_watch_point_ratio"
]
result_df = result_df.fillna(0, subset=metric_cols)
from pyspark.sql.types import DoubleType, FloatType

final_cols = ["user_id", "course_id", "phase"] + metric_cols + ["days_until_first_watch"]
exprs = []
for col_name in final_cols:
    if col_name in metric_cols:
        exprs.append(F.round(F.col(col_name), 2).alias(col_name))
    else:
        exprs.append(F.col(col_name))

result_final = result_df.select(*exprs).orderBy("user_id", "course_id", "phase")
result_final = result_final.fillna({"days_until_first_watch": -1})
```

Kết quả: thu được bộ dữ liệu tương tác video sẵn sàng để tích hợp vào mô hình dự đoán. Dữ liệu cuối cùng tổng hợp toàn bộ hành vi tương tác với video của một học viên trong một khóa học.

2.4. Dữ liệu về các tương tác comment

Dữ liệu bình luận (comment) đóng vai trò quan trọng trong việc đánh giá thái độ và mức độ quan tâm của học viên đối với khóa học. Để đồng bộ với dữ liệu hành vi học tập và đảm bảo tính chính xác cho mô hình dự đoán, quy trình xử lý dữ liệu tương tác comment được thực hiện qua các bước kỹ thuật sau:

a. Hợp nhất và làm sạch dữ liệu

Dữ liệu gốc bao gồm nội dung bình luận, nhãn cảm xúc (sentiment) và thời gian tạo. Nhóm thực hiện việc nối (join) bảng dữ liệu này với bảng thông tin đăng ký (Enrollment) dựa trên khóa chính là cặp (user_id,course_id). Bước này nhằm mục đích

chỉ giữ lại các tương tác thuộc về học viên thực sự của khóa học, đồng thời loại bỏ các dữ liệu nhiễu hoặc không xác định được ngữ cảnh. Sau đó, dữ liệu được tiền xử lý bằng cách chuẩn hóa định dạng thời gian và loại bỏ các trường thông tin không cần thiết để tối ưu hóa hiệu năng xử lý.

b. Phân chia giai đoạn (Phase Splitting)

Để quan sát sự thay đổi hành vi theo thời gian, dữ liệu bình luận cần được ánh xạ vào 4 giai đoạn học tập (Phase 1 - Phase 4) tương ứng với tiến độ của từng học viên.

Thay vì sử dụng các công thức toán học phức tạp, nhóm áp dụng quy tắc gán nhãn dựa trên mốc thời gian thực tế: Mỗi bình luận sẽ được đối chiếu với 4 mốc thời gian (phase_end_date) được cá nhân hóa theo ngày đăng ký của từng học viên. Hệ thống sẽ duyệt tuần tự: nếu thời điểm tạo bình luận (create_time) xảy ra trước hoặc đúng vào ngày kết thúc của một giai đoạn, bình luận đó sẽ được gán nhãn thuộc giai đoạn tương ứng.

Cụ thể:

- **Phase 1:** Được ghi nhận nếu thời gian tạo bình luận nằm trong khoảng từ ngày bắt đầu đến ngày kết thúc của giai đoạn 1.
- **Phase 2, 3, 4:** Được xác định tương tự, nếu thời gian tạo bình luận lớn hơn ngày kết thúc của giai đoạn liền trước và nhỏ hơn hoặc bằng ngày kết thúc của giai đoạn đang xét.

Những tương tác phát sinh ngoài khung thời gian 4 giai đoạn tiêu chuẩn (ví dụ sau khi khóa học đã kết thúc) sẽ được gán nhãn "Out of range" và được lọc bỏ để đảm bảo tính tập trung của dữ liệu huấn luyện.

```
# ================================================================
# 3. XỬ LÝ CHIA PHASE DỰA TRÊN CỘT END_DATE
# ================================================================
print("\n🔄 Đang xử lý phân loại Phase dựa trên mốc thời gian thực tế...")

# Lưu ý: Cần dùng to_date() cho create_time để loại bỏ giờ phút giây trước khi so sánh
df_processed = df.withColumn("phase",
    when(to_date(col("create_time")) <= col("phase_1_end_date"), "phase 1")
    .when(to_date(col("create_time")) <= col("phase_2_end_date"), "phase 2")
    .when(to_date(col("create_time")) <= col("phase_3_end_date"), "phase 3")
    .when(to_date(col("create_time")) <= col("phase_4_end_date"), "phase 4")
    .otherwise("Out of range") # Trường hợp nằm ngoài thời gian học (sau phase 4)
)
```

c. Tính toán chỉ số tích lũy (Cumulative Statistics)

Một thách thức lớn của dữ liệu chuỗi thời gian là tính rời rạc. Để mô hình nắm bắt được lịch sử hoạt động của học viên, nhóm áp dụng kỹ thuật tính tổng dồn (Cumulative Sum) thay vì chỉ đếm số lượng comment tại từng phase riêng lẻ.

Sử dụng kỹ thuật Window functions trong PySpark, nhóm đã tạo ra các đặc trưng tích lũy:

- **sent_1_count, sent_2_count, sent_3_count:** Tổng số lượng bình luận theo từng loại nhãn cảm xúc (Tiêu cực, Trung tính, Tích cực) tính từ lúc bắt đầu học đến hết giai đoạn hiện tại.
- **total_comments:** Tổng số tương tác bình luận tích lũy.

Việc này giúp dữ liệu tại Phase 4 sẽ chứa đựng thông tin tổng hợp của cả quá trình học tập trước đó.

```
# 4) Chuẩn hoá phase và sentiment
# Lọc bỏ các dòng "Out of range" hoặc "N/A" trước khi xử lý
df = (
    df_raw
    .withColumn("phase_int",
                F.regexp_extract(F.col("phase").cast("string"), r"(\d+)", 1).cast("int"))
    .withColumn("sentiment_int", F.col("sentiment").cast("int"))
    .filter(F.col("phase_int").isin(1, 2, 3, 4) & F.col("sentiment_int").isin(1, 2, 3))
)

# 5) Đếm theo (user_id, course_id, phase) và pivot theo sentiment
# Bước này đếm số lượng comment tại CHÍNH phase đó (chưa cộng dồn)
by_keys = ["user_id", "course_id", "phase_int"]
df_counts = (
    df.groupBy(*by_keys)
      .pivot("sentiment_int", [1, 2, 3])
      .agg(F.count(F.lit(1)))
      .na.fill(0)
      .withColumnRenamed("1", "sent_1_count_raw")
      .withColumnRenamed("2", "sent_2_count_raw")
      .withColumnRenamed("3", "sent_3_count_raw")
)

# 6) Tính lũy kế theo phase (Cumulative Sum)
w = (Window
     .partitionBy("user_id", "course_id")
     .orderBy("phase_int")
     .rowsBetween(Window.unboundedPreceding, Window.currentRow))

df_cum = (
    df_counts
    .withColumn("sent_1_count", F.sum("sent_1_count_raw").over(w))
    .withColumn("sent_2_count", F.sum("sent_2_count_raw").over(w))
    .withColumn("sent_3_count", F.sum("sent_3_count_raw").over(w))
    .withColumn("total_comments",
                F.col("sent_1_count") + F.col("sent_2_count") + F.col("sent_3_count"))
)
```

d. Xử lý dữ liệu khuyết thiếu (Handling Missing Data & Silent Users)

Trong thực tế, dữ liệu tương tác thường rất thưa (sparse) do hành vi không đồng đều của người dùng. Nhóm đã xử lý triệt để hai trường hợp phổ biến:

- **Điền khuyết Phase (Fill Missing Phases):** Với các học viên có hoạt động ngắt quãng (ví dụ: có comment ở Phase 1 nhưng không comment ở Phase 2), hệ thống sử dụng kỹ thuật CrossJoin để tạo đủ khung 4 giai đoạn cho mỗi người dùng. Sau đó, áp dụng phương pháp Forward Fill: giá trị tích lũy của giai đoạn trước sẽ được

mang sang giai đoạn sau (ví dụ: nếu Phase 1 có 2 comment và Phase 2 không có hoạt động mới, Phase 2 vẫn ghi nhận tích lũy là 2 comment).

- **Xử lý người dùng "im lặng" (Silent Users):** Nhóm phát hiện một lượng lớn học viên có trong danh sách đăng ký nhưng hoàn toàn không có dữ liệu trong bảng comment. Bằng kỹ thuật Left Anti Join, nhóm đã xác định được danh sách này và bổ sung đầy đủ các bản ghi với giá trị 0 cho tất cả các chỉ số ở cả 4 giai đoạn.

```
# ============================================================
# 4) Kỹ thuật CrossJoin: Tạo đủ 4 phase cho mọi User
# ============================================================
phases_df = spark.createDataFrame([(1,), (2,), (3,), (4,)], ["phase"])
user_course_df = df.select("user_id", "course_id").distinct()

# Tạo lưới: Mỗi User x 4 Phase
full_grid = user_course_df.crossJoin(phases_df)

# Join dữ liệu thực tế vào lưới
joined = full_grid.join(df, on=["user_id", "course_id", "phase"], how="left")

# ============================================================
# 5) Forward Fill: Điền giá trị lũy kế cho ô trống
# ============================================================
w = (Window.partitionBy("user_id", "course_id")
           .orderBy("phase")
           .rowsBetween(Window.unboundedPreceding, 0))

def ffill_from_prev(colname):
    # Logic:
    # 1. Nếu là Phase 1 mà Null -> Nghĩa là chưa có comment -> Gán 0
    # 2. Nếu không phải Phase 1 mà Null -> Giữ nguyên Null để hàm last() xử lý
    # 3. Hàm last(..., ignorenulls=True) sẽ lấy giá trị gần nhất phía trước
    base = F.when((F.col("phase") == 1) & F.col(colname).isNull(), F.lit(0)) \
            .otherwise(F.col(colname))
    return F.last(base, ignorenulls=True).over(w)

filled = (joined
          .withColumn("sent_1_count", ffill_from_prev("sent_1_count"))
          .withColumn("sent_2_count", ffill_from_prev("sent_2_count"))
          .withColumn("sent_3_count", ffill_from_prev("sent_3_count"))
          .withColumn("total_comments", ffill_from_prev("total_comments")))
```

```
# ==================== FIND MISSING PAIRS ====================
print("\n🔍 Đang tìm kiếm các user đăng ký nhưng KHÔNG comment...")
# Lấy danh sách (user, course) đã có trong bảng thống kê
stats_pairs = stats_df.select("user_id","course_id").distinct()

# Tìm những cặp có trong Enroll nhưng KHÔNG có trong Stats (Left Anti Join)
missing_pairs = enroll_df.join(stats_pairs, on=["user_id","course_id"], how="left_anti")

count_missing = missing_pairs.count()
print(f"   -> Phát hiện {count_missing} trường hợp user 'im lặng' (cần bổ sung 4 dòng zero).")

# ==================== BUILD 4 PHASE ROWS FOR MISSING ====================
if count_missing > 0:
    # Tạo dataframe mẫu 4 phase
    phases_df = spark.createDataFrame([(1,), (2,), (3,), (4,)], ["phase"])

    # CrossJoin để tạo đủ 4 dòng cho mỗi user thiếu, gán toàn bộ count = 0
    missing_rows = (missing_pairs.crossJoin(phases_df)
                    .withColumn("sent_1_count", F.lit(0).cast("long"))
                    .withColumn("sent_2_count", F.lit(0).cast("long"))
                    .withColumn("sent_3_count", F.lit(0).cast("long"))
                    .withColumn("total_comments", F.lit(0).cast("long"))
                    .select("user_id","course_id","phase",
                            "sent_1_count","sent_2_count","sent_3_count","total_comments"))

    # Union với dữ liệu gốc
    final_df = stats_df.select("user_id","course_id","phase",
                               "sent_1_count","sent_2_count","sent_3_count","total_comments") \
                       .unionByName(missing_rows)
else:
    print("   -> Không có user nào thiếu, giữ nguyên data.")
    final_df = stats_df

# ==================== SAVE TO DRIVE ====================
print(f"\n💾 Đang lưu file xuống: {out_file}")
```

Kết quả: Bộ dữ liệu cuối cùng đảm bảo tính toàn vẹn và liên tục: mỗi cặp (user_id, course_id) luôn có đủ 4 dòng dữ liệu tương ứng với 4 giai đoạn. Dữ liệu này phản ánh chính xác hành trình tương tác tích lũy của học viên, sẵn sàng cho việc hợp nhất với các nhóm dữ liệu khác để đưa vào mô hình dự đoán.

3. Gán nhãn dữ liệu

3.1. Cấu trúc dữ liệu đầu vào

Dữ liệu gán nhãn được trích xuất từ 5 bảng chính:

- **User:** id, list_course_order
- **User_problem:** log_id, user_id, problem_id, is_correct, attempts, submit_time
- **Problem:** problem_id, exercise_id, context_id
- **Comment:** id, user_id, text, create_time
- **Course:** id, list_resource_id

Ngoài ra còn sử dụng thêm một số bảng relation để nối các bảng lại với nhau, từ đó trích xuất ra các đặc trưng cần thiết cho việc gán nhãn. Để gán nhãn cho từng cặp khóa (user_id, course_id), nhóm sử dụng các thông tin đặc trưng để tạo ra chỉ số đánh giá mức độ hài lòng. Các thông tin đặc trưng được chia thành các nhóm sau:

- **Học tập (L - Learning):** Hiệu quả học tập.
- **Cảm xúc (S - Sentiment):** Bình luận, thái độ.
- **Ngữ cảnh khóa học (C - Course Context):** Các tài nguyên của khóa học.
- **Tính đều đặn (T - Time Regularity)**: Mức độ chăm chỉ và duy trì.

Bảng tổng hợp thông tin input để gán nhãn:

Nhóm	Tên cột	Ý nghĩa
Thông tin tĩnh user và tài nguyên khóa học	user_id	Mã học viên
	course_id	Mã khóa học
	is_preresiquites	Có môn học tiên quyết hay không (0/1)
	num_fields	Số lượng lĩnh vực liên quan
	course_total_exercises	Số lượng exercise khóa học cung cấp
	course_total_videos	Số lượng video khóa học cung cấp
	total_students_enrolled	Số lượng học viên đã tham gia khóa học

Thành phần cảm xúc (comment)	total_comments	Tổng số lượng bình luận của học viên ở khóa học
	sentiment_1_count	Tổng số lượng bình luận mang tính tiêu cực
	sentiment_2_count	Tổng số lượng bình luận mang tính trung tính
	sentiment_3_count	Tổng số lượng bình luận mang tính tích cực
	ratio_sentiment_1	Tỉ lệ bình luận tiêu cực
	ratio_sentiment_2	Tỉ lệ bình luận trung tính
	ratio_sentiment_1	Tỉ lệ bình luận tích cực
Thành phần học tập (problems)	total_exerercises	Số lượng exercise mà học viên đã làm
	total_problems	Số lượng problem mà học viên đã làm
	total_attempts	Tổng số lần nộp bài
	total_earned_score	Tổng điểm học viên nhận được
	total_problem_score	Tổng số điểm của problem trong khóa học
	avg_earned_ratio	Tỉ lệ điểm trung bình đạt được
	correctness_rate	Tỉ lệ làm đúng (is_correct=1)
	active_days	Số ngày hoạt động của học viên (tính theo submit_time)

	avg_gap_days	Số ngày trung bình giữa các lần submit
	start_date	Ngày đầu tiên học viên submit
	end_date	Ngày cuối cùng học viên submit
	total_duration_days	Khoảng cách giữa ngày cuối và ngày đầu tiên submit

3.2. Chiến lược gán nhãn

3.2.1. Tổng quan

Mục tiêu của việc gán nhãn là tổng hợp các hành vi và tương tác của người học thành một chỉ số "satisfaction_score" duy nhất, được lượng hóa trên thang điểm từ 0 đến 1 và phân loại thành 5 cấp độ (1 đến 5). Chỉ số này được xây dựng dựa trên 4 trụ cột chính:

- Học tập (L - Learning): Hiệu quả học tập.
- Cảm xúc (S - Sentiment): Cảm nhận, thái độ.
- Ngữ cảnh khóa học (C - Course Context): Độ phức tạp của khóa học.
- Tính đều đặn (T - Time Regularity): Mức độ chăm chỉ và duy trì.

Công thức tổng quát:

$$\textbf{satisfaction\_score} = wL * L + wS * S + wC * C + wT * T$$

Trong đó wL + wS + wC + wT = 1. Điểm số cuối cùng được ánh xạ thành nhãn thông qua các ngưỡng xác định.

3.2.2. Thành phần học tập (L - Learning)

Nhóm chỉ số L phản ánh mức độ hiệu quả và nỗ lực học tập của người học. Các học viên có tỷ lệ đúng cao, ít thử lại nhiều lần, và duy trì hoạt động ổn định thường thể hiện sự hài lòng và thành công cao hơn.

a/ Các chỉ số đầu vào:

- correctness_rate: tỷ lệ câu trả lời đúng.
- avg_earned_ratio: tỷ lệ điểm trung bình đạt được trên tổng điểm.
- avg_attempts: số lần thử trung bình cho mỗi câu hỏi.
- active_days: số ngày học viên có hoạt động làm bài.
- total_problems và course_total_exercises: dùng để tính độ tin cậy của hành vi học tập

b/ Quy trình xử lý:

Tất cả các chỉ số được chuẩn hóa Min-Max về khoảng [0, 1]. Chỉ số avg_attempts được đảo chiều (avg_attempts_inv = 1 - avg_attempts_norm) vì số lần nộp bài ít hơn thường ám chỉ việc nắm vững kiến thức tốt hơn.

- **Cách tính độ tin cậy của hoạt động học tập:** Thay vì coi tất cả học viên như nhau, mô hình bổ sung thêm một chỉ số độ tin cậy (reliability), thể hiện mức độ bao quát và cường độ của việc làm bài và được xác định dựa trên hai yếu tố chính:
    - **Độ phủ bài tập (exercise coverage):** phản ánh học viên đã hoàn thành bao nhiêu bài tập so với tổng số bài trong khóa:

      coverage = total_exercises/course_total_exercises

    - **Cường độ làm bài (problems per exercise):** thể hiện trung bình mỗi bài tập có bao nhiêu câu hỏi mà học viên đã thực hiện.

      intensity = total_problem/total_exercises

Từ đó, độ tin cậy tổng hợp (reliability) được tính theo công thức:

reliability = 0.7*coverage + 0.3*intensity

Cuối cùng tính chỉ số L qua công thức:

L = ((0.5 + 0.5 * reliability) * (0.5 * correctness_norm + 0.5 * earned_ratio_norm)
+ (0.5 * reliability) * avg_attempts_inv)

- Khi độ tin cậy thấp (học viên làm ít bài), trọng số được dồn vào correctness_rate và avg_earned_ratio - những chỉ số phản ánh trực tiếp chất lượng.
- Khi độ tin cậy cao, chỉ số avg_attempts_inv (phản ánh sự thuần thục) được đưa vào với trọng số cao hơn.
- Cách tiếp cận này giúp điểm số phản ánh chính xác hơn cho cả những học viên mới tham gia.

3.2.3. Thành phần bình luận (S - Sentiment)

Phản ánh trải nghiệm cảm xúc của học viên thông qua nội dung bình luận. Các bình luận tích cực cho thấy sự hài lòng, trong khi tiêu cực thể hiện vấn đề về nội dung hoặc phương pháp giảng dạy.

a/ Các chỉ số đầu vào:

- sentiment_1_count, sentiment_2_count, sentiment_3_count (tiêu cực, trung tính, tích cực)
- total_comments: tổng số bình luận

b/ Quy trình xử lý:

- **Chuẩn hóa và tính toán:** Chênh lệch giữa cảm xúc tích cực và tiêu cực được tính: sentiment_diff = ratio_sentiment_3 - ratio_sentiment_1. Giá trị này được chuyển đổi từ thang [-1, 1] sang [0, 1] để tạo thành S_scaled.
- **Độ tin cậy của cảm xúc (comment_confidence):** Không phải học viên nào cũng bình luận. Độ tin cậy của chỉ số S được tính dựa trên total_comments bằng cách so sánh logarit của số lượng bình luận với phân vị 95. Học viên không có bình luận sẽ có độ tin cậy bằng 0.

Nhận xét: Độ tin cậy này không ảnh hưởng trực tiếp đến giá trị S, mà sẽ ảnh hưởng đến trọng số (wS) của thành phần này trong công thức cuối cùng.

3.2.4. Tài nguyên khóa học (C - Course resource)

Khóa học có nhiều chủ đề, nhiều tài nguyên và có điều kiện tiên quyết thường đòi hỏi nhiều nỗ lực hơn, ảnh hưởng đến cảm nhận hài lòng.

a/ Các chỉ số đầu vào:

- num_fields: Số lĩnh vực kiến thức.
- is_prerequisites: Có điều kiện tiên quyết hay không.
- total_videos: Tổng số video.
- course_total_exercises: Tổng số bài tập.

b/ Quy trình xử lý:

- Tất cả chỉ số được chuẩn hóa Min-Max và kết hợp thành một chỉ số complexity_comp duy nhất, thể hiện độ phức tạp tổng thể.
- Trọng số cố định là **0.15**, đảm bảo yếu tố khóa học có ảnh hưởng vừa phải, không lấn át hành vi học viên. (Khóa học càng nhiều tài nguyên thì khóa học càng phức tạp -> C giảm)

3.2.5. Thành phần thời gian (T - Time)

Thành phần này đánh giá mức độ duy trì và tính kiên trì trong học tập, một yếu tố quan trọng phản ánh sự tận tâm.

a/ Các chỉ số đầu vào:

- total_attempts: Tổng số lần nộp bài.
- active_days: Số ngày hoạt động.
- avg_gap_days: Khoảng cách ngày trung bình giữa các lần nộp bài.

b/ Quy trình xử lý:

- **Mật độ nộp bài (attempt_density_norm):** Được tính bằng total_attempts / (active_days + 1), phản ánh số lần nộp bài trung bình mỗi ngày. Giá trị này càng cao càng tốt.
- **Tính liên tục (gap_inv):** Chỉ số avg_gap_days được chuẩn hóa và đảo chiều. Khoảng cách giữa các lần nộp bài càng nhỏ chứng tỏ học viên càng học tập đều đặn.
- **Tính toán T:** Hai chỉ số trên được kết hợp với trọng số nghiêng về mật độ nộp bài: T = 0.7 * attempt_density_norm + 0.3 * gap_inv

3.2.6. Trọng số các yếu tố

- **Trọng số cơ sở:** (wL_base, wS_base, wC_base, wT_base) = (0.60, 0.15, 0.15, 0.10). Thành phần học tập (L) được ưu tiên cao nhất.
- **Điều chỉnh trọng số động:**
    - Trọng số sentiment (wS) được điều chỉnh trực tiếp bằng độ tin cậy của cảm xúc (comment_confidence). Nếu học viên không có hoặc có ít bình luận, wS sẽ giảm.
    - Phần trọng số bị giảm này (redistribute = wS_base - wS) được phân bổ lại:

        - 80% chuyển sang trọng số Học tập (wL), vì hiệu quả học tập là thước đo quan trọng và đáng tin cậy nhất.
        - 20% chuyển sang trọng số Thời gian (wT), vì sự chăm chỉ có thể phần nào phản ánh sự hài lòng.
    - Trọng số C (Course Context) được giữ nguyên.
- **Chuẩn hóa cuối cùng:** Tổng các trọng số được đảm bảo bằng 1 sau điều chỉnh.

Sau khi tính được satisfaction_score cuối cùng, điểm số được ánh xạ thành 5 nhãn rời rạc bằng cách sử dụng các ngưỡng cố định:

- **Nhãn 1 (Rất không hài lòng):** $0.0 \leq \text{score} < 0.2$
- **Nhãn 2 (Không hài lòng):** $0.2 \leq \text{score} < 0.4$
- **Nhãn 3 (Bình thường):** $0.4 \leq \text{score} < 0.6$
- **Nhãn 4 (Hài lòng):** $0.6 \leq \text{score} < 0.8$
- **Nhãn 5 (Rất hài lòng):** $0.8 \leq \text{score} \leq 1.0$

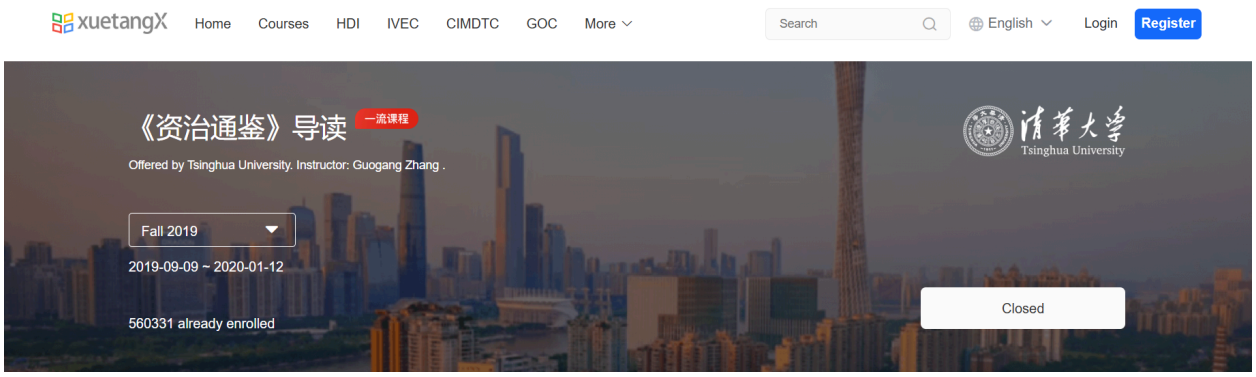
4. Chia tập dữ liệu

4.1. Xây dựng bộ dữ liệu time series

Để phục vụ mục tiêu dự đoán sớm mức độ hài lòng của học viên, nhóm tiến hành xây dựng bộ dữ liệu dạng chuỗi thời gian (time series) phản ánh quá trình học tập và tương tác của từng học viên theo thời gian. Ý tưởng cốt lõi là chia tiến trình tham gia khóa học của mỗi học viên thành 4 giai đoạn (phase) tương ứng với các khoảng thời gian khác nhau giữa ngày đăng ký khóa học (enroll_time) và ngày kết thúc khóa học (end_date).

Thời điểm bắt đầu và kết thúc của khóa học được lấy từ trang web xuetangx.com, nhóm chỉ lấy ra những khóa học có thời gian bắt đầu và kết thúc cụ thể, không sử dụng các khóa có thời gian là “self-paced” để tránh việc các học viên đăng kí khóa học không

có thời hạn kết thúc thì sẽ không chủ động việc học dẫn kết việc trích xuất đặc trưng trở nên khó khăn hơn.



Cụ thể, mỗi khóa học được chia thành 4 phase theo ngày kể từ lúc đăng ký dựa trên thời gian còn lại của người học từ khi đăng kí (remain_day) và chia thành 4 phần bằng nhau.

- **Phase 1:** 0-25% thời gian remain_day (từ enroll_date)
- **Phase 2:** 0-50% thời gian remain_day
- **Phase 3:** 0-75% thời gian remain_day
- **Phase 4:** 0-100% thời gian remain_day (đến end_date của khóa học)

Các giai đoạn có độ dài thời gian linh hoạt dựa trên khoảng thời gian đăng kí của từng học viên và thời gian kết thúc của khóa học, nhằm mô phỏng quá trình học tập theo từng giai đoạn của khóa học. Mỗi phase sẽ tổng hợp các đặc trưng hành vi học tập của học viên dựa trên dữ liệu làm bài tập, xem video và các tương tác bình luận của học viên trên khóa học đó.

Dữ liệu được lọc tiếp tục chỉ giữ ra các cặp khóa (user_id, course_id) có ít nhất một hoạt động trong thời gian học (làm bài tập, xem video hoặc bình luận). Dữ liệu sau cùng có 1350468 dòng (337617 cặp user_course) với 30 features.

	user_id	course_id	phase	exercises_touched	problems_done	total_attempts	correct_submissions	total_earned_score	avg_earned_score	avg_earned_ratio	...	fast_forward_
0	U_1000902	C_697821	1	0	0	0	0	0.0	NaN	0.0000	...	
1	U_1000902	C_697821	2	0	0	0	0	0.0	NaN	0.0000	...	
2	U_1000902	C_697821	3	0	0	0	0	0.0	NaN	0.0000	...	
3	U_1000902	C_697821	4	0	0	0	0	0.0	NaN	0.0000	...	
4	U_1000982	C_947149	1	9	42	42	7	5.6	0.133333	0.1875	...	
...	...	...	...	...	...	...	...	...	...	...	...	
1350463	U_998604	C_934535	4	0	0	0	0	0.0	NaN	0.0000	...	
1350464	U_999821	C_881485	1	0	0	0	0	0.0	NaN	0.0000	...	
1350465	U_999821	C_881485	2	0	0	0	0	0.0	NaN	0.0000	...	
1350466	U_999821	C_881485	3	0	0	0	0	0.0	NaN	0.0000	...	
1350467	U_999821	C_881485	4	0	0	0	0	0.0	NaN	0.0000	...	

1350468 rows × 30 columns

Sau khi hoàn thành bước làm sạch và lọc để giữ lại 337,617 cặp khóa học-học viên có tương tác thực tế (tổng cộng 1,350,468$ dòng), dữ liệu hiện tại đang ở định dạng dài (Long Format), trong đó mỗi cặp (user_id, course_id) được biểu diễn bằng 4 dòng, tương ứng với 4 giai đoạn học tập (phase 1 đến phase 4).

Để chuẩn bị dữ liệu này cho các mô hình học máy truyền thống (như LGBM, XGBoost, SVR…), cần thực hiện bước xoay trục (pivot) dữ liệu thành định dạng rộng (Wide Format). Mục tiêu của việc xoay trục là biến đổi dữ liệu từ 1,350,468 dòng thành 337,617 dòng duy nhất và 110 cột (features), nơi mỗi dòng là một mẫu độc lập đại diện cho một cặp khóa học của một học viên, sẵn sàng để huấn luyện.

	user_id	course_id	gender	total_courses_enrolled	total_students_enrolled	total_videos	total_exercises	num_fields	is_prerequisites	accuracy_p1	...	total_comments_p4	total_earned_score_p1	total_earned_score_p2	total_earned_
0	U_1000902	C_697821	1.0	20	44600	138	69	1	0	0.000000	...	1	0.0	0.0	
1	U_1000982	C_947149	0.0	1	14716	20	14	0	1	0.166667	...	0	5.6	5.6	
2	U_1002814	C_808526	0.0	2	18110	21	18	0	0	0.000000	...	1	0.0	0.0	
3	U_100294	C_682442	0.0	19	10909	113	75	1	0	0.000000	...	1	0.0	0.0	
4	U_10030628	C_936971	1.0	1	231674	38	36	0	0	0.000000	...	12	0.0	0.0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
337612	U_99753	C_1428968	1.0	1	23478	49	5	0	1	0.820000	...	7	0.0	0.0	
337613	U_99772	C_1903985	2.0	7	19649	66	7	0	0	0.838384	...	3	122.0	129.0	
337614	U_9980550	C_936971	1.0	2	231674	38	36	0	0	0.000000	...	12	0.0	0.0	
337615	U_998604	C_934535	0.0	1	11390	35	34	0	1	0.000000	...	5	0.0	0.0	
337616	U_999821	C_881485	0.0	13	16165	30	15	0	1	0.000000	...	1	0.0	0.0	

337617 rows × 110 columns

Bảng thống kê số lượng nhãn:

- **Bộ dữ liệu gốc**

Số dòng	Số feature (không tính các thông tin tĩnh)	Phân phối nhãn
337617	100	1: 6.9% (23123)
		2: 85.9% (290041)
		3: 2.8% (9411)
		4: 2.2% (7359)
		5: 2.3% (7674)

- **Bộ dữ liệu làm giàu bằng Node2Vec**

Số dòng	Số feature (không tính các thông tin tĩnh)	Phân phối nhãn
308936	116	1: 20597
		2: 265184
		3: 8707
		4: 7027
		5: 7421

- **Bộ dữ liệu xử lý bằng SMOTE**

Số dòng	Số feature (không tính các thông tin tĩnh)	Phân phối nhãn
638040	100	1: 87000
		2: 290040
		3: 2.87000
		4: 87000
		5: 87000

- **Bộ dữ liệu kết hợp giữa SMOTE và Node2Vec**

Số dòng	Số feature (không tính các thông tin tĩnh)	Phân phối nhãn
583404	116	1: 79555
		2: 265184
		3: 79555
		4: 79555
		5: 79555

Tổng hợp các feature của bộ dữ liệu:

Nhóm	Tên cột	Ý nghĩa
Thông tin tĩnh user và tài nguyên khóa học	user_id	Mã học viên
	course_id	Mã khóa học
	gender	Giới tính học viên (0,1,2)
	total_courses_enrolled	Tổng số lượng khóa học mà học viên đã đăng kí
	is_preresiquites	Có môn học tiên quyết hay không (0/1)
	num_fields	Số lượng lĩnh vực liên quan
	course_total_exercises	Số lượng exercise khóa học cung cấp

	course_total_videos	Số lượng video khóa học cung cấp
	total_students_enrolled	Số lượng học viên đã tham gia khóa học
Thành phần cảm xúc (comment)	total_comments_p{i}	Tổng số lượng bình luận của học viên ở khóa học trong giai đoạn {i}
	sent_1_count_p{i}	Tổng số lượng bình luận mang tính tiêu cực trong giai đoạn {i}
	sent_2_count_p{i}	Tổng số lượng bình luận mang tính trung tính trong giai đoạn {i}
	sent_3_count_p{i}	Tổng số lượng bình luận mang tính tích cực trong giai đoạn {i}
Thành phần học tập (problems)	exerercises_touched_p{i}	Số lượng exercise mà học viên đã làm trong giai đoạn {i}
	problems_done_p{i}	Số lượng problem mà học viên đã làm trong giai đoạn {i}
	total_attempts_p{i}	Tổng số lần nộp bài trong giai đoạn {i}
	correct_submissions_p{i}	Số lần nộp bài đúng trong giai đoạn {i}
	total_earned_score_p{i}	Tổng điểm học viên nhận được trong giai đoạn {i}
	avg_earned_score_p{i}	Điểm trung bình đạt được trên mỗi bài tập trong giai đoạn {i}

	avg_earned_ratio_p{i}	Tỉ lệ điểm trung bình đạt được trong giai đoạn {i}
	avg_problem_score_p{i}	Điểm trung bình tối đa của các bài tập mà học viên tham gia trong giai đoạn {i}
	accuracy_p{i}	Tỷ lệ chính xác khi làm bài, được tính bằng correct_submissions / total_attempts trong giai đoạn {i}
	problems_per_day_p{i}	Số lượng bài tập trung bình mà học viên làm mỗi ngày trong giai đoạn {i}
	earned_per_attempt_p{i}	Điểm trung bình đạt được trên mỗi lần làm bài trong giai đoạn {i}
Thành phần xem video (video)	Avg_speed_in_course_p{i}	Tốc độ tua trung bình mà học viên đó xem trong giai đoạn {i}
	coverage_ratio_p{i}	Tỉ lệ phần trăm số video học viên đã học trong một khóa trong giai đoạn {i}
	course_watch_ratio_p{i}	Tỉ lệ phần trăm học viên xem được dựa trên thời gian xem thực tế trong giai đoạn {i}
	weighted_watch_ratio_p{i}	Tỉ lệ phần trăm học viên xem được dựa trên thời gian xem có điều chỉnh về tốc độ tua trong giai đoạn {i}
	interactions_per_video_p{i}	Số tương tác trung bình trên mỗi video đã xem trong giai đoạn {i}

	sessions_per_video_p{i}	Số phiên xem lại trên mỗi video đã xem (nếu sessions_per_video > 1 nghĩa là học viên đã xem đi xem lại video nhiều lần) trong giai đoạn {i}
	speed_changes_per_session_p{i}	Số lần thay đổi tốc độ tua trên mỗi phiên xem trong giai đoạn {i}
	fast_forward_per_watch_p{i}	Tỷ lệ tua nhanh trên thời gian xem trong giai đoạn {i} - ~0 → ít tua nhanh - Cao (~3.39) → học viên tua nhanh nhiều (gấp 3.39 lần thời gian xem) - Cực cao (~13.01) → Học viên chỉ lướt (chủ yếu tua video)
	rewind_per_watch_p{i}	Tỷ lệ tua lại trên thời gian xem: học viên xem lại các đoạn nội dung (thường là các khái niệm khó) trong giai đoạn {i}
	max_watch_point_ratio_p{i}	Tỷ lệ mốc xem xa nhất trong giai đoạn {i}: phân biệt người học lướt và học sâu Một học viên có thể có course_watch_ratio (tổng thời gian xem) thấp, nhưng max_watch_point_ratio lại cao bất thường. Điều này xảy ra khi học viên tua nhanh (fast-forward) đến cuối

		video mà không xem toàn bộ nội dung
Các cột được tạo từ Node2Vec	combined_emb_{i} (i = {0-15})	
Label	satisfaction_label	Nhãn hài lòng (1-5)

4.2. Chia dữ liệu train/dev/test

Trước tiên, tập dữ liệu tổng thể được chia ngẫu nhiên thành ba tập độc lập:

- **Tập train - 80%:** Dùng để huấn luyện mô hình.
- **Tập dev - 10%**: Dùng để tinh chỉnh siêu tham số (Hyperparameters) và lựa chọn mô hình tốt nhất.
- **Tập test - 10%**: Dùng để đánh giá hiệu suất cuối cùng của mô hình đã chọn và đặc biệt là cho việc dự báo sớm. Tập này đóng vai trò là dữ liệu "tương lai" chưa từng được mô hình nhìn thấy.

Để mô phỏng dự báo sớm, tập test sẽ được tái cấu trúc thành các phiên bản cắt giảm đặc trưng, tương ứng với các tỷ lệ thời lượng đã hoàn thành của khóa học:

Giai đoạn	Tỉ lệ thời lượng	Đặc trưng sử dụng	Mục đích đánh giá
P1	0-25%	Các đặc trưng trong phase 1 (hậu tố _p1)	Đánh giá khả năng dự đoán rất sớm dựa trên tương tác ban đầu
P2	0-50%	Các đặc trưng trong phase 1 và 2 (hậu tố _p1 và _p2)	Đánh giá khả năng dự đoán giữa kì
P3	0-75%	Các đặc trưng trong phase 1, 2, 3 (hậu tố _p1, _p2, _p3)	Đánh giá khả năng dự đoán gần cuối

P4	0-90%	Các đặc trưng trong phase 1, 2, 3, 4 (hậu tố _p1,_p2, _p3 và _p4)	Đánh giá hiệu suất khi gần như toàn bộ dữ liệu lịch sử đã có

PHÂN TÍCH VẤN ĐỀ

1. Bối cảnh vấn đề và nhu cầu kinh doanh

Các nền tảng giáo dục trực tuyến đại chúng mở (MOOCs) và hệ thống học tập trực tuyến (LMS) đang phải đối mặt với thách thức lớn nhất là duy trì sự tương tác và đảm bảo sự hài lòng của học viên với các khóa học trên nền tảng. Mặc dù dữ liệu tương tác (như xem video, nộp bài, thảo luận) được thu thập liên tục, việc đánh giá mức độ hài lòng thường chỉ được thực hiện thông qua khảo sát ở cuối khóa học. Phương pháp đánh giá chậm trễ này khiến doanh nghiệp mất đi cơ hội can thiệp kịp thời để giữ chân những học viên có nguy cơ bỏ học hoặc không hài lòng. Vấn đề cốt lõi là làm thế nào để chuyển đổi dữ liệu lịch sử thành thông tin dự đoán có giá trị hành động ở các giai đoạn sớm của khóa học.

2. Câu hỏi nghiên cứu và mục tiêu đề tài

Đề tài này tập trung giải quyết khoảng trống thông tin giữa hành vi học tập sớm và kết quả cuối cùng. Câu hỏi nghiên cứu chính là: "Dựa trên các đặc trưng đa dạng và biến đổi theo thời gian (chuỗi thời gian, tài nguyên, nhân khẩu học) được trích xuất từ bộ dữ liệu MOOC CubeX, chúng ta có thể xây dựng mô hình AI để dự đoán mức độ hài lòng cuối cùng của học viên với độ tin cậy cao ngay sau khi họ hoàn thành 25% hoặc 50% thời lượng khóa học hay không?".

Mục tiêu của đề tài được xác định là:

- **Xây dựng mô hình dự đoán sớm (Early Prediction Model):** Sử dụng các thuật toán tiên tiến như XGBoost, LightGBM hoặc TabNet để huấn luyện trên các phiên bản dữ liệu Phase (P1, P2) nhằm đạt được hiệu suất dự đoán cao cho nhãn satisfaction_label.
- **Đánh giá hiệu suất theo thời gian:** Phân tích độ chính xác dự đoán tại các mốc thời gian để xác định thời điểm tối ưu nhất cho việc can thiệp.

- **Xây dựng giải pháp BI hỗ trợ ra quyết định:** Phát triển một Web Dashboard BI để trực quan hóa kết quả dự đoán, giúp các nhà quản lý và đội ngũ hỗ trợ học viên chuyển đổi xác suất rủi ro thành hành động can thiệp.

3. Kết quả đề tài và khả năng ứng dụng

- **Mô hình dự đoán sớm:**

Mô hình này sẽ là sản phẩm cốt lõi của đề tài. Quá trình huấn luyện sẽ tận dụng cấu trúc dữ liệu Wide Format đã được làm giàu, bao gồm các đặc trưng thống kê hành vi theo Phase. Đầu ra của mô hình là nhãn 5 mức độ hài lòng, cho phép phân loại học viên thành các nhóm rủi ro

Việc đánh giá sẽ dựa trên các chỉ số như AUC (Area Under the Curve) và Accuracy-DQ, nhằm đảm bảo mô hình không thiên vị đối với các lớp thiểu số (ví dụ: nhóm không hài lòng).

- **Web Dashboard Business Intelligence (BI):**

Web Dashboard BI đóng vai trò là giao diện ứng dụng hóa kết quả dự đoán. Chức năng chính bao gồm:

  - **Trực quan hóa tỷ lệ rủi ro:** Hiển thị tổng quan tỷ lệ học viên có nguy cơ không hài lòng theo từng khóa học
  - **Phân tích độ quan trọng của đặc trưng:** Dashboard sẽ hiển thị các đặc trưng nào đang là động lực chính của dự đoán. Thông tin này vô giá đối với đội ngũ phát triển nội dung, giúp họ xác định và cải thiện các phần yếu kém trong khóa học.
  - **Danh sách can thiệp**: Cung cấp danh sách các user_id và course_id được xếp hạng theo xác suất rủi ro, cho phép đội ngũ hỗ trợ cá nhân hóa thông điệp và thời điểm can thiệp.

- **Lợi ích trong kinh doanh:**

Việc tích hợp mô hình dự đoán sớm và Dashboard BI mang lại các lợi ích chiến lược sau:

- **Cải thiện khả năng giữ chân:** Dự đoán và can thiệp sớm trực tiếp làm giảm tỷ lệ bỏ học, từ đó tăng tỷ lệ hoàn thành khóa học và tăng mức độ hài lòng tổng thể.
- **Nâng cao hiệu quả vận hành:** Thay vì phân bổ nguồn lực chăm sóc khách hàng một cách ngẫu nhiên, doanh nghiệp có thể tập trung nguồn lực vào nhóm học viên cần thiết nhất, tối ưu hóa chi phí vận hành.
- **Tăng giá trị trọn đời (CLV):** Học viên hài lòng có xu hướng đăng ký các khóa học tiếp theo và trở thành khách hàng trung thành, trực tiếp nâng cao giá trị trọn đời của khách hàng đối với nền tảng MOOC.