

ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH
TRƯỜNG ĐẠI HỌC CÔNG NGHỆ THÔNG TIN



KHAI THÁC DỮ LIỆU VÀ ỨNG DỤNG

BÁO CÁO ĐỀ ÁN MÔN HỌC

GVHD: ThS. Nguyễn Thị Anh Thư

Lớp: CS313.P22

Nhóm 6

Họ và tên	MSSV
Nguyễn Hồng Phát	22521072
Đặng Thanh Ngân	22520929
Phạm Thanh Thảo	22521373
Đinh Hữu Phước	22521150
Lê Dương Minh Thiên	22521386

NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN

....., ngày tháng năm 2025

Người nhận xét

(Ký tên và ghi rõ họ tên)

MỤC LỤC

CHƯƠNG 1. TỔNG QUAN ĐỀ TÀI	13
1.1. Giới thiệu	13
1.2 Ý tưởng đề tài	13
1.3 Định nghĩa bài toán	14
1.4 Đối tượng và phạm vi đề tài	15
1.5 Ứng dụng của bài toán	16
1.6 Khó khăn và thách thức	16
1.7 Mục tiêu thực hiện đề tài	17
CHƯƠNG 2. CÁC CÔNG TRÌNH NGHIÊN CỨU LIÊN QUAN	18
2.1 Phân tích kết quả khảo sát	18
2.2 Hướng phát triển đề tài	25
CHƯƠNG 3. CƠ SỞ LÝ THUYẾT	28
3.1 Nền tảng lý thuyết	28
3.1.1. Hồi quy Softmax Logistic (Logistic Regression)	28
3.1.2. Cây quyết định (Decision Tree)	30
3.1.3. Random Forest	32
3.1.4. REP Tree	34
3.1.5. K-Nearest Neighbors (KNN)	36
3.1.6. SVM	38
3.1.7. Naive Bayes	40
3.1.8. LightGBM	42
3.1.9. XGBoost	44
3.1.10. CatBoost	46
3.1.11. Tabnet Classifier	48
3.1.12. Fully-Connected NN	50
3.1.13. CNN	52
3.1.14. RNN	54
3.1.15. LSTM	56
3.1.16. BiLSTM	58
3.1.17. 4-layer Stacked LSTM	60
3.1.18. Graph NN	61
3.1.19. ANN-LSTM	63
3.1.20. SNN	65
3.2. Công nghệ	67
3.2.1. React	67
3.2.2. Next.js	69
3.2.3. Tailwind	71

3.2.4. Amazon Web Services (AWS)	73
CHƯƠNG 4. PHƯƠNG PHÁP ĐỀ XUẤT	76
4.1 Các bước thực hiện tổng quan từ input đến output của bài toán	76
4.2. Thiết kế kiến trúc dữ liệu lớn có thể triển khai framework	82
4.2.1 Data Ingest (Thu thập dữ liệu)	82
4.2.2 Data Store + Catalog + Transform (Lưu trữ – Biến đổi – Gắn nhãn dữ liệu):	82
4.2.3 Data Enrich (Phân tích thông kê dữ liệu)	83
4.2.4 Tự động hóa & mở rộng truy cập	83
4.3 Thiết kế kiến trúc hệ thống	84
4.3.1 Người dùng (User)	84
4.3.2. Frontend	85
4.3.3 Backend	85
4.3.4 Luồng tương tác với hệ thống	85
CHƯƠNG 5. PHƯƠNG PHÁP THỰC NGHIỆM	86
5.1. Tìm hiểu dữ liệu	86
5.1.1. Giới thiệu chung bộ dữ liệu sử dụng	86
5.1.2. Các đặc điểm chính của bộ dữ liệu trong báo cáo	86
5.1.2.1. Triển Khai MOOCubeX	87
5.1.2.2. Trích Xuất Khái Niệm (Concept Extraction)	87
5.1.2.3. Phát hiện Mối quan hệ Tiên quyết giữa Các Khái niệm	88
5.1.2.4. Quản lý Dữ liệu (Data Curation)	89
5.1.2.5. Hành vi sinh viên (Student Behavior)	90
5.1.2.6. Mô tả tổng quan bộ dữ liệu	91
5.1.3. Mô tả sơ bộ về tập dữ liệu	94
5.1.3.1. Course resources	95
5.1.3.1.1 Course Info (entity)	95
5.1.3.1.2 Video (entity)	97
5.1.3.1.3 Problem (entity)	97
5.1.3.1.4. School (entity)	99
5.1.3.1.5. Teacher (entity)	99
5.1.3.1.6 Course - Field (relation)	100
5.1.3.1.7. Course - School (relation)	101
5.1.3.1.8. Course - Teacher (relation)	101
5.1.3.1.9. Exercise - Problem (relation)	101
5.1.3.1.10. Video ID - CCID (relation)	101
5.1.3.2. Student behaviors.	101
5.1.3.2.1. Student profile (entity)	101
5.1.3.2.2. Comment (entity)	102

5.1.3.2.3. Reply (entity)	103
5.1.3.2.4. User-video (relation)	104
5.1.3.2.5. User-problem	105
5.1.3.2.6. User-xiaomu	106
5.1.3.2.7. Course-comment	106
5.1.3.3. Concepts	107
5.1.3.3.1. Concept	107
5.1.3.3.2. Other	108
5.1.3.3.3. Paper	109
5.1.3.3.4. Concept-Other	110
5.1.3.3.5. Concept-Paper	110
5.1.3.3.6. Concept-Problem	110
5.1.3.3.7. Concept-Video	111
5.1.3.3.8. Concept-Comment	111
5.1.3.4. Prerequisites	111
5.1.3.4.1. CS.json	111
5.1.3.4.2. Math.json	113
5.1.3.4.3. Psy.json	113
5.1.4. Nhận xét bộ dữ liệu và dự đoán mục tiêu sử dụng của bộ dữ liệu	113
5.1.4.1. Nhận xét	113
5.1.4.2. Dự đoán mục tiêu sử dụng bộ dữ liệu	114
5.2. Chuẩn bị dữ liệu	115
5.2.1. Dữ liệu thực nghiệm	115
5.2.2. Khám phá và trích xuất dữ liệu	115
5.2.2.1. Xử lý bộ dữ liệu entities/user.json	115
5.2.2.1.1. Thông kê mô tả về bộ dữ liệu	115
5.2.2.1.2. Xử lý các cột dữ liệu bị thiếu	116
5.2.2.1.3. Trích xuất thông tin từ các cột dạng danh sách	120
a. Thêm tổng số lượng khóa học cho mỗi người dùng	120
b. Tách thông tin ngày, tháng, năm và thời gian từ cột enroll_time	
121	
c. Tính khoảng cách (số ngày) giữa các lần đăng ký khóa học của từng học sinh	122
d. Thêm cột giá trị nhỏ nhất, giá trị lớn nhất, trung bình của mỗi lần đăng ký khóa học của học sinh	124
e. Phân loại thời gian đăng kí theo buổi “sáng”, “chiều”, “tối”	125
f. Thêm cột giá trị trung bình cho tất cả các tháng mỗi người dùng tham gia & năm bắt đầu khóa học đầu tiên	
127	
g. Thêm cột ngày đầu tiên đăng kí và ngày gần nhất đăng kí khóa	

học, và khoảng cách giữa 2 ngày	128
h. Tổng kết các cột sau khi trích xuất thông tin từ list	130
5.2.2.1.4. Tính toán các thống kê số liệu cơ bản	132
5.2.2.1.5. Phân tích phân phối dữ liệu bằng cách sử dụng biểu đồ	135
5.2.2.1.6. Xác định các giá trị ngoại lai (outlier)	137
5.2.2.1.7. Tỷ lệ phần trăm NaN trong từng cột	140
5.2.2.1.8. Phân phối giới tính trong thời gian khảo sát	143
5.2.2.1.9. Phần trăm số lượng học sinh tham gia trong các khoảng thời gian	143
5.2.2.1.10. Phân phối số lượng khóa học của người dùng	146
5.2.2.1.11. Các mối quan hệ trong bộ dữ liệu	148
5.2.2.1.12. Phân phối số lượng khóa học được đăng ký	153
5.2.2.2. Xử lý bộ dữ liệu entities/course.json	161
5.2.2.2.1 Thông kê mô tả về bộ dữ liệu	161
5.2.2.2.2 Xử lý các cột dữ liệu bị thiếu bị thiếu	161
5.2.2.2.3 Tính toán các thống kê cơ bản	163
5.2.2.2.4 Phân tích phân phối dữ liệu	164
5.2.2.2.5. Trích xuất các resourse của khóa học	168
5.2.2.2.6. Lọc những đối tượng không có exercise và video	172
5.2.2.2.7. Trích xuất những exercise là bài kiểm tra	176
5.2.2.2.8. Trích xuất trường field trong phần giới thiệu để fillna	180
5.2.2.2.9. Kết luận các trường được trích xuất	183
5.2.2.3. Xử lý bộ dữ liệu entities/problem.json	184
5.2.2.3.1. Phân phối các dạng câu hỏi trong file problem	184
5.2.2.3.2. Thông kê các dạng câu hỏi trong file problem	186
a. Câu hỏi loại 1	186
b. Câu hỏi loại 2	187
c. Câu hỏi loại 3	189
d. Câu hỏi loại 4	190
e. Câu hỏi loại 5	190
f. Câu hỏi loại 6	191
g. Câu hỏi loại 9	192
5.2.2.4. Xử lý bộ dữ liệu relations/user-problem.json	192
5.2.2.4.1. Gộp dữ liệu user-problem và problem	194
5.2.2.4.2. Đień giá trị từ bảng problem	198
5.2.2.4.3. Gộp dữ liệu user-problem với exercise và course đã được xử lý	204
5.2.2.4.3. Gom nhóm theo use_id và course_id	204
5.2.2.4.5. Phân loại những exercise là bài tập hay bài kiểm tra và tính	

điểm	206
5.2.2.4.6. Chia thành các phase cho quá trình huấn luyện	210
5.2.2.4.7. Tổng kết các đặc trưng	211
5.2.2.5. Xử lý bộ dữ liệu relation/user-video.json	214
5.2.2.5.1. Thống kê các trường dữ liệu	214
5.2.2.5.2. Phân tích Tổng quan về quy mô	214
5.2.2.5.3. Tổng thời gian xem mỗi người dùng	215
5.2.2.5.4. Phân bố tốc độ phát lại video	216
5.2.2.5.5. Tổng thời gian xem mỗi video	217
5.2.2.5.6. Số lượng segment trung bình mỗi video	217
5.2.2.5.7. Tổng số lượt xem theo ngày	218
5.2.2.5.8. Top 10 video được xem nhiều segment nhất	219
5.2.2.5.9. Lọc dữ liệu với các file cần thiết	220
5.2.2.6. Xử lý bộ dữ liệu entities/reply.json	222
5.2.2.6.1. Thống kê các trường dữ liệu	222
5.2.2.6.2. Thống kê số lượng reply theo người dùng	223
5.2.2.6.3. Thống kê reply theo thời gian	223
5.2.2.6.4. Gộp dữ liệu reply và với các file cần thiết	224
5.2.2.7. Xử lý bộ dữ liệu entities/comment.json	225
5.2.2.7.1. Thống kê các trường dữ liệu	225
5.2.2.7.2. Thống kê số lượng comment theo người dùng	225
5.2.2.7.3. Thống kê thời gian bình luận theo năm	226
5.2.2.7.4. Thống kê thời gian bình luận theo giờ trong ngày	227
5.2.2.7.5. Làm sạch dữ liệu	227
5.2.2.7.6. Gộp dữ liệu comment và với các file cần thiết	228
5.2.3. Trích xuất đặc trưng	229
5.2.3.1. Trích xuất đặc trưng từ dữ liệu comment và reply	229
5.2.3.1.1. Comment	229
a. Tiền xử lý	229
b. Trích xuất đặc trưng theo từng phase	231
5.2.3.1.2. Reply	234
a. Tiền xử lý	234
b. Trích xuất đặc trưng theo từng phase	235
5.2.3.1.3. Phân tích cảm xúc từ bình luận (comment) và phản hồi (reply)	
236	
a. Gán nhãn dữ liệu comment	237
b. Gán nhãn dữ liệu reply	239
c. Phối hợp cảm xúc từ bình luận (comment) và phản hồi (reply)	239
5.2.3.2. Trích xuất đặc trưng từ dữ liệu user-video	241

5.2.3.2.1 Tạo đặc trưng duration_seg và avg_duration_seg	241
5.2.3.2.2. Chuyển đổi và Gộp Dữ liệu theo user_id và course_of_watched_video	241
5.2.3.2.3. Tạo Đặc Trung video_watch_count	242
5.2.3.2.4. Tạo Đặc Trung total_videos_for_user và video_watched_percentage	243
5.2.3.2.5. Tạo Đặc Trung max_watch_per_video, watch_percentages, và video_percentage_watch_time	244
5.2.3.2.6. Tạo Đặc Trung video_speed_avg, video_time_between_views_avg, và video_time_between_views_std	245
5.2.3.2.7. Tạo Đặc Trung video_pause_count, video_pause_avg, và video_pause_std	247
5.2.3.2.8. Tạo Đặc Trung video_rewatch_avg và video_rewatch_std	248
5.2.3.2.9. Tạo Đặc Trung ent_seg và entropy_time	249
5.2.3.3. Trích xuất đặc trưng từ dữ liệu user-course	250
5.2.3.3.1. Trích xuất đặc trưng sẵn có từ dữ liệu course	250
5.2.3.3.2. Tạo đặc trưng start_date, end_date và duration_days.	250
5.2.3.3.3. Trích xuất đặc trưng từ resource của course.	251
5.2.3.3.4. Tạo đặc trưng thành phần điểm của khóa học.	251
5.2.3.3.5. Trích xuất đặc trưng sẵn có từ dữ liệu user	252
5.2.3.3.6. Tạo đặc trưng user_past_course_count và user_time_since_last_course	252
5.2.3.4. Trích xuất đặc trưng từ dữ liệu user-problem	253
5.2.3.4.1. Tạo đặc trưng liên quan đến số lượng và phạm vi bài tập	253
5.2.3.4.2. Tạo đặc trưng liên quan đến hiệu suất trả lời đúng	255
5.2.3.4.3. Tạo đặc trưng liên quan đến điểm số đạt được	257
5.2.3.4.4. Tạo đặc trưng liên quan đến mức độ hoàn thành	259
5.2.3.4.5. Tạo đặc trưng liên quan đến số lần làm bài	260
5.2.3.4.6. Tạo đặc trưng liên quan đến thời gian làm bài	261
5.2.3.4.7. Tạo đặc trưng liên quan đến ngôn ngữ bài tập	263
5.2.3.5. Tổng hợp toàn bộ các đặc trưng của bài toán	264
5.2.4. Tổng hợp dữ liệu	269
5.2.4.1. Nguồn dữ liệu chính:	270
5.2.4.2. Các bước hợp nhất dữ liệu:	270
5.2.4.3. Chia dữ liệu thành các tuần và giai đoạn:	273
5.2.4.4. Xử lý dữ liệu số:	275
5.2.5. Thông kê dữ liệu sau khi tổng hợp	278
5.2.5.1. Thông tin cơ bản bộ dữ liệu	278
5.2.5.2. Trực quan hóa phân phố dữ liệu	281

5.2.5.3. Trực quan hóa phân phối dữ liệu	286
5.2.5.4. Tính toán hệ số tương quan	293
5.2.5.5. Loại bỏ những feature có tương quan cao	294
5.2.6. Ghi nhãn dữ liệu	297
5.2.6.1. Cơ sở tính điểm	297
5.2.6.2. Chiến lược xếp loại kết quả của học viên.	298
5.2.7. Chia tập dữ liệu	300
5.2.7.1. Xác định thời gian cho tập dữ liệu test	300
5.2.7.1.1. Thông tin về thời gian bắt đầu, kết thúc khóa học	300
5.2.7.1.2. Chia tập train và test theo thời gian	303
a. Tập test	303
b. Tập train	303
5.2.7.2. Tổng quan tập train và test	304
5.2.7.2.1. Giai đoạn 1 (phase 1)	304
5.2.7.2.2. Giai đoạn 2 (phase 2)	306
5.2.7.2.3. Giai đoạn 3 (phase 3)	307
5.2.7.2.4. Giai đoạn 4 (phase 4)	308
5.2.7.2.5. Tổng hợp các giai đoạn	308
5.2.7.3. Tập dữ liệu validation	309
5.3. Chất lượng dữ liệu và thông tin về dữ liệu	310
5.3.1. Input của bài toán	310
5.3.2. Metadata của dữ liệu	315
5.3.2.1. Đặc điểm dữ liệu	315
5.3.2.2 Thông tin chi tiết của các cột	316
Bảng course (Thông tin khóa học)	316
Bảng resource (Tài liệu khóa học)	317
Bảng score (Thành phần điểm)	317
Bảng user (Thông tin người dùng)	317
Bảng comment (Hoạt động bình luận của user theo đợt 2 tuần)	318
Bảng reply (Hoạt động bình luận của user theo đợt 2 tuần)	318
Bảng exercise (Hành vi làm bài tập của user theo đợt 2 tuần)	319
5.3.3. Phân tích thống kê dữ liệu	322
5.3.3.1. Kiểm tra số giá trị NULL trong từng cột:	323
5.3.3.2 Kiểm tra số lượng giá trị duy nhất (độ phân tán của dữ liệu): df.unique()	324
5.3.3.3. Kiểm tra độ dài trung bình của chuỗi đối với các cột kiểu object (chuỗi văn bản): df.select_dtypes(include=['object']).apply(lambda x: x.str.len().mean())	325
5.3.3.4 Kiểm tra điều kiện dữ liệu có hợp lệ không theo quy tắc nghiệp vụ	

và duy túc kinh doanh:	325
5.3.4. Phát hiện những giá trị bất thường và dữ liệu trùng lặp	327
5.3.4.1. Xác định outliers	327
5.3.4.2. Xác định dữ liệu bị trùng lặp	329
5.3.5. Kiểm tra tính nhất quán giữa nhiều nguồn dữ liệu	330
5.3.6. Accuracy	330
Accuracy theo Object	330
Accuracy toàn bộ dataset	331
5.3.6.1. Reliability:	332
Độ ổn định qua các fold (Consistency)	342
Đánh giá tổng thể	342
Reliability	342
Kết luận	342
5.3.6.2. Relevance	343
Độ tin cậy tổng quan	350
Đánh giá cụ thể từng phase	351
Sự chênh lệch giữa các phase	351
Kết luận tổng quan	351
5.3.7 Completeness	351
5.3.7.1. Completeness theo Object	352
5.3.7.2. Completeness toàn bộ Dataset	353
5.3.8. Consistency	354
5.3.8.1 Tính nhất quán: Dữ liệu phải đồng nhất giữa các nguồn và hệ thống khác nhau, không có xung đột hoặc trùng lặp.	354
5.3.8.2 Ý tưởng đo lường: Được tính bằng tỷ lệ giữa số lượng điều kiện hợp lệ và tổng số điều kiện cần kiểm tra.	354
5.3.8.3 Sử dụng bộ dữ liệu đến giai đoạn 2 tức người dùng đã học 1 tháng của khóa học để đánh giá độ Consistency. Sẽ bổ sung thêm đầy đủ vào báo cáo đồ án.	354
5.3.8.4 Consistency theo Object	354
a. Miền giá trị (Domain Range)	354
b. Dữ liệu không rỗng (Not-Null)	357
c. Loại dữ liệu (Data Type)	358
d. Ràng buộc logic (Logical Constraints)	361
e. Tính duy nhất (Uniqueness)	363
f. Tính khóa ngoại (Foreign Key Integrity)	363
g. Tính Consistency cho một Object	365
5.3.8.5 Consistency toàn bộ Dataset	366
5.3.9. Timeliness	368

5.3.9.1 Khảo sát dữ liệu và xác định thời gian chủ yếu được cập nhật	368
5.3.9.2 Áp dụng công thức tính timeliness giả sử thời gian cập nhật	373
Timeliness theo Object	373
Timeliness toàn bộ Dataset	375
5.4 Trích xuất đặc trưng từ đồ thị (graph) truyền thống	376
5.4.1 Các thành phần cơ bản trong graph	376
5.4.1.1 Tổng quan về Đồ thị (Graph)	376
5.4.1.1.1 Định nghĩa cơ bản:	376
5.4.1.1.2 Lợi ích khi trích xuất đặc trưng từ đồ thị	376
5.4.1.1.3 Phân loại Trích xuất đặc trưng từ đồ thị	378
a) Node-level Features – Đặc trưng tại từng đỉnh	378
b) Edge-level Features – Đặc trưng cho các cạnh	379
c) Graph-level Features – Đặc trưng cho toàn đồ thị	379
5.4.1.1.4 Ví dụ áp dụng: Zachary's Karate Club Graph	379
5.4.1.2 Xây dựng graph theo dữ liệu	381
5.4.1.2.1. Định nghĩa đồ thị:	381
5.4.1.2.2. Biểu đồ đồ thị cho 1000 node đầu tiên	382
5.4.1.2.3. Một biều đồ về node-level và nhận xét	383
Node degree	383
Eigenvector centrality	384
Centrality closeness	385
Clustering	386
Số liệu của các cột	387
5.4.1.3 Tính toán trên toàn bộ dữ liệu	387
a) Node-Level	387
b) Cluster	388
5.4.2 Trích xuất đặc trưng từ đồ thị (graph) sử dụng node2vec	398
5.4.2.1 Định nghĩa đồ thị trong node2vec	398
5.4.2.2 Các bước thực hiện	400
5.4.2.3 Áp dụng vào dữ liệu	401
5.4.2.4 Gộp đặc trưng với dữ liệu	403
5.4.3 Áp dụng kỹ thuật SMOTE	412
5.4.3.1 Động lực sử dụng	412
5.4.3.2 Nguyên lý hoạt động	412
5.4.3.2 Đánh giá bộ dữ liệu sau khi áp dụng SMOTE	413
Bộ dữ liệu Node2Vec:	413
Bộ dữ liệu NodeClusterLevel:	413
5.5 Huấn luyện model	413

5.5.1 Độ đo đánh giá	414
1. Accuracy	414
2. F1-score	415
3. Precision và Recall	415
4. AUC-ROC (Area Under the Curve - Receiver Operating Characteristic)	
416	
5.5.2 Mô hình huấn luyện	417
5.5.2.1 Lý do lựa chọn mô hình	417
5.5.2.2 Giới thiệu mô hình	419
5.5.2.3 Quy trình huấn luyện	420
5.5.2.4 Tham số tốt nhất	421
5.5.2.5 Kết quả	423
Final-data	423
Filtered-final-data	424
SMOTE Filtered-final-data	426
Node2vec	428
SMOTE Node2vec	430
NodeClusterLevel	431
SMOTE NodeClusterLevel	433
5.5.2.6 Nhận xét	435
CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN	439
6.1. Kết quả đạt được	439
6.2. Hạn chế	439
6.3. Hướng phát triển trong tương lai	440
6.3.1. Tối ưu hóa và tự động hóa mô hình	440
6.3.2. Mở rộng và nâng cao chất lượng dữ liệu	440
6.3.3. Xây dựng hệ thống cảnh báo sớm toàn diện	440
6.3.4. Phát triển ứng dụng thực tế	440
TÀI LIỆU THAM KHẢO	441

CHƯƠNG 1. TỔNG QUAN ĐỀ TÀI

1.1. Giới thiệu

Trong kỷ nguyên chuyển đổi số, giáo dục trực tuyến đã và đang trở thành một xu hướng tất yếu, mở ra cơ hội tiếp cận tri thức rộng lớn cho hàng triệu người học trên toàn cầu. Đặc biệt, các khóa học trực tuyến mở đại trà (MOOCs – Massive Open Online Courses) đang dần khẳng định vai trò quan trọng trong việc dân chủ hóa giáo dục, khi mang lại tính linh hoạt, tiện lợi và chi phí thấp cho người học ở nhiều độ tuổi, vùng miền và trình độ khác nhau.

Tuy nhiên, bên cạnh những lợi thế nổi bật, các nền tảng MOOCs vẫn còn tồn tại những hạn chế, trong đó nổi bật nhất là tỷ lệ hoàn thành khóa học thấp và thiếu các cơ chế hỗ trợ học tập cá nhân hóa. Nhiều học viên không thể duy trì động lực hoặc gặp khó khăn trong việc tiếp cận nội dung học tập, dẫn đến kết quả học không như mong đợi hoặc bỏ học giữa chừng. Vấn đề này đặt ra một yêu cầu cấp thiết cho việc phát triển các hệ thống phân tích học tập thông minh, có khả năng dự đoán sớm kết quả học tập và đưa ra cảnh báo kịp thời cho người học.

Đề tài nghiên cứu "**Dự đoán sớm kết quả học tập của học sinh trong các khóa học MOOC bằng trích xuất đặc trưng quan hệ dựa trên đồ thị**" được xây dựng với mục tiêu giải quyết bài toán trên. Đề tài tập trung vào việc ứng dụng các kỹ thuật học máy và học sâu hiện đại để xây dựng mô hình dự đoán kết quả học tập, giúp phát hiện sớm những học viên có nguy cơ gặp khó khăn trong quá trình học và từ đó hỗ trợ đưa ra các biện pháp can thiệp phù hợp.

1.2 Ý tưởng đề tài

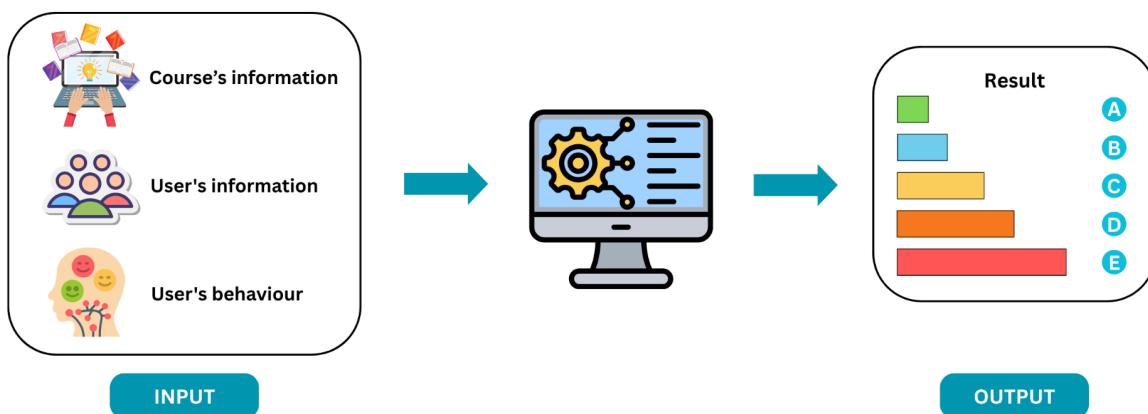
Xây dựng một mô hình dự đoán kết quả học tập cuối cùng của học viên trong mỗi khóa học, đồng thời cung cấp cảnh báo sớm để học viên có thể điều chỉnh phương pháp học tập phù hợp. Mô hình này sẽ dựa trên dữ liệu hành vi học tập của học viên trong suốt khóa học, chẳng hạn như thời gian xem video, số lần tham gia bài kiểm tra, số lượng bình luận trong diễn đàn, và điểm số trong các bài tập. Mục tiêu của mô hình là không chỉ dự đoán chính xác kết quả cuối cùng mà còn có thể phát hiện sớm các dấu hiệu cho thấy học viên có thể gặp khó khăn trong quá trình học. Từ đó, hệ thống có thể đưa ra các khuyến nghị hoặc cảnh báo kịp thời, giúp học viên thay đổi chiến lược học tập để cải thiện kết quả. Ví dụ, nếu một học viên có thời gian xem video quá ít hoặc kết quả bài kiểm tra thấp, hệ thống sẽ cảnh báo để học viên có thể dành nhiều thời gian hơn cho học tập hoặc tham gia thảo luận nhiều hơn.

Mô hình dự đoán sẽ được huấn luyện dựa trên các dữ liệu lịch sử của học viên trong nhiều khóa học khác nhau, sử dụng các thuật toán học máy (machine learning)

như hồi quy tuyến tính, logistic regression, hoặc các mô hình phức tạp hơn như random forest hoặc mạng nơ-ron. Để tối ưu độ chính xác và hiệu quả của cảnh báo sớm, việc sử dụng kỹ thuật phân tích thống kê và khai thác dữ liệu sẽ giúp xác định các yếu tố có ảnh hưởng lớn đến kết quả học tập của học viên.

1.3 Định nghĩa bài toán

Bài toán mà nhóm đặt ra là xây dựng một **hệ thống dự đoán sớm kết quả học tập của học sinh (xếp loại theo 5 mức độ)** trong các khóa học MOOC bằng trích xuất đặc trưng quan hệ dựa trên đồ thị. Để tiết kiệm tài nguyên tính toán, đề tài chỉ sử dụng một phần dữ liệu đầu vào gồm các nhóm thông tin sau:



- **Input:**

- Thông tin học viên
- Thông tin khóa học
- Hành vi hoạt động của người học với khóa học trong 8 tuần đầu, ví dụ: Số lượt đăng nhập, Thời gian học, Số bài học đã hoàn thành, Lượt xem video, Lượt tương tác với bài tập, Số lần nộp bài
- Hành vi hoạt động trên diễn đàn: Số lượng bình luận, và nhận phản hồi.

Các tệp dữ liệu được sử dụng để trích xuất đặc trưng gồm: entities/user.json, entities/course.json, entities/comment.json, entities/reply.json, relations/comment-reply.json, relations/user-comment.json, relations/user-reply.json, relations/user-problem.json, relations/user-video.json.

- **Output:** Nhấn đầu ra là kết quả cuối khóa học của học viên, được phân loại thành 5 mức độ theo thứ tự từ cao đến thấp:

- A: Xuất sắc
- B: Khá
- C: Trung bình
- D: Yếu

■ E: Không hoạt động / bỏ học

1.4 Đối tượng và phạm vi đề tài

Đề tài tập trung vào việc thiết kế và triển khai một hệ thống thông minh có khả năng phân tích và dự đoán kết quả học tập của sinh viên khi tham gia các khóa học trực tuyến mở (MOOCs), đồng thời đưa ra cảnh báo sớm cho những trường hợp có nguy cơ không đạt kết quả như mong đợi. Đặc biệt, hệ thống được xây dựng với trọng tâm hướng đến nhóm đối tượng những học viên có hiệu suất học tập thấp hoặc có nguy cơ bỏ học – nhằm kịp thời phát hiện và hỗ trợ can thiệp sớm, từ đó góp phần nâng cao tỷ lệ hoàn thành khóa học và cải thiện chất lượng đào tạo.

Phạm vi nghiên cứu bao gồm việc thu thập, xử lý và phân tích dữ liệu hành vi học tập của người học trên nền tảng MOOCs; phát triển mô hình phân loại kết quả học tập theo năm mức độ; tích hợp hệ thống cảnh báo sớm; và triển khai thử nghiệm trên môi trường học tập trực tuyến mô phỏng.

Các khía cạnh kỹ thuật của đề tài sẽ bao gồm:

- Xây dựng mô hình học máy và học sâu sử dụng các thư viện như scikit-learn, TensorFlow hoặc PyTorch để dự đoán kết quả học tập.
- Phát triển kiến trúc lưu trữ và xử lý dữ liệu dựa trên công nghệ dữ liệu lớn như Apache Spark.
- Tích hợp hệ thống lên nền tảng điện toán đám mây nhằm bảo đảm khả năng mở rộng và hiệu quả vận hành.

Đối tượng đầu tiên mà hệ thống hướng đến gồm:

- **Các nhà giáo dục và quản lý khóa học:** Hệ thống đóng vai trò như một công cụ hỗ trợ ra quyết định, cung cấp dữ liệu phân tích về hiệu suất học tập của từng sinh viên cũng như toàn bộ lớp học. Nhờ đó, giảng viên có thể phát hiện sớm những sinh viên gặp khó khăn – đặc biệt là những học viên có hiệu suất học tập thấp hoặc có nguy cơ bỏ học để điều chỉnh phương pháp giảng dạy hoặc xây dựng các chương trình hỗ trợ học tập phù hợp. Đồng thời, người quản lý có thể sử dụng hệ thống để giám sát chất lượng đào tạo và tối ưu hóa việc vận hành khóa học.
- **Trong tương lai, đối tượng tiềm năng cho đề tài là người học.** Trong tương lai, hệ thống sẽ mở rộng phạm vi phục vụ đến đối tượng người học. Khi tham gia các khóa học trực tuyến, người học sẽ được hệ thống theo dõi và đánh giá tiến trình học tập thông qua dữ liệu hành vi và kết quả học tập. Dựa trên phân tích đó, hệ thống sẽ đưa ra các cảnh báo sớm trong trường hợp hiệu suất học tập sụt giảm, giúp người học kịp thời điều chỉnh kế hoạch học tập, tập trung vào những phần kiến thức yếu và tăng khả năng hoàn thành khóa học với kết quả tốt.

1.5 Ứng dụng của bài toán

Bài toán dự đoán kết quả học tập và đưa ra cảnh báo sớm trên nền tảng MOOCs mang lại nhiều giá trị thiết thực trong việc nâng cao chất lượng giáo dục trực tuyến, đặc biệt đối với nhóm học viên có nguy cơ bỏ học hoặc đạt kết quả thấp. Hệ thống được thiết kế nhằm hỗ trợ kịp thời cho nhóm đối tượng này thông qua việc phát hiện sớm các dấu hiệu sụt giảm hiệu suất học tập, từ đó giúp người học điều chỉnh chiến lược học tập một cách chủ động và hiệu quả.

Đối với **người học**, hệ thống mang lại khả năng theo dõi tiến trình học tập cá nhân một cách trực quan và khoa học. Khi được cảnh báo sớm về nguy cơ đạt kết quả không tốt, người học có thể chủ động điều chỉnh kế hoạch học tập, cải thiện phương pháp học và tìm kiếm sự hỗ trợ phù hợp. Điều này không chỉ nâng cao khả năng hoàn thành khóa học mà còn góp phần hình thành kỹ năng tự học, tinh thần trách nhiệm và thái độ học tập tích cực – những yếu tố then chốt trong giáo dục trực tuyến.

Đối với **giảng viên và nhà quản lý giáo dục**, hệ thống đóng vai trò như một công cụ hỗ trợ phân tích và ra quyết định hiệu quả. Thông qua các chỉ số học tập và dữ liệu dự đoán, giáo viên có thể xác định được các nhóm sinh viên đang gặp khó khăn, đặc biệt là nhóm có nguy cơ cao, để đưa ra các biện pháp hỗ trợ kịp thời như: tổ chức các buổi ôn tập chuyên sâu, tư vấn học tập cá nhân hoặc điều chỉnh nội dung và phương pháp giảng dạy. Nhà quản lý có thể sử dụng các báo cáo và thống kê học tập để đánh giá chất lượng của từng khóa học, đồng thời tối ưu hóa chương trình đào tạo và hệ thống hỗ trợ học tập theo hướng cá nhân hóa.

Về tổng thể, việc ứng dụng bài toán vào thực tiễn không chỉ góp phần tăng tỷ lệ hoàn thành khóa học mà còn nâng cao hiệu quả học tập và chất lượng giảng dạy trên các nền tảng MOOCs. Quan trọng hơn, đây là một bước tiến trong việc phát triển các **hệ thống học tập thông minh có khả năng cá nhân hóa**, cung cấp hỗ trợ phù hợp theo từng mức độ học lực của người học, đồng thời phản ánh đúng nhu cầu cá nhân hóa của giáo dục trong kỷ nguyên số hóa.

1.6 Khó khăn và thách thức

Khối lượng dữ liệu lớn: Việc xử lý dữ liệu phong phú từ các phản hồi, hiệu suất học tập và yếu tố ảnh hưởng đến sự hoàn thành khóa học là một thách thức. Quá trình làm sạch và phân tích dữ liệu yêu cầu công cụ mạnh mẽ và quy trình hiệu quả, nếu không sẽ ảnh hưởng đến độ chính xác của dự đoán.

Lựa chọn mô hình học máy: Việc chọn mô hình học máy phù hợp rất quan trọng vì các mô hình khác nhau có thể cho kết quả khác nhau. Cần thử nghiệm nhiều thuật toán và tinh chỉnh tham số để đạt được độ chính xác cao nhất.

Triển khai thực tiễn: Áp dụng kết quả nghiên cứu vào thực tế có thể gặp khó khăn do sự kháng cự đối với thay đổi và thiếu đồng thuận trong tổ chức giáo dục. Xây dựng mối quan hệ với các bên liên quan và thuyết phục họ về lợi ích của nghiên cứu là yếu tố quan trọng để thành công.

Tính bảo mật: Việc bảo vệ thông tin học viên và dữ liệu nhạy cảm là một thách thức lớn. Cần đảm bảo các biện pháp bảo mật mạnh mẽ để ngăn ngừa rủi ro bị rò rỉ dữ liệu, đồng thời tuân thủ các quy định về bảo mật và quyền riêng tư trong suốt quá trình xử lý và phân tích dữ liệu.

1.7 Mục tiêu thực hiện đề tài

Mục tiêu tổng quát:

- Xây dựng hệ thống thông minh dự đoán mức độ hoàn thành khóa học của học viên trên nền tảng MOOC.
- Ứng dụng học máy, dữ liệu lớn và điện toán đám mây để dự đoán kết quả học tập, cảnh báo sớm nguy cơ và hỗ trợ cải thiện chất lượng đào tạo.

Mục tiêu cụ thể:

- Thu thập, xử lý và phân tích dữ liệu học tập từ MOOC.
- Xây dựng mô hình dự đoán kết quả học tập sử dụng các thuật toán học máy như:
 - 4-layer stacked LSTM
 - Random Forest
 - LightGBM
- Tối ưu hóa mô hình học máy nhằm nâng cao độ chính xác và tốc độ dự đoán.
- Phát triển hệ thống cảnh báo sớm nhằm:
 - Phát hiện học viên có nguy cơ không hoàn thành hoặc kết quả thấp.
 - Gửi cảnh báo kịp thời đến học viên và giảng viên để có biện pháp can thiệp.
- Xây dựng website có các chức năng:
 - Dự đoán kết quả học tập.
 - Quản lý và phân tích dữ liệu khóa học và học viên.
 - Hỗ trợ trực quan phân tích dữ liệu cho giảng viên và nhà quản lý.
- Triển khai hệ thống trên nền tảng điện toán đám mây để đảm bảo khả năng mở rộng, tốc độ xử lý và độ ổn định cao.

Phạm vi thực hiện:

- Áp dụng trên dữ liệu học viên các khóa học trực tuyến MOOC từ năm 2019 đến giữa năm 2021.
- Tập trung vào việc dự đoán khả năng hoàn thành khóa học và gửi cảnh báo sớm dựa trên dữ liệu thực tế.

CHƯƠNG 2. CÁC CÔNG TRÌNH NGHIÊN CỨU LIÊN QUAN

2.1 Phân tích kết quả khảo sát

"[ANN-LSTM: A deep learning model for early student performance prediction in MOOC](#)" (Fatima Ahmed Al-azazi, Mossa Ghurab, 2023)

Nội dung: Bài báo nghiên cứu về việc dự đoán kết quả học tập của sinh viên trong các Khóa học Trực tuyến Mở Rộng (MOOCs). Mục tiêu chính là xây dựng một mô hình dự đoán đa lớp (phân loại nhiều kết quả học tập như Đỗ, Trượt, Bỏ học, Giỏi) và hoạt động theo từng ngày để có thể can thiệp sớm, sử dụng dữ liệu nhân khẩu học và dữ liệu clickstream (lịch sử tương tác của sinh viên với môi trường học tập ảo). Nghiên cứu nhằm giải quyết các hạn chế của các mô hình trước đây thường chỉ dự đoán nhị phân, áp dụng cho các khóa học cụ thể hoặc chỉ đưa ra kết quả ở cuối khóa học.

Phương pháp: Nghiên cứu đề xuất mô hình học sâu ANN-LSTM (Artificial Neural Network–Long Short-Term Memory) để xử lý dữ liệu chuỗi thời gian từ clickstream hàng ngày, phù hợp với bài toán dự đoán kết quả học tập. Dữ liệu từ bộ OULAD được tiền xử lý bằng cách tổng hợp tương tác theo ngày, hợp nhất với thông tin nhân khẩu học, chuẩn hóa và mã hóa các đặc trưng. Mô hình ANN-LSTM gồm lớp LSTM đầu vào, các lớp Dense ẩn và đầu ra dùng hàm Softmax để phân loại 4 kết quả học tập. Ngoài ra, các mô hình nền như RNN và GRU được huấn luyện với cấu trúc tương tự để so sánh. Cuối cùng, mô hình được đánh giá bằng các chỉ số Precision, Recall, Accuracy và F1-score, và so sánh với RNN, GRU và các mô hình tiên tiến khác như DFFNN, AML, RF.

Kết quả Kết quả thực nghiệm trên tập dữ liệu OULAD cho thấy:

- Mô hình ANN-LSTM đạt hiệu suất tốt nhất trong việc dự đoán kết quả học tập đa lớp theo từng ngày so với các mô hình nền (RNN, GRU) và các mô hình tiên tiến khác.
- Độ chính xác dự đoán của ANN-LSTM tăng dần theo thời gian khóa học, từ 43% vào ngày đầu lên khoảng 56% vào ngày 15 và đạt 72% vào ngày cuối cùng của khóa học.
- Trong 3 tháng đầu khóa học, ANN-LSTM vượt trội hơn đáng kể so với RNN và GRU, với tỷ lệ tăng độ chính xác trung bình khoảng 15% so với GRU và 21% so với RNN.
- So với các mô hình tiên tiến khác ở cuối khóa, ANN-LSTM đạt độ chính xác 72%, cao hơn DFFNN (63%) và RF (66%).
- Mô hình ANN-LSTM có khả năng xử lý tốt dữ liệu chuỗi thời gian và các đặc trưng hành vi theo từng ngày, điều này được cho là lý do chính giúp nó vượt trội.

Kết luận Nghiên cứu đã thành công trong việc xây dựng và đánh giá mô hình học sâu ANN-LSTM để dự đoán kết quả học tập đa lớp, không phụ thuộc khóa học và theo từng ngày cho sinh viên MOOC, sử dụng dữ liệu nhân khẩu học và clickstream. Mô hình này đã chứng minh được hiệu quả vượt trội so với các mô hình nền và các mô hình tiên tiến khác, đặc biệt là trong việc dự đoán sớm kết quả học tập. Điều này mở ra cơ hội để các nhà giáo dục và hệ thống học tập có thể can thiệp kịp thời cho sinh viên có nguy cơ. Tuy nhiên, nghiên cứu cũng chỉ ra những hạn chế về thời gian xử lý dữ liệu lớn của các kỹ thuật học sâu. Các hướng nghiên cứu trong tương lai có thể tập trung vào việc cải thiện độ chính xác dự đoán trong những ngày đầu khóa học và khám phá các kỹ thuật xử lý dữ liệu khác.

"[Analysis of the Factors Influencing Learners' Performance Prediction With Learning Analytics](#)" (PEDRO MANUEL MORENO-MARCOS et al., 2020):

Bài báo nghiên cứu các yếu tố ảnh hưởng đến khả năng dự đoán kết quả học tập của sinh viên trong Khóa học Trực tuyến Mở Rộng (MOOCs). Mục tiêu chính là phân tích các yếu tố như biến số dự đoán, kết quả dự đoán, và bối cảnh khóa học để hiểu rõ hơn tại sao các mô hình dự đoán thường chỉ hiệu quả trong môi trường cụ thể và khó áp dụng cho khóa học khác. Việc này nhằm giúp cải thiện mô hình dự đoán và tăng khả năng tổng quát hóa sang các bối cảnh học tập khác.

Phương pháp

Nghiên cứu sử dụng dữ liệu từ hai MOOC về lập trình Java trên nền tảng edX với thiết kế khác nhau (một khóa 5 tuần - UC3M và một khóa 10 tuần - HKUST). Dữ liệu được lọc để chỉ bao gồm sinh viên có tương tác đáng kể. Các đặc trưng được sử dụng rất đa dạng, bao gồm điểm các bài tập trước, biến liên quan đến diễn đàn, biến liên quan đến bài tập, dữ liệu clickstream (tương tác video, hoạt động nền tảng, chỉ có ở khóa thứ hai), thời lượng khóa học, loại bài tập (khái niệm vs. thực hành/code), và cách thu thập dữ liệu (tích lũy từ đầu khóa vs. theo tuần). Kết quả dự đoán là điểm các bài tập (bao gồm bài thi cuối khóa) hoặc kết quả đỗ/trượt. Bốn thuật toán dự đoán phổ biến (Regression, SVM, Decision Trees, Random Forest) được áp dụng và đánh giá bằng các chỉ số RMSE (cho điểm) và AUC (cho đỗ/trượt). Nghiên cứu được cấu trúc để trả lời bốn câu hỏi nghiên cứu cụ thể.

Kết quả

Kết quả phân tích chỉ ra nhiều điểm quan trọng:

- Điểm các bài tập trước là yếu tố dự đoán mạnh mẽ nhất cho kết quả học tập tương lai.
- Khả năng dự đoán cải thiện theo thời gian, đạt hiệu quả tốt hơn ở các giai đoạn cuối khóa học.

- Biến liên quan đến hoạt động trên diễn đàn chung không hiệu quả trong việc cải thiện khả năng dự đoán trong hai khóa học này.
- Loại bài tập ảnh hưởng đến khả năng dự đoán: các bài tập định hướng khái niệm (trắc nghiệm) dễ dự đoán hơn các bài tập định hướng thực hành/code (lập trình).
- Dữ liệu clickstream (về hoạt động nền tảng và video) không cải thiện đáng kể khả năng dự đoán khi đã có các biến liên quan đến bài tập. Tuy nhiên, chúng hữu ích khi thông tin về bài tập không có sẵn.
- Trong bài thi cuối khóa, các câu hỏi trắc nghiệm dễ dự đoán hơn các câu hỏi lập trình.
- Điểm cuối khóa (trung bình cả quá trình) thường dễ dự đoán hơn điểm bài thi cuối khóa (kiến thức tại một thời điểm), mặc dù hai kết quả này có mối liên hệ rất chặt chẽ.

Kết luận

Nghiên cứu đã xác định được các yếu tố quan trọng ảnh hưởng lớn đến khả năng dự đoán sớm kết quả học tập trong MOOCs. Các yếu tố dự đoán mạnh nhất là các biến liên quan đến tương tác với bài tập, đặc biệt là điểm các bài tập trước. Biến diễn đàn ít có giá trị dự đoán. Dữ liệu clickstream có thể là một thay thế hữu ích khi thiếu dữ liệu bài tập. Loại bài tập và định dạng câu hỏi cũng đóng vai trò quan trọng (bài tập khái niệm dễ dự đoán hơn bài tập thực hành). Điểm trung bình cuối khóa dễ dự đoán hơn điểm bài thi cuối khóa. Những phát hiện này quan trọng cho việc xây dựng các mô hình dự đoán hiệu quả hơn và có khả năng tổng quát hóa tốt hơn trên các MOOC khác nhau, từ đó hỗ trợ can thiệp kịp thời cho sinh viên có nguy cơ. Tuy nhiên, nghiên cứu có hạn chế về số lượng khóa học và nguồn dữ liệu được sử dụng.

"Identifying the Factors Affecting Student Academic Performance and Engagement Prediction in MOOC Using Deep Learning: A Systematic Literature Review" (SHAHZAD RIZWAN et al., 2025):

Bài báo thực hiện đánh giá tổng quan hệ thống (SLR) các nghiên cứu từ 2019-2024 để tìm hiểu các yếu tố ảnh hưởng đến khả năng dự đoán kết quả học tập và mức độ tương tác của sinh viên trong MOOCs, đặc biệt tập trung vào các phương pháp Học Sâu (Deep Learning - DL). Bối cảnh nghiên cứu là những thách thức của MOOCs như tỷ lệ bỏ học cao và sự cần thiết phải cải thiện hiệu quả của chúng thông qua dự đoán sớm hành vi và kết quả của sinh viên.

Phương pháp

Nghiên cứu áp dụng phương pháp SLR theo hướng dẫn PRISMA, tìm kiếm trên năm cơ sở dữ liệu học thuật lớn. Sau quá trình lọc, 70 bài báo từ giai đoạn 2019-2024 đã được chọn để phân tích. Các bài báo này sử dụng các kỹ thuật Học Máy (ML)/Học Sâu

(DL) để dự đoán các kết quả như bỏ học, có nguy cơ, kết quả chung hoặc mức độ tương tác. Phân tích tập trung vào các đặc trưng (features) được sử dụng (nhân khẩu học, hành vi, hoạt động học tập, clickstream, cảm xúc) và các bộ dữ liệu phổ biến (như OULAD, KDD Cup 2015, XuetangX).

Kết quả

Phân tích 70 bài báo cho thấy:

- Lĩnh vực được nghiên cứu nhiều nhất là dự đoán sinh viên bỏ học và có nguy cơ (47%).
- Việc sử dụng Học Sâu (DL) đã tăng đáng kể từ 2019-2024.
- Các đặc trưng clickstream và nhân khẩu học (63%) là phổ biến nhất, tiếp theo là học thuật (14%).
- Các đặc trưng quan trọng để dự đoán thường bao gồm điểm bài tập trước, tương tác bài tập, dữ liệu clickstream và nhân khẩu học.
- Các bộ dữ liệu được dùng nhiều nhất cho dự đoán kết quả học tập là OULAD, KDD Cup 2015 và XuetangX.
- Nhiều mô hình ML và DL đã được áp dụng và cho thấy hiệu quả dự đoán cao.

Kết luận

Nghiên cứu tổng quan đã xác định được nhiều yếu tố dự đoán quan trọng trong MOOCs, đặc biệt là các đặc trưng liên quan đến clickstream, nhân khẩu học và học thuật. Học Sâu thể hiện tiềm năng lớn nhưng vẫn đối mặt với thách thức như hạn chế dữ liệu, lựa chọn đặc trưng và khả năng thích ứng trên quy mô lớn. Nghiên cứu cũng chỉ ra khoảng trống trong việc phát triển môi trường học tập cá nhân hóa, đặc biệt cho người học có nhu cầu đặc biệt. Những phát hiện này là cơ sở để thiết kế mô hình dự đoán và hệ thống hỗ trợ sinh viên hiệu quả hơn trong tương lai.

"[MOOC performance prediction by Deep Learning from raw clickstream data](#)"
(Kőrösi Gábor, Richard Farkas, 2020):

Bài báo giải quyết vấn đề dự đoán sớm kết quả học tập của sinh viên trong các Khóa học Trực tuyến Mở Rộng (MOOCs), nhấn mạnh sự cần thiết do tỷ lệ hoàn thành thấp. Nghiên cứu khám phá khả năng sử dụng Học Sâu (Deep Learning - DL) trực tiếp trên dữ liệu clickstream thô (cấp độ từng dòng log) để dự đoán hiệu suất cuối khóa, khác với cách truyền thống dựa vào thiết kế đặc trưng thủ công. Mục tiêu là xem các mạng nơ-ron sâu có thể tận dụng lợi thế từ dữ liệu thô này hay không.

Phương pháp

Nghiên cứu đề xuất sử dụng Mạng Nơ-ron Hồi quy (RNN) với đơn vị GRU, xử lý trực tiếp chuỗi dữ liệu clickstream thô. Mỗi hành động được biểu diễn cùng thông tin thời

gian. Mô hình GRU áp dụng kỹ thuật Dropout để tránh overfitting. Phương pháp này được so sánh với các mô hình cơ sở dùng thuật toán Machine Learning truyền thống (XGBoost, Ridge Regression) trên các đặc trưng tổng hợp, được thiết kế thủ công. Các mô hình được đánh giá hàng tuần bằng dữ liệu thu thập được đến cuối tuần đó. Dữ liệu log hoạt động từ khóa học của Stanford Lagunita được sử dụng. Đánh giá dựa trên Accuracy (phân loại) và RMSE (hồi quy), kèm kiểm định chéo 4 lần. Chỉ sinh viên hoàn thành bài kiểm tra nhất định mới được phân tích để xử lý dữ liệu thưa thớt.

Kết quả

Thực nghiệm cho thấy mô hình GRU trên dữ liệu thô cho kết quả vượt trội đáng kể so với các mô hình cơ sở dùng đặc trưng tổng hợp. Cả hai loại mô hình đều cải thiện chất lượng dự đoán khi có thêm dữ liệu theo từng tuần, nhưng RNN với dữ liệu thô nhạy hơn trong việc nắm bắt các mẫu hành vi. Mô hình đe xuất vượt qua mô hình cơ sở tốt nhất với cải thiện đáng kể về Accuracy (15%) và RMSE (30%). Hiệu suất dự đoán cải thiện đến khoảng tuần thứ 3 rồi ổn định. Sau 3 tuần, mô hình đạt độ chính xác 54% (phân loại), cao hơn 39% của mô hình cơ sở. Phân tích bổ sung chỉ ra GRU tốt hơn phương pháp truyền thống ở mọi kích thước log trong 2 tuần đầu; từ tuần 3-5, phương pháp truyền thống có thể tốt hơn với chuỗi log ngắn, GRU chỉ vượt trội khi có nhiều dữ liệu từ chuỗi dài. Có tương quan đáng kể giữa độ dài chuỗi log và chất lượng dự đoán của RNN. GRU cũng có khả năng nhận diện các vùng giống lặp trong không gian dự đoán. Mô hình GRU đặc biệt hữu ích trong việc dự đoán hiệu suất của sinh viên làm bài kiểm tra.

Kết luận

Nghiên cứu khẳng định tính khả thi và thành công của việc dùng Mạng Nơ-ron Hồi quy (RNN) trực tiếp trên dữ liệu log thô để dự đoán kết quả học tập trong MOOCs. Lợi ích chính là loại bỏ công đoạn thiết kế đặc trưng thủ công, tiết kiệm thời gian, công sức và tránh sai sót. Kết quả trên bộ dữ liệu Stanford Lagunita cho thấy mô hình đe xuất đạt kết quả tốt hơn đáng kể so với các mô hình cơ sở. Nghiên cứu thừa nhận cần so sánh với các mô hình cơ sở mạnh hơn trong tương lai. Các hướng phát triển tiếp theo bao gồm tối ưu hóa siêu tham số, xử lý mất cân bằng dữ liệu, cải thiện khả năng khai thác dữ liệu và đánh giá trên các bộ dữ liệu khác.

“The Crowd in MOOCs: A Study of Learning Patterns at Scale”(Xin Zhoua , Aixin Suna , Jie Zhang and Donghui Linb, 2024):

Bài báo tiến hành phân tích mô hình học tập của người học trên quy mô lớn trong các Khóa học Trực tuyến Mở Rộng (MOOCs). Nghiên cứu sử dụng một bộ dữ liệu khổng lồ gồm hàng trăm triệu hoạt động học tập từ gần một triệu người học trong hai năm. Mục tiêu là khám phá các mô hình học tập từ góc độ thời gian và đăng ký khóa học để

hiểu rõ hơn về hành vi của đám đông trong MOOCs và ứng dụng vào các tác vụ như đề xuất khóa học.

Phương pháp

Nghiên cứu sử dụng bộ dữ liệu gồm 351 triệu hoạt động từ 772.880 người học trên 1.629 khóa học trong giai đoạn 2015-2017. Phương pháp bao gồm: phân tích thống kê để xác định xu hướng thời gian của hoạt động học tập; áp dụng lý thuyết thông tin tương hỗ (chỉ số Jaccard, thông tin tương hỗ điểm) và khai phá mẫu tuần tự để phân tích mô hình đồng đăng ký và chuyển đổi giữa các khóa học/danh mục khóa học. Một mô hình đơn giản là FrePaPop được đề xuất và đánh giá để chứng minh cách các mô hình đăng ký này có thể cải thiện hệ thống đề xuất khóa học.

Kết quả

Phân tích dữ liệu lớn cho thấy:

- Hoạt động học tập có mô hình định kỳ rõ rệt theo ngày (đỉnh vào buổi tối) và tuần (tăng vào cuối tuần trong học kỳ). Khoảng thời gian giữa các hoạt động liên tiếp tuân theo hồn hợp luật lũy thừa và hàm cosine định kỳ.
- Các khóa học thường được đăng ký cùng nhau chủ yếu thuộc cùng danh mục hoặc cùng trường/tổ chức. Người học có xác suất cao đăng ký ít nhất hai khóa trong cùng danh mục.
- Các mô hình chuyển đổi đăng ký giữa các danh mục cũng được xác định, cho thấy tính bất đối xứng. Các lĩnh vực khoa học tự nhiên có độ gắn kết chuyển đổi nội bộ cao hơn.
- Mô hình đề xuất FrePaPop, sử dụng các mẫu tuần tự này, đạt hiệu suất cạnh tranh so với các mô hình phức tạp hơn nhưng với thời gian huấn luyện nhanh hơn đáng kể (hơn 200 lần so với FPMC).

Kết luận

Nghiên cứu đã thành công trong việc khám phá các mô hình học tập theo thời gian và đăng ký khóa học trên quy mô lớn trong MOOCs, cho thấy tính ổn định hành vi và mối quan hệ giữa các khóa học. Những mô hình này mang lại giá trị ứng dụng quan trọng, đặc biệt là trong việc cải thiện hiệu quả của hệ thống đề xuất khóa học và tiềm năng trong việc suy luận mối quan hệ tiên quyết giữa các khóa học. Mặc dù có những hạn chế về nguồn dữ liệu duy nhất và đặc điểm người học, các phát hiện chung vẫn rất có ý nghĩa.

"[Enhancing academic performance prediction with temporal graph networks for massive open online courses](#)" (Huang và Chen, 2024):

Bài báo tập trung vào việc cải thiện dự đoán kết quả học tập của sinh viên trong các Khóa học Trực tuyến Mở Rộng (MOOCs). Các phương pháp hiện tại dựa trên Học Sâu thường bỏ qua thông tin thời gian và tương tác, vốn rất quan trọng. Để khắc phục, nghiên cứu đề xuất biểu diễn quá trình học tập dưới dạng đồ thị thời gian động và giới thiệu mô hình mới APP-TGN dựa trên Mạng Nơ-ron Đồ thị Thời gian (TGN).

Phương pháp

Mô hình APP-TGN được đề xuất. Nó xây dựng một đồ thị động từ dữ liệu hoạt động học tập của sinh viên. Đồ thị này được xử lý bởi một mạng TGN cải tiến với bộ lọc thấp-cao. Một mô-đun lấy mẫu toàn cục được thêm vào để giảm sai lệch dữ liệu. Biểu diễn từ cả hai nhánh được kết hợp qua multi-head attention để dự đoán kết quả. Mô hình được đánh giá trên tập dữ liệu OULA cho hai nhiệm vụ phân loại nhị phân: Pass/Fail và Pass/Withdrawn. Nghiên cứu so sánh APP-TGN với các baseline truyền thống và dựa trên đồ thị, sử dụng các chỉ số ACC, F1, REL, và đánh giá cả khả năng dự đoán sớm.

Kết quả

APP-TGN thể hiện hiệu suất vượt trội đáng kể so với các baseline trên tập dữ liệu OULA cho cả hai nhiệm vụ, đặc biệt trong việc dự đoán sớm sinh viên có nguy cơ. Các mô hình dựa trên đồ thị tốt hơn các mô hình không dựa trên đồ thị, và việc sử dụng đồ thị thời gian cải thiện đáng kể so với đồ thị tĩnh. Các phân tích chỉ ra rằng mọi thành phần của APP-TGN (lấy mẫu toàn cục, bộ lọc thấp-cao, TGN) đều quan trọng. Tương tác với Quiz và Forumng có ảnh hưởng lớn nhất đến dự đoán. Chi phí tính toán của APP-TGN là cạnh tranh.

Kết luận

Nghiên cứu đã thành công giới thiệu mô hình APP-TGN, một phương pháp hiệu quả để dự đoán kết quả học tập trong MOOCs bằng cách khai thác thông tin thời gian và tương tác thông qua đồ thị thời gian động và TGN. APP-TGN vượt trội đáng kể so với các phương pháp hiện có và có tiềm năng lớn cho các ứng dụng giáo dục thông minh như phản hồi tự động và cá nhân hóa. Hạn chế bao gồm sự phụ thuộc vào dữ liệu từ một nguồn duy nhất và cần cải thiện khả năng khái quát hóa. Công việc tương lai sẽ tập trung vào xử lý dữ liệu đa dạng hơn và dự báo các kết quả giáo dục khác.

“[Meta Transfer Learning for Early Success Prediction in MOOCs](#)”(Vinitra Swamy, Mirko Marras, Tanja Käser, 2024):

Bài báo tập trung vào việc dự đoán sớm kết quả học tập của sinh viên trong MOOCs, với mục tiêu tạo ra các mô hình có khả năng chuyển giao sang các khóa học mới chưa có dữ liệu kết quả. Nghiên cứu đề xuất kết hợp dữ liệu hành vi của sinh viên với thông tin mô tả (meta information) về khóa học sử dụng kỹ thuật Học Chuyển Giao.

Phương pháp

Nghiên cứu sử dụng tập dữ liệu từ 26 MOOC với hơn 145.000 sinh viên ban đầu. Đặc trưng hành vi của sinh viên (dựa trên tương tác video, quiz) và đặc trưng mô tả khóa học (như thời lượng, cấp độ, ngôn ngữ, tiêu đề, mô tả) được trích xuất. Ba kiến trúc mô hình Học Sâu được đánh giá: chỉ dựa trên hành vi (BO), kết hợp hành vi và mô tả theo thời gian (BTM), và kết hợp hành vi và mô tả một cách tinh có cơ chế attention (BSM). Bài toán là phân loại nhị phân (đỗ/trượt), đánh giá khả năng chuyển giao và dự đoán sớm (ở mức 40% và 60% thời lượng khóa học) bằng chỉ số Balanced Accuracy (BAC).

Kết quả

Kết quả cho thấy khả năng chuyển giao phụ thuộc vào đặc điểm khóa học. Mô hình kết hợp đặc trưng hành vi và mô tả một cách tinh (BSM) mang lại hiệu suất chuyển giao tốt nhất và ổn định nhất trên nhiều khóa học khác nhau. Các đặc trưng mô tả như Cấp độ, Tiêu đề và Thời lượng rất quan trọng cho dự đoán sớm (40%), trong khi Duration (Thời lượng) vẫn quan trọng hơn ở các mức dự đoán muộn hơn (60%). Sự hiện diện của các bài kiểm tra (quizzes) trong khóa học cũng có ảnh hưởng tích cực đến khả năng dự đoán. Kỹ thuật fine-tuning chỉ hiệu quả khi các lần chạy trước của khóa học rất giống lần chạy hiện tại.

Kết luận

Nghiên cứu đã chứng minh khả năng chuyển giao hiệu quả của các mô hình dự đoán sớm kết quả học tập trong MOOCs bằng cách kết hợp dữ liệu hành vi và mô tả khóa học. Mô hình BSM cho thấy hiệu suất chuyển giao vượt trội, cho phép "khởi động nhanh" việc dự đoán cho các khóa học mới. Điều này mở ra tiềm năng cho các can thiệp mục tiêu hỗ trợ sinh viên có nguy cơ. Hạn chế bao gồm dữ liệu từ một nguồn duy nhất và việc trích xuất đặc trưng hành vi thủ công.

2.2 Hướng phát triển đề tài

1. Mở rộng và cải tiến mô hình dự đoán

- Thu thập và làm sạch dữ liệu:** Tiến hành thu thập và làm sạch dữ liệu từ các khóa học trực tuyến mở (MOOCs) bao gồm thông tin hành vi học tập, hoạt động trên diễn đàn và kết quả học tập của học viên trong các giai đoạn tiếp theo. Việc này sẽ giúp cập nhật và cải thiện độ chính xác của mô hình dự đoán.
- Nâng cao các tính năng đầu vào:** Cải tiến dữ liệu đầu vào bằng cách tích hợp thêm các đặc trưng khác có thể ảnh hưởng đến kết quả học tập của học viên, như việc tham gia vào các hoạt động nhóm, thời gian học tập ngoài giờ hoặc các yếu tố tâm lý, giúp mô hình dự đoán chính xác hơn.

- **Thử nghiệm các mô hình học máy mới:** Mở rộng phạm vi thử nghiệm với các mô hình phức tạp hơn như mạng nơ-ron sâu (Deep Learning) hoặc các mô hình học máy kết hợp (ensemble models) để tìm ra phương án tối ưu nhất cho dự đoán kết quả học tập.
- **Điều chỉnh và tối ưu hóa mô hình:** Tinh chỉnh tham số và sử dụng các kỹ thuật như Grid Search hoặc Random Search để tối ưu hóa các mô hình học máy, nhằm tăng độ chính xác và hiệu quả của mô hình dự đoán.

2. Phát triển hệ thống cảnh báo sớm và đề xuất điều chỉnh chiến lược học tập

- **Tích hợp hệ thống cảnh báo sớm:** Xây dựng hệ thống cảnh báo sớm có khả năng gửi thông báo tự động đến học viên khi phát hiện những dấu hiệu cho thấy học viên có thể gặp khó khăn trong quá trình học, như thời gian học quá ít, điểm số kiểm tra thấp, hoặc ít tham gia thảo luận.
- **Cung cấp đề xuất cá nhân hóa:** Hệ thống sẽ không chỉ cảnh báo mà còn đưa ra các đề xuất học tập cá nhân hóa, ví dụ như tham gia thảo luận nhiều hơn, dành thời gian học bổ sung cho các bài học khó, hoặc tham gia các buổi ôn tập.

3. Tích hợp hệ thống lên nền tảng điện toán đám mây

- **Tối ưu hóa khả năng mở rộng:** Triển khai mô hình và hệ thống cảnh báo trên nền tảng điện toán đám mây (cloud platform) như AWS, Google Cloud hoặc Azure để đảm bảo khả năng mở rộng và hiệu suất vận hành khi người dùng tăng lên.
- **Bảo mật và quyền riêng tư:** Cải thiện tính bảo mật của hệ thống bằng cách sử dụng các công nghệ bảo mật mạnh mẽ và tuân thủ các quy định bảo mật và quyền riêng tư (GDPR, HIPAA, v.v.) khi xử lý dữ liệu học viên.

4. Đánh giá và cải tiến hệ thống

- **Thử nghiệm và đánh giá mô hình trong môi trường thực tế:** Triển khai thử nghiệm hệ thống trong môi trường thực tế với một nhóm học viên, giảng viên và quản lý khóa học. Thu thập phản hồi từ người dùng để đánh giá tính chính xác của mô hình dự đoán và hiệu quả của hệ thống cảnh báo sớm.
- **Cải tiến liên tục:** Dựa trên kết quả thử nghiệm và phản hồi từ người dùng, liên tục cải tiến mô hình và hệ thống, bao gồm việc tối ưu hóa các thuật toán, điều chỉnh các tính năng đầu vào, và nâng cấp giao diện người dùng của hệ thống.

5. Tích hợp các công nghệ học máy mới

- **Ứng dụng học sâu (Deep Learning):** Xem xét ứng dụng các kỹ thuật học sâu như mạng nơ-ron hồi tiếp (RNN), LSTM hoặc GRU cho các dữ liệu thời gian, giúp cải thiện khả năng dự đoán và phát hiện các vấn đề học tập có tính dài hạn.
- **Khám phá các kỹ thuật phân tích dữ liệu khác:** Nghiên cứu và áp dụng các kỹ thuật phân tích dữ liệu mới như học không giám sát (unsupervised learning), phân tích cú pháp văn bản, và khai thác dữ liệu từ diễn đàn thảo luận để cải thiện hệ thống cảnh báo và đề xuất.

6. Mở rộng đối tượng sử dụng hệ thống

- **Mở rộng hệ thống cho người học:** Sau khi hoàn thiện, hệ thống sẽ được triển khai cho đối tượng học viên, giúp họ chủ động hơn trong việc quản lý tiến trình học tập của bản thân và cải thiện kết quả học tập.
- **Phát triển phiên bản đa ngôn ngữ:** Mở rộng hệ thống sang nhiều ngôn ngữ khác để hỗ trợ học viên toàn cầu, đồng thời điều chỉnh hệ thống để phù hợp với các nền tảng MOOCs khác nhau.

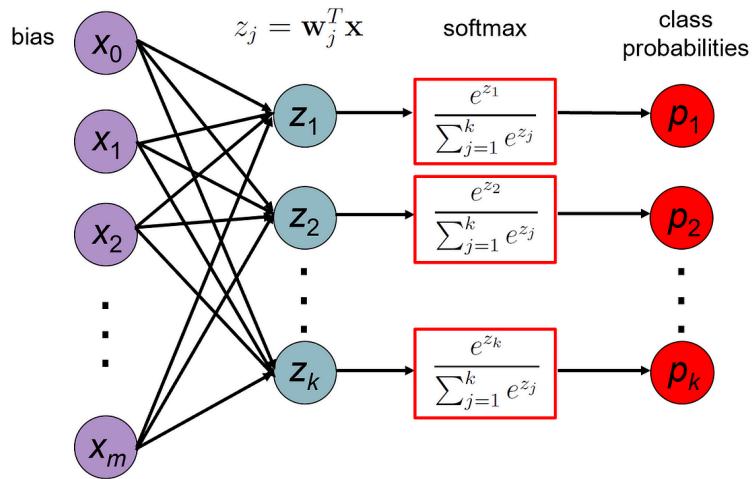
7. Hướng tới việc phát triển hệ thống học tập thông minh

- **Xây dựng hệ thống học tập cá nhân hóa:** Tích hợp thêm các tính năng như gợi ý các khóa học khác, đánh giá sự tiến bộ của học viên, và hỗ trợ học viên thông qua các công cụ học tập thông minh.
- **Ứng dụng phân tích dữ liệu lớn (Big Data) và AI:** Sử dụng các công cụ phân tích dữ liệu lớn và trí tuệ nhân tạo để dự đoán xu hướng học tập, phát hiện mô hình học tập của học viên và cung cấp các giải pháp học tập tối ưu.

CHƯƠNG 3. CƠ SỞ LÝ THUYẾT

3.1 Nền tảng lý thuyết

3.1.1. Hồi quy Softmax Logistic (Logistic Regression)



Giới thiệu mô hình Softmax Logistic (Logistic Regression)

Hồi quy Logistic (Logistic Regression) là một mô hình học máy dùng để giải quyết các bài toán phân loại, đặc biệt là phân loại nhị phân. Mô hình này tính xác suất một mẫu thuộc về một lớp cụ thể. Trong trường hợp phân loại đa lớp, mô hình sẽ sử dụng **Softmax** để tính xác suất thuộc về từng lớp.

Hồi quy Logistic là một mô hình tuyến tính đơn giản, nhưng rất hiệu quả trong các bài toán phân loại, chẳng hạn như phân loại email spam, phân loại bệnh tật (có/không), và phân loại hình ảnh hoặc văn bản. Khi được mở rộng với hàm **Softmax**, mô hình có thể giải quyết các bài toán phân loại với nhiều lớp, giúp phân loại các mẫu vào một trong nhiều lớp có thể có.

Cách hoạt động của Hồi quy Softmax Logistic

- **Phân loại nhị phân (Logistic Regression):**

- Mô hình Logistic sử dụng một hàm sigmoid để chuyển đổi đầu ra thành xác suất giữa 0 và 1. Cụ thể, với một vector đặc trưng x , mô hình tính

toán giá trị:

$$p(y=1|x) = \frac{1}{1 + e^{-(w^T x + b)}}$$

- **Phân loại đa lớp (Softmax Logistic Regression):**

- Đối với các bài toán phân loại đa lớp, hàm **Softmax** sẽ được sử dụng để tính xác suất cho mỗi lớp. Cụ thể, đối với k lớp, Softmax chuyển đổi điểm số z_k của mỗi lớp thành xác suất:

$$P(y=k|x) = \frac{e^{z_k}}{\sum_j e^{z_j}}$$

- **Huấn luyện mô hình:**

- Mô hình học từ dữ liệu huấn luyện bằng cách tối ưu hóa hàm mất mát, thường là hàm **Cross-Entropy**, để tìm các tham số w_{www} và b_{bbb} sao cho sai số giữa xác suất dự đoán và nhãn thực là nhỏ nhất. Quá trình tối ưu hóa có thể sử dụng các thuật toán như **Gradient Descent**.

Ưu điểm của Hồi quy Softmax Logistic

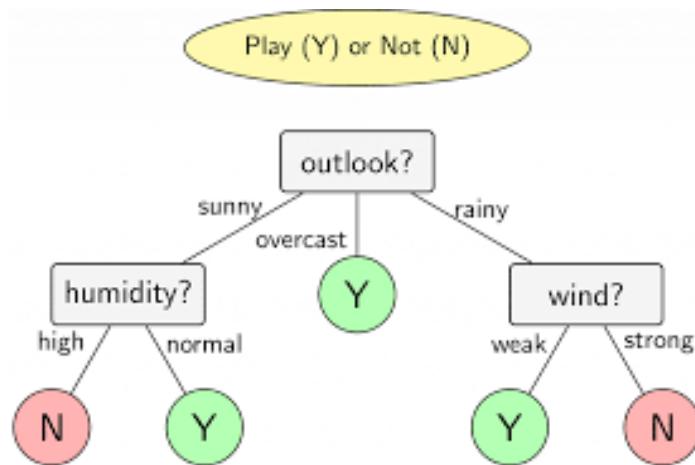
- **Đơn giản và dễ hiểu:** Hồi quy Logistic là một mô hình đơn giản, dễ triển khai và có thể giải thích được các kết quả phân loại.
- **Tính toán nhanh và hiệu quả:** Vì có ít tham số và tính toán đơn giản, Hồi quy Logistic rất hiệu quả trong việc huấn luyện và dự đoán.
- **Khả năng giải thích cao:** Các trọng số của mô hình có thể được giải thích một cách trực quan, giúp hiểu rõ mối quan hệ giữa các đặc trưng đầu vào và quyết định phân loại.
- **Phân loại nhị phân và đa lớp:** Với Softmax, mô hình có thể áp dụng cho cả bài toán phân loại nhị phân và đa lớp.

- **Không yêu cầu chuẩn hóa dữ liệu phức tạp:** Hồi quy Logistic không yêu cầu chuẩn hóa dữ liệu đặc trưng (tuy nhiên, vẫn nên chuẩn hóa nếu đặc trưng có phạm vi giá trị quá khác biệt).

Nhược điểm của Hồi quy Softmax Logistic

- **Giới hạn với dữ liệu phi tuyến tính:** Hồi quy Logistic chỉ có thể học các mối quan hệ tuyến tính. Nếu mối quan hệ giữa các đặc trưng và lớp không phải là tuyến tính, mô hình sẽ không hoạt động tốt.
- **Không xử lý tốt với dữ liệu nhiều đặc trưng hoặc phức tạp:** Với các bài toán có nhiều đặc trưng hoặc mối quan hệ phức tạp, Hồi quy Logistic có thể không hiệu quả bằng các mô hình phức tạp hơn như **Random Forest**, **SVM** hoặc **Mạng nơ-ron sâu**.
- **Dễ bị overfitting:** Nếu dữ liệu huấn luyện có quá nhiều đặc trưng và quá ít mẫu, mô hình có thể học quá mức và kém tổng quát hóa.
- **Không xử lý tốt dữ liệu thiếu:** Mặc dù có thể áp dụng một số phương pháp xử lý dữ liệu thiếu, Hồi quy Logistic không xử lý tốt các tập dữ liệu có nhiều giá trị thiếu.

3.1.2. Cây quyết định (Decision Tree)



Cây quyết định là một mô hình học máy phổ biến trong cả bài toán phân loại và hồi quy. Đây là một mô hình dựa trên cấu trúc cây, trong đó mỗi nút nội bộ đại diện cho

một điều kiện kiểm tra trên một đặc trưng, mỗi nhánh thể hiện kết quả của điều kiện đó, và mỗi nút lá chứa nhãn lớp hoặc giá trị dự đoán. Mô hình này mô phỏng quá trình ra quyết định logic theo dạng cây phân nhánh giúp dễ dàng hiểu và giải thích.

Cách hoạt động

- Mô hình xây dựng cây bằng cách phân tách tập dữ liệu đầu vào thành các tập con dựa trên các đặc trưng nhằm tăng độ “thuần nhất” của các tập con đó.
- Quá trình phân tách dựa trên các tiêu chí như:
 - **Entropy và Gain (thuật toán ID3)**: chọn đặc trưng giảm độ hỗn loạn nhất.
 - **Gini Impurity (thuật toán CART)**: chọn đặc trưng làm giảm độ tạp nhất.
 - **Gain Ratio (C4.5)**: cải thiện ID3 bằng cách chuẩn hóa gain.
- Cây được xây dựng đệ quy bằng cách chọn điều kiện phân tách tốt nhất tại mỗi bước cho đến khi đạt điều kiện dừng như:
 - Tập con thuần nhất (chỉ chứa một lớp).
 - Đạt chiều sâu tối đa.
Số lượng mẫu trong nút con quá nhỏ.
- Khi dự đoán, dữ liệu mới sẽ được so sánh qua các điều kiện từ gốc đến lá để xác định nhãn hoặc giá trị dự đoán.

Ưu điểm

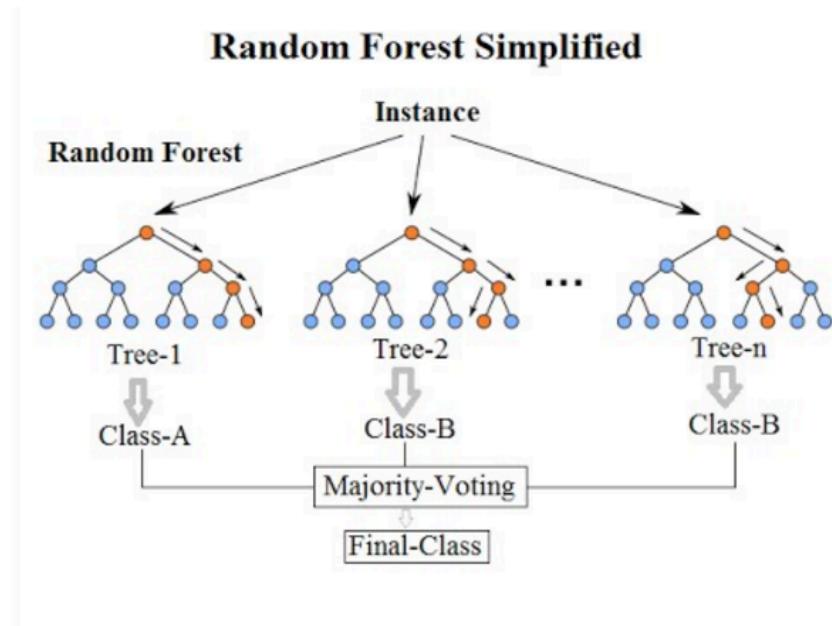
- **Dễ hiểu và trực quan**: Cây quyết định có cấu trúc rõ ràng, dễ dàng giải thích và hình dung.
- **Không yêu cầu chuẩn hóa dữ liệu**: Không cần chuẩn hóa hoặc biến đổi dữ liệu phức tạp.
- **Xử lý dữ liệu phi tuyến tính**: Có khả năng học các ranh giới phân lớp phi tuyến tính.
- **Linh hoạt với cả phân loại và hồi quy**: Có thể áp dụng trong nhiều bài toán khác nhau.

Tự động lựa chọn đặc trưng quan trọng: Quá trình phân tách giúp xác định đặc trưng có ảnh hưởng lớn.

Nhược điểm

- **Dễ bị overfitting:** Nếu không kiểm soát chiều sâu hoặc kích thước cây, mô hình có thể học quá mức trên dữ liệu huấn luyện.
- **Không ổn định:** Thay đổi nhỏ trong dữ liệu có thể làm thay đổi hoàn toàn cấu trúc cây.
- **Ưu tiên các đặc trưng có nhiều giá trị:** Các đặc trưng với nhiều mức giá trị có thể bị ưu tiên trong quá trình phân tách, gây thiên lệch.
Hiệu suất kém trên dữ liệu lớn, phức tạp: Cây đơn lẻ có thể không đạt hiệu quả cao bằng các mô hình phức tạp hơn như Random Forest hoặc Boosting.

3.1.3. Random Forest



Giới thiệu mô hình Random Forest

Random Forest là một mô hình học máy thuộc nhóm ensemble learning (học tập hợp), được xây dựng dựa trên tập hợp nhiều cây quyết định (Decision Trees) để cải thiện độ chính xác và giảm thiểu hiện tượng overfitting của cây đơn lẻ. Mỗi cây trong rừng được huấn luyện trên một tập con dữ liệu khác nhau được lấy mẫu ngẫu nhiên và

sử dụng một tập con các đặc trưng khác nhau để phân tách, tạo ra sự đa dạng trong các cây và tăng cường hiệu quả dự đoán.

Cách hoạt động

- **Tạo nhiều cây quyết định:** Với một tập dữ liệu huấn luyện, thuật toán Random Forest tạo ra nhiều cây quyết định bằng cách lấy mẫu bootstrap (lấy mẫu ngẫu nhiên có hoàn lại) từ dữ liệu ban đầu.
- **Chọn tập con đặc trưng ngẫu nhiên:** Tại mỗi nút phân tách trong cây, thuật toán chỉ xem xét một tập con ngẫu nhiên của các đặc trưng thay vì tất cả các đặc trưng, giúp tăng tính đa dạng và giảm tương quan giữa các cây.
- **Huấn luyện từng cây độc lập:** Mỗi cây được xây dựng hoàn toàn độc lập dựa trên tập dữ liệu và đặc trưng được chọn.
- **Dự đoán:** Với bài toán phân loại, kết quả cuối cùng được dự đoán bằng cách lấy phiếu đa số (majority voting) từ tất cả các cây; với bài toán hồi quy, kết quả là trung bình dự đoán của tất cả các cây.

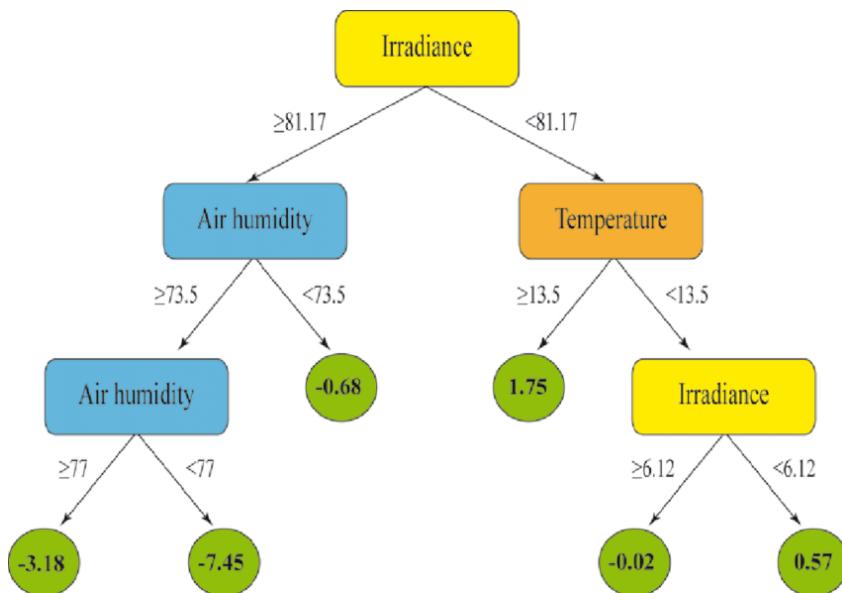
Ưu điểm

- **Giảm overfitting hiệu quả:** Bằng cách kết hợp dự đoán từ nhiều cây, mô hình giảm nguy cơ học quá mức so với cây quyết định đơn lẻ.
- **Độ chính xác cao:** Thường đạt hiệu suất tốt trên nhiều loại dữ liệu và bài toán khác nhau.
- **Xử lý dữ liệu phức tạp và phi tuyến tính tốt:** Có thể học các mối quan hệ phi tuyến tính và tương tác giữa các đặc trưng.
- **Không yêu cầu chuẩn hóa dữ liệu:** Giống cây quyết định, không cần chuẩn hóa đặc trưng.
- **Đánh giá mức độ quan trọng của đặc trưng:** Mô hình có thể cung cấp thông tin về tầm quan trọng của từng đặc trưng trong dự đoán.

Nhược điểm

- **Mô hình lớn và tốn tài nguyên:** Khi số lượng cây lớn, mô hình tiêu tốn nhiều bộ nhớ và thời gian tính toán hơn.
 - **Khó giải thích chi tiết:** Mặc dù dễ hiểu hơn các mô hình phức tạp như mạng nơ-ron sâu, nhưng so với cây quyết định đơn lẻ, Random Forest khó để diễn giải cụ thể từng quyết định.
 - **Không hoạt động tốt với dữ liệu có nhiều giá trị hiếm:** Nếu một số lớp hoặc giá trị quá hiếm, mô hình có thể chưa tối ưu.
- Dự đoán chậm hơn:** So với các mô hình đơn giản, việc dự đoán phải qua nhiều cây khiến thời gian dự đoán lâu hơn.

3.1.4. REP Tree



Giới thiệu mô hình REP Tree

REP Tree (Reduced Error Pruning Tree) là một biến thể của cây quyết định, tập trung vào việc xây dựng cây phân loại hoặc hồi quy với việc áp dụng kỹ thuật **pruning (cắt tỉa cây)** dựa trên lỗi giảm thiểu để tránh overfitting. Mô hình này xây dựng cây quyết định nhanh chóng bằng cách sử dụng các chỉ số thống kê (như Entropy hoặc Gini) để phân tách, sau đó áp dụng kỹ thuật cắt tỉa giảm thiểu lỗi để loại bỏ các nhánh không cần thiết, giúp tăng khả năng tổng quát hóa của cây.

Cách hoạt động

- **Xây dựng cây ban đầu:** REP Tree xây dựng cây quyết định dựa trên tập huấn luyện bằng cách chọn các điểm phân tách dựa trên chỉ số đo độ thuần nhất như Entropy hoặc Gini Impurity.
- **Phân tách nhanh:** Mô hình sử dụng thuật toán tối ưu để xây dựng cây nhanh, thường áp dụng cho dữ liệu lớn.
- **Cắt tỉa cây bằng kỹ thuật Reduced Error Pruning:**
 - Mô hình sử dụng một tập dữ liệu xác thực (validation set) để đánh giá lỗi của cây.
Các nhánh cây được cắt tỉa nếu việc loại bỏ chúng làm giảm hoặc không làm tăng lỗi trên tập xác thực.
 - Quá trình pruning giúp loại bỏ các nhánh phức tạp, giảm overfitting và làm cây đơn giản hơn.
- **Dự đoán:** Dữ liệu mới được dự đoán bằng cách di chuyển theo các nhánh từ gốc đến nút lá tương ứng.

Ưu điểm

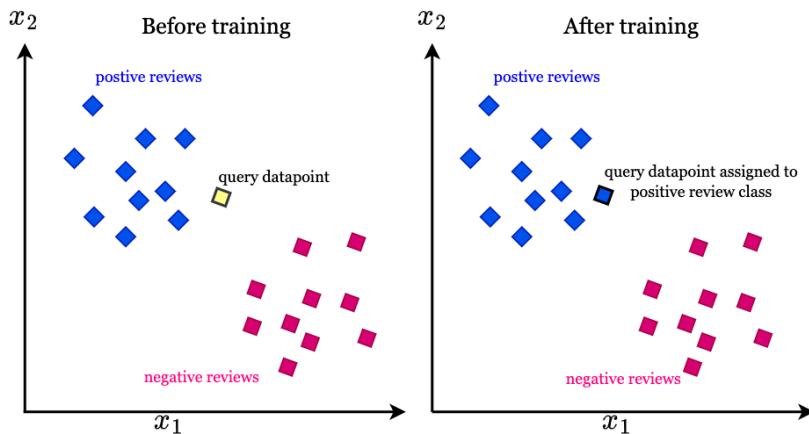
- **Tốc độ xây dựng nhanh:** REP Tree xây dựng cây rất nhanh, phù hợp với dữ liệu lớn.
- **Giảm overfitting:** Kỹ thuật cắt tỉa dựa trên lỗi thiểu giúp cây tổng quát hơn và tránh học quá mức.
- **Dễ hiểu và trực quan:** Mô hình vẫn giữ được cấu trúc cây quyết định dễ giải thích.
- **Áp dụng được cho cả phân loại và hồi quy:** REP Tree linh hoạt cho nhiều bài toán khác nhau.

Nhược điểm

- **Cần tập dữ liệu xác thực:** Quá trình pruning cần có tập xác thực để đánh giá lỗi, làm tăng yêu cầu dữ liệu.
- **Không phù hợp với dữ liệu quá phức tạp:** Nếu dữ liệu có cấu trúc phi tuyến phức tạp hoặc nhiều nhiễu, cây vẫn có thể bị hạn chế về khả năng dự đoán.

- **Có thể mất thông tin:** Quá trình cắt tỉa có thể loại bỏ những nhánh có giá trị nếu không được thực hiện cẩn thận.
- **Không tốt bằng các ensemble:** REP Tree đơn lẻ thường kém hiệu quả hơn so với các mô hình ensemble như Random Forest hay Boosting.

3.1.5. K-Nearest Neighbors (KNN)



Giới thiệu mô hình K-Nearest Neighbors (KNN)

K-Nearest Neighbors (KNN) là một thuật toán học máy đơn giản và trực quan thuộc nhóm mô hình **học gần kề** (instance-based learning). KNN được sử dụng cho cả bài toán phân loại và hồi quy. Ý tưởng chính của KNN là dự đoán nhãn hoặc giá trị của một điểm dữ liệu mới dựa trên nhãn hoặc giá trị của những điểm dữ liệu gần nhất (k láng giềng) trong không gian đặc trưng.

Cách hoạt động

- Khi có một điểm dữ liệu mới cần dự đoán, thuật toán KNN sẽ:
 - Tính khoảng cách từ điểm mới đến tất cả các điểm dữ liệu trong tập huấn luyện. Khoảng cách thường dùng là Euclidean, nhưng cũng có thể là khoảng cách Manhattan, Minkowski, v.v.
 - Xác định k điểm dữ liệu gần nhất (k láng giềng) với điểm cần dự đoán.
- Với bài toán phân loại:

- Thuật toán sẽ lấy nhãn phổ biến nhất trong k láng giềng làm nhãn dự đoán cho điểm mới (bỏ phiếu đa số).
- Với bài toán hồi quy:
 - Giá trị dự đoán sẽ là trung bình hoặc trọng số trung bình của các giá trị k láng giềng gần nhất.

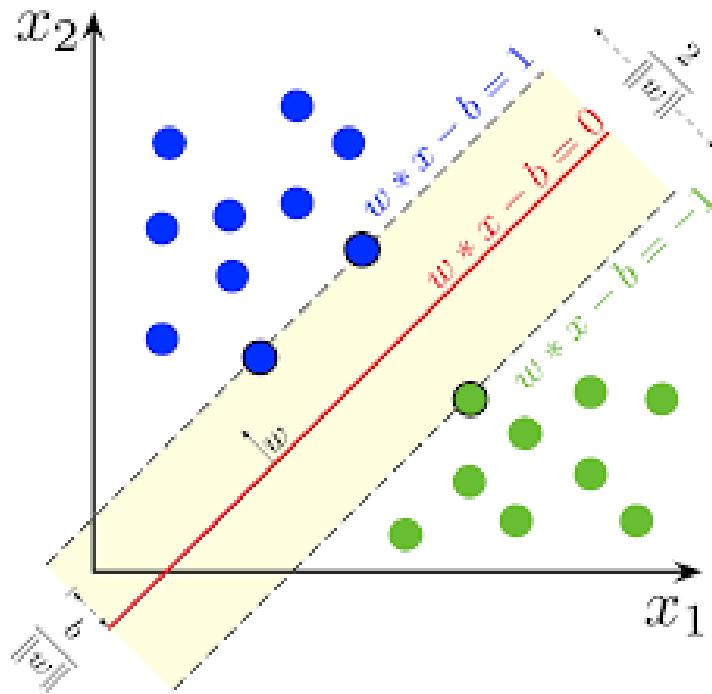
Ưu điểm

- **Đơn giản và dễ hiểu:** KNN không cần giai đoạn huấn luyện phức tạp, ý tưởng trực quan dễ giải thích.
- **Không giả định về phân phối dữ liệu:** Thuật toán không cần giả định tuyến tính hay phân phối chuẩn, linh hoạt với nhiều dạng dữ liệu.
- **Thích hợp với dữ liệu đa dạng:** KNN có thể áp dụng cho dữ liệu phi tuyến tính và các bài toán phân lớp phức tạp.
- **Cập nhật dễ dàng:** Khi có dữ liệu mới, chỉ cần thêm vào bộ dữ liệu mà không cần huấn luyện lại toàn bộ mô hình.

Nhược điểm

- **Tính toán chậm khi dự đoán:** Vì phải tính khoảng cách đến toàn bộ tập dữ liệu huấn luyện, KNN tốn nhiều thời gian khi dự đoán trên tập dữ liệu lớn.
- **Nhạy cảm với dữ liệu nhiễu và đặc trưng không liên quan:** KNN có thể bị ảnh hưởng mạnh bởi dữ liệu nhiễu hoặc các đặc trưng không quan trọng.
- **Chọn tham số k khó:** Giá trị k quá nhỏ dễ dẫn đến overfitting, quá lớn làm mất chi tiết phân lớp.
- **Phụ thuộc vào thang đo đặc trưng:** KNN cần chuẩn hóa hoặc chuẩn hóa các đặc trưng để khoảng cách có ý nghĩa.

3.1.6. SVM



Giới thiệu mô hình Support Vector Machine (SVM)

Support Vector Machine (SVM) là một mô hình học máy mạnh mẽ được sử dụng phổ biến trong các bài toán phân loại và hồi quy. SVM tìm kiếm một siêu phẳng (hyperplane) tối ưu để phân tách các lớp dữ liệu sao cho khoảng cách giữa siêu phẳng này và các điểm dữ liệu gần nhất (gọi là các support vectors) được tối đa hóa. Nhờ cơ chế tối ưu này, SVM có khả năng tạo ra ranh giới phân lớp rõ ràng và hiệu quả ngay cả khi dữ liệu có tính phi tuyến.

Cách Hoạt động

- **Tìm siêu phẳng tối ưu:** SVM cố gắng tìm một siêu phẳng phân tách dữ liệu thành các lớp sao cho khoảng cách (margin) từ siêu phẳng đến các điểm dữ liệu gần nhất là lớn nhất.
- **Support vectors:** Các điểm dữ liệu nằm gần ranh giới phân lớp nhất, ảnh hưởng trực tiếp đến vị trí của siêu phẳng phân tách.

- **Xử lý dữ liệu phi tuyến:** Khi dữ liệu không thể phân tách bằng một siêu phẳng tuyến tính, SVM sử dụng kỹ thuật **kernel trick** để ánh xạ dữ liệu lên không gian chiều cao hơn, nơi có thể tìm được siêu phẳng phân tách tuyến tính.
- Một số kernel phổ biến gồm: kernel tuyến tính, polynomial, Gaussian RBF, sigmoid.
- **Hàm mục tiêu:** SVM tối ưu hàm mục tiêu cân bằng giữa độ chính xác phân loại trên dữ liệu huấn luyện và độ phức tạp của mô hình (qua tham số điều chỉnh C).

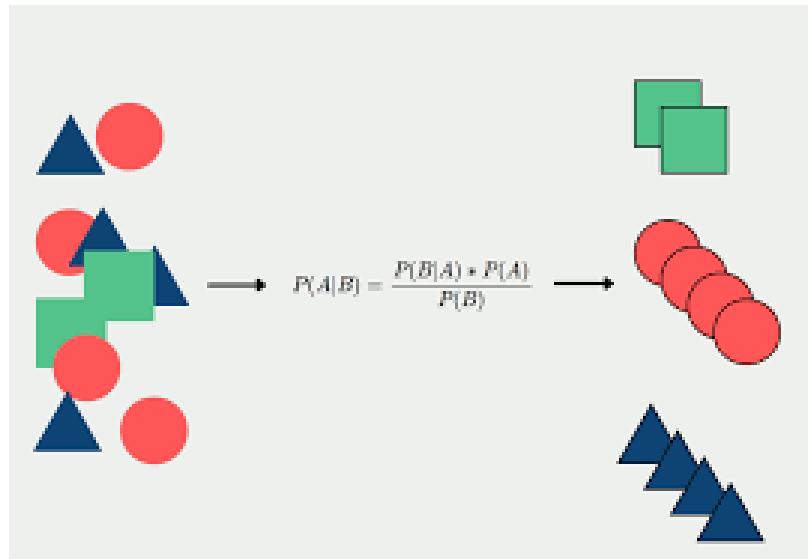
Ưu điểm

- **Hiệu quả trên dữ liệu có ranh giới phân lớp rõ ràng:** SVM tạo ra ranh giới phân lớp tối ưu giúp tăng độ chính xác.
- **Khả năng xử lý phi tuyến mạnh mẽ:** Nhờ kernel trick, SVM có thể xử lý các dữ liệu phức tạp phi tuyến.
- **Ít bị ảnh hưởng bởi số chiều lớn:** SVM hoạt động tốt trong không gian nhiều chiều.

Nhược điểm

- **Tính toán tốn kém:** Khi tập dữ liệu lớn, SVM có thể tốn nhiều thời gian và bộ nhớ để huấn luyện.
- **Khó chọn kernel và tham số:** Việc chọn kernel phù hợp và các tham số như C, gamma đòi hỏi kinh nghiệm và thử nghiệm.
- **Không trực quan:** Khó giải thích mô hình với người không chuyên vì không có cấu trúc dễ nhìn như cây quyết định.
- **Không hiệu quả trên dữ liệu nhiễu nhiều:** SVM có thể bị ảnh hưởng nếu dữ liệu có nhiều nhiễu hoặc không phân tách rõ ràng.

3.1.7. Naive Bayes



Giới thiệu Naive Bayes

Naive Bayes là một nhóm các thuật toán phân loại dựa trên định lý Bayes, với giả định đơn giản rằng các đặc trưng đầu vào là độc lập điều kiện (naive assumption). Mặc dù giả định này thường không đúng trong thực tế, Naive Bayes vẫn là một mô hình rất hiệu quả và được sử dụng rộng rãi trong các bài toán phân loại, đặc biệt là phân loại văn bản, chẳng hạn như lọc thư rác, phân loại cảm xúc, và phân loại các tài liệu.

Naive Bayes dựa trên việc tính toán xác suất có điều kiện của các lớp (labels) dựa trên các đặc trưng đầu vào, và chọn lớp có xác suất cao nhất làm dự đoán cho mẫu dữ liệu mới.

Cách Hoạt động của Naive Bayes

- **Định lý Bayes:** Naive Bayes sử dụng định lý Bayes để tính toán xác suất của mỗi lớp dựa trên các đặc trưng của mẫu. Cụ thể, định lý Bayes phát biểu rằng:

$$P(C_k|X) = \frac{P(X|C_k)P(C_k)}{P(X)}$$

- Giả định độc lập điều kiện (Naive Assumption): Mô hình giả định rằng các đặc trưng x_1, x_2, \dots, x_n , $x_{-1}, x_{-2}, \dots, x_{-n}$ là độc lập điều kiện khi biết lớp C_k . Điều này giúp đơn giản hóa việc tính toán $P(X|C_k)P(X|C_{-k})$ như sau:

$$P(X|C_k) = \prod_{i=1}^n P(x_i|C_k)$$

- Phân loại: Sau khi tính toán xác suất hậu nghiệm cho mỗi lớp, Naive Bayes chọn lớp có xác suất cao nhất làm dự đoán cho mẫu.

Ưu điểm của Naive Bayes

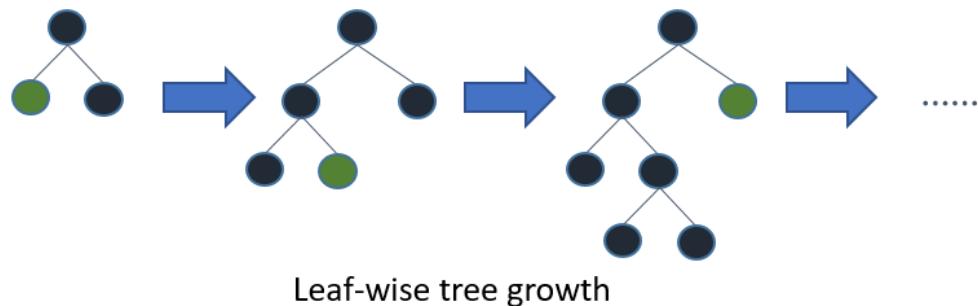
- Đơn giản và nhanh chóng: Naive Bayes rất dễ triển khai và có thể huấn luyện nhanh chóng, thậm chí đối với dữ liệu lớn.
- Hiệu quả với dữ liệu phân loại: Là một mô hình phân loại rất hiệu quả, đặc biệt khi dữ liệu có nhiều đặc trưng không liên quan đến nhau (tức là có thể đáp ứng tốt giả định độc lập điều kiện).
- Khả năng xử lý dữ liệu thiếu: Naive Bayes có thể xử lý tốt các bài toán có dữ liệu thiếu, vì mô hình chỉ cần sử dụng những đặc trưng có mặt.
- Hiệu quả trên dữ liệu văn bản: Naive Bayes là một lựa chọn tuyệt vời cho các bài toán phân loại văn bản, chẳng hạn như phân loại email spam hoặc phân tích cảm xúc, nhờ vào khả năng xử lý tốt các đặc trưng phân loại như từ vựng.
- Dễ giải thích: Các tham số của Naive Bayes có thể được giải thích một cách rõ ràng, giúp hiểu cách mô hình phân loại các mẫu dữ liệu.

Nhược điểm của Naive Bayes

- Giả định độc lập điều kiện không thực tế: Giả định các đặc trưng độc lập điều kiện là một hạn chế lớn, vì trong thực tế, các đặc trưng thường có mối quan hệ phụ thuộc với nhau. Điều này có thể làm giảm độ chính xác của mô hình.

- Không xử lý tốt với dữ liệu có mối quan hệ phi tuyến tính: Nếu các đặc trưng có mối quan hệ phi tuyến tính, Naive Bayes sẽ không thể học được những mối quan hệ này, làm giảm hiệu quả.
- Dễ bị ảnh hưởng bởi dữ liệu không cân bằng: Nếu một lớp chiếm ưu thế trong dữ liệu huấn luyện, mô hình có thể đưa ra các dự đoán sai lệch. Điều này có thể khắc phục bằng cách sử dụng các kỹ thuật cân bằng dữ liệu.
- Ván đề với dữ liệu hiếm: Khi có đặc trưng không xuất hiện trong tập huấn luyện cho một lớp nhất định, xác suất tính toán có thể trở thành 0, gây ra lỗi trong dự đoán. Tuy nhiên, điều này có thể được xử lý bằng cách sử dụng Laplace smoothing.

3.1.8. LightGBM



Giới thiệu mô hình LightGBM

LightGBM (Light Gradient Boosting Machine) là một thư viện học máy thuộc nhóm **boosting**, được phát triển bởi Microsoft. Đây là một thuật toán dựa trên phương pháp **Gradient Boosting Decision Trees (GBDT)**, được tối ưu hóa để xử lý hiệu quả các tập dữ liệu lớn, có nhiều đặc trưng và đạt tốc độ huấn luyện nhanh hơn các thuật toán boosting truyền thống.

LightGBM sử dụng các kỹ thuật tiên tiến như **Leaf-wise tree growth** và **Histogram-based splitting** giúp cải thiện hiệu quả mô hình và giảm chi phí tính toán.

Cách Hoạt động

- LightGBM xây dựng mô hình bằng cách kết hợp nhiều cây quyết định (decision trees) theo phương pháp **gradient boosting**: từng cây được huấn luyện để giảm sai số còn lại từ các cây trước.
- Điểm khác biệt chính của LightGBM là cách xây dựng cây:
 - **Phát triển cây theo kiểu Leaf-wise**: tại mỗi bước, LightGBM mở rộng nhánh lá có sai số lớn nhất thay vì phát triển đồng đều như các phương pháp truyền thống (Level-wise), giúp giảm lỗi nhanh hơn.
- Sử dụng kỹ thuật **Histogram-based splitting**: đặc trưng được chia thành các khoảng giá trị (bins), giảm số phép tính so sánh trong quá trình tìm điểm phân tách.
- Hỗ trợ tốt với dữ liệu lớn, nhiều đặc trưng, và dữ liệu phân tán.
- Áp dụng các kỹ thuật giảm thiểu overfitting như regularization, early stopping.

Ưu điểm

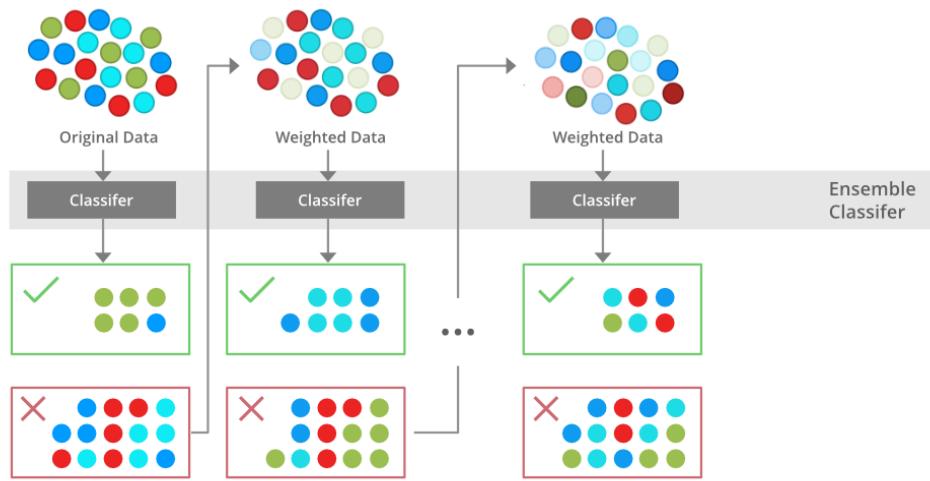
- **Tốc độ huấn luyện rất nhanh**: Nhờ phát triển cây Leaf-wise và histogram, LightGBM thường nhanh hơn các thư viện boosting khác như XGBoost.
- **Hiệu suất cao**: Cải thiện độ chính xác so với nhiều mô hình boosting truyền thống.
- **Xử lý tốt dữ liệu lớn và phức tạp**: Thích hợp với các tập dữ liệu có nhiều đặc trưng và số lượng mẫu lớn.
- **Hỗ trợ các loại dữ liệu đa dạng**: Bao gồm dữ liệu thiếu, dữ liệu phân tán.
- **Tiêu thụ bộ nhớ thấp**: Thiết kế hiệu quả giúp tiết kiệm tài nguyên.

Nhược điểm

- **Dễ bị overfitting nếu không điều chỉnh đúng**: Do phát triển theo lá có sai số lớn nhất, nếu không kiểm soát tốt, mô hình có thể học quá mức trên dữ liệu huấn luyện.
- **Khó điều chỉnh tham số**: LightGBM có nhiều tham số cần tinh chỉnh để đạt hiệu quả tối ưu.

- **Ít trực quan:** Mô hình phức tạp, khó giải thích chi tiết từng quyết định như các cây quyết định đơn lẻ.
- **Cần dữ liệu định dạng tốt:** Mặc dù hỗ trợ nhiều dạng dữ liệu, nhưng dữ liệu cần được chuẩn bị và làm sạch kỹ để tránh lỗi.

3.1.9. XGBoost



Giới thiệu mô hình XGBoost

XGBoost (eXtreme Gradient Boosting) là một thư viện học máy nâng cao thuộc nhóm **gradient boosting** được thiết kế để đạt hiệu suất cao, tốc độ nhanh và khả năng mở rộng tốt. XGBoost mở rộng và tối ưu phương pháp Gradient Boosting Decision Trees (GBDT) truyền thống bằng nhiều cải tiến về thuật toán, cấu trúc dữ liệu và khả năng xử lý song song, giúp mô hình trở thành một trong những công cụ phổ biến nhất trong các cuộc thi khoa học dữ liệu và ứng dụng thực tế.

Cách Hoạt động

- XGBoost xây dựng mô hình bằng cách kết hợp tuần tự nhiều cây quyết định (decision trees) nhỏ, trong đó mỗi cây mới được huấn luyện để giảm sai số còn lại (residual) của các cây trước đó theo hướng gradient.

- Thuật toán sử dụng hàm mất mát và kỹ thuật tối ưu hóa bậc hai (second-order gradient) để tăng tốc quá trình huấn luyện và cải thiện độ chính xác.
- Hỗ trợ nhiều loại hàm mất mát, phù hợp với bài toán phân loại, hồi quy và xếp hạng.
- Tối ưu về bộ nhớ và hỗ trợ tính toán song song giúp tăng tốc độ xử lý.
- Có thể xử lý dữ liệu thiếu và dữ liệu phân tán tốt.

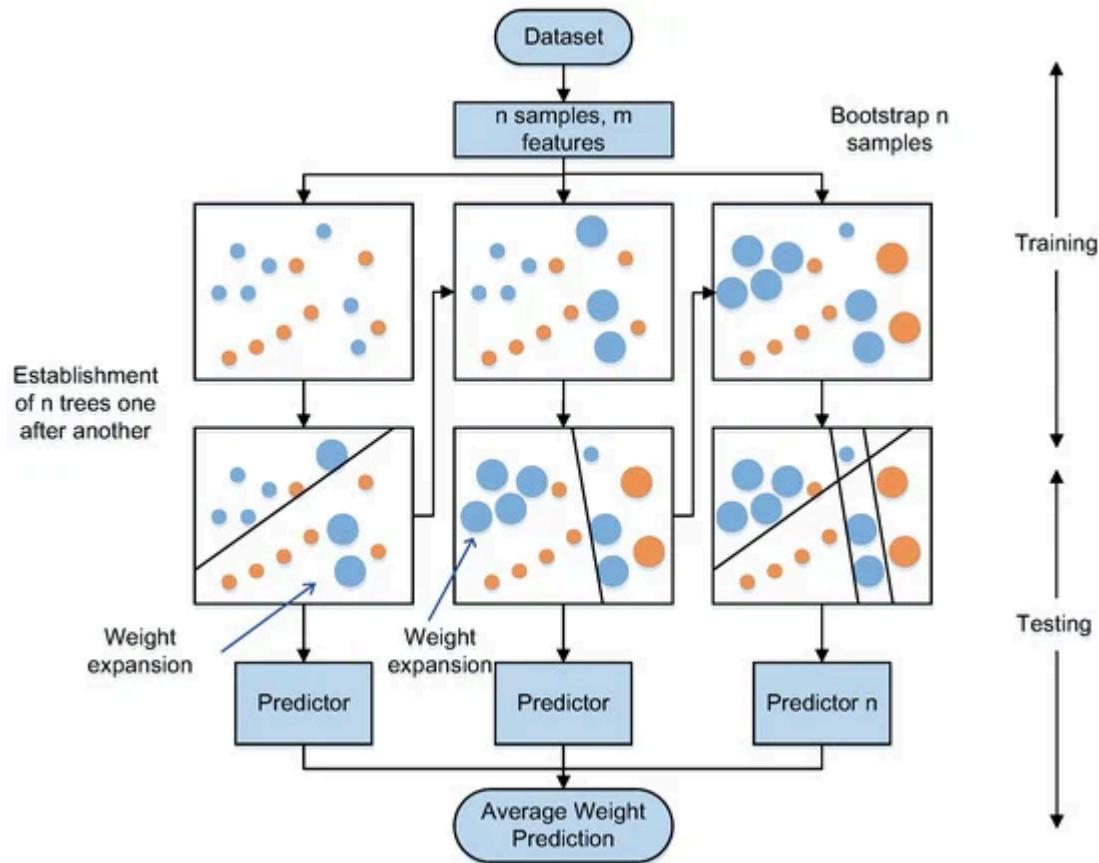
Ưu điểm

- **Hiệu quả cao và độ chính xác tốt:** XGBoost thường cho kết quả dự đoán chính xác hơn nhiều mô hình khác nhờ kỹ thuật boosting và tối ưu hàm mất mát.
- **Tốc độ huấn luyện nhanh:** Các tối ưu thuật toán và tính toán song song giúp tăng tốc độ huấn luyện đáng kể.
- **Khả năng xử lý dữ liệu phức tạp:** Thích hợp với các tập dữ liệu lớn, nhiều đặc trưng và dữ liệu thiếu.
- **Hỗ trợ kỹ thuật regularization:** Giúp giảm overfitting và tăng khả năng tổng quát hóa.
- **Dễ dàng tích hợp và sử dụng:** Hỗ trợ nhiều ngôn ngữ lập trình như Python, R, Java, C++.

Nhược điểm

- **Phức tạp và khó giải thích:** Do tính chất ensemble và nhiều cây kết hợp, mô hình khó để trực tiếp hiểu và giải thích chi tiết từng quyết định.
- **Cần điều chỉnh nhiều tham số:** Để đạt hiệu suất tối ưu, người dùng cần tinh chỉnh nhiều tham số như learning rate, số lượng cây, độ sâu cây,...
- **Tiêu thụ tài nguyên:** Khi làm việc với dữ liệu rất lớn, mô hình có thể yêu cầu bộ nhớ và CPU cao.
- **Có thể bị overfitting nếu không kiểm soát:** Nếu không sử dụng các kỹ thuật regularization hoặc early stopping, mô hình dễ học quá mức.

3.1.10. CatBoost



Giới thiệu mô hình CatBoost

CatBoost (Categorical Boosting) là một thư viện học máy nâng cao thuộc nhóm **gradient boosting** được phát triển bởi Yandex. Điểm nổi bật của CatBoost là khả năng xử lý hiệu quả các biến phân loại (categorical features) mà không cần phải biến đổi thủ công thành dạng số, đồng thời cải thiện hiệu suất và độ chính xác của mô hình. CatBoost được thiết kế để giảm thiểu overfitting và tối ưu tốc độ học, phù hợp với nhiều bài toán phân loại và hồi quy.

Cách Hoạt động

- CatBoost xây dựng mô hình dựa trên phương pháp **gradient boosting** trên cây quyết định.
- Sử dụng kỹ thuật **ordered boosting** để giảm thiểu bias do overfitting trong quá trình huấn luyện.

- Áp dụng phương pháp mã hóa đặc biệt cho biến phân loại (categorical features), tránh gây sai lệch thông tin khi biến đổi dữ liệu.
- Hỗ trợ các hàm mất mát khác nhau phù hợp với phân loại, hồi quy, và các bài toán ranking.
- Tối ưu hóa bộ nhớ và tăng tốc độ huấn luyện thông qua các thuật toán phân tán và song song.

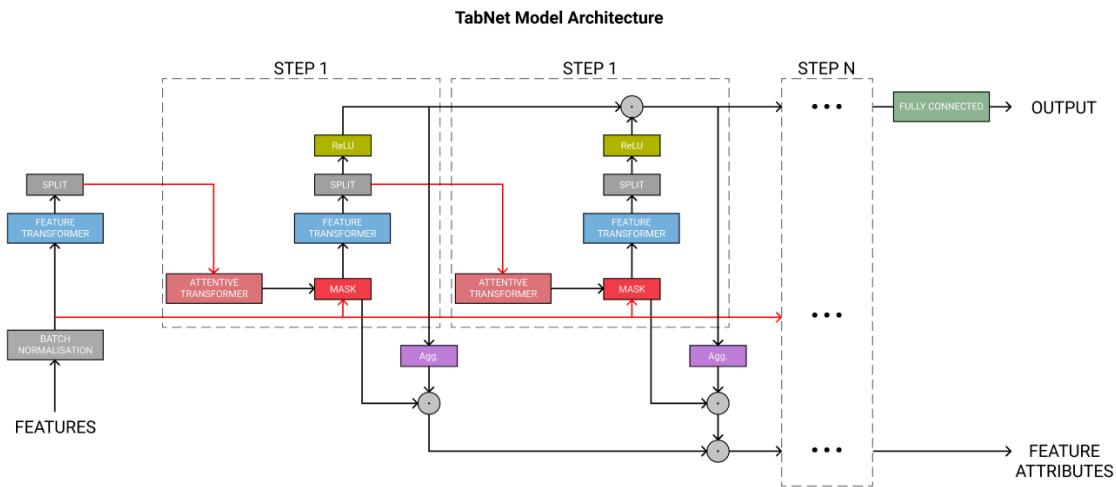
Ưu điểm

- **Xử lý biến phân loại trực tiếp:** Không cần mã hóa thủ công, giúp giảm công sức chuẩn bị dữ liệu và tăng độ chính xác.
- **Giảm overfitting hiệu quả:** Ordered boosting giúp hạn chế hiện tượng học quá mức.
- **Hiệu suất cao và tốc độ nhanh:** Thích hợp cho cả dữ liệu lớn và phức tạp.
- **Tự động xử lý missing values:** Mô hình có khả năng xử lý dữ liệu thiếu hiệu quả.
- **Hỗ trợ nhiều loại bài toán:** Phân loại, hồi quy, ranking,...

Nhược điểm

- **Khó điều chỉnh tham số:** Có nhiều tham số cần tinh chỉnh để đạt hiệu quả tốt nhất.
- **Mô hình phức tạp:** Giống các mô hình boosting khác, CatBoost khó để trực tiếp giải thích và hiểu rõ cấu trúc.
- **Tiêu thụ tài nguyên:** Đối với tập dữ liệu rất lớn, quá trình huấn luyện vẫn có thể đòi hỏi bộ nhớ và CPU cao.
- **Cần dữ liệu chuẩn bị tốt:** Mặc dù xử lý biến phân loại trực tiếp, dữ liệu vẫn cần được làm sạch và kiểm tra kỹ.

3.1.11. Tabnet Classifier



Giới thiệu mô hình TabNet Classifier

TabNet là một kiến trúc mạng nơ-ron sâu chuyên biệt cho dữ liệu bảng (tabular data), được giới thiệu bởi Google Research. TabNet sử dụng cơ chế **attention** để tự động chọn các đặc trưng quan trọng trong từng bước xử lý dữ liệu, giúp mô hình học được các mối quan hệ phức tạp giữa các đặc trưng mà vẫn giữ được tính minh bạch và giải thích được phần nào quyết định của mô hình. TabNet được thiết kế đặc biệt để cạnh tranh và vượt trội so với các mô hình truyền thống như cây quyết định hoặc các mô hình boosting trên dữ liệu bảng.

Cách Hoạt động

- TabNet sử dụng cơ chế **sequential attention** để lựa chọn từng nhóm đặc trưng trong mỗi bước của mạng, thay vì sử dụng toàn bộ đặc trưng một lần.
- Quá trình này giúp mô hình tập trung vào các đặc trưng quan trọng nhất cho từng quyết định, giảm nhiễu và tăng khả năng học sâu.
- Mạng gồm các bước xử lý tuần tự, mỗi bước gồm một module attention và một module feature transformer (dựa trên mạng fully-connected).
- Hệ thống attention được học để quyết định trọng số cho từng đặc trưng, tạo ra sự giải thích về đặc trưng được chọn.

- TabNet kết hợp các đặc trưng đã chọn qua các bước để đưa ra dự đoán cuối cùng.

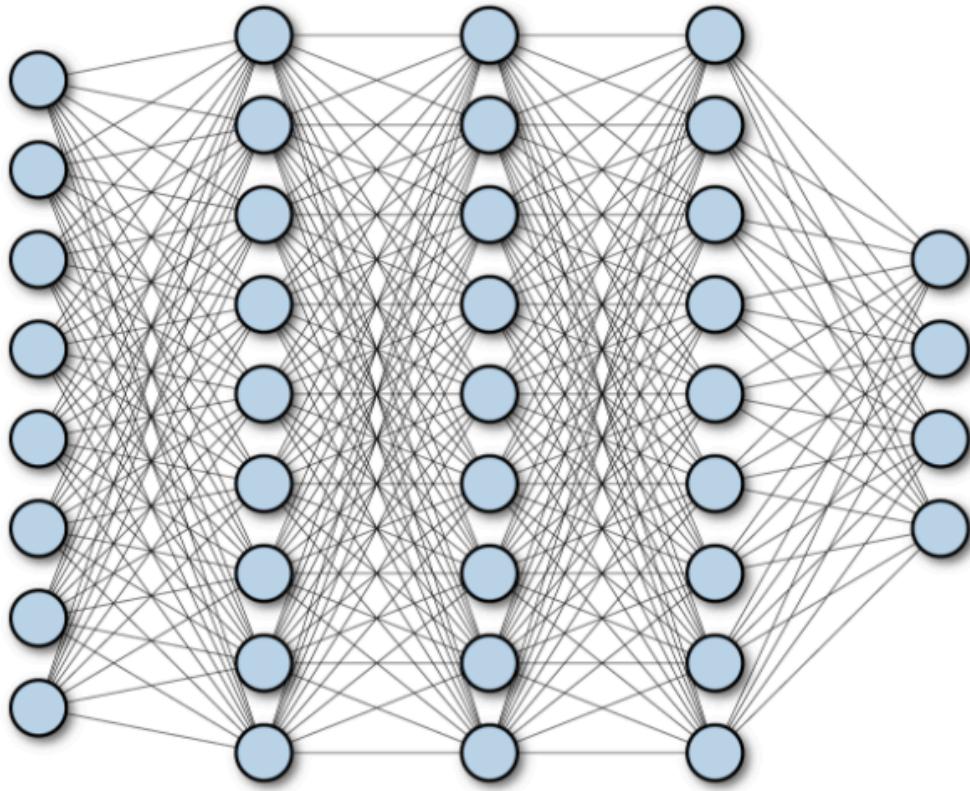
Ưu điểm

- **Hiệu quả trên dữ liệu bảng:** TabNet vượt trội hơn nhiều mô hình truyền thống và mạng nơ-ron thông thường khi xử lý dữ liệu dạng bảng.
- **Khả năng tự chọn đặc trưng:** Cơ chế attention giúp mạng chọn lọc đặc trưng quan trọng tự động, giảm cần thiết phải xử lý đặc trưng phức tạp thủ công.
- **Giải thích được mô hình:** Nhờ trọng số attention, TabNet cung cấp phần nào khả năng hiểu được ảnh hưởng của các đặc trưng đến kết quả.
- **Khả năng mở rộng:** Có thể áp dụng cho các bài toán phân loại, hồi quy trên tập dữ liệu lớn.
- **Không cần nhiều bước tiền xử lý:** Giảm yêu cầu chuẩn hóa và biến đổi phức tạp.

Nhược điểm

- **Mô hình phức tạp:** TabNet có kiến trúc khá phức tạp, đòi hỏi kiến thức sâu về mạng nơ-ron và attention để tinh chỉnh tốt.
- **Thời gian huấn luyện lâu hơn:** So với các mô hình cây hoặc boosting truyền thống, TabNet thường cần thời gian huấn luyện nhiều hơn.
Yêu cầu tài nguyên cao: Mạng nơ-ron sâu như TabNet thường cần GPU và bộ nhớ lớn để huấn luyện hiệu quả.
- **Cần dữ liệu đủ lớn:** Để mô hình phát huy tối đa hiệu quả, dữ liệu huấn luyện cần đủ phong phú và đa dạng.

3.1.12. Fully-Connected NN



Giới thiệu mô hình Fully-Connected Neural Network (FCNN)

Fully-Connected Neural Network (FCNN), còn gọi là Mạng nơ-ron đa lớp (Multilayer Perceptron - MLP), là một kiến trúc mạng nơ-ron nhân tạo cơ bản và phổ biến trong học sâu. FCNN bao gồm các lớp neuron được kết nối hoàn toàn (fully connected) với lớp kế tiếp, nghĩa là mỗi neuron ở lớp trước liên kết với tất cả các neuron ở lớp sau. Mô hình này được sử dụng rộng rãi trong các bài toán phân loại, hồi quy và nhiều tác vụ học máy khác.

Cách Hoạt động

- FCNN gồm các lớp:
 - **Lớp đầu vào (Input layer):** nhận dữ liệu đầu vào dưới dạng vector đặc trưng.

- **Các lớp ẩn (Hidden layers):** mỗi neuron trong lớp ẩn tính toán tổng trọng số có trọng số và áp dụng hàm kích hoạt (activation function) như ReLU, Sigmoid hoặc Tanh để tạo ra đầu ra phi tuyến.
- **Lớp đầu ra (Output layer):** tùy thuộc vào bài toán, có thể dùng hàm Softmax cho phân loại đa lớp hoặc hàm tuyến tính cho hồi quy.
- Quá trình huấn luyện sử dụng thuật toán **lai truyền ngược (backpropagation)** kết hợp với tối ưu hóa (như Gradient Descent) để cập nhật trọng số nhằm giảm thiểu hàm mất mát.
- Mạng học được các mối quan hệ phi tuyến phức tạp giữa các đặc trưng nhờ các lớp ẩn với các hàm kích hoạt phi tuyến.

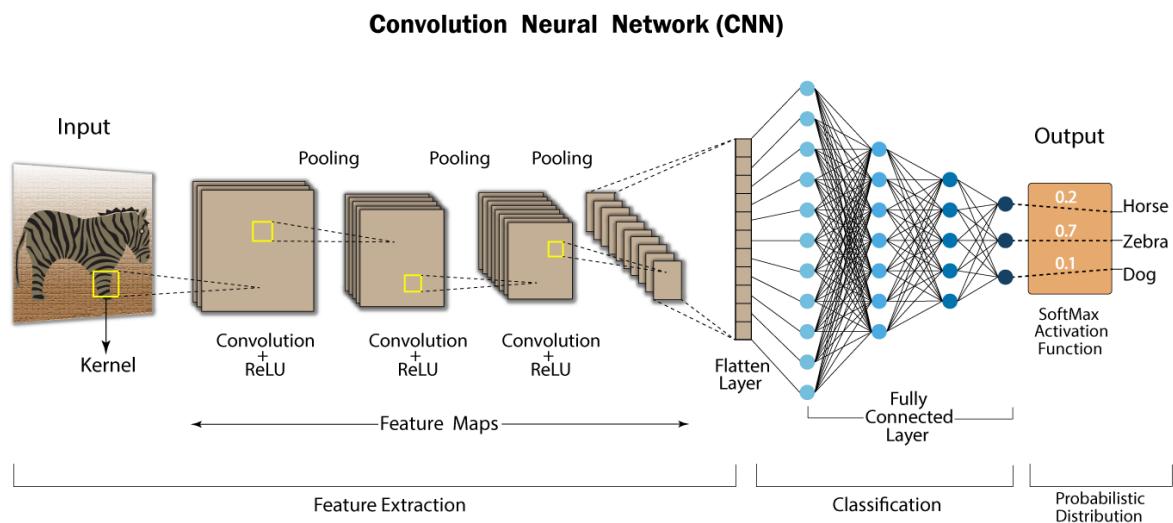
Ưu điểm

- **Khả năng biểu diễn phi tuyến:** FCNN có thể học và biểu diễn các mối quan hệ phi tuyến phức tạp trong dữ liệu.
- **Linh hoạt:** Dễ dàng mở rộng về kiến trúc (số lớp, số neuron) tùy theo bài toán.
- **Ứng dụng rộng rãi:** Phù hợp với nhiều dạng dữ liệu và bài toán khác nhau.
- **Dễ tích hợp với các kỹ thuật học sâu khác:** Có thể kết hợp với CNN, RNN hoặc các kiến trúc phức tạp hơn.

Nhược điểm

- **Dễ bị overfitting:** Đặc biệt khi mạng quá sâu hoặc dữ liệu ít, cần kỹ thuật regularization như dropout, early stopping.
- **Yêu cầu dữ liệu lớn:** Hiệu quả tốt khi có lượng dữ liệu huấn luyện lớn.
- **Tính toán phức tạp:** Mạng sâu lớn cần nhiều tài nguyên tính toán và thời gian huấn luyện.
- **Thiếu khả năng xử lý cấu trúc dữ liệu đặc biệt:** FCNN không phù hợp trực tiếp với dữ liệu có cấu trúc không gian (hình ảnh) hoặc dữ liệu tuần tự (chuỗi thời gian) nếu không kết hợp các mô hình chuyên biệt.

3.1.13. CNN



Giới thiệu mô hình Convolutional Neural Network (CNN)

Convolutional Neural Network (CNN) là một loại mạng nơ-ron sâu được thiết kế đặc biệt để xử lý dữ liệu có cấu trúc dạng lưới như hình ảnh, video hoặc chuỗi thời gian. CNN nổi bật với khả năng tự động học các đặc trưng quan trọng qua các lớp tích chập (convolutional layers), giúp mô hình nhận dạng các mẫu phức tạp trong dữ liệu mà không cần phải xử lý thủ công.

Cách Hoạt động

- Lớp tích chập (Convolutional layer):** Sử dụng các bộ lọc (filters) trượt trên dữ liệu đầu vào để tạo ra các bản đồ đặc trưng (feature maps). Các bộ lọc này học được các đặc trưng như cạnh, góc, hoặc các cấu trúc phức tạp hơn ở các lớp sâu hơn.
- Lớp pooling (Pooling layer):** Giảm kích thước không gian của bản đồ đặc trưng, giữ lại các đặc trưng quan trọng, giúp giảm số lượng tham số và tính toán, đồng thời chống overfitting.
- Lớp Fully-connected:** Các lớp cuối cùng thường là các lớp kết nối đầy đủ, dùng để tổng hợp các đặc trưng đã học và thực hiện phân loại hoặc dự đoán.
- Hàm kích hoạt (Activation function):** Thường dùng ReLU để thêm tính phi tuyến vào mô hình.

- Mạng CNN được huấn luyện bằng thuật toán lan truyền ngược (backpropagation) và tối ưu hóa để cập nhật trọng số các bộ lọc.

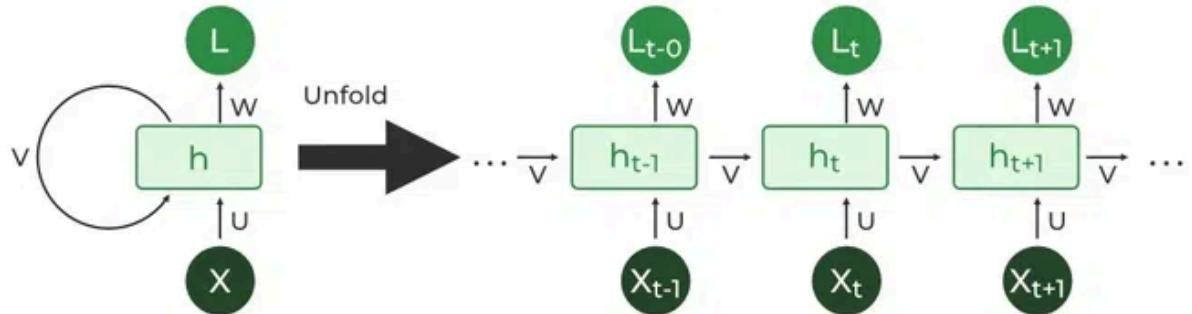
Ưu điểm

- **Tự động học đặc trưng:** CNN không cần phải thiết kế đặc trưng thủ công mà tự động học các bộ lọc đặc trưng quan trọng.
- **Hiệu quả trên dữ liệu hình ảnh và không gian:** Đặc biệt mạnh với các tác vụ nhận dạng hình ảnh, xử lý video, nhận dạng giọng nói.
- **Giảm số lượng tham số:** Nhờ dùng bộ lọc chia sẻ trọng số và pooling, CNN có số lượng tham số ít hơn so với mạng fully-connected cùng kích thước.
- **Khả năng tổng quát hóa tốt:** Có thể học các mẫu phức tạp và giảm thiểu overfitting khi được huấn luyện đúng cách.

Nhược điểm

- **Yêu cầu dữ liệu lớn:** CNN thường cần lượng dữ liệu lớn để học hiệu quả
- **Tính toán phức tạp:** Việc huấn luyện CNN đòi hỏi tài nguyên tính toán mạnh, đặc biệt là GPU.
- **Khó giải thích:** Mặc dù có thể hiểu được qua visualization, nhưng CNN vẫn còn là “hộp đen” khó giải thích chi tiết quyết định.
- **Chỉ phù hợp với dữ liệu có cấu trúc lưới:** CNN không phù hợp cho dữ liệu phi cấu trúc hoặc dữ liệu dạng bảng nếu không biến đổi phù hợp

3.1.14. RNN



Giới thiệu mô hình Recurrent Neural Network (RNN)

Recurrent Neural Network (RNN) là một loại mạng nơ-ron sâu được thiết kế đặc biệt để xử lý dữ liệu tuần tự như chuỗi thời gian, ngôn ngữ tự nhiên, hay các tín hiệu âm thanh. Điểm đặc biệt của RNN là khả năng **ghi nhớ thông tin từ các bước trước đó** thông qua các kết nối vòng lặp, giúp mô hình hiểu và xử lý mối quan hệ tuần tự trong dữ liệu.

Cách Hoạt động

- RNN nhận đầu vào theo từng bước thời gian t , mỗi bước gồm một vector đặc trưng x_{t-1} .
- Mạng duy trì một trạng thái ẩn h_{t-1} tại bước thời gian $t-1$, được tính dựa trên trạng thái ẩn trước đó h_{t-1} và đầu vào hiện tại x_{t-1} :

$$h_t = f(W_{xh}x_t + W_{hh}h_{t-1} + b_h)$$

trong đó f là hàm kích hoạt phi tuyến như tanh hoặc ReLU.

- Dựa trên trạng thái ẩn h_{t-1} , mạng có thể dự đoán đầu ra y_{t+1} tại bước thời gian $t+1$.

- RNN được huấn luyện qua thuật toán lan truyền ngược qua thời gian (Backpropagation Through Time - BPTT), cập nhật các trọng số dựa trên lỗi tổng hợp từ toàn bộ chuỗi.

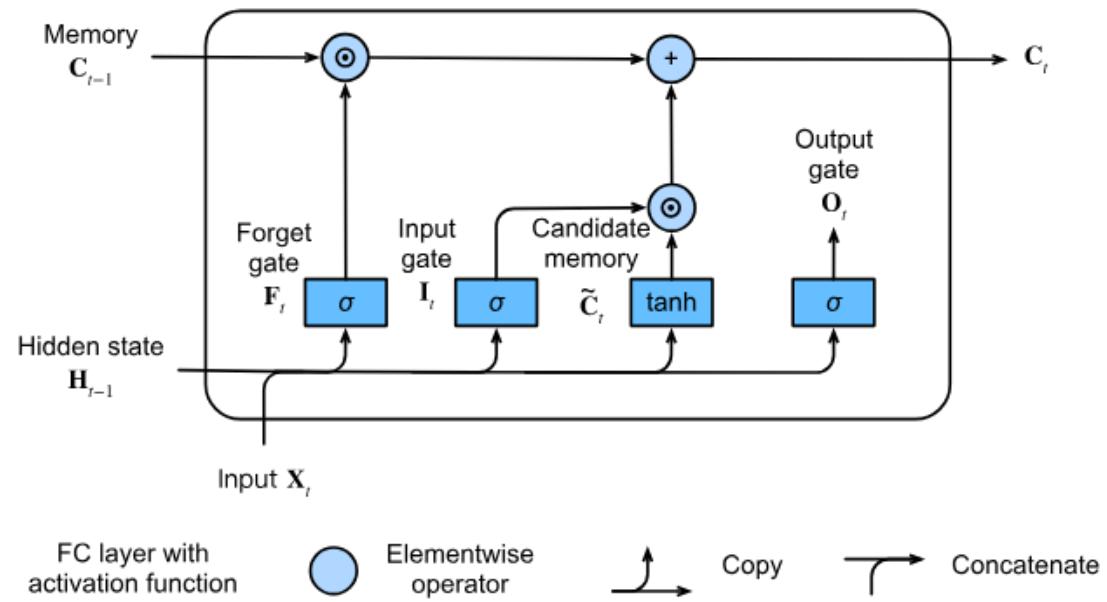
Ưu điểm

- **Xử lý tốt dữ liệu tuần tự:** RNN có khả năng ghi nhớ thông tin từ các bước trước để dự đoán bước hiện tại.
Áp dụng rộng rãi: Dùng trong dịch máy, nhận dạng giọng nói, phân tích chuỗi thời gian, sinh văn bản,...
- **Kiến trúc đơn giản:** Dễ dàng mở rộng và kết hợp với các mô hình khác.

Nhược điểm

- **Khó học các phụ thuộc dài hạn:** RNN truyền thông gấp vấn đề biến mất hoặc bùng nổ gradient khi chuỗi dài, khiến việc học các mối quan hệ dài hạn rất khó khăn.
- **Tính toán chậm:** Phải xử lý tuần tự từng bước thời gian, không dễ dàng song song hóa.
- **Dễ bị overfitting:** Nếu không có kỹ thuật regularization thích hợp.
- **Cần dữ liệu lớn và huấn luyện kỹ:** Để đạt hiệu quả tốt, mạng cần nhiều dữ liệu và thời gian huấn luyện.

3.1.15. LSTM



Giới thiệu mô hình LSTM

LSTM (Long Short-Term Memory) là một biến thể nâng cao của mạng Recurrent Neural Network (RNN), được thiết kế để giải quyết vấn đề **biến mất gradient** trong RNN truyền thống khi học các phụ thuộc dài hạn trong chuỗi dữ liệu. LSTM có cấu trúc đặc biệt với các **cổng điều khiển** giúp mô hình có thể giữ hoặc loại bỏ thông tin quan trọng qua nhiều bước thời gian, từ đó cải thiện khả năng ghi nhớ lâu dài và xử lý dữ liệu tuần tự phức tạp.

Cách Hoạt động

- LSTM gồm các **cell** chứa trạng thái nội bộ (cell state) C_t và trạng thái ẩn h_t .
- Ba cổng chính điều khiển thông tin đi qua cell:
 - **Forget gate (Cổng quên)**: quyết định thông tin nào từ trạng thái cũ sẽ bị loại bỏ.
 - **Input gate (Cổng đầu vào)**: quyết định thông tin mới nào sẽ được thêm vào trạng thái.

- **Output gate (Cổng đầu ra):** quyết định phần thông tin nào từ trạng thái cell sẽ được xuất ra làm trạng thái ẩn hiện tại.
- Các cổng này được tính toán dựa trên đầu vào hiện tại và trạng thái ẩn trước đó, sử dụng các hàm sigmoid và tanh để điều chỉnh giá trị.
- Trạng thái cell được cập nhật qua các bước thời gian, cho phép LSTM giữ lại thông tin quan trọng lâu dài.
- Quá trình huấn luyện LSTM cũng dùng thuật toán lan truyền ngược qua thời gian (BPTT).

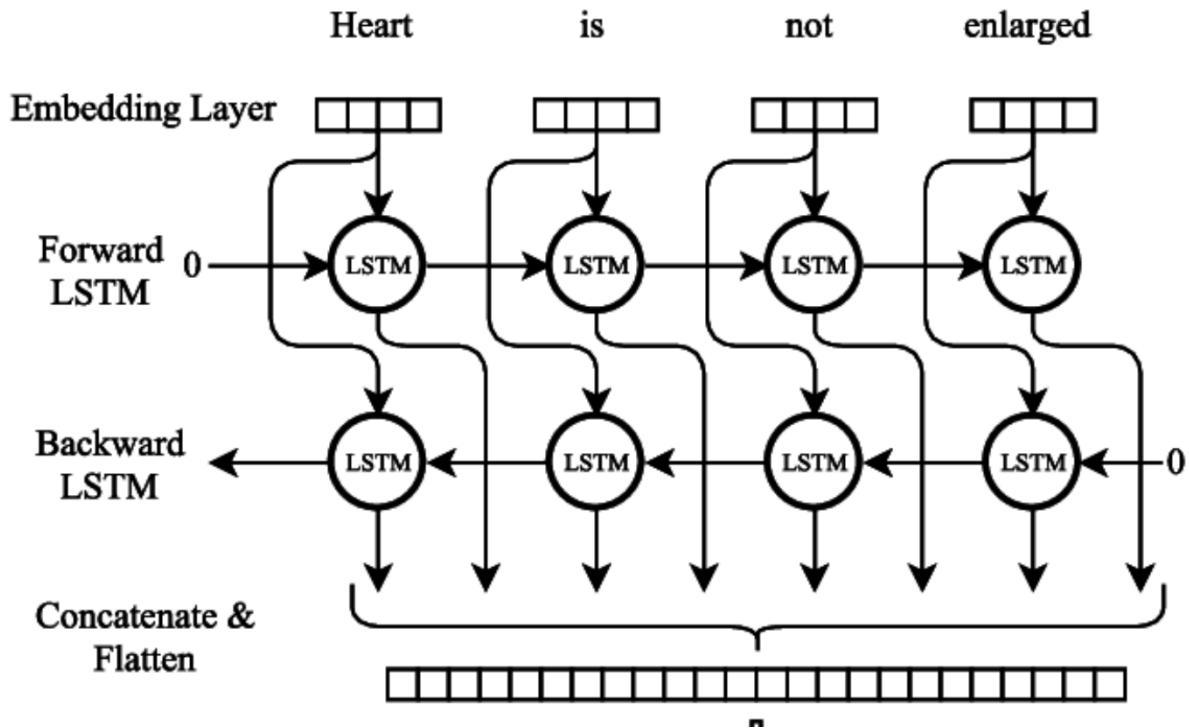
Ưu điểm

- **Giải quyết được vấn đề biến mất gradient:** Giúp mô hình học được các phụ thuộc dài hạn trong chuỗi dữ liệu.
- **Hiệu quả trên nhiều bài toán tuần tự:** Dịch máy, nhận dạng giọng nói, phân tích chuỗi thời gian, sinh văn bản,...
- **Khả năng ghi nhớ thông tin lâu dài:** LSTM có thể duy trì thông tin quan trọng qua nhiều bước thời gian.
- **Kiến trúc linh hoạt:** Có thể kết hợp với các mô hình khác như CNN, Attention,...

Nhược điểm

- **Mô hình phức tạp hơn RNN truyền thống:** Cấu trúc nhiều cổng làm tăng số lượng tham số và tính toán.
- **Thời gian huấn luyện lâu:** Cần tài nguyên tính toán lớn và thời gian huấn luyện dài hơn.
- **Dễ bị overfitting:** Cần sử dụng các kỹ thuật regularization như dropout.
- **Khó điều chỉnh tham số:** Việc tối ưu hóa LSTM đòi hỏi kinh nghiệm và thử nghiệm.

3.1.16. BiLSTM



Giới thiệu mô hình BiLSTM

BiLSTM (Bidirectional Long Short-Term Memory) là một biến thể mở rộng của mạng LSTM, cho phép mô hình học thông tin tuần tự theo cả hai chiều: từ quá khứ đến tương lai (forward direction) và từ tương lai đến quá khứ (backward direction). Điều này giúp mô hình nắm bắt được ngữ cảnh đầy đủ hơn trong dữ liệu tuần tự, đặc biệt hữu ích trong các bài toán như xử lý ngôn ngữ tự nhiên, nhận dạng giọng nói và phân tích chuỗi thời gian.

Cách Hoạt động

- BiLSTM gồm hai mạng LSTM chạy song song:
 - **Mạng LSTM tiến (forward LSTM):** xử lý chuỗi dữ liệu theo thứ tự thời gian từ bước 1 đến bước T.
 - **Mạng LSTM ngược (backward LSTM):** xử lý chuỗi dữ liệu theo chiều ngược lại từ bước T đến bước 1.

- Ở mỗi bước thời gian ttt, BiLSTM kết hợp đầu ra của cả hai mạng LSTM này (thường bằng cách ghép nối hoặc cộng) để tạo thành biểu diễn đặc trưng đầy đủ hơn.
- Kết quả này được sử dụng làm đầu vào cho các lớp tiếp theo hoặc cho quá trình dự đoán.

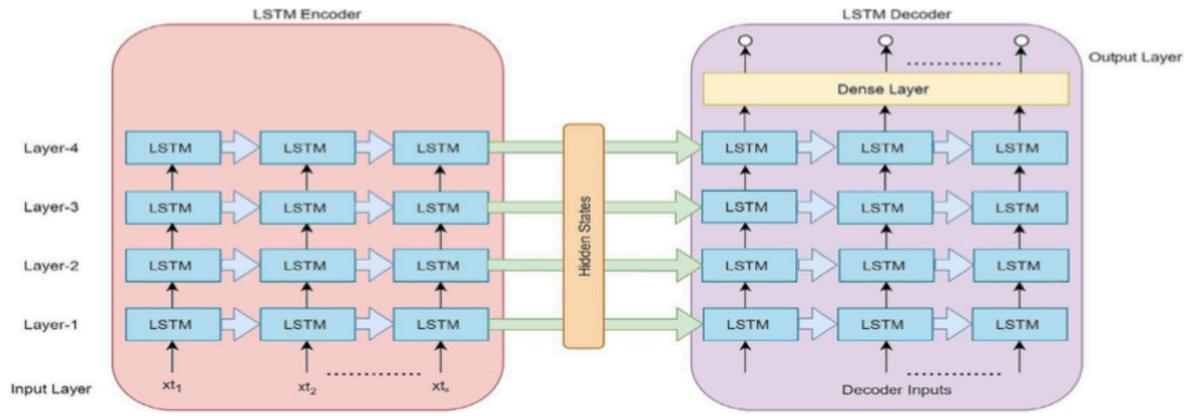
Ưu điểm

- **Hiểu ngữ cảnh toàn diện hơn:** Mô hình sử dụng thông tin cả quá khứ và tương lai trong chuỗi dữ liệu, cải thiện độ chính xác dự đoán.
- **Hiệu quả trên nhiều bài toán tuần tự:** Đặc biệt hữu ích với dữ liệu có tính ngữ cảnh phức tạp như xử lý ngôn ngữ tự nhiên, phân tích cảm xúc, nhận dạng giọng nói.
- **Dễ dàng tích hợp:** Có thể kết hợp với các mô hình khác như Attention, CNN để tăng hiệu quả.

Nhược điểm

- **Tính toán phức tạp và tốn kém hơn LSTM đơn chiều:** Vì phải huấn luyện và dự đoán song song hai mạng LSTM.
- **Yêu cầu tài nguyên lớn:** Cần nhiều bộ nhớ và thời gian huấn luyện hơn.
- **Khó điều chỉnh:** Việc tối ưu tham số mô hình phức tạp hơn do cấu trúc đôi chiều.
- **Không phù hợp cho các ứng dụng cần dự đoán theo thời gian thực với độ trễ thấp:** Vì phải xử lý toàn bộ chuỗi cả chiều thuận và nghịch.

3.1.17. 4-layer Stacked LSTM



Giới thiệu mô hình 4-layer Stacked LSTM

Mạng **Stacked LSTM** là một kiến trúc mở rộng của mạng LSTM cơ bản, trong đó nhiều lớp LSTM được xếp chồng lên nhau (stacked) để tăng khả năng học các đặc trưng phức tạp và biểu diễn dữ liệu sâu hơn. Mô hình **4-layer Stacked LSTM** bao gồm bốn lớp LSTM liên tiếp, mỗi lớp lấy đầu ra của lớp trước làm đầu vào cho lớp kế tiếp. Kiến trúc này thường được sử dụng trong các bài toán tuần tự phức tạp như dịch máy, nhận dạng giọng nói, và phân tích chuỗi thời gian dài.

Cách Hoạt động

- Dữ liệu tuần tự đầu vào được truyền qua lớp LSTM đầu tiên, lớp này sẽ học các đặc trưng ở mức độ thấp hơn.
- Đầu ra của lớp LSTM thứ nhất được chuyển tiếp làm đầu vào cho lớp LSTM thứ hai, giúp học các đặc trưng phức tạp hơn.
- Quá trình này tiếp tục đến lớp LSTM thứ tư, lớp cuối cùng trong chuỗi, cho ra biểu diễn sâu sắc nhất.
- Ở mỗi bước thời gian, trạng thái ẩn và trạng thái cell được cập nhật qua từng lớp.
- Mạng được huấn luyện bằng phương pháp lan truyền ngược qua thời gian (BPTT), cập nhật trọng số của tất cả các lớp cùng lúc.

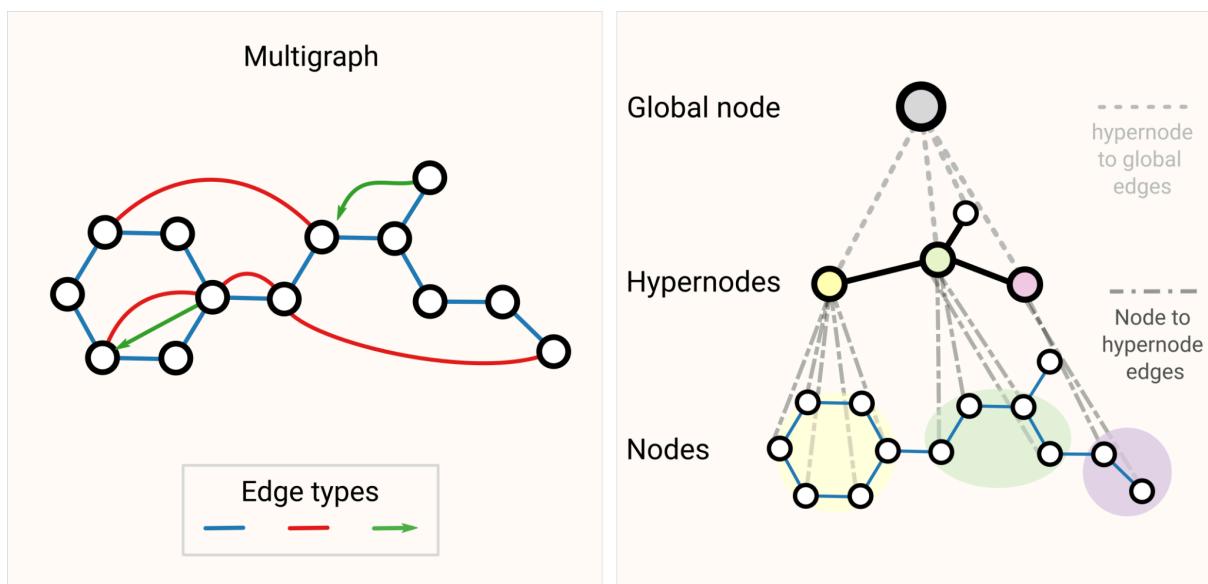
Ưu điểm

- **Khả năng học biểu diễn phức tạp:** Việc xếp nhiều lớp giúp mô hình nắm bắt các mối quan hệ dài hạn và cấu trúc sâu trong dữ liệu tuần tự.
- **Hiệu quả cao trên dữ liệu dài:** Thích hợp với các chuỗi dữ liệu có độ dài và độ phức tạp lớn.
- **Linh hoạt trong thiết kế:** Có thể tùy chỉnh số lượng lớp, kích thước ẩn phù hợp với bài toán.

Nhược điểm

- **Tăng độ phức tạp tính toán:** Nhiều lớp LSTM dẫn đến mô hình nặng nề, tốn nhiều tài nguyên tính toán và thời gian huấn luyện.
- **Dễ bị overfitting:** Mạng sâu có nguy cơ học quá mức nếu không có đủ dữ liệu hoặc kỹ thuật điều chỉnh.
- **Khó tối ưu và điều chỉnh:** Việc huấn luyện nhiều lớp LSTM cần kỹ thuật và kinh nghiệm để chọn tham số phù hợp.
- **Yêu cầu bộ nhớ lớn:** Đặc biệt khi xử lý chuỗi dài và kích thước lớp lớn.

3.1.18. Graph NN



Giới thiệu mô hình Graph Neural Network (GNN)

Graph Neural Network (GNN) là một loại mạng nơ-ron sâu được thiết kế đặc biệt để xử lý dữ liệu có cấu trúc đồ thị (graph-structured data), nơi dữ liệu được biểu diễn dưới dạng các nút (nodes) và các cạnh (edges) kết nối giữa chúng. GNN cho phép học biểu diễn hiệu quả cho các đối tượng có mối quan hệ phức tạp, như mạng xã hội, phân tử hóa học, hệ thống giao thông, và nhiều ứng dụng khác.

Cách Hoạt động

- GNN hoạt động bằng cách lặp đi lặp lại quá trình **lan truyền thông tin** (message passing) giữa các nút trong đồ thị:
 - Mỗi nút nhận thông tin từ các nút láng giềng qua các cạnh.
 - Thông tin này được tổng hợp và kết hợp với đặc trưng hiện tại của nút để cập nhật biểu diễn (embedding) của nút đó.
- Quá trình này được thực hiện qua nhiều tầng (layers), cho phép nút học biểu diễn dựa trên ngữ cảnh của các nút láng giềng trong đồ thị.
- Các hàm tổng hợp phổ biến gồm: hàm tổng (sum), trung bình (mean), hoặc hàm max.
- Cuối cùng, các biểu diễn nút có thể được sử dụng cho các tác vụ phân loại nút, phân loại cạnh, hoặc phân loại đồ thị.

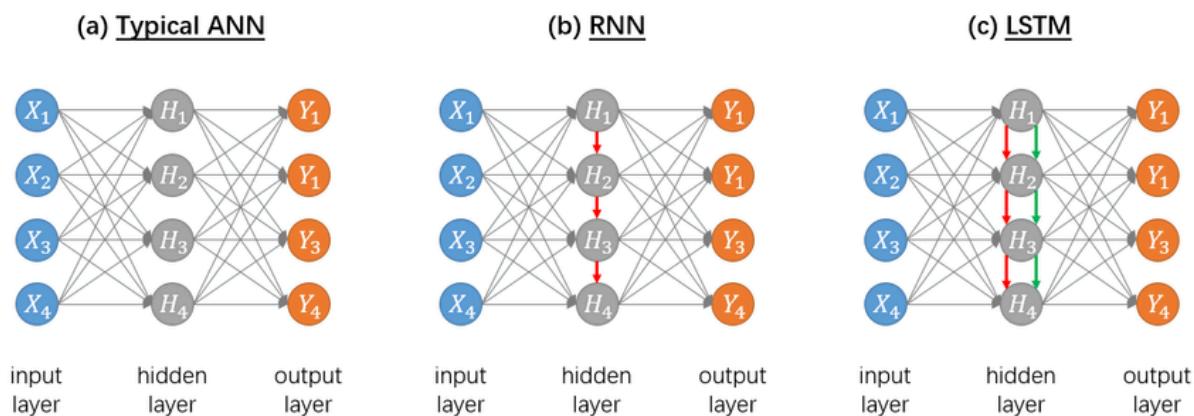
Ưu điểm

- **Xử lý hiệu quả dữ liệu đồ thị:** GNN khai thác được cấu trúc quan hệ phức tạp trong dữ liệu không theo dạng bảng hay lưới.
- **Linh hoạt:** Có thể áp dụng cho nhiều dạng đồ thị khác nhau, từ đồ thị có hướng, vô hướng đến đồ thị có trọng số.
- **Hiệu quả trong nhiều ứng dụng thực tế:** Như dự đoán thuộc tính phân tử, phân tích mạng xã hội, hệ thống đề xuất, và nhiều lĩnh vực khoa học khác.
- **Tự động học biểu diễn:** Giúp thay thế các đặc trưng thủ công bằng biểu diễn học sâu phù hợp với cấu trúc dữ liệu.

Nhược điểm

- **Phức tạp về mặt tính toán:** Việc lan truyền và cập nhật thông tin trong đồ thị lớn có thể tốn nhiều tài nguyên.
- **Khó mở rộng:** Đồ thị rất lớn hoặc có cấu trúc phức tạp gây khó khăn cho việc huấn luyện và triển khai.
- **Thiếu dữ liệu nhãn:** Các bài toán đồ thị thường thiếu dữ liệu nhãn, làm giảm hiệu quả huấn luyện.
- **Khó giải thích:** Các biểu diễn ẩn của nút không dễ để trực tiếp giải thích ý nghĩa.

3.1.19. ANN-LSTM



Giới thiệu mô hình ANN-LSTM

Mô hình **ANN-LSTM** là sự kết hợp giữa mạng nơ-ron nhân tạo truyền thống (Artificial Neural Network - ANN) và mạng LSTM (Long Short-Term Memory), nhằm tận dụng ưu điểm của cả hai loại mạng để xử lý dữ liệu có tính tuần tự phức tạp. Cấu trúc này thường được dùng trong các bài toán dự đoán chuỗi thời gian, phân tích dữ liệu tuần tự kết hợp với các đặc trưng tĩnh hoặc phi tuần tự.

Cách Hoạt động

- **Phản ANN:** Đầu vào ban đầu (có thể là dữ liệu phi tuần tự, đặc trưng tĩnh) được xử lý qua các lớp fully connected của ANN để trích xuất đặc trưng và biến đổi dữ liệu.

- **Phản LSTM:** Dữ liệu sau khi xử lý qua ANN được đưa vào mạng LSTM để học các phụ thuộc tuần tự, ghi nhớ thông tin quan trọng theo thời gian.
- Kết quả của LSTM sẽ được đưa qua các lớp fully connected cuối cùng để thực hiện dự đoán hoặc phân loại.
- Mô hình được huấn luyện bằng thuật toán lan truyền ngược và tối ưu hóa để giảm sai số dự đoán.

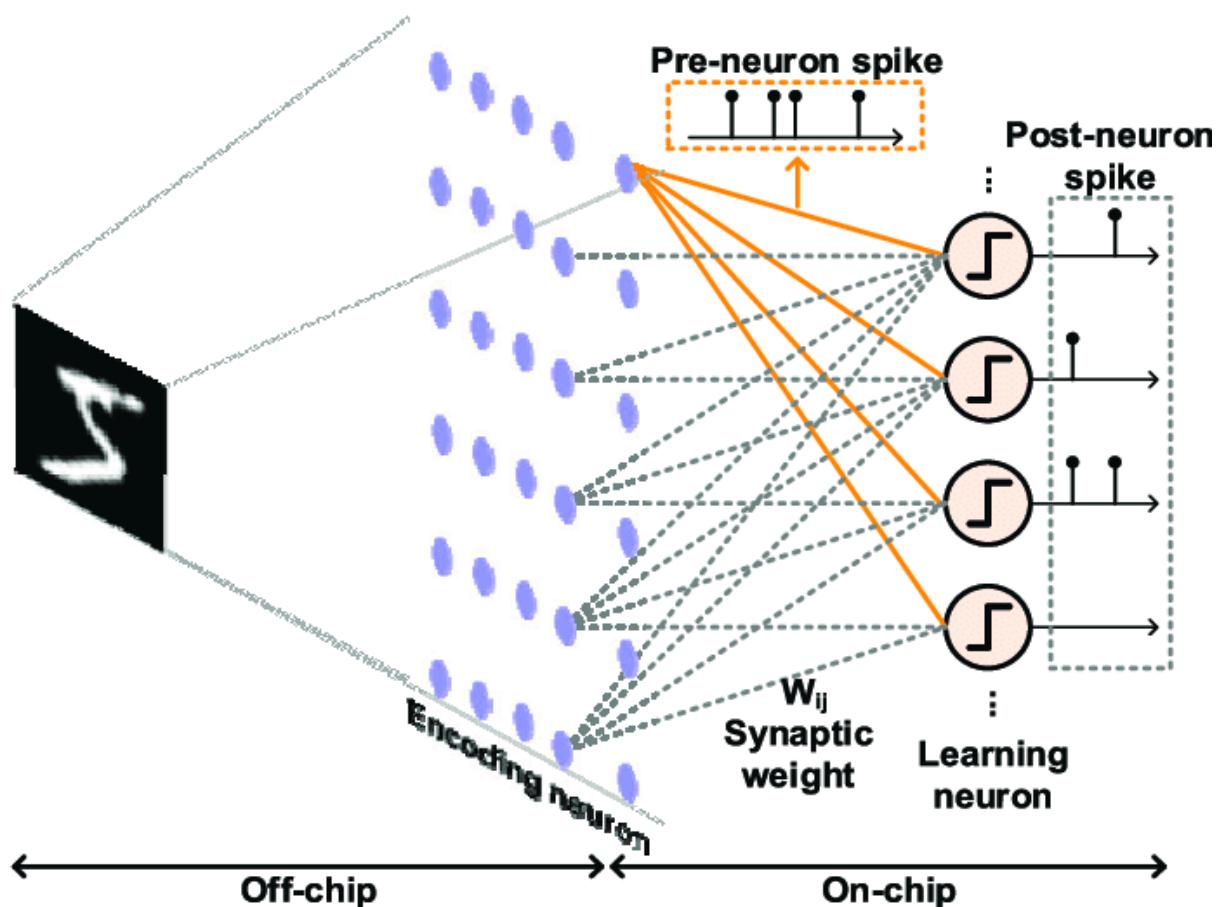
Ưu điểm

- **Kết hợp tốt đặc trưng phi tuần tự và tuần tự:** ANN giúp trích xuất đặc trưng mạnh mẽ, trong khi LSTM học được mối quan hệ thời gian.
- **Hiệu quả trên dữ liệu phức tạp:** Đặc biệt hữu ích với dữ liệu kết hợp dạng bảng và chuỗi thời gian.
- **Linh hoạt:** Có thể điều chỉnh cấu trúc để phù hợp với nhiều loại dữ liệu và bài toán.
- **Tăng khả năng dự đoán:** So với sử dụng riêng lẻ ANN hoặc LSTM, mô hình kết hợp thường cho hiệu quả tốt hơn.

Nhược điểm

- **Tăng độ phức tạp mô hình:** Việc kết hợp hai loại mạng làm tăng số lượng tham số và độ phức tạp tính toán.
- **Yêu cầu dữ liệu lớn:** Để mô hình học tốt, cần lượng dữ liệu đủ lớn và đa dạng.
- **Khó điều chỉnh tham số:** Cần thử nghiệm kỹ để chọn cấu trúc lớp, số lượng neuron, và tham số huấn luyện phù hợp.
- **Thời gian huấn luyện dài hơn:** Mô hình phức tạp hơn nên mất nhiều thời gian huấn luyện.

3.1.20. SNN



Giới thiệu mô hình Spiking Neural Network (SNN)

Spiking Neural Network (SNN) là một loại mạng nơ-ron sinh học mô phỏng hoạt động điện sinh học của các tế bào thần kinh trong não bộ. Khác với các mạng nơ-ron truyền thống sử dụng các giá trị liên tục, SNN truyền tín hiệu dưới dạng các **xung điện (spikes)** theo thời gian, giúp mô hình có khả năng xử lý thông tin theo cách gần giống với hệ thần kinh sinh học thật. SNN được xem là bước tiến trong lĩnh vực tính toán thần kinh và có tiềm năng lớn trong các ứng dụng đòi hỏi xử lý thông tin thời gian thực và tiết kiệm năng lượng.

Cách Hoạt động

- Mỗi neuron trong SNN duy trì một trạng thái điện thế màng (membrane potential).

- Khi điện thế màng vượt qua ngưỡng định trước, neuron phát ra một xung điện (spike) và reset trạng thái.
- Các xung điện truyền từ neuron này sang neuron khác qua các kết nối synapse với trọng số tương ứng.
- Thông tin được mã hóa theo thời gian và tần suất của các xung, thay vì giá trị liên tục.
- Mạng được huấn luyện bằng các thuật toán đặc biệt như Spike-Timing-Dependent Plasticity (STDP) hoặc các biến thể của gradient descent thích ứng với tín hiệu spike.

Ưu điểm

- **Tiết kiệm năng lượng:** Nhờ truyền tín hiệu rời rạc và chỉ hoạt động khi có xung, SNN rất tiết kiệm năng lượng, phù hợp cho các thiết bị nhúng và phần cứng neuromorphic.
- **Xử lý thông tin thời gian thực:** SNN có khả năng xử lý dữ liệu tuần tự và tín hiệu thời gian thực hiệu quả.
- **Mô phỏng sinh học chính xác hơn:** Gần gũi với cách hoạt động của não bộ, có tiềm năng mở rộng trong nghiên cứu trí tuệ nhân tạo sinh học.
- **Khả năng biểu diễn mạnh:** Có thể biểu diễn và học các mẫu phức tạp theo không gian và thời gian.

Nhược điểm

- **Khó huấn luyện:** Do tín hiệu rời rạc và phi tuyến mạnh, việc huấn luyện SNN phức tạp hơn nhiều so với mạng nơ-ron truyền thống.
- **Thiếu công cụ và framework phát triển:** Hiện nay các thư viện hỗ trợ SNN còn hạn chế so với các mạng deep learning phổ biến.
- **Hiệu suất thấp trên một số bài toán:** Trên các tập dữ liệu chuẩn, SNN thường chưa vượt trội hoặc bằng các mạng deep learning thông thường.

Yêu cầu phần cứng đặc biệt: Để tận dụng tối đa lợi thế tiết kiệm năng lượng và tốc độ, SNN cần các phần cứng neuromorphic chuyên dụng.

3.2. Công nghệ

3.2.1. React



1. Giới thiệu React

React là một thư viện JavaScript mã nguồn mở được phát triển bởi Facebook để xây dựng giao diện người dùng (UI) cho các ứng dụng web. React cho phép tạo ra các ứng dụng web động và tương tác cao bằng cách xây dựng giao diện dưới dạng các thành phần (components). Mỗi thành phần trong React có thể tái sử dụng, dễ dàng bảo trì và tối ưu hiệu suất. React chủ yếu được sử dụng để phát triển các ứng dụng một trang (SPA - Single Page Applications), nơi mà các trang không cần tải lại toàn bộ khi người dùng tương tác với ứng dụng.

2. Những điểm nổi bật của React

- **Component-based architecture (Kiến trúc dựa trên thành phần):** React chia giao diện người dùng thành các thành phần độc lập, dễ dàng quản lý và tái sử dụng. Mỗi thành phần có thể quản lý trạng thái riêng biệt và tương tác với các thành phần khác.
- **Virtual DOM (Virtual Document Object Model):** React sử dụng Virtual DOM để tối ưu hóa việc cập nhật giao diện người dùng. Khi có thay đổi trong trạng thái, React chỉ cập nhật phần DOM thay đổi thay vì làm mới toàn bộ giao diện, giúp tăng tốc độ và hiệu suất ứng dụng.

- **Unidirectional Data Flow (Dòng dữ liệu một chiều):** Dữ liệu trong React luôn chảy theo một chiều từ cha đến con, giúp dễ dàng quản lý và kiểm soát trạng thái ứng dụng.
- **JSX (JavaScript XML):** JSX là một cú pháp mở rộng của JavaScript cho phép kết hợp mã HTML và JavaScript trong cùng một tập tin. Điều này làm cho việc xây dựng giao diện trở nên dễ dàng và trực quan hơn.
- **React Hooks:** React cung cấp các hooks (như useState, useEffect, v.v.) giúp tối giản mã và cải thiện khả năng tái sử dụng logic trong các thành phần chức năng mà không cần sử dụng class-based components.
- **Rich Ecosystem and Community:** React có một hệ sinh thái phong phú với rất nhiều thư viện và công cụ hỗ trợ, cộng đồng lớn mạnh và tài liệu phong phú giúp dễ dàng học hỏi và phát triển.

3. Những ứng dụng của React

- **Single Page Applications (SPA):** React là công nghệ phổ biến để xây dựng các ứng dụng một trang, nơi người dùng có thể tương tác với ứng dụng mà không cần tải lại trang, mang đến trải nghiệm mượt mà và nhanh chóng.
- **Web và Mobile Applications:** React Native, một phiên bản mở rộng của React, cho phép phát triển ứng dụng di động cho cả iOS và Android với mã nguồn chung. Điều này giúp giảm chi phí phát triển và bảo trì.
- **Dynamic Websites:** React rất phù hợp cho các trang web động và tương tác cao, chẳng hạn như các trang web có nội dung được cập nhật thường xuyên như mạng xã hội, hệ thống quản lý nội dung, hoặc các trang thương mại điện tử.
- **Real-time Applications:** Các ứng dụng yêu cầu đồng bộ hóa thời gian thực, chẳng hạn như chat hoặc thông báo, có thể được xây dựng hiệu quả với React nhờ khả năng cập nhật giao diện nhanh chóng và không cần tải lại trang.
- **Dashboards and Analytics Tools:** React cũng được sử dụng để xây dựng các công cụ phân tích và bảng điều khiển (dashboards) vì khả năng xử lý giao diện động và yêu cầu tái sử dụng nhiều thành phần.

3.2.2. Next.js



1. Giới thiệu Next.js

Next.js là một framework JavaScript mã nguồn mở, được xây dựng trên nền tảng **React**, giúp phát triển các ứng dụng web với tính năng **Server-Side Rendering (SSR)**, **Static Site Generation (SSG)** và **Client-Side Rendering (CSR)**. Next.js cung cấp một cách tiếp cận đơn giản và mạnh mẽ để xây dựng các ứng dụng web hiện đại, với các tính năng tích hợp sẵn như phân trang, tối ưu hóa hiệu suất, và hỗ trợ cho việc phát triển ứng dụng đa trang (multi-page applications).

Với Next.js, các nhà phát triển có thể dễ dàng tạo ra các trang web có hiệu suất cao, dễ bảo trì và tối ưu cho SEO (Search Engine Optimization), nhờ vào khả năng render trang trên server trước khi gửi nó tới client.

2. Những điểm nổi bật của Next.js

- **Server-Side Rendering (SSR):** Next.js hỗ trợ SSR, giúp trang web được render ở phía server trước khi gửi đến client. Điều này cải thiện thời gian tải trang và tối ưu SEO.
- **Static Site Generation (SSG):** Next.js cung cấp khả năng tạo ra các trang tĩnh trong quá trình build, giúp tăng tốc độ tải trang và giảm thiểu yêu cầu tài nguyên máy chủ.

- **Automatic Code Splitting:** Next.js tự động chia mã (code splitting) cho từng trang để chỉ tải mã cần thiết cho mỗi trang cụ thể. Điều này giúp giảm kích thước tải trang ban đầu và cải thiện hiệu suất.
- **File-based Routing:** Next.js sử dụng hệ thống routing tự động dựa trên cấu trúc thư mục của dự án. Mỗi tệp JavaScript trong thư mục pages sẽ tự động trở thành một route trong ứng dụng, giúp việc cấu hình và quản lý routing trở nên dễ dàng.
- **API Routes:** Next.js hỗ trợ API routes, cho phép bạn dễ dàng xây dựng API server-side mà không cần cấu hình một máy chủ riêng biệt. Điều này rất hữu ích khi cần thêm tính năng backend vào ứng dụng.
- **Image Optimization:** Next.js cung cấp tính năng tối ưu hóa hình ảnh tự động, giúp giảm thời gian tải trang và cải thiện trải nghiệm người dùng.
- **Hot Module Replacement (HMR):** Tính năng này giúp việc phát triển và cập nhật ứng dụng trở nên mượt mà hơn, khi các thay đổi trong mã sẽ được cập nhật ngay lập tức mà không cần tải lại trang.
- **SEO-friendly:** Vì hỗ trợ SSR và SSG, Next.js giúp cải thiện khả năng tối ưu hóa công cụ tìm kiếm (SEO), làm cho các trang web dễ dàng được công cụ tìm kiếm thu thập và lập chỉ mục.
- **Built-in CSS and Sass Support:** Next.js hỗ trợ tích hợp sẵn CSS và Sass, đồng thời cho phép sử dụng các phương thức CSS-in-JS như styled-components hoặc emotion.

3. Những ứng dụng của Next.js

- **Trang web tĩnh (Static Websites):** Next.js rất phù hợp để xây dựng các trang web tĩnh với khả năng SSG, cung cấp thời gian tải trang cực kỳ nhanh và tối ưu cho SEO.
- **Ứng dụng đơn trang (SPA):** Với tính năng SSR và CSR, Next.js có thể xây dựng các ứng dụng đơn trang, nơi mà các trang được render động trên client nhưng vẫn tối ưu cho SEO và hiệu suất.
- **E-commerce Websites:** Next.js giúp xây dựng các trang web thương mại điện tử với tốc độ tải nhanh và khả năng tối ưu hóa SEO tốt hơn so với các

framework client-side thuận túy, điều này rất quan trọng cho việc tiếp cận người dùng qua tìm kiếm.

- **Blog và CMS (Content Management Systems):** Next.js là một lựa chọn tuyệt vời để xây dựng các trang web blog hoặc CMS vì khả năng render trang tĩnh nhanh chóng và tích hợp dễ dàng với các hệ thống quản lý nội dung (Content APIs).
- **Bảng điều khiển (Dashboards):** Next.js có thể được sử dụng để xây dựng các ứng dụng quản lý và phân tích dữ liệu (dashboard), nhờ vào khả năng tạo trang động hiệu quả kết hợp với tối ưu hóa cho trải nghiệm người dùng.
- **Web Applications with Dynamic Content:** Next.js phù hợp cho các ứng dụng web động, nơi dữ liệu được thay đổi liên tục, như các ứng dụng mạng xã hội, nền tảng học trực tuyến, hệ thống quản lý khách hàng (CRM), v.v.

3.2.3. Tailwind



1. Giới thiệu Tailwind CSS

Tailwind CSS là một framework CSS utility-first mã nguồn mở, cho phép các nhà phát triển xây dựng giao diện người dùng tùy chỉnh và linh hoạt mà không cần phải viết nhiều CSS tùy chỉnh. Thay vì cung cấp các thành phần UI sẵn có như các framework khác (ví dụ: Bootstrap), Tailwind cung cấp các lớp tiện ích (utility classes) để điều chỉnh kiểu dáng trực tiếp trong HTML, giúp tạo ra các thiết kế hoàn toàn linh hoạt và tối ưu.

Với Tailwind, bạn có thể tạo ra giao diện người dùng theo nhu cầu cụ thể mà không phải lo lắng về việc tạo các lớp CSS phức tạp hoặc viết mã thừa, từ đó giúp tăng hiệu suất phát triển và tối ưu hóa mã nguồn.

2. Những điểm nổi bật của Tailwind CSS

- **Utility-first CSS Framework:** Tailwind hoạt động theo phương pháp utility-first, tức là thay vì xây dựng các thành phần UI đầy đủ, bạn sẽ sử dụng các lớp tiện ích (utilities) để áp dụng trực tiếp các kiểu dáng như padding, margin, màu sắc, font-size, v.v. Điều này giúp bạn kiểm soát chi tiết từng phần trong giao diện.
- **Tùy chỉnh linh hoạt:** Tailwind cho phép bạn dễ dàng tùy chỉnh cấu hình mặc định thông qua tập tin cấu hình tailwind.config.js, từ đó bạn có thể thay đổi màu sắc, kích thước, kiểu phông chữ, breakpoints, v.v.
- **Responsive design (Thiết kế phản hồi):** Tailwind cung cấp các lớp tiện ích hỗ trợ thiết kế responsive ngay lập tức, với các lớp như sm:, md:, lg:, xl: để tạo giao diện linh hoạt trên các kích thước màn hình khác nhau mà không cần viết media queries thủ công.
- **Purge CSS tích hợp:** Tailwind tích hợp công cụ purge CSS giúp loại bỏ các lớp không sử dụng khỏi tệp CSS cuối cùng, giúp giảm dung lượng của tệp CSS và cải thiện hiệu suất.
- **Tối ưu hóa phát triển:** Các lớp tiện ích của Tailwind cho phép nhà phát triển xây dựng giao diện nhanh chóng mà không cần phải quay lại viết mã CSS mới cho từng thành phần. Điều này giúp tiết kiệm thời gian và tăng tốc quá trình phát triển.
- **Hỗ trợ với các công cụ hiện đại:** Tailwind tích hợp tốt với các công cụ hiện đại như PostCSS, PurgeCSS, và các framework frontend như React, Vue, và Angular.
- **Hệ thống thiết kế sẵn có:** Tailwind giúp bạn dễ dàng tạo ra các hệ thống thiết kế riêng, cho phép bạn duy trì tính nhất quán trong giao diện và dễ dàng chia sẻ lại các thành phần UI.

3. Những ứng dụng của Tailwind CSS

- **Trang web tùy chỉnh và nhanh chóng:** Tailwind rất lý tưởng cho các ứng dụng hoặc trang web tùy chỉnh, nơi bạn cần kiểm soát chi tiết từng thành phần mà không phải phụ thuộc vào các thành phần UI sẵn có.
- **Web applications (Ứng dụng web):** Tailwind giúp tạo các giao diện người dùng đẹp mắt và đáp ứng nhanh, phù hợp cho các ứng dụng web cần có giao diện tùy chỉnh mạnh mẽ và dễ dàng bảo trì.
- **Prototyping:** Với các lớp tiện ích dễ sử dụng, Tailwind là công cụ tuyệt vời để tạo nguyên mẫu giao diện nhanh chóng, cho phép các nhà thiết kế và phát triển thử nghiệm và điều chỉnh giao diện dễ dàng trong thời gian ngắn.
- **Trang web tĩnh (Static Websites):** Tailwind rất phù hợp để xây dựng các trang web tĩnh hoặc các trang blog với các yêu cầu tùy chỉnh cao về giao diện, nhưng lại không muốn bị ràng buộc bởi các thiết kế UI có sẵn của các framework CSS khác.
- **Hệ thống thiết kế nội bộ:** Tailwind cho phép bạn xây dựng hệ thống thiết kế với các lớp CSS có thể tái sử dụng, dễ dàng duy trì và mở rộng.
- **Landing pages (Trang đích):** Các trang đích yêu cầu một giao diện nhẹ, nhanh chóng và dễ dàng tùy chỉnh, Tailwind là sự lựa chọn tuyệt vời cho các dự án này.

3.2.4. Amazon Web Services (AWS)



Amazon Web Services (AWS) là một nền tảng điện toán đám mây toàn diện, cung cấp hơn 200 dịch vụ đầy đủ tính năng từ các trung tâm dữ liệu trên toàn cầu. AWS cung cấp mọi thứ, từ cơ sở hạ tầng cơ bản như máy chủ ảo và lưu trữ, đến các công nghệ tiên tiến như trí tuệ nhân tạo (AI), học máy (Machine Learning) và Internet vạn vật (IoT).

AWS cung cấp một loạt các dịch vụ được phân loại thành các lĩnh vực chính sau:

1. Tính toán:

- Amazon Elastic Compute Cloud (EC2): Cung cấp máy chủ ảo (virtual machines) cho phép bạn chạy ứng dụng trên đám mây. Ưu điểm: Linh hoạt, khả năng mở rộng, chi phí hiệu quả.
- Amazon Elastic Container Service (ECS) & Elastic Kubernetes Service (EKS): Dịch vụ quản lý container cho phép bạn chạy và quản lý ứng dụng containerized. Ưu điểm: Triển khai nhanh chóng, dễ dàng quản lý, khả năng mở rộng cao.
- AWS Lambda: Dịch vụ tính toán không máy chủ (serverless) cho phép bạn chạy mã mà không cần quản lý máy chủ. Ưu điểm: Tiết kiệm chi phí, tự động mở rộng, dễ sử dụng.

2. Lưu trữ:

- Amazon Simple Storage Service (S3): Dịch vụ lưu trữ đối tượng cho phép bạn lưu trữ và truy xuất bất kỳ lượng dữ liệu nào từ bất kỳ đâu trên web. Ưu điểm: Độ bền cao, khả năng mở rộng, chi phí thấp.
- Amazon Elastic Block Store (EBS): Cung cấp ổ đĩa cứng ảo có thể gắn vào các instance EC2. Ưu điểm: Hiệu suất cao, độ tin cậy cao.
- Amazon Relational Database Service (RDS): Dịch vụ cơ sở dữ liệu quan hệ được quản lý, hỗ trợ nhiều engine cơ sở dữ liệu phổ biến. Ưu điểm: Dễ sử dụng, khả năng mở rộng, độ tin cậy cao.

3. Mạng:

- Amazon Virtual Private Cloud (VPC): Cho phép bạn tạo một mạng riêng ảo trên đám mây AWS. Ưu điểm: Kiểm soát hoàn toàn mạng, bảo mật cao.

- Amazon Route 53: Dịch vụ DNS được quản lý. Ưu điểm: Độ tin cậy cao, khả năng mở rộng.

4. Cơ sở dữ liệu:

- Amazon DynamoDB: Dịch vụ cơ sở dữ liệu NoSQL được quản lý. Ưu điểm: Hiệu suất cao, khả năng mở rộng.
- Amazon Redshift: Dịch vụ kho dữ liệu (data warehouse). Ưu điểm: Phân tích dữ liệu lớn, hiệu suất cao.

5. Học máy:

- Amazon SageMaker: Nền tảng học máy được quản lý đầy đủ. Ưu điểm: Dễ sử dụng, khả năng mở rộng.
- Amazon Rekognition: Dịch vụ nhận dạng hình ảnh và video. Ưu điểm: Phân tích hình ảnh và video dễ dàng.

6. Bảo mật, danh tính và tuân thủ:

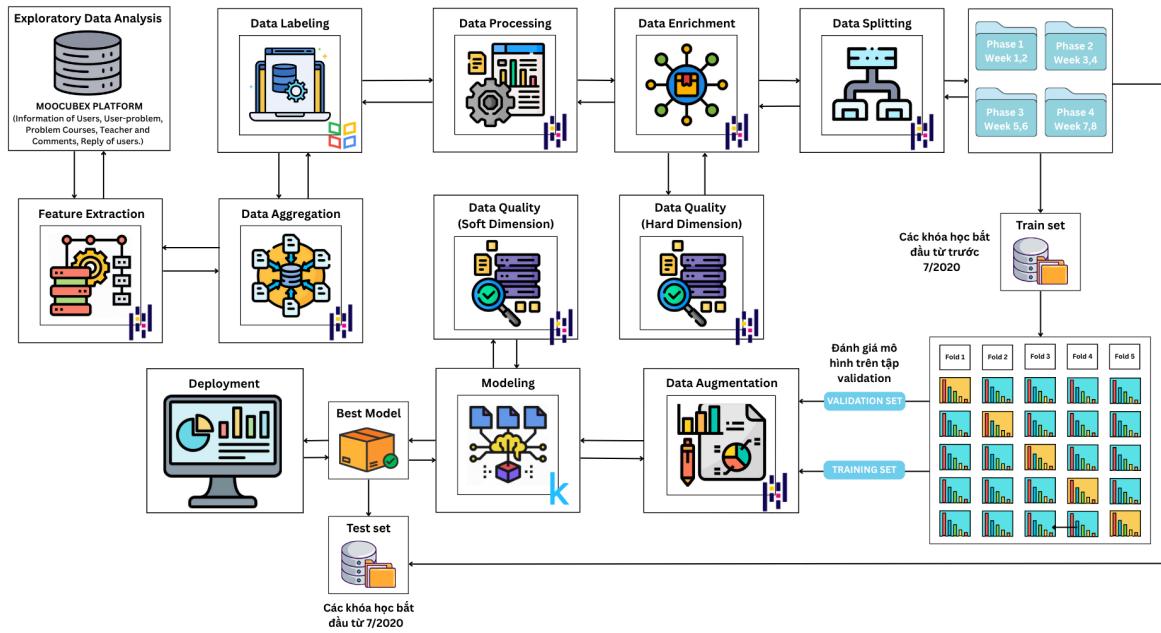
- AWS Identity and Access Management (IAM): Cho phép bạn quản lý quyền truy cập vào tài nguyên AWS. Ưu điểm: Bảo mật cao, kiểm soát truy cập chi tiết.
- Amazon Key Management Service (KMS): Dịch vụ quản lý khóa mã hóa. Ưu điểm: Bảo mật dữ liệu mạnh mẽ.

Ưu điểm của việc sử dụng AWS:

- Khả năng mở rộng và linh hoạt: Dễ dàng mở rộng hoặc thu hẹp tài nguyên theo nhu cầu.
- Chi phí hiệu quả: Chỉ phải trả tiền cho những gì bạn sử dụng.
- Độ tin cậy cao: Cơ sở hạ tầng đáng tin cậy với khả năng phục hồi cao.
- Bảo mật: Các dịch vụ bảo mật mạnh mẽ để bảo vệ dữ liệu.
- Đa dạng dịch vụ: Cung cấp một loạt các dịch vụ đáp ứng mọi nhu cầu.
- Cộng đồng và hỗ trợ lớn: Cộng đồng người dùng lớn và hỗ trợ kỹ thuật chuyên nghiệp.

CHƯƠNG 4. PHƯƠNG PHÁP ĐỀ XUẤT

4.1 Các bước thực hiện tổng quan từ input đến output của bài toán



MOOCubeX PLATFORM (Nguồn dữ liệu):

- Bộ dữ liệu MOOCubeX:** Là một bộ dữ liệu quy mô lớn, được thiết kế đặc biệt cho nghiên cứu về phân tích hành vi học tập và hỗ trợ học tập thông minh. Nó phù hợp cho các bài toán như dự đoán kết quả học tập, phát hiện học viên có nguy cơ bỏ học, và cá nhân hóa nội dung.
- Dữ liệu đầu vào được sử dụng (Input):** Do hạn chế tài nguyên tính toán, để tài tập trung vào một phần dữ liệu, bao gồm:
 - Thông tin học viên:** (từ entities/user.json)
 - Thông tin khóa học:** (từ entities/course.json)
 - Hành vi hoạt động của người học với khóa học trong 8 tuần đầu:** Số lượt đăng nhập, thời gian học, số bài học hoàn thành, lượt xem video (từ relations/user-video.json), lượt tương tác với bài tập (từ relations/user-problem.json), số lần nộp bài.
 - Hành vi hoạt động trên diễn đàn:** Số lượng bình luận (từ entities/comment.json, relations/user-comment.json) và nhận phản hồi (từ entities/reply.json, relations/user-reply.json, relations/comment-reply.json).
- Các tệp dữ liệu cụ thể được sử dụng để trích xuất đặc trưng gồm: entities/user.json, entities/course.json, entities/comment.json,

entities/reply.json, relations/comment-reply.json, relations/user-comment.json, relations/user-reply.json, relations/user-problem.json, relations/user-video.json.

Exploratory Data Analysis (EDA - Phân tích Khám phá Dữ liệu):

- Dữ liệu từ MOOCubeX được đưa vào phân tích để hiểu rõ hơn về cấu trúc, xu hướng, phân phối và mối quan hệ giữa các yếu tố.
- Sử dụng các công cụ trực quan hóa như **Matplotlib** và **Seaborn** để thực hiện việc này.
- Mục tiêu là phát hiện các mẫu, điểm bất thường và kiểm tra các giả định ban đầu.

Feature Extraction (Trích xuất Đặc trưng):

- Từ dữ liệu thô, bước này tạo ra các đặc trưng có ý nghĩa cho mô hình. Các đặc trưng được tạo bao gồm: tổng thời gian học mỗi tuần, số lượt truy cập video, tỷ lệ hoàn thành bài tập, số lượng tương tác diễn đàn, v.v.
- **Thông tin Input cuối cùng (Các đặc trưng chi tiết):**
 - **Thông tin khóa học (course_*):** course_id, num_prerequisites, field_x, num_field_x, start_date, end_date, duration_days.
 - **Tài liệu khóa học (resource_*):** video_count, exercise_count, chapter_count.
 - **Thành phần điểm (score_*):** (Thông tin này có thể được dùng để tạo nhãn hoặc làm đặc trưng nếu có) assignment, exam, video, certificate.
 - **Thông tin người dùng (user_*):** user_id, school, user_enroll_time, user_past_course_count, user_time_since_last_course.
 - **Hành vi học tập - Bình luận (comment_*) - Theo từng đợt (phase{i}):** comment_count_phase{i}, total_words_phase{i}_x, entropy_time_comment_phase{i}.
 - **Hành vi học tập - Phản hồi (reply_*) - Theo từng đợt (phase{i}):** reply_count_phase{i}, total_words_phase{i}_y, entropy_time_reply_phase{i}.
 - **Hành vi học tập - Bài tập (exercise_*) - Theo từng đợt ({i}):** Rất nhiều đặc trưng chi tiết như exercise_id_count_{i}, exercise_correct_sum_{i}, exercise_diff_mean_{i}, exercise_hour_entropy_{i}, v.v. (như liệt kê trong bảng).
 - **Hành vi học tập - Video (video_*) - Theo từng đợt ({i}):** video_watched_count_{i}, video_watched_percentage_{i}, video_watch_time_{i}, video_pause_count_{i}, video_speed_avg_{i}, video_entropy_time_{i}, video_final_score_percentage_{i}, v.v. (như liệt kê trong bảng).

- Việc này giúp đơn giản hóa quá trình truy xuất, xử lý và trích xuất đặc trưng từ cấu trúc rõ ràng của MOOCubeX.

Data Aggregation (Tổng hợp Dữ liệu):

- Các đặc trưng đã trích xuất từ nhiều nguồn và bảng khác nhau (ví dụ: thông tin người dùng, thông tin khóa học, log tương tác) được kết hợp lại để tạo thành một bộ dữ liệu thống nhất cho mỗi thực thể (ví dụ: mỗi lượt đăng ký của học viên trong một khóa học).

Data Labeling (Gán nhãn Dữ liệu):

Output (Nhận đầu ra): Kết quả cuối khóa học của học viên, được phân loại thành 5 mức độ từ cao đến thấp: **A (Xuất sắc), B (Khá), C (Trung bình), D (Yếu), E (Không hoạt động / bỏ học).**

- Quy trình gán nhãn:**

- Lọc dữ liệu:**

- Loại bỏ những khóa học có ít học viên đăng ký.
 - Loại bỏ những khóa học có ít hoặc quá nhiều tài liệu.
 - Loại bỏ những khóa học không có bài tập và không có video.
 - Loại bỏ những khóa học không giới hạn thời gian.
 - Loại bỏ những học viên chỉ đăng ký nhưng không làm bài tập.

- Gán nhãn dựa trên điểm số (theo mô hình XueTangX):**

- Nhóm xác định 3 thành phần điểm chính: **Assignment** (bài tập trắc nghiệm, tự luận, câu hỏi chương, gộp cả điểm Discussion nếu có), **Exam** (chỉ tính điểm Final Exam, bỏ qua điểm giữa kỳ nếu có), **Video** (tỷ lệ xem video, mức độ hoàn thành tài liệu học, gộp thêm phần Reading nếu có).
 - Phân loại nhãn dựa trên tổng điểm (Score):
 - A (Excellent):** $85 \leq \text{Score} \leq 100$
 - B (Good):** $70 \leq \text{Score} < 85$
 - C (Pass):** $60 \leq \text{Score} < 70$
 - D (Fail):** $30 \leq \text{Score} < 60$
 - E (Inactive):** $0 \leq \text{Score} < 30$

Data Processing (Xử lý Dữ liệu):

- Làm sạch dữ liệu:** Sử dụng các công cụ như **Pandas (Python)** để loại bỏ các bản ghi trùng lặp, thiếu giá trị hoặc sai định dạng.
- Chuẩn hóa dữ liệu:** Thực hiện các kỹ thuật như **Min-Max Scaling** hoặc **Standardization** để đảm bảo dữ liệu đầu vào đồng nhất cho các mô hình.

Data Splitting (Phân chia Dữ liệu):

- Đặc trưng hành vi được tính theo **4 giai đoạn, mỗi giai đoạn gồm 2 tuần** (trong 8 tuần đầu).
- Train set (Tập huấn luyện):** Dữ liệu từ **các khóa học bắt đầu từ trước 7/2020.**
- Test set (Tập kiểm tra):** Dữ liệu từ **các khóa học bắt đầu từ 7/2020.** Tập này được giữ riêng cho đánh giá cuối cùng.
- Tập huấn luyện (Train set) sẽ được tiếp tục chia nhỏ bằng Stratified K-Fold ($K=5$) trong giai đoạn Modeling để tạo các tập huấn luyện nội bộ và tập kiểm định.

Data Quality (Hard Dimensions - Chất lượng Dữ liệu theo Chiều cứng):

Mục tiêu: Đảm bảo tính chính xác, đầy đủ, nhất quán và kịp thời của dữ liệu *trên tập huấn luyện (Train set)* trước khi đưa vào mô hình. Các kiểm tra cụ thể:

- Phân tích thống kê cơ bản:
 - Kiểm tra số giá trị NULL trong từng cột.
 - Kiểm tra số lượng giá trị duy nhất (độ phân tán): `df.nunique()`.
 - Kiểm tra độ dài trung bình của chuỗi (đối với các cột object).
- Kiểm tra quy tắc nghiệp vụ và kinh doanh: Đảm bảo dữ liệu tuân thủ logic thực tế.
- Phát hiện bất thường (Outliers) và dữ liệu trùng lặp.
- Kiểm tra tính nhất quán giữa nhiều nguồn dữ liệu (nếu có).
- Accuracy (Độ chính xác):** Dữ liệu phản ánh đúng thực tế.
 - Kiểm tra định dạng:** `user_id` bắt đầu bằng `U_`, `course_id` bằng `C_`.
 - Kiểm tra miền giá trị:** `grade_label` là một trong {A, B, C, D, E}, các cột số liệu phải lớn hơn hoặc bằng 0 (ví dụ: `course_prerequisite_count` từ 0-10).
- Completeness (Tính đầy đủ):** Tất cả dữ liệu cần thiết đã được thu thập.
 - Đo lường:** Tỷ lệ hoàn chỉnh theo từng hàng (Object) và toàn bộ Dataset. (Ví dụ: ghi nhận trường school đôi khi thiếu, làm giảm completeness của hàng đó xuống 98.36%).
 - Kiểm tra non-null:** Các trường quan trọng không được để trống.
 - Kiểm tra data type:** Dữ liệu có đúng kiểu mong muốn (int, float, string, datetime).
- Consistency (Tính nhất quán):** Dữ liệu không mâu thuẫn.
 - Ràng buộc logic:** Ví dụ, tổng điểm thành phần nếu có ràng buộc, `course_days >= video_watch_time`.

- **Tính duy nhất (Uniqueness):** Kiểm tra các dòng hoàn toàn giống nhau hoặc các dòng có đặc trưng trùng lặp (bỏ qua ID).
- **Tính khóa ngoại (Foreign Key Integrity):** user_id phải tồn tại trong user.json, course_id mà user đăng ký phải có trong course.json.
- **Timeliness (Tính kịp thời):** Dữ liệu được cập nhật và không lỗi thời.
 - **Kiểm tra:** Dữ liệu có được cập nhật đúng hạn, không nằm ngoài vùng khảo sát (ví dụ: course_year không quá xa, course_days không quá dài/ngắn bất thường, điểm số nằm trong khoảng [0,100], user_enroll_time không quá xa hiện tại).

Data Augmentation (Tăng cường Dữ liệu):

- Áp dụng cho phần huấn luyện nội bộ trong mỗi fold của Stratified K-Fold (trên "Train set" trước 7/2020).
- Phương pháp: SMOTE-SVM.
- Hiệu quả được đánh giá trên phần kiểm định tương ứng của fold đó.

Modeling (Xây dựng Mô hình):

- Sử dụng tập huấn luyện (đã được tăng cường nếu có) để huấn luyện các mô hình.
- Thuật toán truyền thống: Logistic Regression, Decision Tree, Random Forest, REP Tree, KNN, SVM, Naive Bayes, LightGBM, XGBoost, CatBoost.
- Mô hình học sâu: ANN, CNN, RNN, LSTM, BiLSTM, 4-layer Stacked LSTM, GNN, ANN-LSTM hybrid.
- Tinh chỉnh tham số (Hyperparameter Tuning): Dùng Grid Search hoặc Bayesian Optimization.
- Đánh giá mô hình (trong K-Fold Cross-Validation):
 - Sử dụng các chỉ số: Accuracy, Precision, Recall, F1-score, ROC-AUC.
 - Stratified Cross-Validation (K-Fold) được áp dụng để kiểm tra khả năng tổng quát hóa.

Data Quality (Soft Dimensions - Chất lượng Dữ liệu theo Chiều mềm):

- **Mục tiêu:** Đánh giá chất lượng dữ liệu dựa trên khả năng của nó trong việc hỗ trợ mô hình tạo ra kết quả đáng tin cậy, phù hợp và ổn định. Bước này thực hiện sau khi đã có các mô hình được huấn luyện và đánh giá.
- **Các đánh giá cụ thể:**
 - **Reliability (Độ tin cậy của dữ liệu cho mô hình):**
 - **Đánh giá qua hiệu suất mô hình:** Accuracy và F1-score cao trên các tập kiểm định (validation sets trong K-Fold) cho thấy dữ liệu

có độ tin cậy tốt để mô hình học. Accuracy cao nhưng F1-score thấp có thể chỉ ra vấn đề mất cân bằng lớp trong dữ liệu ảnh hưởng đến độ tin cậy cho các lớp thiểu số.

- **Kiểm tra độ ổn định (Consistency of Reliability) bằng Cross-Validation:** Kết quả mô hình (ví dụ: F1-score) ổn định, ít biến động (variance thấp) qua các fold khác nhau của cross-validation cho thấy dữ liệu có độ tin cậy cao, không bị phụ thuộc quá nhiều vào một cách chia cụ thể. Nếu kết quả biến động nhiều, dữ liệu có thể chứa lỗi hoặc không đủ đại diện.

- **Relevance (Độ liên quan của dữ liệu đối với bài toán):**

- **Đánh giá qua Feature Importance:** Các đặc trưng có độ quan trọng cao (từ các mô hình như Random Forest, LightGBM) cho thấy chúng liên quan mật thiết đến việc dự đoán nhãn đầu ra. Đặc trưng có độ quan trọng thấp có thể không cần thiết hoặc nhiễu.
- **Kiểm tra qua AUC-ROC:** Giá trị AUC-ROC gần 1.0 cho thấy các đặc trưng trong dữ liệu có độ liên quan cao, giúp mô hình phân biệt tốt giữa các lớp.

- **Validity, Uniqueness (theo ngữ cảnh mô hình):** Mặc dù một số khía cạnh của Validity và Uniqueness được kiểm tra ở Hard Dimensions, ở đây có thể xem xét liệu các đặc trưng có thực sự hợp lệ và mang lại thông tin duy nhất hữu ích cho *mô hình cụ thể* hay không, dựa trên phân tích kết quả và lỗi của mô hình.

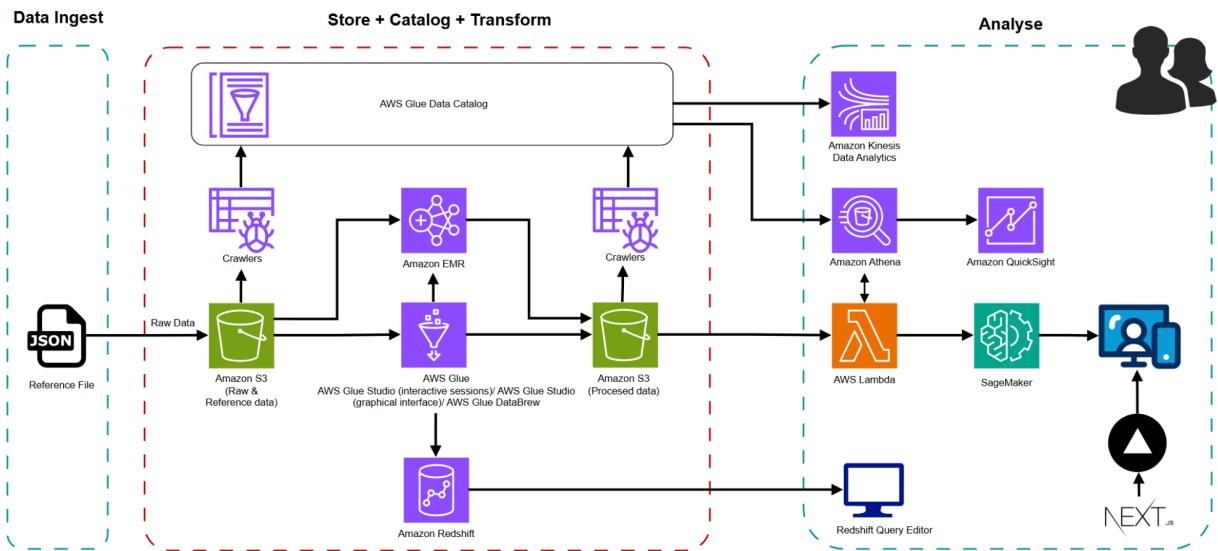
Best Model (Mô hình Tốt nhất):

- Dựa trên kết quả đánh giá K-Fold Cross-Validation và các phân tích về Soft Dimensions (độ tin cậy, liên quan), mô hình có hiệu suất tốt nhất và ổn định nhất được lựa chọn.

Kiểm thử Cuối cùng và Triển khai

- **Test set (Đánh giá trên Tập kiểm tra cuối cùng):** "Best Model" được đánh giá lần cuối trên "Test set" (dữ liệu từ các khóa học bắt đầu từ 7/2020).
- **Deployment (Triển khai):** Nếu "Best Model" đạt hiệu suất tốt, nó sẽ được triển khai vào môi trường thực tế.

4.2. Thiết kế kiến trúc dữ liệu lớn có thể triển khai framework



4.2.1 Data Ingest (Thu thập dữ liệu)

Nguồn dữ liệu đầu vào: Bộ dữ liệu MOOCCubeX, định dạng các tệp JSON.

4.2.2 Data Store + Catalog + Transform (Lưu trữ – Biến đổi – Gắn nhãn dữ liệu):

Data Store – Lưu trữ dữ liệu

- Amazon S3 được sử dụng làm kho lưu trữ chính cho toàn bộ dữ liệu thô (raw data), dữ liệu đã qua xử lý (processed data), và dữ liệu đầu ra phân tích (analytics-ready).
- Tổ chức dữ liệu theo tầng (Data Lake Layers):
 - Raw Layer: chứa dữ liệu gốc (JSON) từ MOOCCubeX.
 - Processed Layer: chứa dữ liệu đã được làm sạch, phân loại, định dạng, và sẵn sàng cho phân tích hoặc huấn luyện mô hình.

Data Catalog – Lập danh mục dữ liệu

AWS Glue Data Catalog:

- Tự động tạo metadata (schema, bảng, cột) thông qua crawler.
- Hỗ trợ quản lý và khám phá dữ liệu trong S3 cho các dịch vụ như Athena, Redshift Spectrum.

Data Transform – Chuyển đổi và chuẩn hóa dữ liệu

AWS Glue Studio:

- Viết và chạy các job ETL (Extract – Transform – Load) bằng giao diện kéo-thả hoặc mã Python.
- Giám sát và trực quan hóa luồng ETL.

Glue Interactive Sessions + Jupyter Notebook:

- Tạo, thử nghiệm và gỡ lỗi các script ETL tương tác trực tiếp trong môi trường phát triển quen thuộc.

AWS Glue DataBrew:

- Giao diện không cần mã hóa (no-code) để làm sạch, chuẩn hóa và phân tích dữ liệu.
- Hỗ trợ hơn 250 phép biến đổi dữ liệu phổ biến.

Amazon EMR với Apache Spark:

- Xử lý các tập dữ liệu lớn, phức tạp, đòi hỏi khả năng xử lý phân tán cao.
- Chạy job Spark để thực hiện các bước biến đổi nặng hoặc phức tạp hơn so với Glue.

Upload lên Amazon Redshift:

- Dữ liệu đã được xử lý, tổ chức được tải lên Redshift để phục vụ cho truy vấn nhanh và phân tích đa chiều.

4.2.3 Data Enrich (Phân tích thông kê dữ liệu)

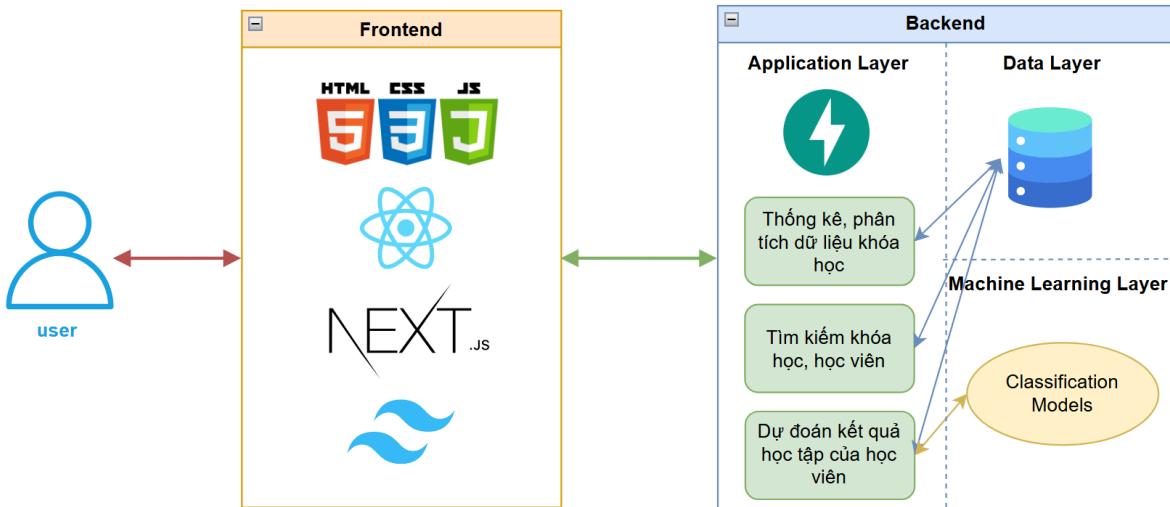
- Amazon Athena:
 - Truy vấn dữ liệu trực tiếp trên S3 bằng cú pháp SQL.
 - Kết hợp với Glue Catalog để định danh bảng.
- Amazon QuickSight:
 - Tạo báo cáo, dashboard tương tác dựa trên dữ liệu từ Athena hoặc Redshift.
 - Chia sẻ trực tiếp dashboard BI với người dùng cuối.

4.2.4 Tự động hóa & mở rộng truy cập

- AWS Lambda:
 - Viết các hàm không máy chủ để tự động hóa truy vấn (Athena SQL) hoặc đọc dữ liệu từ S4.
 - Có thể kích hoạt từ sự kiện hoặc API Gateway.
- Amazon SageMaker:
 - Huấn luyện và triển khai các mô hình máy học (Machine Learning).
 - Dự đoán hành vi người dùng, điểm số học tập, cá nhân hóa nội dung, v.v.
- Triển khai ứng dụng BI trên Vercel (Next.js):

- Phát triển ứng dụng web trực quan hóa dữ liệu bằng Next.js.
- Kết nối dữ liệu với API (Lambda hoặc GraphQL).
- Triển khai dễ dàng qua Vercel, đảm bảo hiệu suất và khả năng mở rộng.

4.3 Thiết kế kiến trúc hệ thống



4.3.1 Người dùng (User)

Người dùng là các cá nhân tương tác trực tiếp với hệ thống để sử dụng các tính năng đã được thiết kế. Cụ thể:

- Phân tích dữ liệu khóa học:
Người dùng có thể xem các kết quả phân tích về khóa học, như số lượng học viên tham gia, đánh giá tổng quan, hay thông tin chi tiết về khóa học.
- Tìm kiếm khóa học theo tên:
Người dùng nhập tên hoặc từ khóa liên quan để tra cứu thông tin của một hoặc nhiều khóa học.
- Dự đoán mức độ hoàn thành khóa học của các học viên:
Người dùng cung cấp thông tin đầu vào (ví dụ: tên học viên, khóa học mà học viên đang học,...) để hệ thống đưa ra dự đoán về mức độ hoàn thành khóa học của học viên.

Các thao tác mà người dùng thực hiện bao gồm:

- Nhập thông tin vào giao diện.
- Gửi yêu cầu tới hệ thống để phân tích, tìm kiếm, hoặc dự đoán.
- Nhận và xem kết quả trả về từ hệ thống.

4.3.2. Frontend

- **Công nghệ sử dụng:** HTML, CSS, JS, React, Next.js, tailwind.
- **Chức năng:** Giao diện người dùng (UI) được xây dựng để tương tác với người dùng.

4.3.3 Backend

- **Phân chia thành hai lớp:**
 - **Application Layer:**
 - Thông kê, phân tích dữ liệu khoa học.
 - Tìm kiếm thông tin, học
 - Dự đoán kết quả học tập.
 - **Data Layer:** Lưu trữ dữ liệu.
- **Machine Learning Layer:**
 - Chứa các **Classification Models** (Mô hình phân loại) để hỗ trợ phân tích và dự đoán.

4.3.4 Luồng tương tác với hệ thống

1. Người dùng tương tác với Frontend.
2. Frontend kết nối với Backend để xử lý dữ liệu và truy cập mô hình học máy.
3. Dữ liệu từ Data Layer được sử dụng để huấn luyện và chạy các mô hình phân loại.

CHƯƠNG 5. PHƯƠNG PHÁP THỰC NGHIỆM

5.1. Tìm hiểu dữ liệu

5.1.1. Giới thiệu chung bộ dữ liệu sử dụng

MOOCCubeX là một bộ dữ liệu MOOC (Massive Open Online Courses - Khóa Học Trực Tuyến Mở) được thu thập bởi Nhóm Kỹ Thuật Tri Thức của Đại học Thanh Hoa và thu thập từ nền tảng XuetangX, một trong những đối tác chính của edX. Mặc dù hệ thống XuetangX ra mắt vào tháng 10 năm 2013, đến ngày 31 tháng 5 năm 2021, nền tảng này đã cung cấp hơn 6.000 khóa học, bao gồm các khóa từ Đại học Thanh Hoa, Đại học Bắc Kinh, cũng như các khóa học edX từ MIT, Stanford, UC Berkeley, thu hút hơn 4,5 triệu người dùng đăng ký.

XuetangX cung cấp đa dạng tài nguyên học tập, cho phép người dùng tự do ghi danh và tham gia đầy đủ vào quá trình học, bao gồm xem video, làm bài tập và thảo luận. Các dữ liệu này có mối liên hệ chặt chẽ và được quản lý tốt, tạo nên một cơ sở dữ liệu phong phú về hành vi học tập của sinh viên. Nhờ đó, MOOCCubeX trở thành một nguồn dữ liệu lý tưởng cho nghiên cứu về cách thức học viên tiếp cận, tương tác và hoàn thành các khóa học trực tuyến.

MOOCCubeX có các đặc điểm sau:

- Phạm vi bao phủ cao: MOOCCubeX thu thập nhiều tài nguyên MOOC và tài nguyên giáo dục bên ngoài, cũng như hồ sơ dữ liệu về việc học tập, thực hành và thảo luận của học viên.
- Quy mô lớn: So với kho dữ liệu giáo dục truy cập mở khác, quy mô của MOOCCubeX lớn hơn, do đó hỗ trợ việc khám phá các mô hình sâu với yêu cầu dữ liệu cao.
- Lấy khái niệm làm trung tâm: Dữ liệu không đồng nhất được tổ chức bằng các khái niệm chi tiết, giúp tài nguyên có liên quan hơn và dễ biểu diễn, tìm kiếm và mô hình hóa hơn.

5.1.2. Các đặc điểm chính của bộ dữ liệu trong báo cáo

MOOCCubeX được xây dựng qua ba giai đoạn chính:

1. Xử lý Dữ liệu:

- Phân loại khóa học theo lĩnh vực để tạo cây phân loại khóa học.
- Loại bỏ trùng lặp trong tài nguyên khóa học (video, bài tập).
- Tích hợp hành vi học tập của sinh viên từ dữ liệu thô để dễ sử dụng hơn.

2. Trích Xuất Khái Niệm Chi Tiết:

- Sử dụng phương pháp học supervise learning để xác định khái niệm từ nội dung video.
- Xây dựng đồ thị (graph) quan hệ giữa các khái niệm để hỗ trợ học tập thích ứng (adaptive learning).

3. Xây Dựng & Liên Kết Dữ Liệu:

- Gán nhãn khái niệm chi tiết cho tài nguyên khóa học.
- Liên kết tài nguyên trong khóa học với dữ liệu bên ngoài để tạo hệ thống tri thức mở rộng.

5.1.2.1. Triển Khai MOOCubeX

1. Xử lý Dữ liệu

- **Phân loại khóa học:** Nhóm khóa học thành 88 lĩnh vực dựa trên hệ thống phân loại.
- **Loại bỏ trùng lặp:** Xác định và hợp nhất tài nguyên bị lặp, giảm dữ liệu thừa xuống 10.3%.
- **Tích hợp hành vi học tập:** Chuyển đổi dữ liệu xem video của sinh viên từ log 5 giây thành các đoạn xem có ý nghĩa hơn.

2. Trích Xuất Khái Niệm Chi Tiết

- Xác định các khái niệm quan trọng từ phụ đề video.
- Sử dụng cơ chế học tập tương tác để khám phá quan hệ giữa các khái niệm.
- Chỉ sử dụng phụ đề video vì dữ liệu văn bản từ các nguồn khác quá ngắn.

5.1.2.2. Trích Xuất Khái Niệm (Concept Extraction)

Quá trình trích xuất khái niệm được chia thành hai giai đoạn:

1. Trích Xuất Ứng Viên (Candidate Extraction - Tăng độ bao phủ)

- **Phrase Mining:** Sử dụng cụm danh từ từ tiêu đề Wikipedia tiếng Trung làm bảng thuật ngữ, chọn các cụm xuất hiện trong phụ đề video làm ứng viên.
- **Entity Linking:** Ánh xạ thực thể vào cơ sở tri thức ngoài (XLink) để mở rộng khái niệm từ cơ sở dữ liệu lớn (Xlore).
- **Named Entity Recognition (NER):** Sử dụng mô hình RoBERTa được tinh chỉnh để nhận diện thực thể có nhãn duy nhất ("concept"), chọn những cụm có độ tin cậy > 0.85 .

2. Xếp Hạng Khái Niệm (Concept Ranking - Tăng độ chính xác)

- Áp dụng K-means để chia khái niệm của mỗi khóa học thành 15 cụm dựa trên embedding BERT.
- Chọn hai cụm có điểm số cao nhất làm khái niệm chính của khóa học, giúp nâng cao độ chính xác của trích xuất.

$$score(j) = \min_{1 \leq i \leq 10} d(s_i, c_j)$$

Ý nghĩa công thức:

- s_i là trung tâm của cụm hạt giống thứ i .
- c_j là khái niệm ứng viên thứ j .
- $d(s_i, c_j)$ là hàm đo độ tương đồng cosine giữa embedding BERT của s_i và c_j .
- Các khái niệm hạt giống được gán nhãn của mỗi lĩnh vực được gom thành 10 cụm.

Cách tính điểm:

- Mỗi khóa học có 15 cụm khái niệm.
- Điểm của cụm j được tính dựa trên khoảng cách nhỏ nhất giữa 10 hạt giống hàng đầu và trung tâm cụm c_j .
- Hai cụm có điểm cao nhất sẽ được chọn làm khái niệm chính của khóa học.

5.1.2.3. Phát hiện Mối quan hệ Tiên quyết giữa Các Khái niệm

Trong MOOCCubeX, mối quan hệ tiên quyết giữa các khái niệm giúp xác định liệu khái niệm A có cần thiết để hiểu khái niệm B hay không. Tuy nhiên, do dữ liệu về mối quan hệ này khá thưa thớt, nhóm nghiên cứu đã đề xuất một phương pháp **huấn luyện đồng bộ tương tác (interactive co-training)** kết hợp hai cách tiếp cận chính:

1. Phương pháp dựa trên văn bản (Text-based Method)

- Sử dụng mạng nơ-ron đơn giản để phân loại mối quan hệ giữa hai khái niệm.
- Mã hóa văn bản bằng **BERT**, lấy vector embedding từ token cuối cùng hoặc token "[CLS]".
- Kết hợp hai embedding của cặp khái niệm và dự đoán nhãn nhị phân (có hoặc không có mối quan hệ tiên quyết).

2. Phương pháp dựa trên đồ thị (Graph-based Method)

- Sử dụng **Graph Attention Networks (GAT)** để tạo embedding cho các khái niệm dựa trên mối quan hệ đồ thị.

- Kết hợp embedding từ bộ mã hóa văn bản với embedding từ đồ thị để dự đoán mối quan hệ tiên quyết.
- Tận dụng thứ tự video trong một khóa học để suy luận mối quan hệ tiên quyết (khái niệm trong video trước có thể là tiền đề cho khái niệm trong video sau).

3. Gán nhãn tương tác (Interactive Labeling)

- Kết hợp kết quả từ hai mô hình trên để tạo danh sách ứng viên có thể có mối quan hệ tiên quyết.
- Nhóm annotator giàu kinh nghiệm sẽ chọn lọc cặp khái niệm có khả năng cao là đúng.
- Một annotator khác kiểm tra lại và loại bỏ các vòng lặp trong đồ thị để đảm bảo cấu trúc hợp lý.
- Quá trình này lặp lại nhiều lần cho đến khi thu thập đủ cặp khái niệm có mối quan hệ tiên quyết.

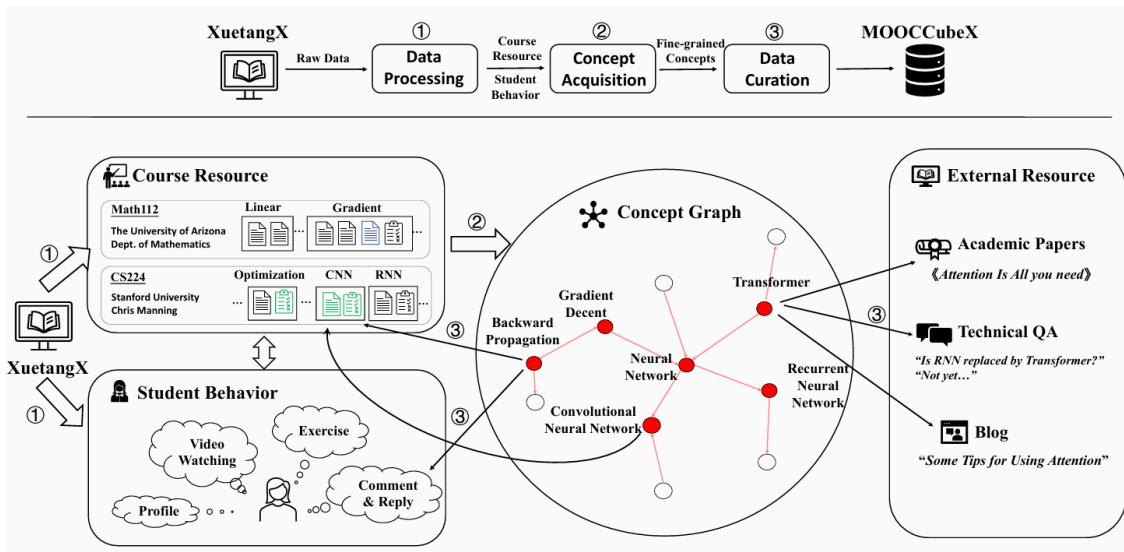


Figure 1: The structure of the MOOCubeX repository and the construction framework.

5.1.2.4. Quản lý Dữ liệu (Data Curation)

Quá trình xây dựng đồ thị khái niệm chi tiết giúp liên kết các tài nguyên MOOC không đồng nhất với nhau và tích hợp nhiều loại tài nguyên bên ngoài. Quản lý dữ liệu trong **MOOCubeX** gồm hai phần chính:

1. Gán Nhãn Khái Niệm cho Nhiều Loại Tài Nguyên (Multi-resource Concept Annotation)

- Các tài nguyên MOOC hiện tại chỉ có liên kết thưa thớt qua cấu trúc khóa học.
- Đồ thị khái niệm giúp liên kết các tài nguyên dựa trên mức độ tri thức.

- Video được gán nhãn khái niệm một cách tự nhiên vì khái niệm được trích xuất từ phụ đề video.
- Các bài tập, bình luận và phản hồi không có nhãn khái niệm trực tiếp. Để giải quyết:
 - Lấy các khái niệm từ video trong cùng chương làm ứng viên.
 - Dùng BERT embedding để so khớp top 1-3 khái niệm có độ tương đồng cosine ≥ 0.8 .
- Kết quả:
 - Mỗi khóa học liên kết với **26 khóa học khác** qua 10 khái niệm chung.
 - Mỗi video có **422 video liên quan** dựa trên 3 khái niệm chung.
 - Mỗi bài tập có **364 mối quan hệ** qua 1 khái niệm chung.

2. Tích Hợp Tài Nguyên Bên Ngoài (External Resource Curation)

- Tìm kiếm tài nguyên bên ngoài để bổ sung vào hệ thống:
 - **Bài báo học thuật:** Tìm 10 bài báo liên quan từ **ArnetMiner** cho mỗi khái niệm.
 - **Blog và Hỏi-Đáp Kỹ thuật:**
 - Tìm bài viết liên quan từ **CSDN** (blog kỹ thuật).
 - Thu thập câu hỏi và câu trả lời từ **Zhihu** (diễn đàn hỏi-đáp).
- Dữ liệu này được lưu trữ vào **MOOCubeX** và liên kết với tài nguyên trong hệ thống thông qua các khái niệm.

5.1.2.5. Hành vi sinh viên (Student Behavior)

1. Xem video (Video Watching)

- Dữ liệu vẫn có độ thưa thớt do hành vi sinh viên không đồng đều.
- **Phân phối tần suất xem video có dạng đuôi dài (long-tail distribution):**
 - **28.5% sinh viên** xem video phổ biến nhất.
 - **39.2% video** chỉ được xem đúng một lần.
- Cho thấy sự mất cân bằng lớn trong hành vi xem video.

2. Làm bài tập & giải quyết vấn đề (Exercising and Problem Solving)

- Phân phối **cân bằng hơn**, có dạng gần với **phân phối chuẩn (normal distribution)**.
- Khi phân tích sâu hơn về tỷ lệ hoàn thành bài tập:
 - Mỗi khoảng tần suất xuất hiện với độ cân bằng cao.
- Một nguyên nhân tiềm năng:
 - Điểm số cuối khóa và **chứng chỉ MOOCs** thường liên quan chặt chẽ đến mức độ tham gia bài tập.

- Dữ liệu này có thể hỗ trợ nghiên cứu giáo dục để phát triển **mô hình giảng dạy** giúp tăng mức độ tham gia khóa học.

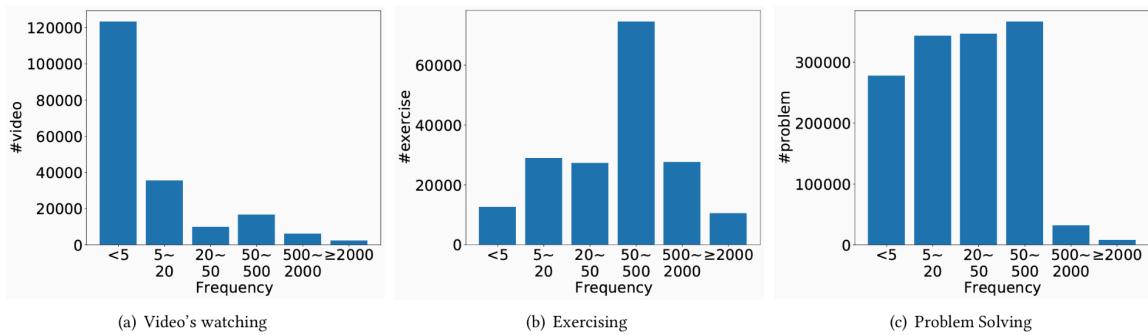


Figure 4: The distribution of student behavioral data.

Hình 1: Biểu đồ thể hiện sự phân phối dữ liệu trong video watching behaviors và the exercising

5.1.2.6. Mô tả tổng quan bộ dữ liệu

Bộ dữ liệu MOOCubeX chứa 4.216 khóa học, 230.263 video giảng dạy, 358.265 bài tập, 637.572 khái niệm chi tiết (fine-grained), hơn 296 triệu bản ghi thô về hành vi của người học và ghi nhận hành vi học tập của 3.330.294 sinh viên được chia thành 2 phần chính: Tài nguyên khóa học (Course resources) và Hành vi người học (Student behavior).

Course resources: chứa thông tin chi tiết về các tài nguyên học tập trong từng khóa học, bao gồm:

- Thông tin khóa học: Bao gồm tên khóa học, mã khóa học, lĩnh vực, và độ dài khóa học.
- Nội dung học tập: Danh sách các tài nguyên như video bài giảng, tài liệu tham khảo, bài tập, và đề thi.
- Thời gian và tiến trình học: Ghi lại các mốc thời gian bắt đầu và kết thúc của các khóa học cũng như các yêu cầu về tiến trình học.
- Cấu trúc khóa học: Sự phân chia thành các phần hoặc tuần học, số lượng bài giảng và bài tập yêu cầu.
- Thông tin liên quan khác: thông tin về giáo viên giảng dạy, trường học Course resources giúp xác định rõ ràng khối lượng và nội dung học tập mà sinh viên phải hoàn thành trong mỗi khóa học. Thông tin này có thể hỗ trợ phân tích mức độ khó dễ

của khóa học và các yếu tố ảnh hưởng đến kết quả học tập. Student behaviors: tập trung vào việc ghi lại thông tin người học và các hành vi học tập cũng như tương tác của người dùng trên nền tảng XuetangX, bao gồm:

- Thông tin người học: Bao gồm tên người dùng, mã người dùng, giới tính, trường,...
- Hoạt động xem video: Tổng thời gian xem video, tần suất truy cập các bài giảng video trong từng khóa học.
- Tham gia diễn đàn thảo luận: Số lượng bài viết, bình luận, và trả lời của sinh viên trong các diễn đàn khóa học.
- Làm bài tập và bài kiểm tra: Ghi nhận các nỗ lực làm bài tập, bài kiểm tra và điểm số đạt được.
- Tương tác trên Xiaomu: Sự tương tác của người học với Xiaomu (QA bot của XueTangX). Student behaviors phản ánh cách thức sinh viên tương tác với khóa học, giúp xác định các hành vi học tập tích cực hay tiêu cực. Dữ liệu đã qua xử lý để giảm việc tiết lộ thông tin cá nhân nhạy cảm của người dùng. Hai phần dữ liệu này kết hợp với nhau tạo nên một bức tranh toàn diện về cả nội dung học tập và cách sinh viên tiếp cận, từ đó hỗ trợ phân tích và dự đoán hiệu suất học tập.
- Bảng thống kê số lượng chi tiết của từng loại tài nguyên có trong bộ dữ liệu:

Tên tài nguyên	Số lượng
Tài nguyên khóa học	3,781 khóa học
Tài nguyên video	59,581 video
Tài nguyên vấn đề	2,454,422 vấn đề
Tài nguyên trường học	429 trường học
Tài nguyên giảng viên	17,018 giảng viên
Tài nguyên Trường học – Lĩnh vực	632 quan hệ
Tài nguyên Khóa học – Trường học	3,983 quan hệ

Tài nguyên Khóa học – Giảng viên	97,192 quan hệ
Tài nguyên phản hồi bình luận	331,011 phản hồi
Tài nguyên User – Xiaomu	108,351 quan hệ
Tài nguyên Course – Comment	10,181,950 quan hệ
Tài nguyên User – Comment	8,422,134 quan hệ
Tài nguyên User – Reply	331011 quan hệ
Tài nguyên Comment - Reply	370,493 quan hệ
Tài nguyên Concepts	637,572 concepts
Tài nguyên Other	210,349 mẫu
Tài nguyên Concept - Other	379,926 quan hệ
Tài nguyên Concept - Paper	5,410,752 quan hệ
Tài nguyên Concept-Problem	33,180 quan hệ
Tài nguyên Concept-Video	624,683 quan hệ
Tài nguyên Concept-Comment	31,074 quan hệ
Tài nguyên CS	492,102 mẫu
Tài nguyên Math	331202 mẫu
Tài nguyên Psy	757,771 mẫu

Bảng 1.1 Bảng thống kê mô tả bộ dữ liệu sử dụng

Mô tả chi tiết của các bảng dữ liệu trong bộ dữ liệu MOOCubeX được trình bày trong các mục sau.

5.1.3. Mô tả sơ bộ về tập dữ liệu

Bộ dữ liệu MOOCCubeX chứa 4.216 khóa học, 230.263 video giảng dạy, 358.265 bài tập, 637.572 khái niệm chi tiết (fine-grained), hơn 296 triệu bản ghi thô về hành vi của người học và ghi nhận hành vi học tập của 3.330.294 sinh viên được chia thành 2 phần chính: Tài nguyên khóa học (Course resources) và Hành vi người học (Student behavior).

- Course Resource:**

Phần Tài nguyên khóa học của MOOCCubeX bắt đầu bằng việc thu thập dữ liệu khóa học từ XuetangX. Sau khi loại bỏ các khóa học thử nghiệm và khóa học không còn hoạt động, thông tin chi tiết về 4.216 khóa học đã được thu thập. Ở giai đoạn này, tên và mô tả của mỗi khóa học được lưu trữ dưới dạng văn bản, và mỗi khóa học được gán một mã id. Các khóa học trong MOOCs không độc lập với nhau. Một khóa học bao gồm nhiều chương giảng dạy, và một chương thường bao gồm một loạt video và bài tập. Thông tin có cấu trúc như vậy cũng rất quan trọng, do đó, việc thu thập thông tin liên quan đến khóa học, bao gồm giáo trình của khóa học và danh sách tài nguyên bao gồm (video, bài tập, và bình luận) được lưu trữ dưới dạng danh sách. Ngoài ra, thông tin về giáo viên và trường đại học của khóa học, cùng với giới thiệu về họ được thu thập từ web. Loại thông tin này có thể xây dựng các mối liên kết cho các khóa học và hỗ trợ các nhiệm vụ liên quan như phát hiện phong cách giảng dạy.

- Student behaviours:**

Ngoài các nguồn tài nguyên tĩnh, các loại hành vi của sinh viên cũng rất quan trọng cho nghiên cứu học tập thích nghi, giúp mô hình hóa ý định học tập của sinh viên ở các cấp độ nhận thức và các hoạt động xã hội. Do đó, tác giả thu thập các bản ghi chi tiết từ XuetangX, bao gồm: hồ sơ sinh viên, hành vi xem video, bài tập và thảo luận. Các hành vi này tự nhiên liên kết với các nguồn tài nguyên của khóa học. Mặc dù đã có giấy phép từ nền tảng, tác giả vẫn cần thực hiện các hoạt động giảm nhẹ cảm như ẩn danh trong quá trình xử lý dữ liệu.

Mô tả chi tiết của các bảng dữ liệu trong MOOCCubeX như các mục sau:

5.1.3.1. Course resources

5.1.3.1.1 Course Info (entity)

Mô tả: Thông tin của các khóa học và các thông tin có liên quan đến khóa học.

Tên file: **entities/course.json**

Số lượng mẫu: **3781** mẫu

Trường	Nội dung	Kiểu dữ liệu	Miền giá trị
about	Giới thiệu khóa học	string	
id	id của khóa học	string	
field	Danh sách các lĩnh vực mà khóa học thuộc về	array của các string	
name	Tên của khóa học	string	
prerequisites	Mô tả những kiến thức tiên quyết	string	
resource	Danh sách các tài nguyên, có thể là một video hoặc một nhóm các bài tập. Object gồm các trường: - resource_id (string): ID của khóa học. Nếu bắt đầu bằng “V_” thì nó là một Video (resource_id được xem như video_id), nếu bắt đầu bằng “Ex_” thì nó là một nhóm Exercise (resource_id được xem như exercise_id). Xem thêm lưu ý	list của các object	

	<p>dưới bảng này.</p> <ul style="list-style-type: none"> - chapter (string): số chapter - titles (list<string>): danh sách các tựa đề, gồm chapter title, video title, v.v, có tối đa 3 cấp độ tiêu đề. 		
--	---	--	--

Mỗi tài nguyên là một video hoặc một bộ bài tập. Các trường dữ liệu được hiển thị trong bảng dưới đây.

Trường	Mô tả
resource_type	Loại tài nguyên: video (một video) hoặc exercise (một nhóm bài tập)
resource_id	ID của tài nguyên
chapter	Số chương
titles	Danh sách tiêu đề, bao gồm tiêu đề chương, tiêu đề video, v.v. Có tối đa 3 cấp độ tiêu đề.

Video

Nếu **resource_type** là video, thì nó sẽ có **video_id** bắt đầu bằng **V_**. Trong bài viết này, **video_id** được gọi là **video_id**.

- Nhiều **video_id** có thể tương ứng với một ccid, và ccid xác định duy nhất một video.
- Các **video_id** này thể hiện cùng một video ccid nhưng tại các thời điểm bắt đầu khác nhau (ví dụ: Xuân 2018 / Thu 2020, v.v.).
- Mối quan hệ giữa **video_id** và **ccid** có thể được tìm thấy trong tệp **relations/video_id-ccid.txt**.
- Phụ đề của video có thể được tìm thấy trong **entities/video.json** thông qua ccid.

Bài tập (Exercise)

- Mỗi bộ bài tập (exercise) tương ứng với nhiều câu hỏi (problem).
- Mỗi quan hệ cụ thể có thể được tìm thấy trong tệp **relations/exercise-problem.json**, nơi mỗi exercise sẽ có danh sách các **problem_id** tương ứng.

5.1.3.1.2 Video (entity)

- Mô tả: Tên video và chú thích. Nội dung của video, khóa học của nó, chương và thứ tự có thể tìm thấy trong file course.json
- Tên file: **entities/video.json**
- Số lượng mẫu: **59581** mẫu

Trường	Nội dung	Kiểu dữ liệu	Miền giá trị
ccid	id duy nhất của video	string	
name	tên của video	string	
start	thời gian bắt đầu mỗi câu phụ đề của video	list<float>	[0, +∞)
end	thời điểm mỗi câu của phụ đề video kết thúc	list<float>	[0, +∞)
text	nội dung phụ đề từng câu trong video	list<string>	

5.1.3.1.3 Problem (entity)

- Mô tả: Chứa nội dung cụ thể của bài tập trong khóa học. Mỗi nhóm bài tập (excercise) sẽ tương ứng với nhiều câu hỏi (problem).
- Tên file: **entities/problem.json**
- Số lượng mẫu: **2454422** mẫu

- Dung lượng: 2.1Gb

Trường	Nội dung	Kiểu dữ liệu	Miền giá trị
problem_id	id của vấn đề, bắt đầu bằng Pm_	int	
excercise_id	id của bài tập, bắt đầu bằng Ex_	string	
language	ngôn ngữ mô tả của vấn đề	string	{'English', 'Chinese'}
title	tựa của bài tập	string	
content	mô tả vấn đề	string	
option	lựa chọn của vấn đề	object chứa các cặp key, value. Key có kiểu string, là ký hiệu A, B, C, ... của lựa chọn. Value có kiểu string, là mô tả của lựa chọn	Key nhận các giá trị như: "A", "B", "C", "D", ...
answer	câu trả lời cho câu hỏi	string	
score	điểm cho câu hỏi	float	
type	loại câu hỏi	int	
typetext	loại câu hỏi	string	

location	chương của vấn đề	string	
context_id	các leaf_id liên quan đến vấn đề	list<int>	

5.1.3.1.4. School (entity)

- Mô tả: Thông tin chi tiết về các trường đại học có dạy các khóa học.
- Tên file: **entities/school.json**
- Số lượng mẫu: **429** mẫu
- Dung lượng: 613Kb

Trường	Nội dung	Kiểu dữ liệu	Miền giá trị
id	ID trường, bắt đầu với S_	string	
name	Tên tiếng Trung của trường	string	
name_en	Tên tiếng Anh của trường	string	
sign	Tên viết tắt của tên tiếng Anh của trường	string	
about	Giới thiệu	string	
motto	Châm ngôn của trường	string	

5.1.3.1.5. Teacher (entity)

- Mô tả: Thông tin về các giảng viên tham gia giảng dạy các khóa học.
- Tên file: **entities/teacher.json**
- Số lượng mẫu: **17018** mẫu.

Trường	Nội dung	Kiểu dữ liệu	Miền giá trị
id	ID của giáo viên, bắt đầu với T_	string	
name	Tên tiếng Trung của giáo viên	string	
name_en	Tên tiếng Anh của giáo viên	string	
about	Hồ sơ của giáo viên	string	
job_title	chức danh công việc	string	
org_name	tổ chức liên kết	string	

5.1.3.1.6 Course - Field (relation)

- Mô tả: Các lĩnh vực của khóa học được xác định thủ công dựa trên 88 lĩnh vực trong *Danh mục ngành, chuyên ngành cấp bằng tiến sĩ, thạc sĩ và đào tạo sau đại học* do Bộ Giáo dục ban hành năm 1997. Mỗi khóa học có thể thuộc một hoặc nhiều lĩnh vực.

- Tên file: **relations/course-field.json**

- Số lượng mẫu: **632** mẫu

Trường	Nội dung	Kiểu dữ liệu	Miền giá trị
course_id	ID của khóa học	int	
course_name	Tên của khóa học	string	
field	Danh sách lĩnh vực được đánh nhãn thủ công	list<str>, mỗi string đại diện cho một lĩnh vực	

5.1.3.1.7. Course - School (relation)

- Mô tả: Trường mà khóa học tương ứng được dạy.
- Định dạng: {course ID}\t{school ID}
- Tên file: **relations/course-school.txt**
- Số lượng mẫu: **3983** mẫu

5.1.3.1.8. Course - Teacher (relation)

- Mô tả: Giảng viên của khóa học.
- Định dạng: {course ID}\t{teacher ID}
- Tên file: **relations/course-teacher.txt**
- Số lượng mẫu: **97192** mẫu

5.1.3.1.9. Exercise - Problem (relation)

- Mô tả: Tập hợp các vấn đề (câu hỏi) chứa trong bài tập.
- Định dạng: {exercise ID}\t{question ID}
- Mỗi dòng trong bộ Exercise là một bộ gồm bài tập (exercise) tương ứng với câu hỏi (problem). Ví dụ: Ex_143 Pm_1
- Tên file: **relations/exercise-problem.txt**
- Số lượng mẫu: **6252830** mẫu

5.1.3.1.10. Video ID - CCID (relation)

- Mô tả: Video và ccid tương ứng của nó.
- Định dạng: {Video ID}\t{ccid}
- Tên file: **relations/video_id-ccid.txt**
- Số lượng mẫu: **2798892** mẫu

5.1.3.2. Student behaviors.

5.1.3.2.1. Student profile (entity)

- Mô tả: Chứa thông tin cá nhân của người dùng và thông tin về các khóa

học mà người dùng đã đăng ký.

- Tên file: **entities/user.json**

- Số lượng mẫu: 3330294 mẫu

Trường	Nội dung	Kiểu dữ liệu	Miền giá trị
id	Id người dùng, bắt đầu bằng “U_”	string	
name	Tên người dùng	string	
gender	Giới tính	int	
school	Tên trường	string	
year_of_birth	Năm sinh	int	
course_order	Các mã khóa học đã chọn	list<int>	
enroll_time	Thời gian đăng ký tương ứng với từng khoá học	list<DateTime>. DateTime có định dạng “YYYY-MM-DD HH:MM:SS”	

5.1.3.2.2. Comment (entity)

- Mô tả: Thông tin các bình luận của các học sinh (user) đăng tải lên.

- Tên file: **entities/comment.json**

- Dung lượng file: 2.1GB

Trường	Nội dung	Kiểu dữ liệu	Miền giá trị
id	Comment ID, bắt đầu bằng "Cm_"	String	
user_id	ID của người dùng đã bình luận, bắt đầu bằng "U_"	Int	
text	Nội dung bình luận	String	
create_time	Thời gian bình luận	DateTime, có định dạng “YYYY-MM-DD HH:MM:SS”	
resource_id	ID của tài nguyên (như video, exercise) mà user bình luận	String	Có thể nhận giá trị null

Lưu ý: Thực tế, trường resource_id không được nhắc đến trong file user-en.md trên github của nhóm tác giả nhưng dữ liệu thực lại có thêm trường này.

```
[{"id": "Cm_188", "user_id": 11731, "text": "资质统建", "resource_id": "V_454874", "create_time": "2019-09-05 17:12:24"}, {"id": "Cm_190", "user_id": 11731, "text": "资质统建", "resource_id": "V_454874", "create_time": "2019-09-05 17:12:29"}, {"id": "Cm_192", "user_id": 11731, "text": "资质统建", "resource_id": "V_454874", "create_time": "2019-09-05 17:12:29"}]
```

5.1.3.2.3. Reply (entity)

- Mô tả: Thông tin của phần trả lời bình luận (reply) của học sinh (user)
- Tên file: **entities/reply.json**
- Số lượng mẫu: 331011 mẫu
- Dung lượng: 50 MB

Trường	Nội dung	Kiểu dữ liệu	Miền giá trị
id	Reply ID, bắt đầu bằng “Rp_”	string	
user_id	ID của người dùng đã bình luận, bắt đầu bằng "U_"	string	
text	Nội dung phản hồi	string	
create_time	Thời gian phản hồi	DateTime, có định dạng “YYYY-MM-DD HH:MM:SS”	

5.1.3.2.4. User-video (relation)

- Mô tả: Lưu trữ thông tin chi tiết về hành vi xem video của người dùng gồm tốc độ và các bước nhảy thời gian của người dùng khi xem video.
- Tên file: **relations/user-video.json**
- Dung lượng: **3 GB**

Trường	Nội dung	Kiểu dữ liệu	Miền giá trị
user_id	ID của user, bắt đầu bằng “U_”	string	
seq	Mảng, trình tự người dùng xem video, mỗi đối tượng trong mảng là trình tự thời gian người dùng xem một video nhất định, bao gồm thời gian xem video,	list<object>. Mỗi object sẽ gồm 2 trường video_id (string) và segment (list<object>). Mỗi phần tử trong	

	thời gian bắt đầu và kết thúc của video, và tốc độ xem video, v.v.	segment bao gồm các trường start_point (float), end_point (float), speed (float), local_start_time (int)	
--	--	--	--

Lưu ý: Ví dụ về 1 phần tử trong seq.

```
{'video_id': 'V_1395639', 'segment': [ {'start_point': 100.0, 'end_point': 106.25, 'speed': 1.25, 'local_start_time': 1588438980}, {'start_point': 180.0, 'end_point': 186.25, 'speed': 1.25, 'local_start_time': 1588439045} ]}
```

5.1.3.2.5. User-problem

- Mô tả: Ghi lại tất cả các vấn đề, câu hỏi hoặc lỗi mà người học gặp phải trong quá trình học tập.
- Tên file: **relations/user-problem.json**
- Dung lượng: **21 GB**

Trường	Nội dung	Kiểu dữ liệu	Miền giá trị
log_id	ID của bản ghi câu hỏi của người dùng, kết hợp với khóa duy nhất của user_id và problem_id	string	
user_id	ID người dùng, bắt đầu bằng U_	string	
problem_id	ID vấn đề, bắt đầu bằng Pm_	string	
is_correct	Câu hỏi có đúng không	bool	0 hoặc 1
attempts	Số lượng câu hỏi đã thử	int	

score	Điểm của người dùng	float	
submit_time	Thời gian làm câu hỏi	DateTime, có định dạng “YYYY-MM-DD HH:MM:SS”	

5.1.3.2.6. User-xiaomu

- Mô tả: Tương tác của người dùng với Xiaomu (bot QA của XuetangX).

- Tên file: **relations/user-xiaomu.json**

- Số lượng mẫu: **108351** mẫu

Trường	Nội dung	Kiểu dữ liệu	Miền giá trị
user_id	ID của user, bắt đầu bằng “U_”	string	
question_type	Loại câu hỏi của user	string	
question	Câu hỏi hỏi bởi user	string	

- Question_type unique: 26, question_unique: 21449, số lượng loại câu hỏi và số lượng câu hỏi khá nhỏ so với số lượng mẫu

- Có 32 mẫu không có question

- Số câu hỏi ở mỗi dạng câu hỏi có sự chênh lệch lớn.

- Độ dài của các câu hỏi cũng có sự chênh lệch lớn: số câu hỏi độ dài > 1 là 1525, số câu hỏi có độ dài <=1 là 106826.

5.1.3.2.7. Course-comment

- Mô tả: Bình luận của người dùng liên quan đến khóa học.

- Định dạng: {course ID}\t{review ID}.

- Tên file: **relations/course-comment.txt**

- Số lượng mẫu: **10181950** mẫu

1.3.2.8. User-comment

- Mô tả: Bình luận của Người dùng.
- Định dạng: {User ID}\t{Comment ID}.
- Tên file: **relations/user-comment.txt**
- Số lượng mẫu: **8422134** mẫu

1.3.2.9. User-reply

- Mô tả: Phản hồi Bình luận của Người dùng.
- Định dạng: {User ID}\t{Reply ID}.
- Tên file: **relations/user-reply.txt**
- Số lượng mẫu: **331011** mẫu

1.3.2.10. Comment-reply

- Mô tả: Phản hồi bình luận liên quan đến khái niệm.
- Định dạng là {Concept ID}\t{Reply ID}.
- Tên file: **relations/comment-reply.txt**
- Số lượng mẫu: **370493** mẫu

5.1.3.3. Concepts

5.1.3.3.1. Concept

- Mô tả: Một thực thể chính trong hệ thống, đại diện cho các khái niệm hoặc chủ đề mà khóa học trực tuyến đề cập.
- Tên file: **entities/concept.json**
- Số mẫu dữ liệu: **637572** mẫu

Trường	Nội dung	Kiểu dữ liệu	Miền giá trị

id	Mã khái niệm, thể hiện theo format K_{tên khái niệm}_{trường khái niệm}	string	
name	Tên khái niệm (mặc định giống với tên khái niệm trong mã khái niệm)	string	
context	Ngữ cảnh khái niệm xuất hiện trong một phần của Bách khoa toàn thư Wiki/Baidu, câu hỏi và câu trả lời Zhihu(50 ký tự trước và sau)	string	

- Không có mẫu null
- Có 556606 mẫu name phân biệt, trong đó name có độ dài > 1 chỉ có 56634 mẫu
- context_length = 0 chiếm đa số: 536000/637572 mẫu, dài nhất là 216

5.1.3.3.2. Other

- Mô tả: Các tài liệu liên quan được thu thập bên ngoài những khoá học
- Tên file: **entities/other.json**
- Số mẫu dữ liệu: **210349** mẫu

Trường	Nội dung	Kiểu dữ liệu	Miền giá trị
id	Mã dữ liệu	string	
concept	khái niệm của thông tin của tài liệu thu thập được	string	
type	Nguồn dữ liệu, bao gồm [“zhihu”, “baike”, “wiki”]	string	[“zhihu”, “baike”, “wiki”]

content	Nội dung của tài liệu	string	
---------	-----------------------	--------	--

5.1.3.3.3. Paper

- Mô tả: Đại diện cho các bài nghiên cứu, tài liệu học thuật hoặc bài báo liên quan đến các khái niệm trong hệ thống.
- Tên file: **entities/paper.json**
- Dung lượng: **6.8 GB**

Trường	Nội dung	Kiểu dữ liệu	Miền giá trị
id	Mã bài báo	string	
concept	Mã khái niệm	string	
abstract	Phản giới thiệu (abstract) của bài báo	string	[“zhihu”, “baike”, “wiki”]
author	Nội dung của tài liệu	string	
lang	Ngôn ngữ của bài báo	string	“en”: Tiếng Anh “zh”: Tiếng Trung
pages	Số lượng trang	Integer	
num_citation	Số lượng citation (trích dẫn từ bài báo khác) tính trong năm 2020	Integer	

score	Điểm số tương đồng giữa bài báo và khái niệm. Điểm càng cao thì càng liên quan nhiều đến khái niệm	Float	
sourcetype	Nguồn của bài báo (hiện tại tất cả là publication)	String	
title	Tên bài báo	String	
venue	Diễn đàn bài báo được đăng tải	String	
urls	các đường link dẫn đến bài báo	List <String>	
year	năm xuất bản	Year	

5.1.3.3.4. Concept-Other

- Mô tả: Khái niệm liên quan với tài nguyên ngoài môn học
- Định dạng: {concept ID}\t{resource ID}
- Tên file: **relations/concept-other.txt**
- Số lượng mẫu: **379926** mẫu

5.1.3.3.5. Concept-Paper

- Mô tả: Khái niệm liên quan với các bài báo ngoài môn học
- Định dạng: {concept ID}\t{paper ID}
- Tên file: **relations/concept-paper.txt**
- Số lượng mẫu: **5410752** mẫu

5.1.3.3.6. Concept-Problem

- Mô tả: Khái niệm liên quan với các vấn đề

- Định dạng: {Concept ID}\t{Question ID}

- Tên file: **relations/concept-problem.txt**

- Số lượng mẫu: **33180** mẫu

5.1.3.3.7. Concept-Video

- Mô tả: Khái niệm liên quan đến video

- Định dạng: {concept ID}\t{ccid}

- Tên file: **relations/concept-video.txt**

- Số lượng mẫu: **624683** mẫu

5.1.3.3.8. Concept-Comment

- Mô tả: Khái niệm liên quan đến bình luận

- Định dạng: {concept ID}\t{review ID}

- Tên file: **relations/concept-comment.txt**

- Số lượng mẫu: **31074** mẫu

5.1.3.4. Prerequisites

5.1.3.4.1. CS.json

- Mô tả: Dự đoán và chú thích của con người về các tiên điều kiện của môn Khoa học Máy tính.

- Tên file: **prerequisites/cs.json**

- Số lượng mẫu: **492102** mẫu

Trường	Nội dung	Kiểu dữ liệu	Miền giá trị
c1	Khái niệm điều kiện tiên quyết	string	
c2	Khái niệm điều kiện sau sửa	string	

	chưa		
ground_truth	Chỉ ra có mối quan hệ sửa chữa tuần tự hay không, 1 có nghĩa là có, 0 có nghĩa là không.	int	0 hoặc 1 hoặc -1
text_predict	Cung cấp kết quả dự đoán sử dụng đặc điểm văn bản.	list<float>	
graph_predict	Mức độ tin cậy của dự đoán được đạt được bằng các đặc điểm đồ thị.	list<float>	

Ví dụ: { "c1": "操作命令", "c2": "重新启动", "ground_truth": 1, "text_predict": [0.015044071711599827, 0.984955906867981], "graph_predict": [0.00342249171808362, 0.9965775012969971] }

c1: "操作命令" (Lệnh thao tác)

→ Đây là khái niệm tiên quyết, tức là một người cần biết về "操作命令" trước khi học "重新启动".

c2: "重新启动" (Khởi động lại)

→ Đây là khái niệm hậu kỳ, tức là chỉ có thể hiểu và sử dụng "重新启动" nếu đã biết về "操作命令".

ground_truth: 1

→ Nhẫn thực tế cho thấy có quan hệ tiên quyết giữa hai khái niệm này. Nói cách khác, để học "重新启动" (khởi động lại), trước tiên cần hiểu về "操作命令" (lệnh thao tác).

text_predict: [0.015, 0.985]

→ Đây là kết quả dự đoán từ mô hình sử dụng đặc trưng văn bản (text features).

- 0.015 ($\approx 1.5\%$) là xác suất **không có quan hệ** tiên quyết.

- 0.985 ($\approx 98.5\%$) là xác suất **có quan hệ** tiên quyết.
→ Kết quả này cho thấy mô hình rất tin rằng có mối quan hệ giữa hai khái niệm.

graph_predict: [0.003, 0.997]

→ Đây là kết quả dự đoán từ mô hình sử dụng đặc trưng đồ thị (graph features).

- 0.003 ($\approx 0.3\%$) là xác suất **không có quan hệ** tiên quyết.
- 0.997 ($\approx 99.7\%$) là xác suất **có quan hệ** tiên quyết.
→ Dự đoán này còn chắc chắn hơn so với mô hình dùng text, khẳng định mạnh mẽ rằng "操作命令" là tiên quyết của "重新启动".

5.1.3.4.2. Math.json

- Mô tả: Chú thích và dự đoán các khái niệm trong lĩnh vực toán học, theo định dạng giống CS.json.

- Tên file: **prerequisites/math.json**

- Số lượng mẫu: **331202** mẫu

5.1.3.4.3. Psy.json

- Mô tả: Chú thích và dự đoán các khái niệm trong lĩnh vực tâm lý học, theo định dạng giống CS.json.

- Tên file: **prerequisites/psy.json**

- Số lượng mẫu: **757771** mẫu

5.1.4. Nhận xét bộ dữ liệu và dự đoán mục tiêu sử dụng của bộ dữ liệu

5.1.4.1. Nhận xét

Sau khi đọc và khảo sát sơ lược bộ dữ liệu, nhóm rút ra được một số nhận xét sau:

- Đây là một bộ dữ liệu có kích thước lớn, do đó từ bộ dữ liệu này có thể hỗ trợ việc khám phá dữ liệu phục vụ cho các mục đích hỗ trợ học tập với các phương pháp tiếp cận học máy, học sâu, ...
- Các khóa học không tồn tại độc lập mà được tổ chức theo chương, bài giảng và tài nguyên liên quan. Điều này giúp dễ dàng phân tích cấu trúc nội dung giảng dạy cũng như mối quan hệ giữa các khóa học.

- Hành vi sinh viên được ghi nhận chi tiết, liên kết với tài nguyên khóa học, giúp nghiên cứu chuyên sâu về hành vi học tập thích nghi, dự đoán kết quả học tập và cải thiện trải nghiệm học trực tuyến.
- Nhìn chung, đây là một bộ dữ liệu không đồng nhất nhưng được tổ chức một cách bài bản, linh hoạt với mức độ chi tiết rất cao. Điều này giúp cho việc sử dụng tài nguyên có thể linh hoạt với nhiều mục đích sử dụng khác nhau, đồng thời việc tìm kiếm dữ liệu cũng như thiết lập các mô hình để khai phá dữ liệu cũng dễ dàng hơn.

5.1.4.2. Dự đoán mục tiêu sử dụng bộ dữ liệu

- MOOCCubeX là một bộ dữ liệu có liên quan đến chủ đề học tập nên nhóm dự định sử dụng bộ dữ liệu để thực hiện “*Dự đoán kết quả học tập của học viên đối với một khóa học cụ thể theo 5 cấp độ*”.

- **Mục tiêu sử dụng bộ dữ liệu:** Nhóm tập trung vào dữ liệu liên quan đến người học như nhân khẩu học và hành vi tương tác của người dùng với khóa học trong bộ dữ liệu **student behavior (User)**. Với dữ liệu liên quan đến khóa học **Course Resources (Course, Exercise, Problem)** như dữ liệu **exercising** chứa dữ liệu bài tập và điểm số của người học, nhóm có thể tính được **điểm số của bài final exam** của khóa học và tiến hành phân loại gán nhãn theo thang điểm đã khảo sát. Để đưa ra cảnh báo sớm cho người học, nhóm tập trung xây dựng mô hình máy học dự đoán tốt cho 2 mức độ thấp nhất ,

- **Mục tiêu của bài toán** là xây dựng một mô hình dự đoán kết quả học tập của học viên đối với một khóa học cụ thể dựa trên dữ liệu từ bộ MOOCCubeX. Bộ dữ liệu này bao gồm các thông tin quan trọng như hành vi học tập, thời gian xem video, tần suất làm bài tập, mức độ tương tác trong thảo luận, và kết quả bài kiểm tra.

Mô hình sẽ phân tích và nhận diện các yếu tố ảnh hưởng lớn đến kết quả học tập, từ đó đưa ra dự đoán về khả năng hoàn thành khóa học.

Kết quả ứng dụng từ mô hình có thể hỗ trợ các nền tảng giáo dục trong việc cá nhân hóa lộ trình học tập, giúp giảng viên điều chỉnh nội dung và phương pháp giảng dạy,

đồng thời tối ưu hóa trải nghiệm học tập, nâng cao tỷ lệ hoàn thành khóa học và chất lượng giáo dục trực tuyến.

5.2. Chuẩn bị dữ liệu

5.2.1. Dữ liệu thực nghiệm

Nhóm xác định các file dữ liệu chính cần được xử lý trước khi kết hợp thành bộ dữ liệu hoàn chỉnh bao gồm:

- entities/user.json
- entities/course.json
- entities/problem.json
- relations/user-problem.json
- relations/user-video.json
- entities/reply.json
- entities/comment.json

Ngoài ra, nhóm còn sử dụng thêm các bộ dữ liệu khác như entities/teacher.json, relations/comment-reply.txt, relations/exercise-problem.txt, relations/video_id-ccid.txt,... để điền khuyết và xác minh tính chính xác của dữ liệu.

5.2.2. Khám phá và trích xuất dữ liệu

5.2.2.1. Xử lý bộ dữ liệu entities/user.json

5.2.2.1.1. Thông kê mô tả về bộ dữ liệu

id	name	gender	school	year_of_birth	course_order	enroll_time
U_22	我	0.0		2015.0	[682129, 2294668]	[2019-10-12 10:28:02, 2020-11-21 14:03:28]
U_24	王帅国	1.0	清华大学	6558.0	[597214, 605512, 597211, 597314, 597208, 62950...]	[2019-05-20 16:06:48, 2019-05-24 19:34:43, 201...]
U_25	王帅国	0.0	清华大学	NaN	[1903985]	[2020-08-07 18:59:13]
U_53	于歆杰	1.0	清华大学	1973.0	[696679, 1704639, 943255, 1729417, 682164, 177...]	[2020-03-01 21:24:30, 2020-03-12 16:17:02, 202...]
U_54	马昱春	2.0	清华大学	NaN	[682442, 682164, 1748240, 1778890, 1829031, 17...]	[2019-10-09 02:17:49, 2019-11-08 00:49:03, 202...]

- Xác định kích thước dữ liệu:
 - Mô tả: Để biết được số hàng, số cột của dữ liệu.
 - Hàm sử dụng: dataframe.shape
 - Input: dữ liệu được lưu dưới dạng dataframe
 - Output: Một tập hợp gồm hai phần tử. Phần tử đầu tiên là số hàng, phần tử thứ hai là số cột.
 - Nhận xét: Dữ liệu có 3330294 hàng và 7 cột.

`df.shape`

(3330294, 7)

Xác định kích thước dữ liệu

5.2.2.1.2. Xử lý các cột dữ liệu bị thiếu

- Xác định kiểu dữ liệu và các cột bị thiếu dữ liệu:
 - Mô tả: Xác định kiểu dữ liệu của từng cột trong dữ liệu và các cột nào bị thiếu dữ liệu.
 - Hàm sử dụng: dataframe.info(show_counts = True)
 - Input: dữ liệu được lưu dưới dạng dataframe
 - Output: Kiểu dữ liệu và số giá trị không phải null tương ứng với từng cột trong dataframe
 - Nhận xét:
 - Từ kết quả ta có thể thấy 5 cột (id, name, school, course_order, enroll_time) có kiểu dữ liệu là object, và cột gender, year_of_birth có kiểu dữ liệu là float64. Object là đối tượng mô tả chung cho nhiều dạng dữ liệu, do đó ta có thể dựa vào phần mô tả dữ liệu để xác định được liệu dữ liệu cụ thể và có chiến lược khai thác hiệu quả (ví dụ cột cột enroll_time lưu ngày đăng ký khóa học của học sinh có thể khai thác dưới dạng dữ liệu Datetime).
 - Dữ liệu có 3330294 hàng nhưng có 3 cột (name, gender, school) có giá trị “None-Null Count” nhỏ hơn 3330294 nên ta có thể suy ra 3 cột này bị thiếu dữ liệu. Đặc biệt, cột year_of_birth có ít giá trị không Null trên tổng thể dữ liệu. Giá trị “None-Null Count” giúp ta xác định nhanh được cột nào bị thiếu dữ liệu, để xử lý sâu hơn ta nên dùng các hàm khác để xác định cụ thể số dữ liệu bị thiếu trong từng cột để có chiến lược xử lý phù hợp.

`df.info(show_counts=True)`

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3330294 entries, 0 to 3330293
Data columns (total 7 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   id          3330294 non-null   object 
 1   name         3330240 non-null   object 
 2   gender        3330240 non-null   float64
 3   school        3330240 non-null   object 
 4   year_of_birth 48530 non-null   float64
 5   course_order  3330294 non-null   object 
 6   enroll_time   3330294 non-null   object 
dtypes: float64(2), object(5)
memory usage: 177.9+ MB

```

- Xác định số dữ liệu bị thiếu trong từng cột:
 - Mô tả: Xác định cụ thể số dữ liệu bị thiếu trong từng cột
 - Thư viện: pandas
 - Hàm sử dụng: dataframe.isnull().sum()
 - Input: dữ liệu được lưu dưới dạng dataframe
 - Output: Số dữ liệu bị thiếu trong từng cột.
 - Nhận xét: Với giá trị “Non-Null Count” của hàm dataframe.info(show_counts = True) ta có thể xác định nhanh được cột nào bị thiếu dữ liệu, tuy nhiên để có chiến lược xử lý phù hợp ta cần biết số lượng bị thiếu cụ thể trong từng cột.
 - Số lượng dữ liệu name, gender và school giống nhau và bằng 54 trên tổng số 3330294 hàng.
 - Dữ liệu year_of_birth có đến 3281764 dữ liệu bị thiếu.

```
df.isnull().sum()
```

id	0
name	54
gender	54
school	54
year_of_birth	3281764
course_order	0
enroll_time	0
dtype:	int64

- Mô tả: Xác định cụ thể phần trăm dữ liệu bị thiếu trong từng cột
- Thư viện: pandas
- Hàm sử dụng: dataframe.isnull().mean()
- Input: dữ liệu được lưu dưới dạng dataframe
- Output: Phần trăm dữ liệu bị thiếu trong từng cột.
- Nhận xét: Cột year_of_birth có 98.54% giá trị bị thiếu (Null). Do đó, trong giai đoạn xử lý dữ liệu, chúng ta có thể loại bỏ cột này để thuận tiện hơn cho quá trình phân tích và xây dựng mô hình dự đoán.

```
df.isnull().mean()
```

```
id          0.000000
name        0.000016
gender      0.000016
school      0.000016
year_of_birth 0.985428
course_order 0.000000
enroll_time 0.000000
dtype: float64
```

Xác định phần trăm dữ liệu bị thiếu

- Xác định cụ thể những hàng có name, gender, school có dữ liệu bị thiếu: Nhóm in những hàng bị thiếu dữ liệu tại cột name trước.

- Mô tả: Xác định cụ thể những hàng có name có dữ liệu bị thiếu
- Thư viện: Pandas
- Hàm sử dụng: dataframe[datafram["thuoc_tinh"].isna()]
- Input: Dữ liệu được lưu dưới dạng dataframe
- Output: Những hàng có giá trị trong cột “thuoc_tinh” là NaN
- Nhận xét: Tổng quan, khi giá trị name bị thiếu thì các cột gender và school cũng bị thiếu. Điều này có thể cho thấy rằng những bản ghi thiếu thông tin về tên thường đi kèm với việc thiếu thông tin về giới tính và trường học. Trong quá trình xử lý dữ liệu, chúng ta cần xem xét cách xử lý các bản ghi này, có thể loại bỏ hoặc điền giá trị thay thế tùy theo mục đích phân tích và mô hình dự đoán.

```
# Lọc các dòng có name bị NaN
nan_name_df = df[df["name"].isna()]
```

	id	name	gender	school	year_of_birth	course_order	enroll_time
973231	U_13894470	None	NaN	None	NaN	[735031, 681299, 677038]	[2019-11-05 23:06:24, 2019-12-16 19:19:59, 201...]
1682593	U_19334113	None	NaN	None	NaN	[681655]	[2020-02-11 20:32:39]
1724824	U_19951280	None	NaN	None	NaN	[696911, 676658]	[2020-02-12 20:46:40, 2020-02-17 20:43:12]
1858592	U_22799383	None	NaN	None	NaN	[680788]	[2020-06-04 14:13:54]
1873334	U_22854528	None	NaN	None	NaN	[682222]	[2020-02-16 19:57:53]
1881143	U_22890552	None	NaN	None	NaN	[696884]	[2020-02-16 22:01:16]
1896142	U_23170740	None	NaN	None	NaN	[927963]	[2020-02-18 11:42:40]
1900764	U_23283404	None	NaN	None	NaN	[696994, 697188, 676932]	[2020-02-17 15:00:30, 2020-02-18 14:08:54, 202...]
1900765	U_23289891	None	NaN	None	NaN	[677097]	[2020-02-17 15:14:54]
1903337	U_23346151	None	NaN	None	NaN	[948259]	[2020-05-01 15:09:34]
1913718	U_23532734	None	NaN	None	NaN	[629559]	[2020-02-17 20:35:29]
1916216	U_23565866	None	NaN	None	NaN	[866770]	[2020-02-17 22:28:12]
1916217	U_23565873	None	NaN	None	NaN	[799800]	[2020-02-17 23:25:58]
1923522	U_23909831	None	NaN	None	NaN	[682671, 707081]	[2020-02-25 08:14:21, 2020-02-25 08:21:02]

- Xác định các hàng có dữ liệu bị thiếu giống nhau ở các cột name, gender, school
 - Phương pháp thực hiện:
 - Kiểm tra các hàng có giá trị thiếu (NaN) trong ba cột name, gender, và school.
 - Nhóm các hàng bị thiếu theo cùng một mẫu để xác định liệu có mối quan hệ giữa việc thiếu dữ liệu ở các cột này hay không.
 - Đếm số lượng các hàng có cùng kiểu thiếu dữ liệu để rút ra nhận xét.
 - Nhận xét:
 - Khi cột name bị thiếu dữ liệu (NaN), thì cột gender và school cũng bị thiếu cùng một lúc.
 - Điều này cho thấy rằng dữ liệu của ba cột này có mối liên kết với nhau, có thể do nguồn dữ liệu bị thiếu hoặc cách thu thập dữ liệu đã làm cho thông tin bị mất đồng loạt.
 - Nếu cần xử lý, có thể loại bỏ các hàng bị thiếu này hoặc cố gắng khôi phục dữ liệu bằng cách điền giá trị phù hợp.

```
# Kiểm tra gender và school có NaN khi name NaN không
nan_counts = nan_name_df[["gender", "school"]].isna().sum()

# Hiển thị kết quả
print("Số lượng dòng có 'name' bị NaN và các cột khác bị NaN:")
print(nan_counts)
```

Số lượng dòng có 'name' bị NaN và các cột khác bị NaN:

```
gender    54
school    54
dtype: int64
```

5.2.2.1.3. Trích xuất thông tin từ các cột dạng danh sách

Các cột chứa danh sách dữ liệu (ví dụ: danh sách thời gian đăng ký khóa học) thường khó xử lý trực tiếp khi phân tích dữ liệu hoặc xây dựng mô hình. Việc trích xuất các đặc trưng quan trọng giúp:

- Dễ dàng phân tích xu hướng và hành vi của người dùng theo thời gian.
- Giảm độ phức tạp khi làm việc với dữ liệu dạng danh sách.
- Tạo thêm các đặc trưng hữu ích để cải thiện chất lượng mô hình dự đoán.

a. *Thêm tổng số lượng khóa học cho mỗi người dùng*

Mô tả: Trong dataset, cột course_order chứa danh sách các ID khóa học mà mỗi người dùng đã đăng ký. Ta cần thêm một cột mới number_of_courses để biểu thị tổng số khóa học mà mỗi người dùng đã tham gia.

Ví dụ, nếu course_order = [682129, 2294668], thì number_of_courses = 2.

Hàm sử dụng:

1. apply(): Duyệt qua từng phần tử của một cột và áp dụng một hàm tùy chỉnh.
2. len(): Tính độ dài của một list
3. isinstance(courses, list): Kiểm tra nếu course_order là danh sách (list), thì tính len(). Nếu giá trị là NaN hoặc kiểu khác, gán giá trị 0.

Input: Mỗi hàng trong cột course_order là một danh sách.

Output: Số lượng khóa học từng học sinh đã đăng kí.

Nhận xét (Lợi ích):

- Hỗ trợ phân tích dữ liệu người dùng
 - Xác định nhóm người dùng đăng ký nhiều khóa học nhất.
 - Phân tích thói quen học tập dựa trên số lượng khóa học đã tham gia.
- Dễ dàng trực quan hóa dữ liệu
 - Biểu đồ histogram về phân phối số lượng khóa học trên mỗi người dùng.
 - So sánh số lượng khóa học giữa các nhóm người dùng khác nhau.

```
# Giả định df là DataFrame ban đầu
df["number_of_courses"] = df["course_order"].apply(lambda courses: len(courses) if isinstance(courses,
```

	id	name	gender	school	year_of_birth	course_order	enroll_time	number_of_courses
0	U_22	我	0.0		2015.0	[682129, 2294668]	[2019-10-12 10:28:02, 2020-11-21 14:03:28]	2
1	U_24	王帅国	1.0	清华大学	6558.0	[597214, 605512, 597211, 597314, 597208, 62950...]	[2019-05-20 16:06:48, 2019-05-24 19:34:43, 201...]	65
2	U_25	王帅国	0.0	清华大学	Nan	[1903985]	[2020-08-07 18:59:13]	1
3	U_53	于歆杰	1.0	清华大学	1973.0	[696679, 1704639, 943255, 1729417, 682164, 177...]	[2020-03-01 21:24:30, 2020-03-12 16:17:02, 202...]	8
4	U_54	马昱春	2.0	清华大学	Nan	[682442, 682164, 1748240, 1778890, 1829031, 17...]	[2019-10-09 02:17:49, 2019-11-08 00:49:03, 202...]	9

b. Tách thông tin ngày, tháng, năm và thời gian từ cột enroll_time

Mô tả:

- Cột enroll_time chứa danh sách các thời điểm người dùng đăng ký khóa học dưới dạng chuỗi datetime (YYYY-MM-DD HH:MM:SS).
- Để phân tích và trực quan hóa dữ liệu dễ dàng hơn, ta cần tách các thành phần: năm (year_enroll), tháng (month_enroll), ngày (day_enroll), và thời gian (time_enroll) thành các cột riêng biệt.

Các hàm sử dụng:

1. apply(): Duyệt qua từng phần tử của một cột và áp dụng một hàm tùy chỉnh.
2. split(): Chia chuỗi theo dấu phân cách bao gồm “ ” (khoảng trắng) và “-”, .

Nhận xét:

1. Hỗ trợ phân tích xu hướng đăng ký:
 - Xác định năm nào, tháng nào có nhiều người đăng ký nhất.
 - Tìm hiểu xem người dùng có xu hướng đăng ký vào ngày cụ thể nào trong tháng không.
2. Dễ dàng trực quan hóa dữ liệu:
 - Có thể vẽ biểu đồ theo từng năm, từng tháng để thấy sự thay đổi theo thời gian.
 - Kiểm tra xem có mối quan hệ giữa thời gian trong ngày (time_enroll) và số lượng đăng ký không.
3. Hỗ trợ xây dựng mô hình dự đoán:
 - Các mô hình Machine Learning có thể sử dụng thông tin này để dự đoán xu hướng đăng ký trong tương lai.

```

# Tách năm (year_enroll) từ enroll_time và chuyển về dạng số nguyên
df["year_enroll"] = df["enroll_time"].apply(lambda dates: [int(date.split(" ")[0].split("-")[0])])

# Tách tháng (month_enroll) từ enroll_time và chuyển về dạng số nguyên
df["month_enroll"] = df["enroll_time"].apply(lambda dates: [int(date.split(" ")[0].split("-")[1])])

# Tách ngày (day_enroll) từ enroll_time và chuyển về dạng số nguyên
df["day_enroll"] = df["enroll_time"].apply(lambda dates: [int(date.split(" ")[0].split("-")[2])])

# Tách thời gian (time_enroll) từ enroll_time (giữ nguyên dạng chuỗi HH:MM:SS)
df["time_enroll"] = df["enroll_time"].apply(lambda dates: [date.split(" ")[1] for date in dates])

```

id	name	gender	school	year_of_birth	course_order	enroll_time	year_enroll	month_enroll	day_enroll	time_enroll	
U_22	我	0.0		2015.0	[682129, 2294668]	[2019-10-12 10:28:02, 2020-11-21 14:03:28]	[2019, 2020]	[10, 11]	[12, 21]	[10:28:02, 14:03:28]	
U_24	王帅国	1.0	清华大学	6558.0	[597214, 605512, 597211, 597314, 597208, 62950...]	[2019-05-20 16:06:48, 2019-05-24 19:34:43, 201...]	[2019, 2019, 2019, 2019, 2019, 201...]	[5, 5, 6, 6, 8, 8, 9, 10, 10, 10, 10, 10...]	[20, 24, 11, 12, 17, 15, 15, 15, 11, 6, 8, 9, ...]	[16:06:48, 19:34:43, 02:50:04, 17:22:07, 15:22...]	
U_25	王帅国	0.0	清华大学	NaN	[1903985]	[2020-08-07 18:59:13]	[2020]	[8]	[7]	[18:59:13]	
U_53	于歆杰	1.0	清华大学	1973.0	[696679, 1704639, 943255, 1729417, 682164, 177...]	[2020-03-01 21:24:30, 2020-03-12 16:17:02, 202...]	[2020, 2020, 2020, 2020, 2020, 2020]	[3, 3, 3, 3, 4, 4, 6, 6]	[1, 12, 17, 25, 12, 30, 8, 18]	[21:24:30, 16:17:02, 08:46:12, 19:27:50, 07:47...]	
U_54	马昱春	2.0	清华大学	NaN	[682442, 682164, 1748240, 1778890, 1829031, 17...]	[2019-10-09 02:17:49, 2019-11-08 00:49:03, 202...]	[2019, 2020, 2020, 2020, 2020, 202...]	[10, 11, 4, 5, 6, 6, 6, 7]	[9, 8, 15, 6, 8, 12, 13, 13, 20]	[02:17:49, 00:49:03, 09:12:56, 19:30:08, 20:37...]	

c. Tính khoảng cách (số ngày) giữa các lần đăng ký khóa học của từng học sinh

Mô tả

- Tính khoảng thời gian (tính theo số ngày) giữa các lần đăng ký khóa học của mỗi người dùng.
- Sử dụng các thông tin đã tách trước đó từ cột enroll_time, bao gồm:
 - year_enroll (năm đăng ký)
 - month_enroll (tháng đăng ký)
 - day_enroll (ngày đăng ký)
- Sau đó, tính toán khoảng cách giữa các ngày liên tiếp để xác định khoảng thời gian giữa các lần đăng ký.
- Các bước:
 - Kiểm tra dữ liệu đầu vào:

- Nếu danh sách years rỗng hoặc có ít hơn 2 phần tử (tức là người dùng chỉ có một lần đăng ký), thì trả về None vì không thể tính khoảng cách giữa các lần đăng ký.

2. Chuyển đổi thành datetime:

- Kết hợp year, month, day để tạo danh sách các đối tượng datetime.
- Sắp xếp danh sách theo thứ tự thời gian để đảm bảo tính toán chính xác.

3. Tính khoảng cách giữa các lần đăng ký:

- Sử dụng vòng lặp lặp để tính số ngày giữa từng cặp ngày liên tiếp.

Input: Cột ["year_enroll"] ["month_enroll"] ["day_enroll"] trong dữ liệu.

Output: Danh sách khoảng cách (ngày) giữa 2 khóa học liên tiếp của từng học sinh.

Nhận xét: Có nhiều dữ liệu bị NaN trong cột mới tạo vì có nhiều học sinh chỉ đăng ký một khóa.

1. Phân tích khoảng cách giữa các lần đăng ký khóa học

- Xác định liệu người dùng có xu hướng đăng ký liên tiếp hay không.
- Kiểm tra tần suất đăng ký khóa học theo thời gian.

2. Xác định hành vi người dùng

- Nếu khoảng cách nhỏ → Người dùng có thể đăng ký nhiều khóa học trong thời gian ngắn.
- Nếu khoảng cách lớn → Người dùng có thể quay lại sau một thời gian dài để đăng ký khóa mới.

3. Hỗ trợ dự đoán hành vi đăng ký

- Có thể sử dụng thông tin này để dự đoán khi nào người dùng có thể đăng ký khóa học tiếp theo.
- Giúp hệ thống gợi ý khóa học phù hợp tại thời điểm thích hợp dựa trên khoảng cách giữa các lần đăng ký trước đó.

```
# Hàm tính khoảng cách (số ngày) giữa các lần đăng ký khóa học
def compute_intervals(years, months, days):
    if not years or len(years) < 2: # Xử lý trường hợp danh sách rỗng hoặc người dùng chỉ có một lần đăng ký
        return None

    # Chuyển đổi thành các đối tượng datetime, đảm bảo kiểu dữ liệu chính xác
    dates = sorted([pd.Timestamp(year=int(y), month=int(m), day=int(d))
                   for y, m, d in zip(years, months, days)])

    # Tính khoảng cách (số ngày) giữa các lần đăng ký liên tiếp
    intervals = [(dates[i+1] - dates[i]).days for i in range(len(dates)-1)]

    return intervals

# Áp dụng hàm compute_intervals() cho từng dòng của DataFrame để tạo cột intervals_course
df["intervals_course"] = df.apply(lambda row: compute_intervals(row["year_enroll"], row["month_enroll"], row["day_enroll"]), axis=1)
```

year_of_birth	course_order	enroll_time	year_enroll	month_enroll	day_enroll	time_enroll	number_of_courses	intervals_course	
2015.0	[682129, 2294668]	[2019-10-12 10:28:02, 2020-11-21 14:03:28]	[2019, 2020]	[10, 11]	[12, 21]	[10:28:02, 14:03:28]	2	[406]	
6558.0	[597214, 605512, 597211, 597314, 597208, 62950...]	[2019-05-20 16:06:48, 2019-05-24 19:34:43, 201...]	[2019, 2019, 2019, 2019, 201...]	[5, 5, 6, 6, 8, 8, 8, 9, 10, 10, 10, 10, 10...]	[20, 24, 11, 12, 17, 15, 15, 15, 11, 6, 8, 9, ...]	[16:06:48, 19:34:43, 02:50:04, 17:22:07, 15:22...]	65	[4, 18, 1, 5, 59, 0, 0, 27, 25, 2, 1, 2, 2, 5,...]	
NaN	[1903985]	[2020-08-07 18:59:13]	[2020]	[8]	[7]	[18:59:13]	1	None	
1973.0	[696679, 1704639, 943255, 1729417, 682164, 177...]	[2020-03-01 21:24:30, 2020-03-12 16:17:02, 202...]	[2020, 2020, 2020, 2020, 2020]	[3, 3, 3, 3, 4, 4, 6, 6]	[1, 12, 17, 25, 12, 30, 8, 18]	[21:24:30, 16:17:02, 08:46:12, 19:27:50, 07:47...]	8	[11, 5, 8, 18, 18, 39, 10]	
NaN	[682442, 682164, 1748240, 1778890, 1829031, 17...]	[2019-10-09 02:17:49, 2019-11-08 00:49:03, 202...]	[2019, 2020, 2020, 2020, 2020]	[10, 11, 4, 5, 6, 6, 6, 7]	[9, 8, 15, 6, 8, 12, 13, 13, 20]	[02:17:49, 00:49:03, 09:12:56, 19:30:08, 20:37...]	9	[30, 159, 21, 33, 4, 1, 0, 37]	

d. Thêm cột giá trị nhỏ nhất, giá trị lớn nhất, trung bình của mỗi lần đăng ký khóa học của học sinh

Mô tả: Tính toán các giá trị **khoảng cách lớn nhất, nhỏ nhất và trung bình** giữa các lần đăng ký khóa học của mỗi người dùng.

Nhận xét:

- Xác định tần suất học tập:
 - Nếu min_interval nhỏ, người dùng có xu hướng đăng ký liên tục trong thời gian ngắn.
 - Nếu max_interval lớn, có thể họ chỉ học theo từng giai đoạn dài.
- Đánh giá mức độ cam kết:
 - Nếu avg_interval nhỏ, người dùng có thể rất tích cực học tập.
 - Ngược lại, nếu avg_interval lớn, họ có thể học gián đoạn hoặc mất hứng thú.
- Gợi ý thời điểm nhắc nhở học tập:
 - Nếu một người thường đăng ký khóa mới sau khoảng X ngày, hệ thống có thể gửi thông báo gợi ý vào đúng thời điểm đó.
- Tối ưu hóa chương trình khuyến mãi:
 - Nếu người dùng có xu hướng học sau một khoảng thời gian dài (max_interval lớn), nên tăng có thể gửi khuyến mãi để kích thích họ quay lại sớm hơn.
- Dự đoán khả năng bỏ học:
 - Nếu khoảng cách trung bình (avg_interval) tăng dần theo thời gian, có thể người dùng đang mất hứng thú.

- Mô hình có thể dự đoán những ai có nguy cơ rời bỏ nền tảng học tập.
 - Phân nhóm người dùng theo hành vi học tập:
 - Nhóm học liên tục (low avg_interval)
 - Nhóm học ngắt quãng (high avg_interval)
 - Nhóm học theo mùa (high max_interval nhưng low min_interval)
 - Phân tích thời gian tối ưu giữa các khóa học:
 - Nếu phần lớn người dùng có avg_interval khoảng 30 ngày, có thể khóa học tiếp theo nên được đề xuất sau khoảng thời gian này.
 - Điều chỉnh độ khó của nội dung:
 - Nếu min_interval quá nhỏ, có thể khóa học quá dễ hoặc không đủ thử thách.
 - Nếu max_interval quá lớn, có thể nội dung chưa hấp dẫn đủ để giữ chân người học.

```

# Hàm tính khoảng cách lớn nhất, nhỏ nhất và trung bình giữa các lần đăng ký
def calculate_stats(intervals):
    if intervals is None or len(intervals) == 0:
        return None, None, None # Xử lý trường hợp không có dữ liệu hoặc danh sách rỗng

    return max(intervals), min(intervals), round(np.mean(intervals), 2) # Tính giá trị lớn nhất, nhỏ nhất và trung bình

# Áp dụng hàm calculate_stats để tạo các cột mới trong DataFrame
df[['max_interval', 'min_interval', 'avg_interval']] = df['intervals_course'].apply(lambda x: pd.S

```

e. Phân loại thời gian đăng ký theo buổi “sáng”, “chiều”, “tối”

Mô tả: Phân loại thời gian đăng ký khóa học theo buổi trong ngày

Nhận xét:

- Hiểu rõ thói quen học tập của người dùng:
 - Xác định thời điểm người dùng thường đăng ký khóa học nhất.
 - Hỗ trợ cá nhân hóa trải nghiệm, như gợi ý học tập vào giờ cao điểm.
 - Cải thiện chiến lược quảng cáo & tiếp cận người dùng:
 - Đầu xuất khóa học vào thời gian phù hợp để tăng tỷ lệ đăng ký.
 - Gửi email nhắc nhở hoặc thông báo vào thời điểm người dùng hay hoạt động nhất.
 - Tối ưu hóa lịch trình khóa học & tài nguyên hệ thống:
 - Phân bổ tài nguyên giảng dạy, máy chủ vào khung giờ cao điểm.
 - Điều chỉnh lịch học trực tuyến để phù hợp với xu hướng người học.

```
def classify_time_of_day(times):
    """Phân loại từng thời điểm trong danh sách thành buổi sáng, chiều hoặc tối."""
    if not times or not isinstance(times, list):
        return None # Xử lý trường hợp danh sách rỗng hoặc dữ liệu không hợp lệ

    time_categories = []
    for time in times:
        if isinstance(time, str) and ":" in time: # Đảm bảo dữ liệu là chuỗi thời gian hợp lệ
            hour = int(time.split(":")[0]) # Trích xuất giờ từ chuỗi thời gian
            if 5 <= hour < 12:
                time_categories.append("morning") # Từ 5h đến trước 12h là buổi sáng
            elif 12 <= hour < 18:
                time_categories.append("afternoon") # Từ 12h đến trước 18h là buổi chiều
            else:
                time_categories.append("night") # Từ 18h trở đi là buổi tối
        else:
            time_categories.append(None) # Xử lý dữ liệu thời gian bị lỗi hoặc không hợp lệ

    return time_categories

# Áp dụng hàm để phân loại thời gian trong từng dòng của DataFrame
df["time_of_day"] = df["time_enroll"].apply(classify_time_of_day)
```

Dữ liệu sau khi thêm buổi đăng ký trong ngày

f. Thêm cột giá trị trung bình cho tất cả các tháng mỗi người dùng tham gia & năm bắt đầu khóa học đầu tiên

Mô tả: Tính năm đầu tiên đăng ký và trung bình tháng đăng ký

Nhận xét:

- Phân tích hành vi học tập của người dùng:
 - Xác định thời điểm người dùng bắt đầu học tập.
 - Đánh giá xu hướng đăng ký khóa học theo thời gian.
- Hỗ trợ chiến lược tiếp thị và cải thiện trải nghiệm người dùng:
 - Nhắm mục tiêu người dùng lâu năm với các chương trình ưu đãi phù hợp.
 - Xác định thời điểm cao điểm đăng ký để tối ưu chiến dịch quảng cáo.
- Cải thiện thiết kế khóa học và tối ưu tài nguyên:
 - Điều chỉnh lịch khai giảng theo xu hướng đăng ký.
 - Phân bổ tài nguyên hệ thống hợp lý theo mùa cao điểm.

```
# Hàm tính năm đầu tiên đăng ký và trung bình tháng đăng ký
def calculate_year_month(years, months):
    year_start = min(years) if years else None # Tìm năm đăng ký sớm nhất
    avg_month = round(np.mean(months), 2) if months else None # Tính trung bình tháng đăng ký (làm tròn)
    return year_start, avg_month

# Áp dụng hàm calculate_year_month cho từng dòng để tạo các cột year_start và avg_month_enroll
df[['year_start', 'avg_month_enroll']] = df.apply(lambda row: pd.Series(calculate_year_month(row['
```

1]	[12, 21]	[10:28:02, 14:03:28]	[406]	406.0	406.0	406.00	[morning, afternoon]	2019.0	10.50	
8, 0, ...]	[20, 24, 11, 12, 17, 15, 15, 11, 6, 8, 9, ...]	[16:06:48, 19:34:43, 02:50:04, 17:22:07, 15:22...]	[4, 18, 1, 5, 59, 0, 0, 27, 25, 2, 1, 2, 2, 5,...]	65.0	0.0	8.61	[afternoon, night, night, afternoon, afternoon...]	2019.0	6.38	
8]	[7]	[18:59:13]	None	NaN	NaN	NaN	[night]	2020.0	8.00	
4, 6]	[1, 12, 17, 25, 12, 30, 8, 18]	[21:24:30, 16:17:02, 08:46:12, 19:27:50, 07:47...]	[11, 5, 8, 18, 18, 39, 10]	39.0	5.0	15.57	[night, afternoon, morning, night, morning, af...]	2020.0	4.00	
6, 7]	[9, 8, 15, 6, 8, 12, 13, 13, 20]	[02:17:49, 00:49:03, 09:12:56, 19:30:08, 20:37...]	[30, 159, 21, 33, 4, 1, 0, 37]	159.0	0.0	35.62	[night, night, morning, night, night, afternoo...]	2019.0	6.78	

Dữ liệu sau khi thêm tháng trung bình đăng ký khóa học

g. Thêm cột ngày đầu tiên đăng kí và ngày gần nhất đăng kí khóa học, và khoảng cách giữa 2 ngày

Nhận xét: Lợi ích của việc xác định lần đăng ký đầu tiên và gần nhất

- Phân tích hành vi học tập của người dùng:
 - Xác định người dùng mới so với người dùng đã tham gia lâu năm.
 - Theo dõi quá trình học tập và mức độ trung thành của người dùng.
 - Hỗ trợ chiến lược tiếp thị và giữ chân người dùng:
 - Gửi ưu đãi đặc biệt cho người dùng mới để khuyến khích tham gia nhiều hơn.
 - Nhắc nhở hoặc tái kích hoạt người dùng không đăng ký khóa học trong thời gian dài.
 - Cải thiện trải nghiệm người dùng:
 - Điều chỉnh nội dung và lịch học dựa trên thời gian hoạt động của người dùng.
 - Đưa ra gợi ý khóa học phù hợp với người học dựa trên khoảng thời gian giữa các lần đăng ký.

```
def safe_timestamp(year, month, day):
    """Tạo timestamp hợp lệ, xử lý các giá trị ngày không hợp lệ."""
    try:
        return pd.Timestamp(year, month, day) # Tạo đối tượng timestamp
    except ValueError: # Nếu giá trị ngày không hợp lệ
        print(year, month, day) # In ra để kiểm tra lỗi
        return pd.NaT # Trả về NaT (Not a Time) để xử lý dữ liệu bị lỗi
```

intervals_course	max_interval	min_interval	avg_interval	time_of_day	year_start	avg_month_enroll	first_enroll	latest_enroll
[406]	406.0	406.0	406.00	[morning, afternoon]	2019.0	10.50	2019-10-12	2020-11-21
[4, 18, 1, 5, 59, 0, 0, 27, 25, 2, 1, 2, 2, 5,...]	65.0	0.0	8.61	[afternoon, night, night, afternoon, afternoon...]	2019.0	6.38	2019-05-20	2020-11-21
None	NaN	NaN	NaN	[night]	2020.0	8.00	2020-08-07	2020-08-07
[11, 5, 8, 18, 18, 39, 10]	39.0	5.0	15.57	[night, afternoon, morning, night, morning, af...]	2020.0	4.00	2020-03-01	2020-06-18
[30, 159, 21, 33, 4, 1, 0, 37]	159.0	0.0	35.62	[night, night, morning, night, night, afternoon...]	2019.0	6.78	2019-10-09	2020-07-20

Dữ liệu sau khi thêm ngày đầu tiên và ngày cuối cùng đăng ký khóa học

Mô tả: Tính khoảng thời gian giữa lần đăng ký đầu tiên và lần đăng ký gần nhất

Nhận xét:

- Đánh giá mức độ gắn bó của người dùng:
 - Người dùng có khoảng thời gian dài giữa lần đầu và lần gần nhất có xu hướng học tập liên tục.
 - Người dùng có thời gian ngắn có thể chỉ tham gia một khóa học rồi rời đi.
- Phân tích mô hình học tập:
 - Phát hiện nhóm người dùng học theo đợt (ví dụ: tham gia khóa học theo mùa).
 - Đánh giá mức độ quay lại học tập của người dùng.
- Hỗ trợ chiến lược giữ chân người học:
 - Nhắc nhở người dùng tiếp tục học nếu khoảng thời gian đăng ký ngắn.
 - Gợi ý nội dung phù hợp dựa trên thói quen học tập.

```
: df["enrollment_duration_days"] = (df["latest_enroll"] - df["first_enroll"]).dt.days
```

terval	min_interval	avg_interval	time_of_day	year_start	avg_month_enroll	first_enroll	latest_enroll	enrollment_duration_days
406.0	406.0	406.00	[morning, afternoon]	2019.0	10.50	2019-10-12	2020-11-21	406
65.0	0.0	8.61	[afternoon, night, night, afternoon, afternoon...]	2019.0	6.38	2019-05-20	2020-11-21	551
Nan	Nan	Nan	[night]	2020.0	8.00	2020-08-07	2020-08-07	0
39.0	5.0	15.57	[night, afternoon, morning, night, morning, af...]	2020.0	4.00	2020-03-01	2020-06-18	109
159.0	0.0	35.62	[night, night, morning, night, night, afternoon...]	2019.0	6.78	2019-10-09	2020-07-20	285

Dữ liệu sau khi thêm khoảng cách giữa ngày đầu tiên và cuối cùng đăng ký

h. Tổng kết các cột sau khi trích xuất thông tin từ list

Tổng cộng có cột, ngoài 7 cột ban đầu thì có những cột mới sau:

Trường	Nội dung	Kiểu dữ liệu	Miền giá trị
number_of_courses	Số lượng khóa học đăng ký của người dùng	Integer	Số nguyên không âm (≥ 0)
year_enroll	Năm đăng ký khóa học	Integer	4 chữ số (VD: 2019, 2020, 2021)
month_enroll	Tháng đăng ký khóa học	Integer	1 - 12
day_enroll	Ngày đăng ký khóa học	Integer	1 - 31

time_enroll	Thời gian đăng ký khóa học (ngày giờ)	Datetime (String)	YYYY-MM-DD HH:MM:SS
intervals_course	Khoảng thời gian giữa các lần đăng ký khóa học liên tiếp	List[Float]	Danh sách khoảng thời gian (ngày) giữa các lần đăng ký
max_interval	Khoảng thời gian dài nhất giữa hai lần đăng ký liên tiếp	Float	Số dương hoặc NaN nếu chỉ có 1 khóa học đăng ký
min_interval	Khoảng thời gian ngắn nhất giữa hai lần đăng ký liên tiếp	Float	Số dương hoặc NaN nếu chỉ có 1 khóa học đăng ký
avg_interval	Khoảng thời gian trung bình giữa các lần đăng ký	Float	Số dương hoặc NaN nếu chỉ có 1 khóa học đăng ký
time_of_day	Khoảng thời gian trong ngày đăng ký (sáng, chiều, tối)	List[String]	["morning", "afternoon", "night"]
year_start	Năm đầu tiên đăng ký khóa học	Integer	4 chữ số (VD: 2019, 2020)
avg_month_enroll	Trung bình số tháng đăng ký mỗi năm	Float	Số dương hoặc NaN nếu chỉ có 1 năm đăng ký

first_enroll	Ngày đăng ký khóa học đầu tiên	Date (String)	YYYY-MM-DD
latest_enroll	Ngày đăng ký khóa học gần nhất	Date (String)	YYYY-MM-DD
enrollment_duration_days	Khoảng thời gian từ lần đăng ký đầu tiên đến lần gần nhất	Integer	Số nguyên không âm (≥ 0)

5.2.2.1.4. Tính toán các thống kê số liệu cơ bản

- **Dữ liệu dạng số**

Nhận xét:

- Giá trị gender thuộc phân loại nhưng được tính trong hàm describe() nên giá trị ngày không nên xem xét
- Nhận xét thường trường:

1. Gender

- Giá trị min: **0.0**, max: **232.0**
- **Bất thường:** Giới tính thường chỉ có hai hoặc ba giá trị (ví dụ: 0 = Nam, 1 = Nữ, 2 = Khác), nhưng max lên đến **232.0** → Có thể có lỗi nhập dữ liệu.
- **Giải pháp:** Kiểm tra phân phối dữ liệu, loại bỏ giá trị vượt quá phạm vi hợp lý.

2. Year of Birth

- Giá trị min: **1111.0**, max: **9989.0**
- **Bất thường:**
 - Min **1111.0** → Không thể có năm sinh trước thế kỷ 20.
 - Max **9989.0** → Không hợp lý vì chưa đến năm đó.
- **Giải pháp:**
 - Giới hạn năm sinh hợp lý (ví dụ: **1920 ≤ year_of_birth ≤ 2025**).
 - Thay giá trị bất thường bằng giá trị trung bình hợp lý.

3. Number of Courses

- Max = **3715.0**
- **Bất thường:** Trung bình **3.54** nhưng max lên đến **3715** → Có thể là ngoại lệ hoặc lỗi dữ liệu.
- **Giải pháp:** Xác minh người dùng thực sự có thể đăng ký số lượng khóa học lớn như vậy không.

4. Max Interval, Min Interval, Avg Interval

- Giá trị max: **456.0** (tương tự ở cả 3 cột)

- Không có bất thường rõ ràng, vì số ngày giữa các lần đăng ký có thể dao động lớn.

5. Year Start

- Min = **2019.0**, Max = **2020.0** → Hợp lý nếu dữ liệu chỉ từ giai đoạn 2019-2020.

6. Avg Month Enroll

- Min = **1.0**, Max = **12.0** → Hợp lý vì tháng trong năm từ 1 đến 12.

7. Enrollment Duration Days

- Min = **0.0**, Max = **679.0**
- Không có bất thường rõ ràng, vì có thể có người chỉ đăng ký một khóa học trong một ngày, hoặc có người học kéo dài gần **2 năm**.

	count	mean	min	25%	50%	75%	max	std
gender	3330240.0	0.945575	0.0	0.0	1.0	2.0	232.0	0.83211
year_of_birth	48530.0	2039.016299	1111.0	2020.0	2020.0	2020.0	9989.0	358.674303
number_of_courses	3330294.0	3.54536	1.0	1.0	1.0	2.0	3715.0	10.480855
max_interval	1198696.0	50.242424	0.0	0.0	17.0	72.0	456.0	71.070381
min_interval	1198696.0	18.757496	0.0	0.0	0.0	5.0	456.0	50.418957
avg_interval	1198696.0	28.127005	0.0	0.0	6.92	30.43	456.0	52.326759
year_start	3330294.0	2019.858755	2019.0	2020.0	2020.0	2020.0	2020.0	0.348275
avg_month_enroll	3330294.0	5.062096	1.0	2.0	4.0	7.87	12.0	3.202856
first_enroll	3330294	2020-03-27 02:31:48.220715520	2019-01-24 00:00:00	2020-02-10 00:00:00	2020-02-29 00:00:00	2020-05-12 00:00:00	2020-12-08 00:00:00	NaN
latest_enroll	3330294	2020-04-26 02:54:30.392343552	2019-02-27 00:00:00	2020-02-17 00:00:00	2020-03-21 00:00:00	2020-07-01 00:00:00	2020-12-08 00:00:00	NaN
enrollment_duration_days	3330294.0	30.015766	0.0	0.0	0.0	1.0	679.0	82.35594

- **Dữ liệu là phân loại**

Hàm `.value_counts()` trong **pandas** được sử dụng để đếm số lần xuất hiện của từng giá trị trong một cột của DataFrame hoặc một Series. Tham số `dropna=False` giúp bao gồm cả các giá trị **NaN** trong kết quả thống kê.

```

import pandas as pd

# Lọc danh sách các cột có kiểu dữ liệu object hoặc category
categorical_columns = df.select_dtypes(include=["object", "category"]).columns
print("Categorical Columns:", categorical_columns)

# Đếm số lần xuất hiện của từng giá trị trong các cột liên quan
gender_counts = df["gender"].value_counts(dropna=False)
school_counts = df["school"].value_counts(dropna=False)
year_start_counts = df["year_start"].value_counts(dropna=False)

# Hiển thị kết quả
print("\nGender Counts:\n", gender_counts)
print("\nSchool Counts:\n", school_counts)
print("\nYear Start Counts:\n", year_start_counts)

```

Gender Counts:		School Counts:	
		school	
gender		NaN	2201897
0.0	1221931	清华大学	18412
1.0	1067858	昆明理工大学	15351
2.0	1040449	湖南大学	13388
NaN	54	河南工业大学	10198
232.0	1	...	
3.0	1	faw-vw	1
		Xinxiang Medical University	1
		现在工程学院	1
		长征职业技术学院	1
		厦门市第十中学	1
		Name: count, Length: 24971, dtype: int64	

Year Start Counts:

year_start	count
2020.0	2859905
2019.0	470389
	Name: count, dtype: int64

Nhận xét:

1. Cột gender (Giới tính)

- Có ba giá trị phổ biến: 0.0 (1,221,931 lần xuất hiện), 1.0 (1,067,858 lần xuất hiện), 2.0 (1,040,449 lần xuất hiện). Đây có thể là các mã giới tính thông thường (ví dụ: Nam, Nữ, Khác).
- Có 54 giá trị NaN, tức là thiếu thông tin về giới tính.
- Xuất hiện hai giá trị bất thường: 232.0 và 3.0, mỗi giá trị chỉ xuất hiện một lần. Đây có thể là lỗi nhập liệu hoặc sai sót trong dữ liệu.

Dữ liệu bất thường: Các giá trị 232.0 và 3.0 cần được kiểm tra và xử lý, có thể loại bỏ hoặc thay thế.

2. Cột school (Trường học)

- Có **2,201,897 trường hợp NaN**, tức là dữ liệu bị thiếu rất nhiều.
- Một số trường đại học có số lượng lớn người đăng ký, như 清华大学 (18,412), 昆明理工大学 (15,351), 湖南大学 (13,388), 河南工业大学 (10,198).

- Xuất hiện một số giá trị không phải tên trường, ví dụ như 蔡志杰, có thể là lỗi nhập liệu hoặc dữ liệu không đồng nhất.

Dữ liệu bất thường: Cần xem xét có nên loại bỏ hoặc thay thế các giá trị NaN bằng "Unknown". Đồng thời, kiểm tra và làm sạch các giá trị không phải tên trường học.

3. Cột year_start (Năm bắt đầu đăng ký)

- Phần lớn người đăng ký vào năm 2020 (2,859,905 trường hợp), còn năm 2019 có 470,389 trường hợp.
- Không có giá trị NaN hoặc dữ liệu bất thường trong cột này.

Dữ liệu hợp lý: Không cần xử lý thêm, nhưng có thể phân tích sâu hơn để xem xu hướng đăng ký giữa các năm.

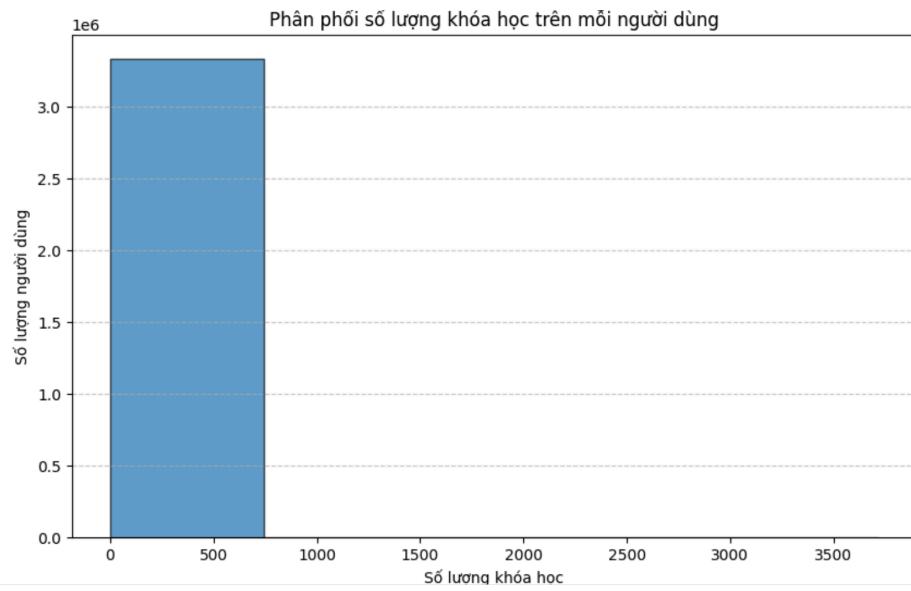
5.2.2.1.5. Phân tích phân phối dữ liệu bằng cách sử dụng biểu đồ

Biểu đồ **Histogram** thường được sử dụng để thể hiện sự phân bố của các giá trị số trong dữ liệu. Lý do sử dụng Histogram cho các trường trên là vì:

- Histogram giúp trực quan hóa **tần suất xuất hiện** của các giá trị trong một khoảng cụ thể, giúp dễ dàng quan sát sự phân bố tổng thể của dữ liệu.
- Các cột như **number_of_courses**, **enrollment_duration_days** và **intervals_course** đều là dữ liệu dạng số (numeric) có phạm vi rộng, phù hợp để biểu diễn bằng histogram thay vì các loại biểu đồ khác.
- Histogram giúp nhận diện các khoảng giá trị có nhiều người dùng hoặc khoảng giá trị hiếm khi xuất hiện.

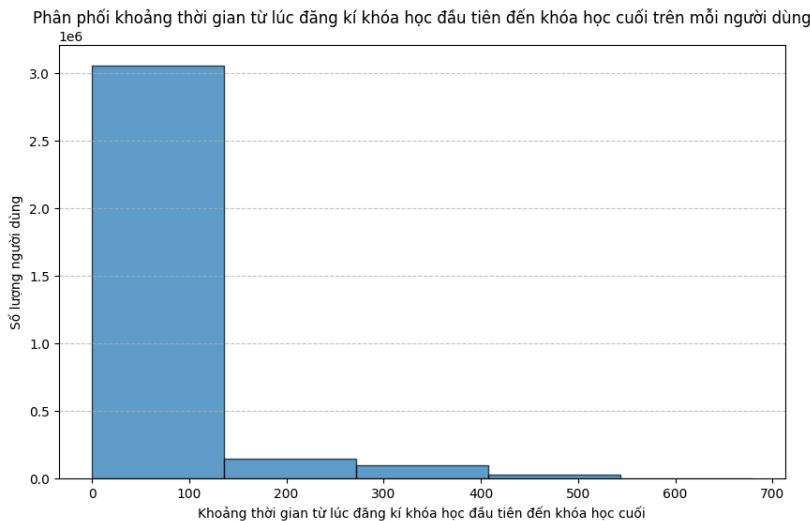
```
# Vẽ histogram
plt.figure(figsize=(10, 6))
plt.hist(df["number_of_courses"], bins=5, edgecolor="black", alpha=0.7)

# Thiết lập tiêu đề và nhãn trục
plt.xlabel("Số lượng khóa học")
plt.ylabel("Số lượng người dùng")
plt.title("Phân phối số lượng khóa học trên mỗi người dùng")
plt.grid(axis="y", linestyle="--", alpha=0.7)
```



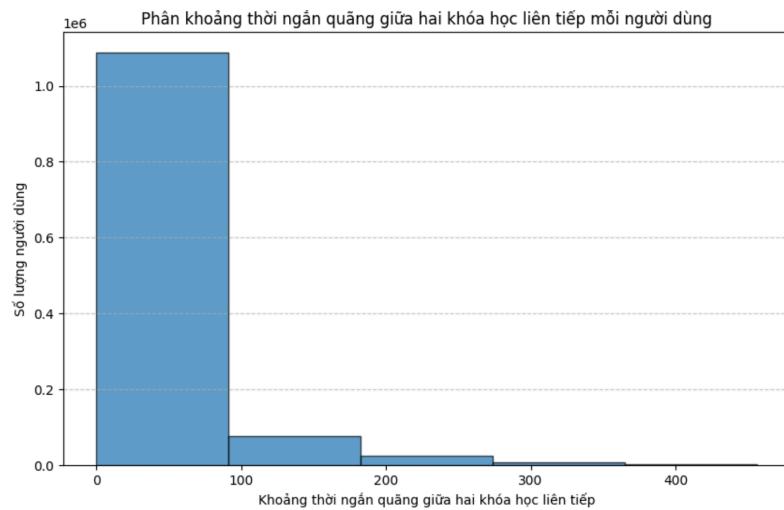
Nhận xét:

- Số lượng khóa học của người học tập trung chủ yếu ở khoảng **dưới 500 khóa học**.
- Không có trường hợp nào đăng ký trên **1000 khóa học**, cho thấy phần lớn học sinh chỉ đăng ký một vài khóa học.
- Dữ liệu có xu hướng giảm dần khi số lượng khóa học tăng, cho thấy phần lớn người dùng chỉ tham gia số ít khóa học.



Nhận xét:

- Đa số người dùng có khoảng thời gian học tập trong khoảng **0-100 ngày**.
- Rất ít người dùng có khoảng thời gian học tập từ **200 đến 500 ngày**, cho thấy phần lớn học viên hoàn thành các khóa học trong thời gian ngắn.
- Điều này có thể phản ánh thói quen học tập ngắn hạn hoặc do nội dung khóa học không đủ hấp dẫn để giữ chân học viên trong thời gian dài.



Nhận xét:

- Thời gian ngắn quãng giữa hai khóa học thường nằm trong khoảng **0-100 ngày**, cho thấy người học có xu hướng tham gia liên tục.
- Số lượng người dùng quay lại học sau **100 ngày** là rất ít, cho thấy nền tảng cần có các chiến lược giữ chân người học.
- Kết quả này có thể giúp nền tảng học tập điều chỉnh các chương trình ưu đãi hoặc gửi thông báo nhắc nhở để khuyến khích người dùng quay lại học

5.2.2.1.6. Xác định các giá trị ngoại lai (outlier)

Trước khi sử dụng nhóm loại bỏ một số hàng có dữ liệu là NaN và chỉ lấy những trường nào có giá trị là số.

Sử dụng độ lệnh chuẩn

Những giá trị nằm ngoài khoảng trên được coi là **ngoại lai**.

```
# Tính mean, std, lower_bound, upper_bound bằng numpy để tối ưu hóa
mean_vals = df[numerical_cols].mean()
std_vals = df[numerical_cols].std()
lower_bounds = mean_vals - 3 * std_vals
upper_bounds = mean_vals + 3 * std_vals

# Xác định ngoại lai
outliers = (df[numerical_cols] < lower_bounds) | (df[numerical_cols] > upper_bounds)

# Thống kê số lượng ngoại lai trong từng cột
outlier_counts = outliers.sum()
print("Số lượng ngoại lai trong từng cột:")
print(outlier_counts)
```

```
Số lượng ngoại lai trong từng cột:  
gender           1  
year_of_birth   170  
number_of_courses 77699  
max_interval    13107  
min_interval    26522  
avg_interval    22009  
year_start       0  
avg_month_enroll 0  
enrollment_duration_days 96209  
dtype: int64
```

Nhận xét:

1. Cột gender (Giới tính): Có 1 giá trị ngoại lai, đó là 232.0, đây có thể là lỗi nhập dữ liệu hoặc lỗi mã hóa giới tính. Các giá trị hợp lệ nên chỉ bao gồm 0, 1, 2.
2. Cột year_of_birth (Năm sinh): Tuy có số lượng ít nhưng số lượng ngoại lai cũng nhiều.
3. Cột number_of_courses (Số lượng khóa học): Có 77,699 giá trị ngoại lai, điều này cho thấy có những người học đăng ký số lượng khóa học rất lớn (có thể lên đến hàng ngàn khóa học). Việc một người đăng ký quá nhiều khóa học (hàng trăm hoặc hàng ngàn) có thể là dấu hiệu của tài khoản spam hoặc lỗi nhập liệu.
4. Cột max_interval, min_interval, avg_interval (Khoảng thời gian giữa các khóa học): Có số lượng ngoại lai lớn khi tìm bằng phương pháp độ lệch chuẩn. Tùy vào trình độ, quá trình của mỗi người thì sẽ có thời gian đăng ký học tiếp theo là khác nhau nên không phải ngoại lai.
5. Cột year_start (Năm bắt đầu học): Không có ngoại lai. Điều này hợp lý vì dữ liệu chỉ có các năm hợp lệ (2019 và 2020), không có sai số lớn.
6. Cột avg_month_enroll (Trung bình tháng đăng ký): Không có ngoại lai. Điều này chứng tỏ tháng đăng ký khóa học phân bố đồng đều, không có giá trị bất thường.
7. Cột enrollment_duration_days: Có 96,209 giá trị ngoại lai, nghĩa là

Kết luận về các cột không phải ngoại lai

Các cột không có giá trị ngoại lai gồm: year_of_birth (Năm sinh), max_interval (Khoảng thời gian lớn nhất giữa hai khóa học), min_interval (Khoảng thời gian nhỏ nhất giữa hai khóa học), avg_interval (Khoảng thời gian trung bình giữa hai khóa học), year_start (Năm bắt đầu học), avg_month_enroll (Trung bình tháng đăng ký)

Hướng xử lý cho các cột có ngoại lai

- gender: Loại bỏ hoặc sửa giá trị 232.0.
- number_of_courses: Kiểm tra xem các tài khoản đăng ký hàng trăm hoặc hàng ngàn khóa học có phải là tài khoản spam hay không.

Sử dụng z-score

z-score: Xác định ngoại lai khi $|z\text{-score}| > 3$.

```
from scipy.stats import zscore

# Tính Z-score cho từng cột số
z_scores = df[numerical_cols].apply(zscore)

# Định nghĩa ngưỡng ngoại lai (thường là  $|Z| > 3$ )
threshold = 3
outliers = (np.abs(z_scores) > threshold)

# Thống kê số lượng ngoại lai trong từng cột
outlier_counts = outliers.sum()
print("Số lượng ngoại lai trong từng cột:")
print(outlier_counts)
```

Số lượng ngoại lai trong từng cột:

```
gender                  1
year_of_birth            0
number_of_courses        77699
max_interval              0
min_interval              0
avg_interval              0
year_start                0
avg_month_enroll          0
enrollment_duration_days 96209
dtype: int64
```

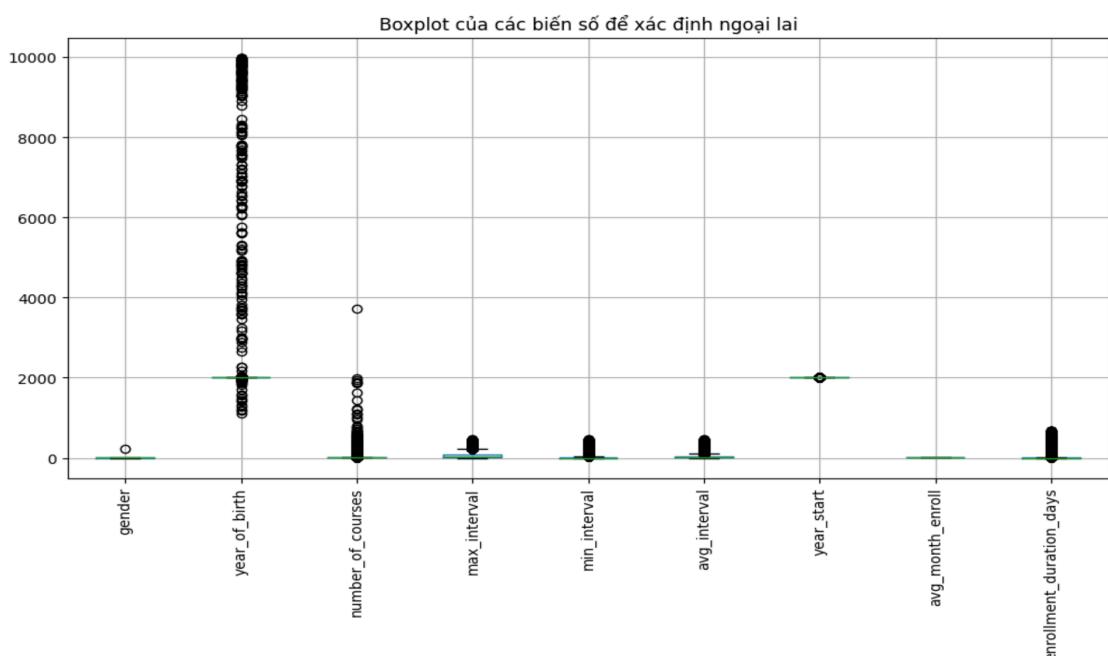
Nhận xét:

- Cột gender có 1 giá trị ngoại lai, có thể là do giá trị không hợp lệ (ví dụ: giá trị 232.0 trong dữ liệu ban đầu).
- Cột year_of_birth không có ngoại lai, cho thấy năm sinh của người dùng khá ổn định, không có giá trị bất thường quá xa trung bình.
- Cột number_of_courses có 77,699 ngoại lai, cho thấy có nhiều người đăng ký số lượng khóa học vượt mức bình thường.
- Cột max_interval, min_interval, avg_interval, year_start, avg_month_enroll không có ngoại lai, nghĩa là các khoảng thời gian giãn cách giữa các khóa học và thời điểm đăng ký không có giá trị quá bất thường.

- Cột enrollment_duration_days có 96,209 ngoại lai, nghĩa là có nhiều người dùng có thời gian đăng ký trải dài hơn mức trung bình rất nhiều, có thể do một số người duy trì việc học lâu dài.
- Phương pháp z-score cho thấy số hàng ngoại lai thấp hơn phương pháp chỉ sử dụng độ lệch chuẩn.

Sử dụng boxplot để trực quan các giá trị ngoại lai

```
plt.figure(figsize=(12, 6))
df[numerical_cols].boxplot(rot=90) # Xoay nhãn trực x để dễ đọc
plt.title("Boxplot của các biến số để xác định ngoại lai")
plt.show()
```



Nhận xét:

- **Year_of_birth** có mức độ phân tán lớn và chứa nhiều giá trị nằm ngoài khoảng phân vị. Vì vậy, cần chuẩn hóa trước khi huấn luyện mô hình để đảm bảo tính ổn định.
- **Number_of_courses** cho thấy đa số học viên đăng ký ít khóa học, trong khi một số ít người đăng ký số lượng lớn (trên 100 khóa học), tạo ra sự chênh lệch đáng kể trong dữ liệu.

5.2.2.1.7. Tỷ lệ phần trăm NaN trong từng cột

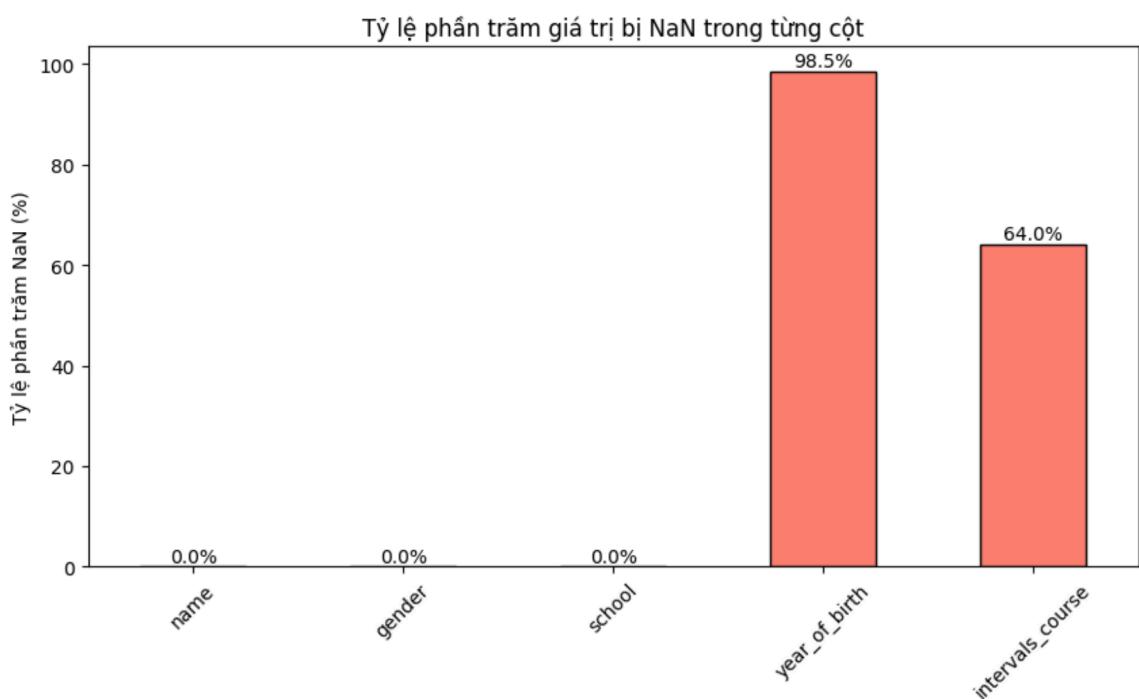
Bộ dữ liệu sử dụng trực quan hóa dữ liệu là **entities/user.json**. Một số thuộc tính được sử dụng lại từ thống kê dữ liệu.

Dữ liệu bị thiếu

```

import matplotlib.pyplot as plt
# Tính tỷ lệ phần trăm NaN của từng cột
nan_percentage = (df.isna().sum() / len(df)) * 100
# Chỉ lấy những cột có NaN
nan_percentage = nan_percentage[nan_percentage > 0]
# Vẽ biểu đồ cột
plt.figure(figsize=(10, 5))
nan_percentage.plot(kind="bar", color="salmon", edgecolor="black")
# Thêm nhãn và tiêu đề
plt.xlabel("Tên cột")
plt.ylabel("Tỷ lệ phần trăm NaN (%)")
plt.title("Tỷ lệ phần trăm giá trị bị NaN trong từng cột")
plt.xticks(rotation=45) # Xoay tên cột để dễ đọc
# Hiển thị giá trị trên mỗi cột
for index, value in enumerate(nan_percentage):
    plt.text(index, value + 1, f"{value:.1f}%", ha="center", fontsize=10)
plt.show()

```



```

import matplotlib.pyplot as plt

column_name = "year_of_birth" # Thay bằng tên cột cần kiểm tra

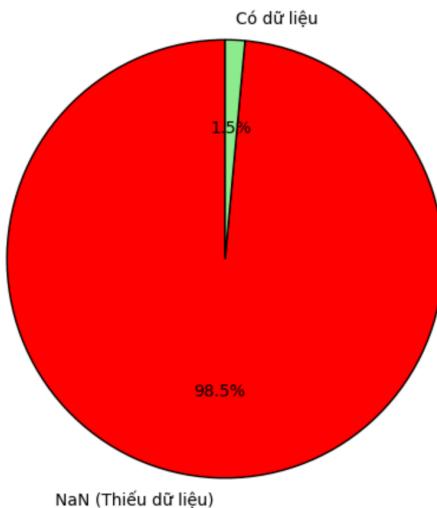
# Tính số lượng NaN và không NaN
nan_count = df[column_name].isna().sum()
non_nan_count = df[column_name].notna().sum()

# Tạo dữ liệu cho biểu đồ
labels = ["NaN (Thiếu dữ liệu)", "Có dữ liệu"]
sizes = [nan_count, non_nan_count]
colors = ["red", "lightgreen"]

# Vẽ biểu đồ tròn
plt.figure(figsize=(6, 6))
plt.pie(sizes, labels=labels, autopct="%1.1f%%", colors=colors, startangle=90, wedgeprops={"edgecolor": "black", "width": 1})
plt.title(f"Tỷ lệ dữ liệu NaN trong cột '{column_name}'")
plt.show()

```

Tỷ lệ dữ liệu NaN trong cột 'year_of_birth'



Nhận xét:

Cột intervals_course (khoảng cách giữa các lần đăng ký khóa học):

- Có 64% học sinh chỉ đăng ký một khóa học.
- Điều này khiến nhiều hàng trong intervals_course có giá trị NaN, vì không thể tính khoảng cách giữa các lần đăng ký khi chỉ có một khóa học.

Cột year_of_birth (năm sinh):

- 98.5% giá trị trong cột này bị thiếu (NaN), chỉ 1.5% hàng có giá trị hợp lệ.
- Ngoài ra, khi mô tả dữ liệu (describe()), có một số giá trị lớn hơn 2021 - năm khảo sát, cho thấy có nhiều trong dữ liệu.
- Vì dữ liệu trong cột này gần như không sử dụng được, nên nên loại bỏ year_of_birth để tránh ảnh hưởng đến hiệu suất của mô hình.

Các cột còn lại: Số lượng giá trị bị thiếu là không đáng kể và không ảnh hưởng lớn đến quá trình xử lý dữ liệu.

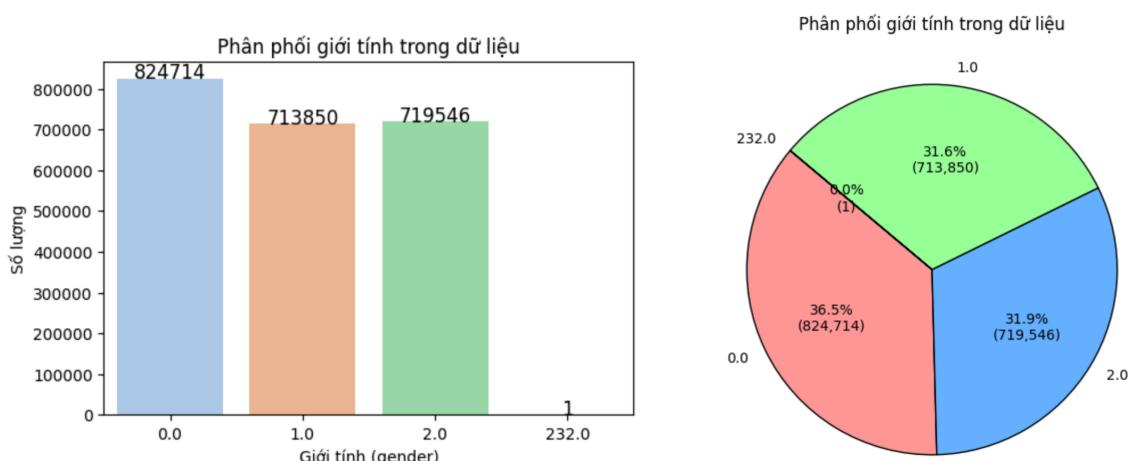
5.2.2.1.8. Phân phối giới tính trong thời gian khảo sát

```
# Đếm số lượng từng loại gender
gender_counts = df["gender"].value_counts().sort_index()

# Vẽ biểu đồ cột
plt.figure(figsize=(6, 4))
sns.barplot(x=gender_counts.index, y=gender_counts.values, palette="pastel")

# Thêm tiêu đề và nhãn trục
plt.xlabel("Giới tính (gender)")
plt.ylabel("Số lượng")
plt.title("Phân phối giới tính trong dữ liệu")
plt.pie(
    gender_counts,
    labels=gender_counts.index,
    autopct=lambda p: '{:.1f}\n({:,.0f})'.format(p, p * sum(gender_counts) / 100),
    colors=["#ff9999", "#66b3ff", "#99ff99"],
    startangle=140,
    wedgeprops={'edgecolor': 'black'}
)
# Hiển thị số lượng trên cột
for i, v in enumerate(gender_counts.values):
    plt.text(i, v + 1, str(v), ha="center", fontsize=12)

plt.show()
```



Nhận xét:

- Giới tính 0 chiếm tỷ lệ cao nhất trong dữ liệu với 36.5%, cao hơn so với các giới tính khác. Tuy nhiên, sự chênh lệch giữa các nhóm không quá lớn (36.5% cho giới tính 0, 31.6% cho giới tính 1, và 31.9% cho giới tính 2). Do đó, dữ liệu có sự phân bố khá đồng đều, giúp giảm nguy cơ mất cân bằng và hạn chế hiện tượng bias trong quá trình huấn luyện mô hình.

5.2.2.1.9. Phần trăm số lượng học sinh tham gia trong các khoảng thời gian

Phần trăm số lượng học sinh tham gia theo từng năm

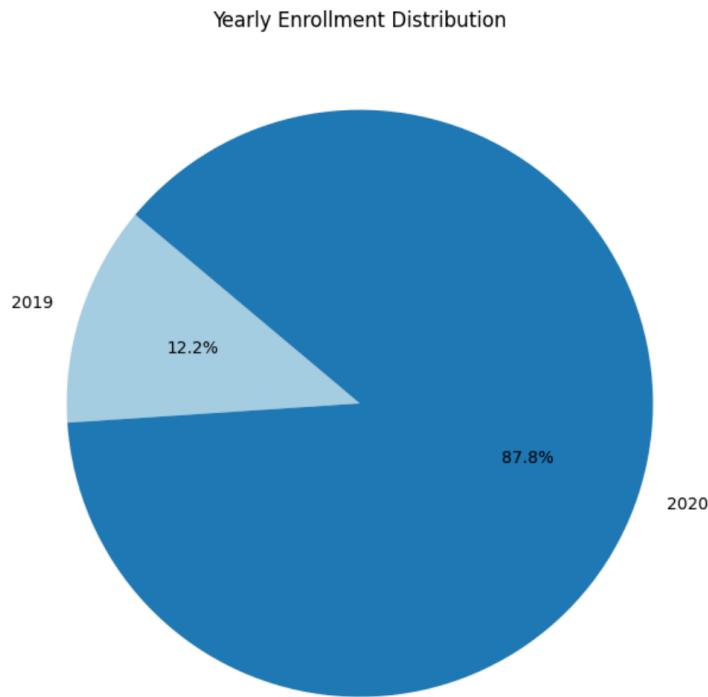
```

import matplotlib.pyplot as plt

# Assuming 'df' has a column 'year_enroll' containing lists of years
yearly_enroll_counts = df['year_enroll'].explode().value_counts().sort_index()

# Plot pie chart
plt.figure(figsize=(8, 8))
plt.pie(yearly_enroll_counts, labels=yearly_enroll_counts.index, autopct='%1.1f%%', startangle=140)
plt.title('Yearly Enrollment Distribution')
plt.show()

```



Nhận xét

- Năm **2019**, chỉ có **12,2%** học sinh tham gia khóa học.
- Năm **2020**, tỷ lệ học sinh tham gia tăng lên **87,8%**, tức là **lớn gấp khoảng 7,2 lần so với năm 2019**.
- Điều này cho thấy sự gia tăng mạnh mẽ về số lượng học sinh đăng ký khóa học trong năm 2020.
- Cũng cho thấy mất cân bằng khá lớn về dữ liệu, khó huấn luyện mô hình theo thời gian vì dữ liệu không có sự đa dạng cho các năm.

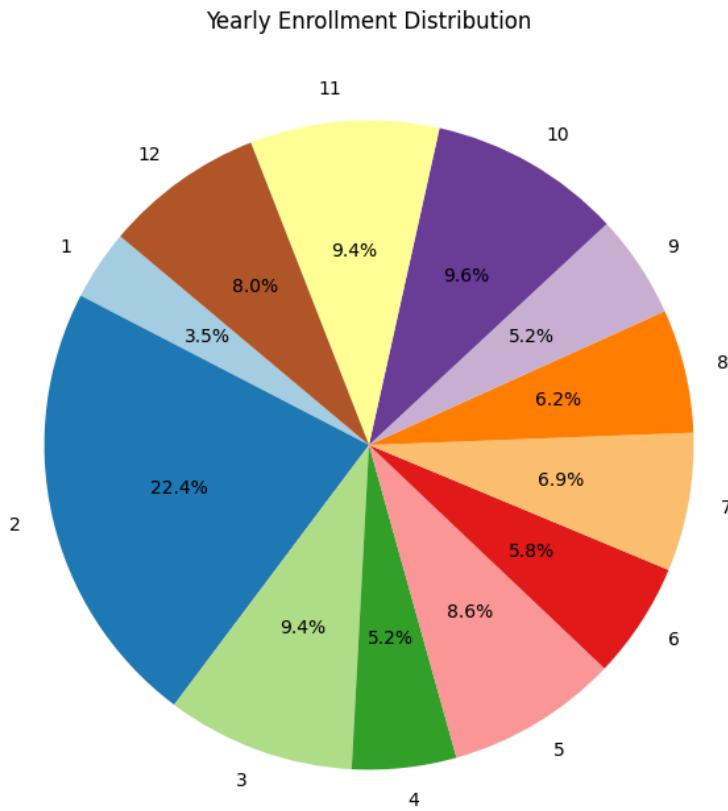
Phản trǎm số lượng học sinh tham gia theo từng tháng

```

# Assuming 'df' has a column 'month_enroll' containing lists of months
month_enroll_counts = df['month_enroll'].explode().value_counts().sort_index()

# Plot pie chart
plt.figure(figsize=(8, 8))
plt.pie(month_enroll_counts, labels=month_enroll_counts.index, autopct='%1.1f%%', startangle=140)
plt.title('Yearly Enrollment Distribution')
plt.show()

```



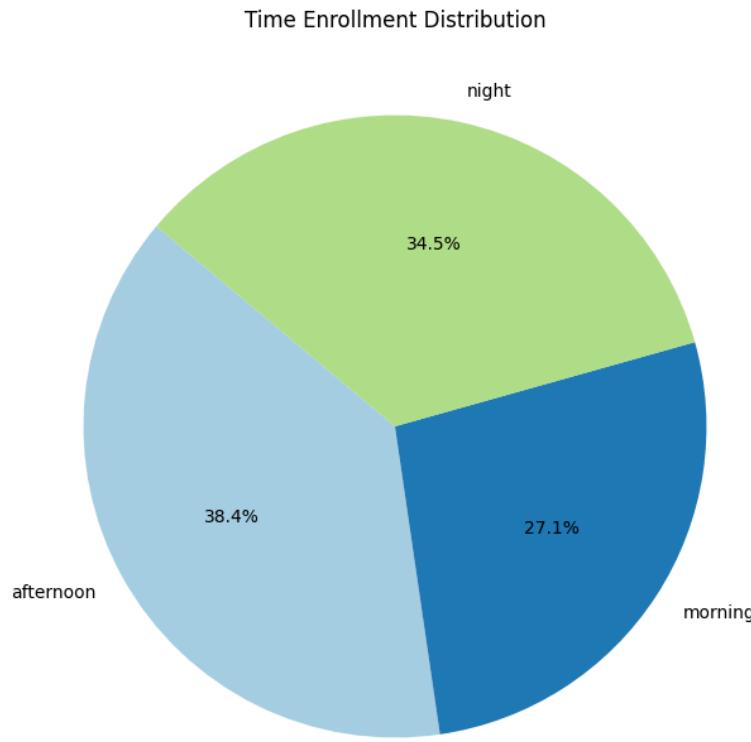
Nhận xét

- **Tháng 2, 3 và 11, 10** là những tháng có nhiều học sinh đăng ký nhất, với tỷ lệ lần lượt là **22,4%, 9,4%,** và **9,4%, 9,6%**. Điều này cho thấy có thể có các chương trình ưu đãi hoặc nhu cầu học tăng cao trong các tháng này.
- **Tháng 1** có số lượng học sinh đăng ký mới thấp nhất, chỉ **3,2%**, có thể do thời điểm này rơi vào dịp nghỉ lễ hoặc chưa phải là giai đoạn cao điểm để bắt đầu khóa học.
- Các tháng còn lại có tỷ lệ đăng ký dao động trong khoảng **5-8%**, phản ánh mức độ ổn định của số lượng học viên đăng ký trong phần lớn thời gian trong năm.
- Xu hướng đăng ký cao vào tháng 2 và 3 có thể liên quan đến nhu cầu học tập đầu năm, trong khi tháng 10, 11 có thể là thời điểm học sinh chuẩn bị cho các kỳ thi hoặc các chương trình đào tạo cuối năm.

Phản trǎm số lượng học sinh đăng kí theo buổi thời gian trong ngày

```
# Assuming 'df' has a column 'year_enroll' containing lists of years
time_of_day_counts = df['time_of_day'].explode().value_counts().sort_index()

# Plot pie chart
plt.figure(figsize=(8, 8))
plt.pie(time_of_day_counts, labels=time_of_day_counts.index, autopct='%.1f%%', startangle=140, counterclockwise=False)
plt.title('Time Enrollment Distribution')
plt.show()
```



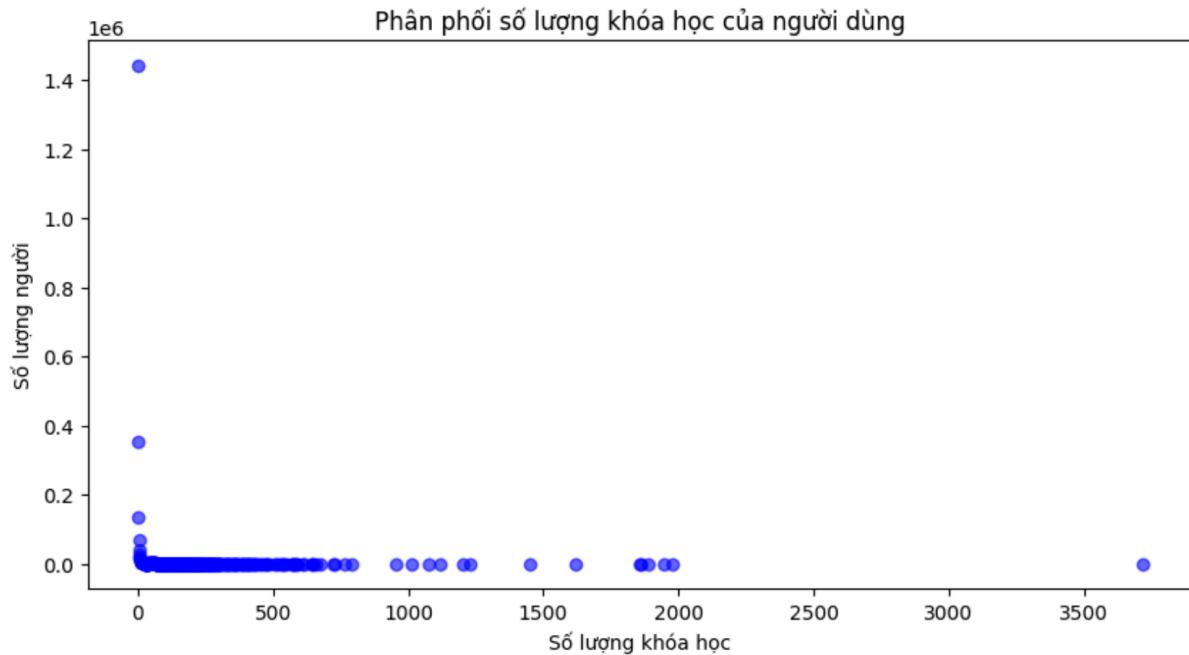
Nhận xét:

- **Buổi chiều (38.4%)** và **buổi tối (34.5%)** có tỷ lệ học sinh đăng ký khóa học gần như ngang nhau, cho thấy đây là khoảng thời gian phổ biến để đăng ký. Điều này có thể do học sinh và người đi làm có thời gian rảnh sau giờ học hoặc giờ làm việc.
- **Buổi sáng (27.1%)** có tỷ lệ đăng ký thấp hơn đáng kể so với buổi chiều và buổi tối. Nguyên nhân có thể do nhiều học sinh hoặc người đi làm bận rộn vào buổi sáng, ít có thời gian để thực hiện việc đăng ký khóa học.
- Xu hướng này cho thấy thời gian đăng ký chủ yếu tập trung vào các khung giờ ngoài giờ hành chính, có thể giúp các nền tảng giáo dục điều chỉnh chiến lược quảng bá và hỗ trợ tư vấn phù hợp với thời điểm có nhiều người đăng ký nhất.

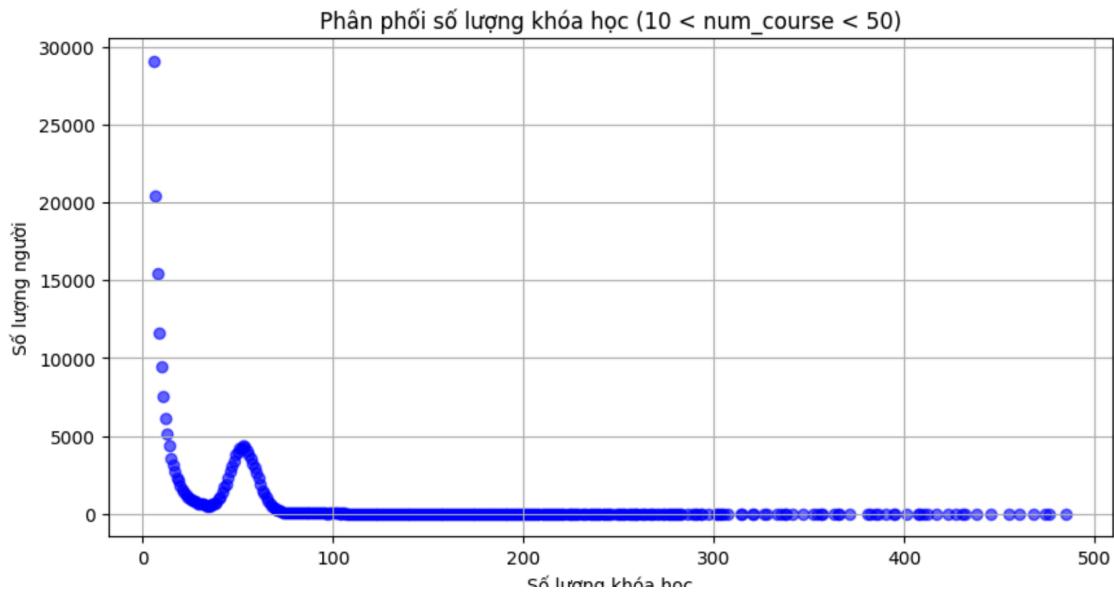
5.2.2.1.10. Phân phối số lượng khóa học của người dùng

```
# Vẽ biểu đồ cột
plt.figure(figsize=(10, 5))
# Vẽ scatter plot
plt.scatter(course_counts[ "number_of_courses" ], course_counts[ "count" ], color="b", alpha=0.6)

# Thêm tiêu đề và nhãn
plt.xlabel("Số lượng khóa học")
plt.ylabel("Số lượng người")
plt.title("Phân phối số lượng khóa học của người dùng")
plt.xticks(rotation=0) # Giữ nhãn trục X thẳng đứng
plt.show()
```



Biểu đồ scatter thể hiện phân phối số lượng khóa học của người dùng



Biểu đồ scatter thể hiện phân phối số lượng khóa học của người dùng có đăng ký từ 10 đến 50 khóa

Nhật xét:

1. Xu hướng chung:

- Nếu đường biểu đồ có xu hướng giảm dần, điều đó cho thấy số lượng người học nhiều khóa học ít đi, nghĩa là hầu hết người dùng chỉ đăng ký một số ít khóa học.
- Nếu đường biểu đồ có nhiều dao động, có thể có một số lượng khóa học nhất định thu hút nhiều người đăng ký hơn hẳn so với các khóa học khác.

2. Sự phân bố số lượng khóa học đã đăng ký:

- Nếu ở đầu biểu đồ có giá trị cao và sau đó giảm dần, điều này gợi ý rằng đa số người học chỉ đăng ký ít khóa học.
- Nếu biểu đồ xuất hiện nhiều đỉnh cao tại một số điểm cụ thể (ví dụ: 5, 10, 20 khóa học), có thể một số lượng khóa học nhất định có sức hút mạnh mẽ, khiến nhiều người đăng ký hơn.

3. Ngưỡng 500 khóa học:

- Biểu đồ giúp tập trung phân tích vào nhóm người học có số lượng đăng ký hợp lý.
- Nếu đường biểu đồ giảm mạnh khi số khóa học tăng, có thể kết luận rằng đa số người học đăng ký dưới 500 khóa học, và số người học đăng ký nhiều khóa học giảm dần theo số lượng khóa học.

5.2.2.1.11. Các mối quan hệ trong bộ dữ liệu

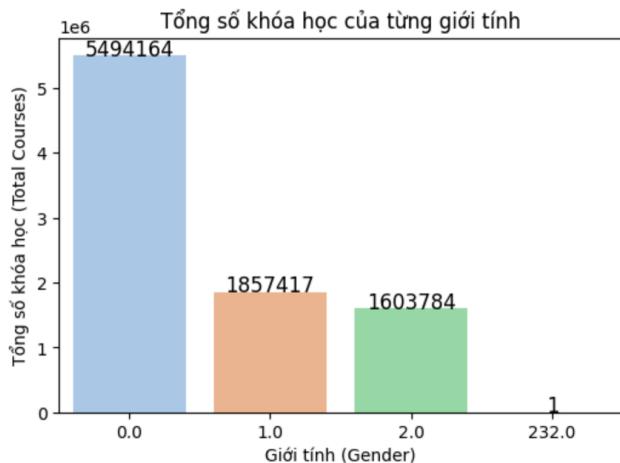
Mối quan hệ giữa gender và số lượng khóa học

```
# Vẽ biểu đồ cột
plt.figure(figsize=(6, 4))
sns.barplot(x=gender_course_sum.index, y=gender_course_sum.values, palette="pastel")

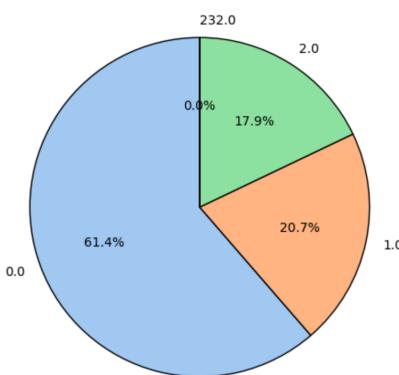
# Thêm tiêu đề và nhãn trục
plt.xlabel("Giới tính (Gender)")
plt.ylabel("Tổng số khóa học (Total Courses)")
plt.title("Tổng số khóa học của từng giới tính")

# Hiển thị số lượng trên cột
for i, v in enumerate(gender_course_sum.values):
    plt.text(i, v + 1, str(v), ha="center", fontsize=12)

plt.show()
```



Tỷ lệ phần trăm tổng số khóa học của từng giới tính



Phần trăm tổng số lượng khóa học được đăng kí bởi ba giới tính

Nhận xét:

- Tuy phân trong phối dữ liệu phần trăm giới tính 0, 1, 2 không có sự chênh lệch nhiều (36,5% cho 0, 31,6% cho 1 và 31,9% cho 2) nhưng tổng khóa học được đăng kí bởi những người có giới tính 0 là nhiều nhất. Điều này có thể do outliers chưa được loại bỏ
- Phần trăm số lượng khóa học của giới tính 2 là 17,9% và 1 là 20,7%.

Mối quan hệ giữa gender với các khoảng thời gian

a) Mối quan hệ giữa gender với khoảng thời gian ngày đầu tiên đăng ký và ngày cuối cùng đăng ký

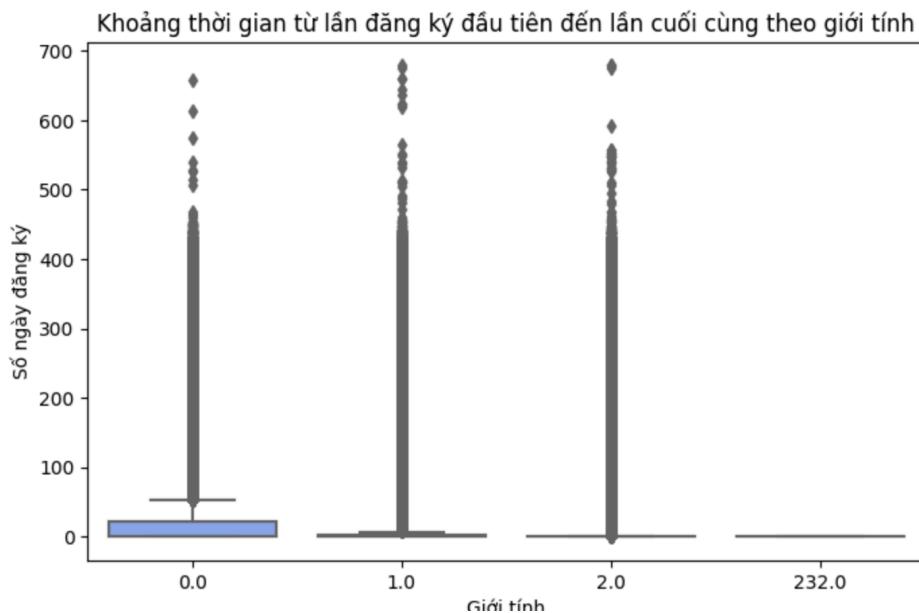
Vì đây là một khoảng thời gian liên tục (enrollment_duration_days), có thể dùng **boxplot** để xem phân bố hoặc **scatter plot** để thấy xu hướng.

```

plt.figure(figsize=(8, 5))
sns.boxplot(data=df, x="gender", y="enrollment_duration_days", palette="coolwarm")

plt.title("Khoảng thời gian từ lần đăng ký đầu tiên đến lần cuối cùng theo giới tính")
plt.xlabel("Giới tính")
plt.ylabel("Số ngày đăng ký")
plt.show()

```



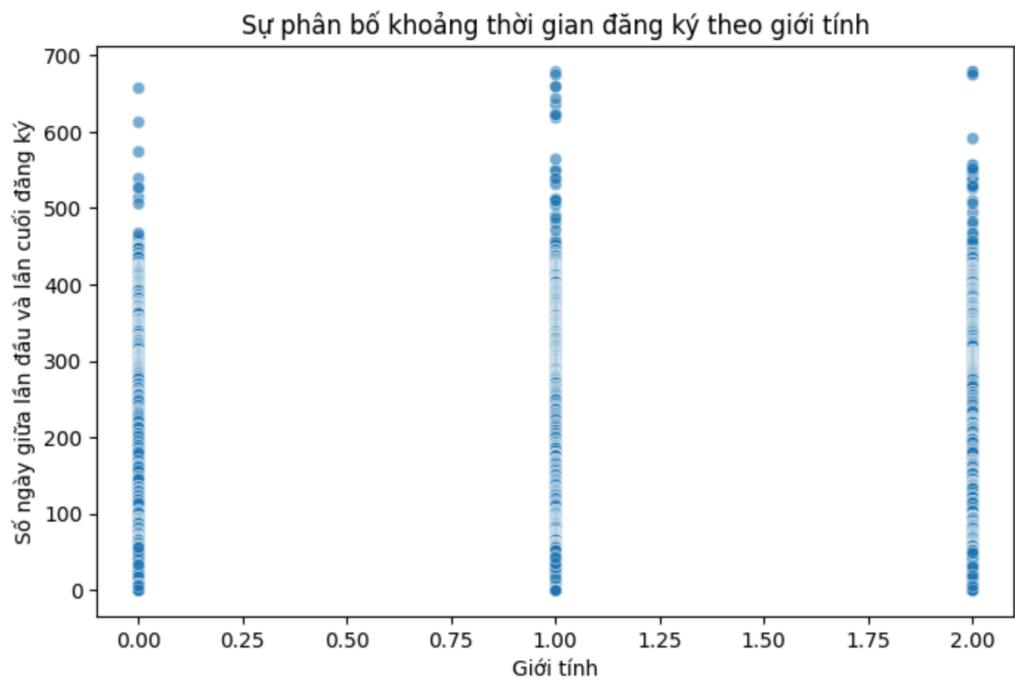
Biểu đồ boxplot thể hiện khoảng thời gian từ lần đăng ký đầu tiên và gần nhất theo giới tính

```

plt.figure(figsize=(8, 5))
sns.scatterplot(data=df, x="gender", y="enrollment_duration_days", alpha=0.6)

plt.title("Sự phân bố khoảng thời gian đăng ký theo giới tính")
plt.xlabel("Giới tính")
plt.ylabel("Số ngày giữa lần đầu và lần cuối đăng ký")
plt.show()

```



Biểu đồ scatter biểu diễn sự phân bố khoảng thời gian đăng ký đầu tiên và gần nhất

Nhận xét:

- Biểu đồ scatter cho thấy rằng một số người thuộc giới tính 1 có thời gian chờ lâu hơn trước khi đăng ký khóa học đầu tiên so với những người thuộc giới tính 0 và 2.
- Ngoài ra, nhóm giới tính 0 có xu hướng đăng ký khóa học đầu tiên chủ yếu trong khoảng 0 đến 490 ngày trước ngày hiện tại, cho thấy họ thường tham gia khóa học sớm hơn so với các nhóm giới tính khác.

b) Mối quan hệ giữa gender với các thời điểm trong ngày

Sử dụng **heatmap** (`plt.imshow()`) để thể hiện tần suất đăng ký theo từng khung giờ và **bar stack** thể hiện sự phân bố của từng giới tính đăng ký theo thời gian trong ngày.

```

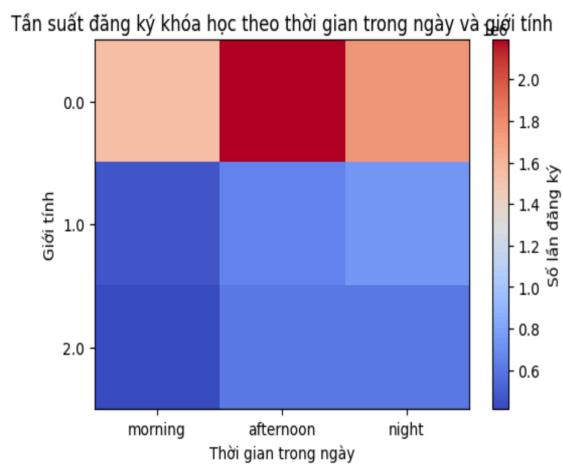
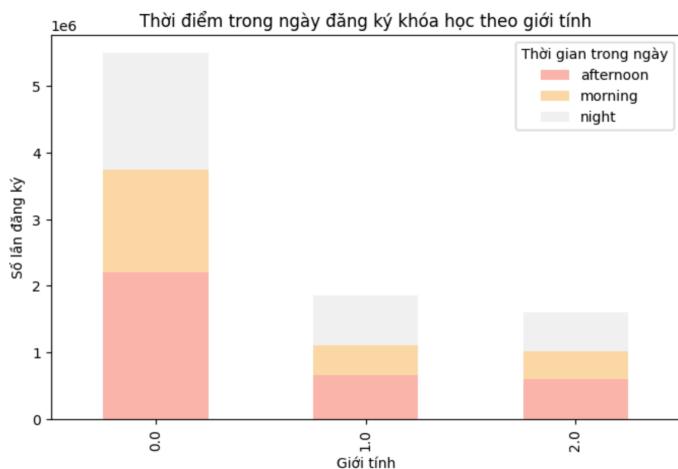
# Định nghĩa thứ tự của các khung giờ trong ngày
time_of_day_order = ["morning", "afternoon", "night"]

# Đếm số lượng đăng ký theo gender và time_of_day
heatmap_data = df_exploded.groupby(["gender", "time_of_day"]).size().unstack().reindex(columns=time_of_day_order)

# Chuyển đổi dữ liệu về dạng numpy để vẽ imshow
plt.figure(figsize=(6, 4))
plt.imshow(heatmap_data, cmap="coolwarm", aspect="auto")

# Gắn nhãn trực
plt.xticks(range(len(time_of_day_order)), time_of_day_order)
plt.yticks(range(len(heatmap_data.index)), heatmap_data.index)
plt.colorbar(label="Số lần đăng ký")
plt.title("Tần suất đăng ký khóa học theo thời gian trong ngày và giới tính")
plt.xlabel("Thời gian trong ngày")
plt.ylabel("Giới tính")
plt.show()

```



Biểu đồ stackbar và heatmap thể hiện tần suất đăng ký khóa học trong ngày theo giới tính

Nhận xét:

- Những người thuộc **giới tính 0** có xu hướng đăng ký khóa học chủ yếu vào **buổi chiều**.
- Những người thuộc **giới tính 1** thường đăng ký vào **buổi tối**.

- Trong khi đó, những người thuộc **giới tính 2** có thời gian đăng ký phân bố **đều giữa ban ngày và ban đêm**.

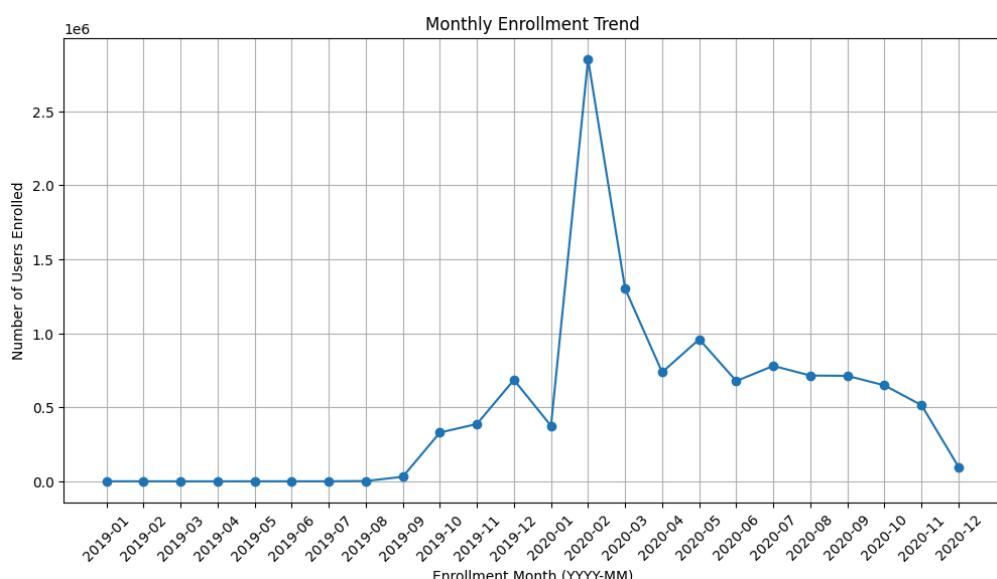
5.2.2.1.12. Phân phối số lượng khóa học được đăng ký

Số lượng khóa học được đăng kí theo từng tháng

```
df['enroll_date'] = df.apply(lambda row: f'{y}-{m:02d}' for y, m in zip(row['year_enroll'], row['month_enroll']))
# Gán dữ liệu dates và số lượng user tham gia mỗi tháng
monthly_enroll_counts = pd.Series([date for dates in df['enroll_date'] for date in dates]).value_counts()
# Vẽ chart timeline
plt.figure(figsize=(12, 6))
plt.plot(monthly_enroll_counts.index, monthly_enroll_counts.values, marker='o', linestyle='-' )
plt.xticks(rotation=45)
plt.xlabel("Enrollment Month (YYYY-MM)")
plt.ylabel("Number of Users Enrolled")
plt.title("Monthly Enrollment Trend")
plt.grid(True)
plt.show()
```

2019-01	20
2019-02	29
2019-03	9
2019-04	3
2019-05	174
2019-06	137
2019-07	94
2019-08	2123
2019-09	30772
2019-10	329482
2019-11	387493
2019-12	685544
2020-01	375011
2020-02	2852458
2020-03	1301506
2020-04	737032
2020-05	960092
2020-06	677531
2020-07	780129
2020-08	714792
2020-09	712614
2020-10	648466
2020-11	514516
2020-12	97063

Name: count, dtype: int64



Thể hiện số lượng khóa học được đăng kí theo từng tháng

Nhận xét:

- **Tăng trưởng mạnh:** Trong giai đoạn **từ tháng 8/2019 đến 2/2020**, số lượng học sinh đăng ký khóa học tăng đáng kể. Đặc biệt, **tháng 1/2020** có mức tăng đột biến từ **0.4 lên gần 2.8 triệu học viên** (1e6). Điều này có thể do nhu cầu học tập tăng cao sau kỳ nghỉ lễ hoặc do các chương trình khuyến mãi đầu năm của các nền tảng giáo dục.
- **Giảm dần sau đỉnh cao:** Từ **tháng 4/2020 đến 11/2020**, số lượng học sinh tham gia khóa học bắt đầu có xu hướng giảm, từ **1 triệu xuống còn khoảng 0.5 triệu**. Sự suy giảm này có thể liên quan đến việc học sinh đã ổn định lịch trình học tập hoặc các yếu tố bên ngoài ảnh hưởng đến nhu cầu đăng ký khóa học.
- **Giảm sâu về cuối năm:** Cuối cùng, số lượng học viên tiếp tục giảm xuống khoảng **0.2 triệu**, cho thấy sự suy giảm rõ rệt trong việc đăng ký khóa học. Điều này có thể phản ánh sự thay đổi trong xu hướng học tập, kết thúc các chương trình ưu đãi hoặc học sinh đã hoàn thành các khóa học cần thiết trước đó.

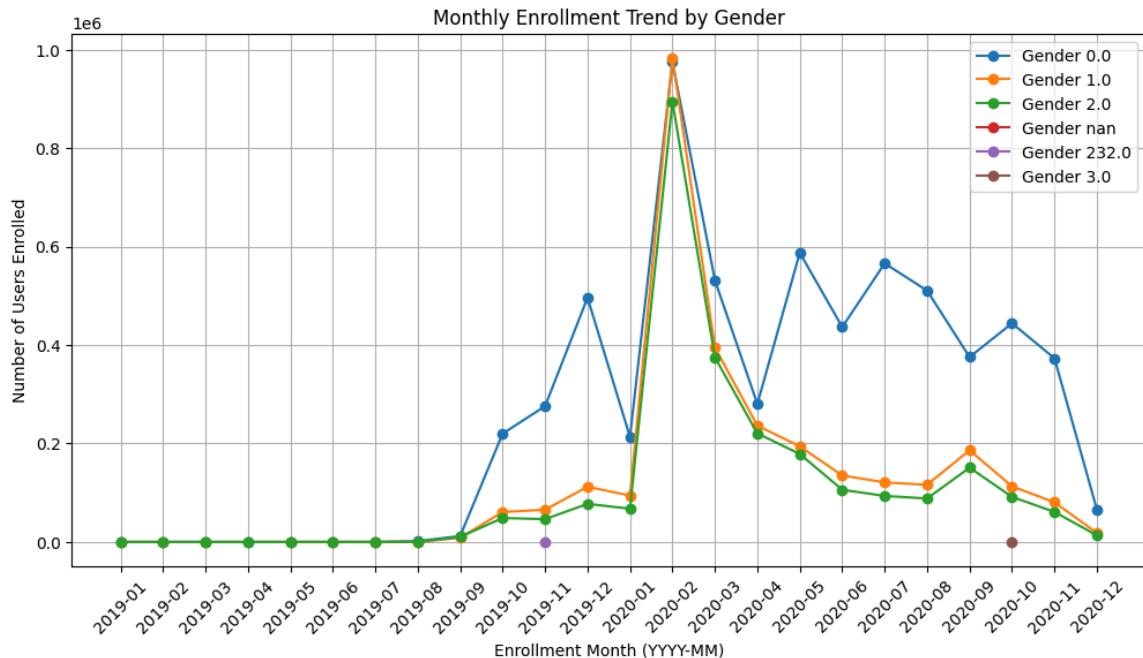
Số lượng khóa học đăng kí theo từng tháng, chia theo gender

Tương tự như số lượng khóa học được đăng kí theo từng tháng, sau đây là biểu đồ cho xu hướng số lượng đăng kí khóa học nhưng chia theo gender.

```
# Khởi tạo từ điển gender_enroll_counts để lưu số lượng lượt đăng ký theo tháng cho từng giới tính.
gender_enroll_counts = {}
# Duyệt qua từng giới tính trong cột gender
for gender in df['gender'].unique():
    # Lọc các dòng có giới tính tương ứng.
    gender_df = df[df['gender'] == gender]

    # Làm phẳng danh sách ngày đăng ký (enroll_date) vì mỗi người dùng có thể đăng ký nhiều lần.
    enroll_dates = [date for dates in gender_df['enroll_date'] for date in dates]

    # Đếm số lần xuất hiện của mỗi tháng đăng ký.
    gender_enroll_counts[gender] = pd.Series(enroll_dates).value_counts().sort_index()
# Vẽ biểu đồ xu hướng đăng ký theo thời gian
plt.figure(figsize=(12, 6))
# Sử dụng plt.plot() để vẽ số lượng đăng ký theo tháng cho từng giới tính.
for gender, counts in gender_enroll_counts.items():
    plt.plot(counts.index, counts.values, marker='o', linestyle='-', label=f"Gender {gender}")
plt.xticks(rotation=45)
plt.xlabel("Enrollment Month (YYYY-MM)")
plt.ylabel("Number of Users Enrolled")
plt.title("Monthly Enrollment Trend by Gender")
plt.legend()
plt.grid(True)
plt.show()
```



Hình biểu diễn số lượng đăng ký của từng giới tính theo từng tháng

Nhận xét:

- **Tăng mạnh vào tháng 1:** Tương tự như xu hướng tổng thể, số lượng học sinh thuộc **cả ba nhóm giới tính (gender 0, gender 1, gender 2)** đều tăng mạnh trong tháng 1. Điều này cho thấy không có sự khác biệt đáng kể giữa các giới tính trong giai đoạn này.
- **Xu hướng tương tự giữa gender 1 và gender 2:** Số lượng học sinh thuộc **gender 1 và gender 2** có xu hướng biến động tương tự nhau theo thời gian, cho thấy có thể hai nhóm này có hành vi đăng ký khóa học giống nhau.
- **Gender 0 ổn định hơn:** Học sinh thuộc nhóm **gender 0** có số lượng đăng ký **ổn định hơn**, không tăng hoặc giảm quá mạnh. Đặc biệt, trong **hai giai đoạn từ 9/2019 đến 1/2020 và từ 3/2020 đến 12/2020**, số lượng học sinh **gender 0** thường cao hơn so với hai nhóm còn lại.
- **Dữ liệu bị nhiễu:** Xuất hiện một số giá trị **bất thường trong cột giới tính**, như **gender 3 và gender 232**, có thể là dữ liệu sai hoặc bị nhập lỗi. Những giá trị này cần được xử lý hoặc loại bỏ trong quá trình làm sạch dữ liệu để đảm bảo mô hình huấn luyện chính xác.

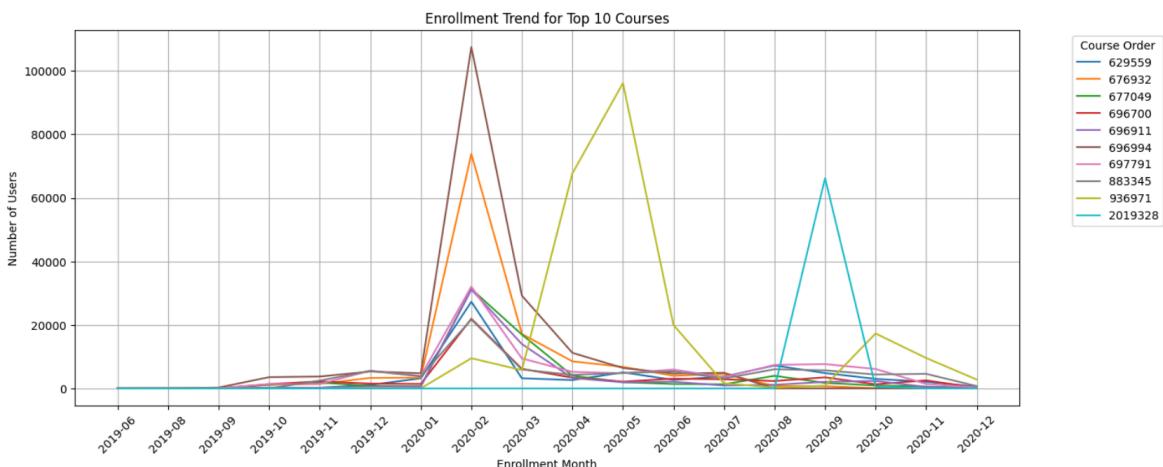
Số lượng khóa học được đăng ký của 10-20 khóa học phổ biến

- Biểu diễn lượt đăng ký của top 10 khóa học nhiều nhất được đăng ký theo tháng

```

# Lấy 10 course có nhiều người học nhất
top_courses = course_enroll_df.groupby("course_order")["num_users"].sum().nlargest(10).index
# Lọc dữ liệu
filtered_df = course_enroll_df[course_enroll_df["course_order"].isin(top_courses)]
# Chuyển thành dạng pivot
pivot_df = filtered_df.pivot(index="enroll_date", columns="course_order", values="num_users").fillna(0)
# Vẽ hình
plt.figure(figsize=(15, 6))
sns.lineplot(data=pivot_df, dashes=False)
plt.xticks(rotation=45)
plt.xlabel("Enrollment Month")
plt.ylabel("Number of Users")
plt.title("Enrollment Trend for Top 10 Courses")
plt.legend(title="Course Order", bbox_to_anchor=(1.05, 1), loc='upper left')
plt.grid(True)
plt.show()

```



Biểu diễn lượt đăng ký của top 10 khóa học nhiều nhất được đăng ký theo tháng

Nhận xét:

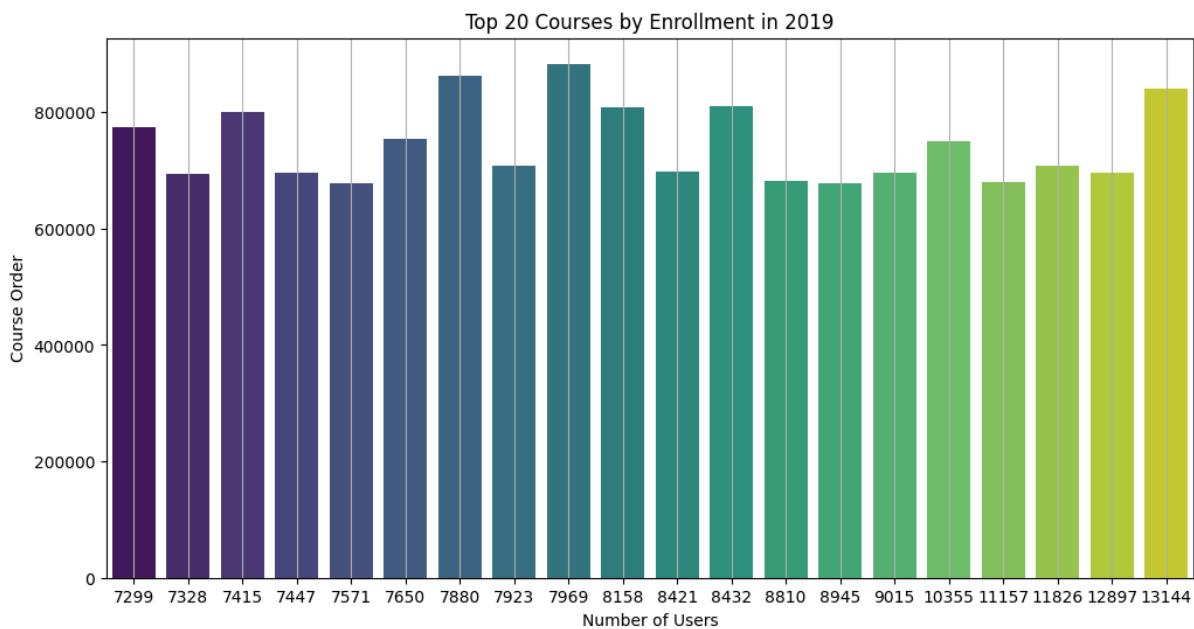
- **Xu hướng đăng ký theo thời điểm:** Các khóa học phổ biến thường chỉ được đăng ký nhiều trong một **giai đoạn nhất định**, sau đó dần giảm và hiếm khi tăng trở lại. Điều này cho thấy rằng **sự quan tâm đến khóa học có tính thời điểm**, có thể do chương trình đào tạo, nhu cầu học tập hoặc các sự kiện đặc biệt trong từng giai đoạn.
- **Ví dụ cụ thể:** Môn học mã số **696911** có số lượng đăng ký cao nhất trong khoảng **tháng 1/2020 đến tháng 3/2020**, sau đó số lượng đăng ký **giảm dần và không có dấu hiệu tăng trở lại**. Điều này có thể phản ánh rằng khóa học chỉ thu hút sự quan tâm vào một khoảng thời gian nhất định, sau đó học viên chuyển sang các khóa học khác phù hợp hơn với nhu cầu.
- **Tác động đến chiến lược giảng dạy:** Xu hướng này có thể hữu ích cho các tổ chức giáo dục trong việc **định kỳ mở lại các khóa học phổ biến** hoặc **điều chỉnh nội dung khóa học** để thu hút học viên trong các giai đoạn tiếp theo.

- b) Biểu hiện số lượng người đăng ký của top 20 khóa năm 2019 và năm 2020

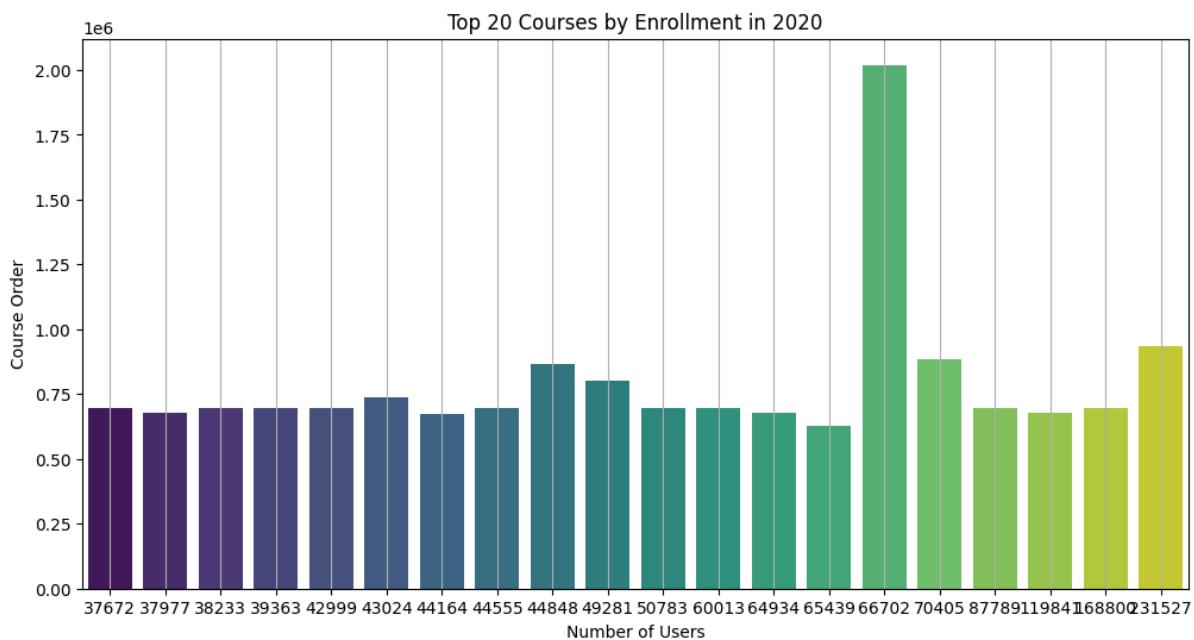
```

# Lọc dữ liệu course trong năm 2019
df_course_2019 = course_enroll_df[course_enroll_df["enroll_date"].str.startswith("2019")]
# Thống kê số lượng người học của từng course
course_counts_2019 = df_course_2019.groupby("course_order")["num_users"].sum().sort_values(ascending=True)
# Vẽ chart
plt.figure(figsize=(12, 6))
sns.barplot(x=course_counts_2019.values, y=course_counts_2019.index, palette="viridis")
plt.xlabel("Number of Users")
plt.ylabel("Course Order")
plt.title("Top 20 Courses by Enrollment in 2019")
plt.grid(axis="x")
plt.show()

```



Biểu hiện số lượng người đăng ký của top 20 khóa năm 2019



Hình thể hiện số lượng người đăng ký của top 20 khóa năm 2020

Nhận xét:

- **Khóa học 66702 có số lượng học sinh tham gia nhiều nhất trong năm 2020.**
Điều này cho thấy khóa học này có thể phù hợp với xu hướng học tập trong năm, đáp ứng tốt nhu cầu của học viên.
- **Sự thay đổi xu hướng khóa học theo năm:** Các khóa học phổ biến vào năm 2019 hiếm khi giữ được độ phổ biến vào năm 2020. Điều này có thể phản ánh sự thay đổi về nhu cầu học tập, chương trình đào tạo, hoặc xu hướng mới xuất hiện trong năm 2020.
- **Nguyên nhân có thể xảy ra:**

Chương trình học có sự **cập nhật**, khiến các môn học cũ ít được quan tâm hơn.

- Sự thay đổi trong sở thích của học viên, có thể do **xu hướng nghề nghiệp, các chứng chỉ mới, hoặc công nghệ thay đổi**.
- Một số khóa học chỉ phù hợp với **một nhóm đối tượng cụ thể** và không còn thu hút học viên mới vào năm sau.

a) Giới hạn một số user ngoại lai

Đếm số lượt đặt hàng cho mỗi khóa học.

```
frequency = course_count.values
counts = pd.Series(frequency).value_counts()
counts

1      907
2      124
10     45
3      40
4      35
...
4307    1
4314    1
4325    1
4337    1
2241    1
Name: count, Length: 2190, dtype: int64
```

```
counts.describe()
```

```
count    2190.000000
mean      2.146575
std       19.671351
min      1.000000
25%      1.000000
50%      1.000000
75%      1.000000
max      907.000000
Name: count, dtype: float64
```

Phân tích và trực quan hóa phân phối của số lượt đặt hàng (sử dụng thống kê mô tả, histogram, boxplot) để hiểu xem có bao nhiêu khóa học có ít lượt đặt hàng.

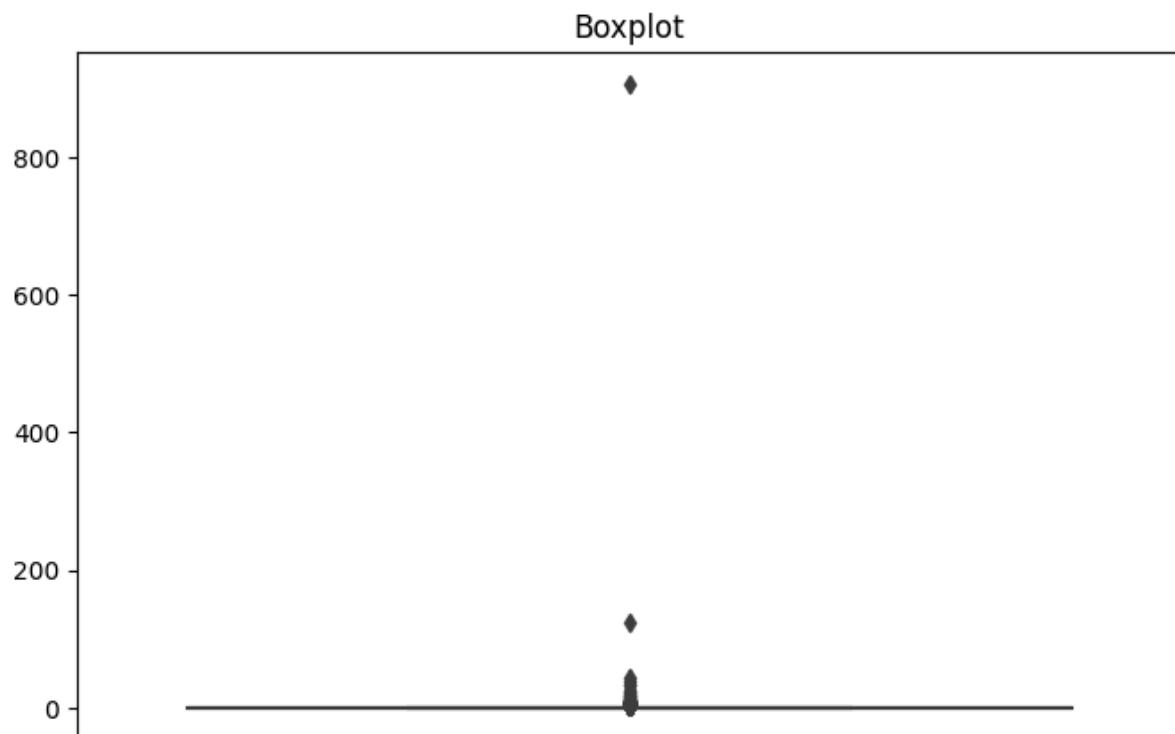
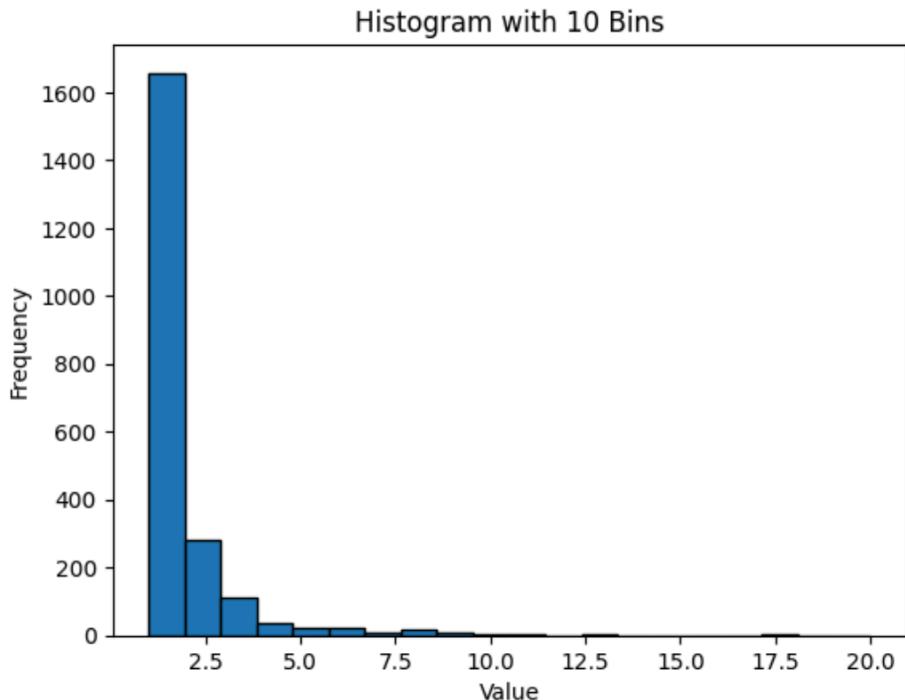
```

# Plot histogram
plt.hist(counts, bins=20, edgecolor='black', range=(1, 20)) # 10 bins, range from 1 to 500

# Labels and title
plt.xlabel("Value")
plt.ylabel("Frequency")
plt.title("Histogram with 10 Bins")

Text(0.5, 1.0, 'Histogram with 10 Bins')

```



Lọc ra những khóa học có số lượt đặt hàng từ 11 trở lên.

```

# Assuming 'data' is your NumPy array
filtered_data = course_count[course_count >= 11] # Remove values equal to 1

filtered_data

course_order
936971      231674
696994      181697
676932      125789
697791      96210
883345      78374
...
2141537      11
1926748      11
2342491      11
2199259      11
2179942      11
Name: count, Length: 3425, dtype: int64

```

5.2.2.2. Xử lý bộ dữ liệu entities/course.json

5.2.2.2.1 Thông kê mô tả về bộ dữ liệu

id	name	field	prerequisites	about	resource	num_resources	about_length
0	C_584313	《资治通鉴》导读	[历史学, 中国语言文学]	通过老师导读，同学们可深入这一经典文本内部，得以纵览千年历史，提升国学素养、体味人生智慧。	[[titles: [第一课 导论与三家分晋, 导论, 导论], 'resou...]	91	45
1	C_584329	微积分——极限理论与一元函数	[应用经济学, 数学, 物理学, 理论经济学]	本课程是理工科的一门数学基础课，系统、全面地介绍了一元函数微积分学。课程既保持了数学的严谨和...	[[titles: [序言, 序言, 序言], 'resource_id': ...]	170	70
2	C_584381	新闻摄影	[艺术学, 新闻传播学]	掌握基本的摄影技能，了解图片新闻的工作方式，训练对生活的观察和热爱，发展对图像的审美和批评...	[[titles: [第一章 绪论, 第一讲 引言1, 引言1], 'res...]	127	61
3	C_597208	数据挖掘：理论与算法	[计算机科学与技术]	最有趣的理论+最有用的算法+不得不学的数据科学。	[[titles: [走进数据科学: 博大精深, 美不胜收, 整装待发, Video...]	125	24
4	C_597225	大学计算机	□	大学计算机课程将以计算思维为导向，以计算机原理、概念为基础，以新技术新方法为牵引，以创新思维...	[[titles: [第1周: 基于计算机的问题求解, 课程介绍, 开篇...]	165	82
5	C_597229	财务分析与决策	[应用经济学, 管理科学与工程]	这门课程用财务语言解构企业的价值创造过程，从而帮助学习者理解影响价值创造的各种因素，建立财务...	[[titles: [资金的运用——认识资产, '1.1 绪论', '绪论'], ...]	95	65
6	C_597291	高级英语写作	□	本课程能够帮助学生掌握英语段落、短文、图表作文、应用文的写作方法和技巧，学习地道、规范的学术...	[[titles: [Chapter One Paragraph Writing', ...]	54	84
7	C_597307	大唐兴衰	[历史学]	隋唐五代史是史学名著《资治通鉴》中分量最重、史料价值最高的部分。通过老师导读，同学们可深入...	[[titles: [第一课、隋朝开基, 第一节 隋帝杨坚, 第一节 隋帝杨坚...]	65	66
8	C_597365	五分钟轻松搞定职场礼仪 (2019卷)	□	职场‘礼’为先，成功的未来不是梦！	[[titles: [课程介绍动画: 职场 礼'为先, 成功的未来不是梦', '课程介绍...]	88	17
9	C_597367	时尚化妆造型 (2018秋)	□	针对爱美人士讲解时尚生活、新娘和商业化妆造型基本原理及方法，使其能快速掌握现在时尚流行妆容造...	[[titles: [第一章 化妆基础, '1.1 化妆品与化妆工具', Video...]	24	55

- Xác định kích thước dữ liệu:
 - Mô tả: Để biết được số hàng, số cột của dữ liệu.
 - Hàm sử dụng: dataframe.shape
 - Input: dữ liệu được lưu dưới dạng dataframe
 - Output: Một tập hợp gồm hai phần tử. Phần tử đầu tiên là số hàng, phần tử thứ hai là số cột.
 - Nhận xét: Dữ liệu có 3781 hàng và 6 cột.

df.shape

(3781, 6)

5.2.2.2.2 Xử lý các cột dữ liệu bị thiếu bị thiếu

- Xác định kiểu dữ liệu và các cột bị thiếu dữ liệu:
 - Mô tả: Xác định kiểu dữ liệu của từng cột trong dữ liệu và các cột nào bị thiếu dữ liệu.
 - Hàm sử dụng: dataframe.info()

- Input: dữ liệu được lưu dưới dạng dataframe
- Output: Kiểu dữ liệu và số giá trị không phải null tương ứng với từng cột trong dataframe

```
df.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 3781 entries, 0 to 3780
Data columns (total 6 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   id          3781 non-null    object 
 1   name         3781 non-null    object 
 2   field        3781 non-null    object 
 3   prerequisites 3779 non-null    object 
 4   about         3779 non-null    object 
 5   resource      3781 non-null    object 
dtypes: object(6)
memory usage: 177.4+ KB
```

Nhận xét:

- Dữ liệu có cấu trúc tốt với hầu hết các cột đầy đủ giá trị.
 - Có 2 dòng bị thiếu ở cột prerequisites và about, nhưng tỷ lệ thiếu là rất nhỏ (0.05%), có thể xử lý dễ dàng bằng phương pháp điền giá trị mặc định hoặc loại bỏ nếu cần thiết.
- Xác định số dữ liệu bị thiếu trong từng cột:
 - Mô tả: Xác định cụ thể số dữ liệu bị thiếu trong từng cột
 - Thư viện: pandas
 - Hàm sử dụng: dataframe.isnull().sum()
 - Input: dữ liệu được lưu dưới dạng dataframe
 - Output: Số dữ liệu bị thiếu trong từng cột.
 - **Nhận xét:** Sau khi kiểm tra các giá trị null trong bộ dữ liệu thì chúng ta thu được 2 cột bị thiếu đó là cột ‘prerequisites’ và ‘about’. Với mỗi cột đều thiếu 2 giá trị được thể hiện trong hình.

```
# Kiểm tra giá trị thiếu
print("\nGiá trị thiếu trong dữ liệu:")
print(df.isnull().sum())
```

```
Giá trị thiếu trong dữ liệu:
id          0
name        0
field       0
prerequisites 2
about       2
resource     0
dtype: int64
```

- Xử lý các giá trị bị thiếu trong từng cột:
 - Mô tả: Xác định cụ thể số dữ liệu bị thiếu trong từng cột
 - Thư viện: pandas
 - Hàm sử dụng: dataframe.fillna("")

```
# Bước 2: Làm sạch dữ liệu trước khi phân tích
# Xử lý giá trị None hoặc rỗng trong cột 'about'
df['about'] = df['about'].fillna("") # Thay None bằng chuỗi rỗng
df['prerequisites'] = df['prerequisites'].fillna("Không có") # Thay None bằng "Không có"

# Kiểm tra giá trị thiếu
print("\nGiá trị thiếu trong dữ liệu:")
print(df.isnull().sum())
```

Giá trị thiếu trong dữ liệu:

	id	name	field	prerequisites	about	resource
	0	0	0	0	0	0

dtype: int64

Nhận xét: Sau khi thay thế các dòng bị thiếu thì bộ dữ liệu đã được làm sạch, sẵn sàng cho việc trích xuất các đặc trưng sau này

5.2.2.3 Tính toán các thống kê cơ bản

- Xác định mục tiêu tính toán dựa trên 2 cột ‘resource’ và ‘about’

Thống kê cơ bản cho số lượng tài nguyên:

	count	mean	std	min	25%	50%	75%	max
	3781.000000	71.685533	74.802345	1.000000	38.000000	59.000000	88.000000	2728.000000

Name: num_resources, dtype: float64

Thống kê cơ bản cho độ dài mô tả:

	count	mean	std	min	25%	50%	75%	max
	3781.000000	121.144142	100.863319	0.000000	66.000000	98.000000	142.000000	512.000000

Name: about_length, dtype: float64

Trung vị số lượng tài nguyên: 59.0
 Độ lệch chuẩn số lượng tài nguyên: 74.80234466137358
 Trung vị độ dài mô tả: 98.0
 Độ lệch chuẩn độ dài mô tả: 100.86331896103769

Nhận xét:

Số lượng tài nguyên của các khóa học có mức độ phân tán cao ($std = 74.80$) và phân phối lệch phái, thể hiện qua:

- Trung bình (71.69) lớn hơn trung vị (59).
- Giá trị cực đại rất lớn (2.728), vượt xa 75% phân vị (88), cho thấy sự xuất hiện của giá trị ngoại lai.

Hầu hết các khóa học có khoảng 38–88 tài nguyên, nhưng có một số ít khóa học vượt xa mức này.

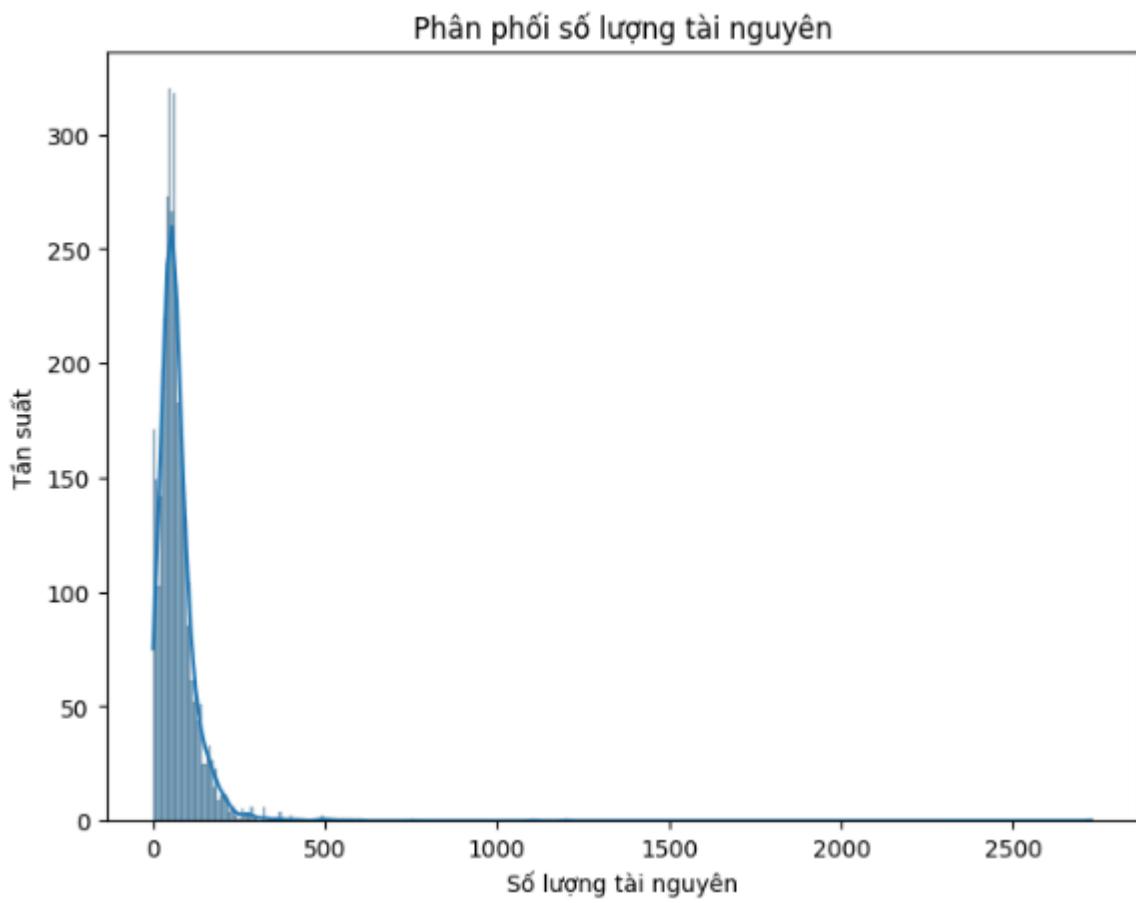
Mô tả khóa học có độ dài trung bình khoảng 121 ký tự, nhưng phân bố rất không đồng đều:

- Một số mô tả gần như rỗng (0 ký tự).
- Sự chênh lệch giữa trung vị (98) và trung bình (121) cho thấy phân phối lệch phái.
- Có mô tả dài lên đến 512 ký tự, thể hiện một vài khóa học cung cấp thông tin chi tiết hơn hẳn phần lớn còn lại.

Độ lệch chuẩn lớn (100.86) phản ánh mức độ biến thiên cao giữa các mô tả.

5.2.2.4 Phân tích phân phối dữ liệu

Sử dụng **biểu đồ Histogram** để thể hiện phân phối số lượng tài nguyên

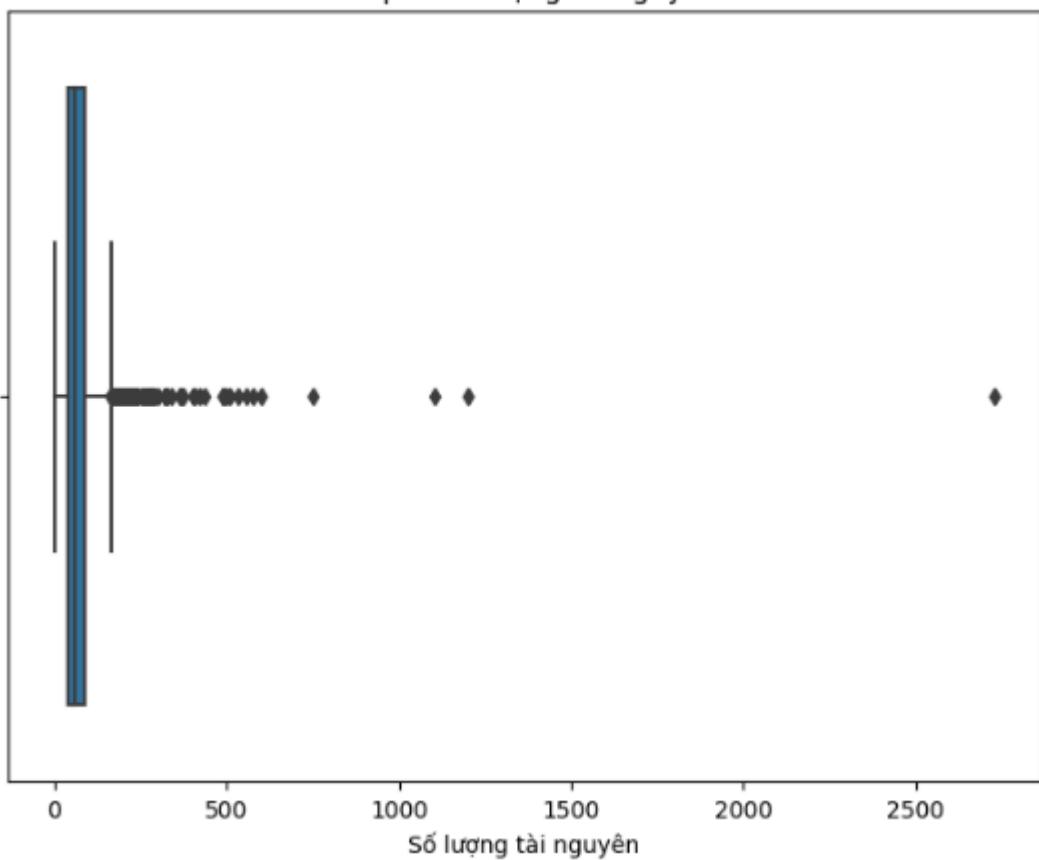


Nhận xét:

- Phân phối nghiêng về bên phải (right-skewed): Phần lớn các khóa học có số lượng tài nguyên tương đối thấp, tập trung nhiều ở khoảng từ 0 đến 100.
- Đuôi dài: Có một số khóa học có số lượng tài nguyên rất cao (lên đến hơn 2500), tuy nhiên đây là các trường hợp hiếm.
- Dạng phân phối không chuẩn (non-normal): Dữ liệu không phân phối chuẩn mà lệch, cho thấy có sự mất cân đối trong phân phối tài nguyên giữa các khóa học.

Sử dụng **biểu đồ Boxplot** để phát hiện các giá trị ngoại lai (outliers)

Boxplot số lượng tài nguyên



Nhận xét:

Hộp chính (box) đại diện cho khoảng từ phân vị (IQR), phần lớn dữ liệu nằm trong khoảng từ 0 đến ~50 tài nguyên.

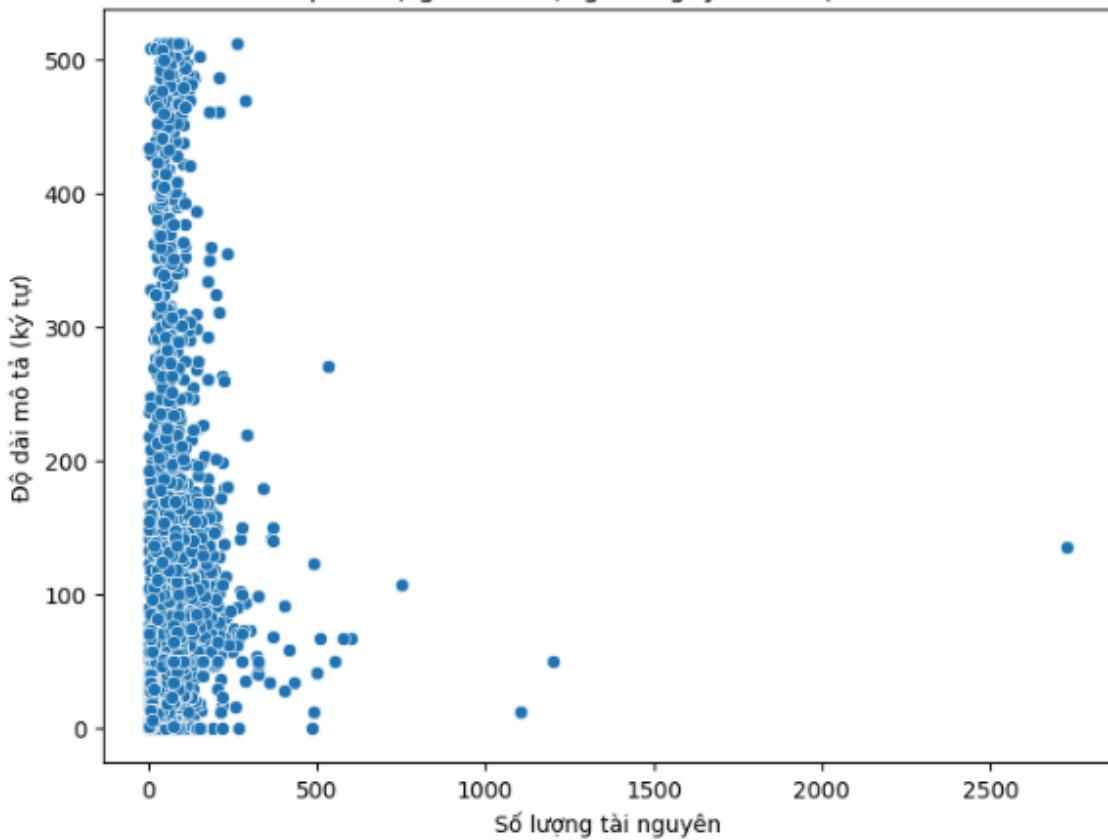
Nhiều giá trị ngoại lai (outliers) nằm rải rác phía bên phải, có thể thấy nhiều điểm vượt xa so với phần lớn dữ liệu:

- Các chấm rời bên phải biểu diễn những khóa học có lượng tài nguyên cực kỳ lớn, vượt quá ngưỡng bình thường.
- Một số điểm nằm trên 1000, thậm chí gần 3000.

2.2.2.5 Mối quan hệ giữa số lượng tài nguyên và độ dài mô tả

Biểu đồ scatter cho chúng ta thấy được 1 cách trực quan về mối quan hệ của 2 biến

Mối quan hệ giữa số lượng tài nguyên và độ dài mô tả



Nhận xét:

Phân bố dữ liệu tập trung ở giá trị thấp: Phần lớn các khóa học có số lượng tài nguyên dưới 500 và mô tả ngắn hơn 200 ký tự. Điều này phù hợp với các thống kê mô tả trước đó, trong đó trung vị của số lượng tài nguyên là 59 và trung vị độ dài mô tả là 98 ký tự.

Không tồn tại mối quan hệ tuyến tính rõ rệt giữa hai biến. Quan sát biểu đồ cho thấy dữ liệu có xu hướng phân tán ngẫu nhiên, không hình thành mô hình hoặc xu hướng cụ thể nào giữa độ dài mô tả và số lượng tài nguyên. Điều này cho thấy độ dài mô tả không phải là yếu tố quyết định số lượng tài nguyên, và ngược lại.

Xuất hiện một số điểm ngoại lai ở cả hai chiều:

- Một số khóa học có số lượng tài nguyên vượt quá 1000, nhưng đi kèm mô tả ngắn.
- Một số mô tả dài trên 400 ký tự lại gắn với số lượng tài nguyên trung bình hoặc thấp.

Những giá trị này có thể ảnh hưởng đến các phép phân tích thống kê và nên được xử lý hoặc đánh giá kỹ hơn trong giai đoạn tiền xử lý dữ liệu.

5.2.2.5. Trích xuất các resourse của khóa học

Các trường trong resource của mỗi khóa học bao gồm: title, resource_id, chapter.

```
: resource_df = course_info["resource"].apply(pd.Series)
```

```
: course_info_exploded = course_info.explode("resource")
course_info_exploded
```

	id	name	field	prerequisites	about	resource
0	C_584313	《资治通鉴》导读	[历史学,中国语言文学]		通过老师导读，同学们可深入这一经典文本内部，得以纵览千年历史，提升国学素养，体味人生智慧。	{"titles": ["第一课 导论与三家分晋", "导论", "导论"], "resou...
0	C_584313	《资治通鉴》导读	[历史学,中国语言文学]		通过老师导读，同学们可深入这一经典文本内部，得以纵览千年历史，提升国学素养，体味人生智慧。	{"titles": ["第一课 导论与三家分晋", "智伯的覆亡", "智伯的覆亡"], ...}
0	C_584313	《资治通鉴》导读	[历史学,中国语言文学]		通过老师导读，同学们可深入这一经典文本内部，得以纵览千年历史，提升国学素养，体味人生智慧。	{"titles": ["第一课 导论与三家分晋", "智伯悲剧的反思", "智伯的覆亡讨论..."]}
0	C_584313	《资治通鉴》导读	[历史学,中国语言文学]		通过老师导读，同学们可深入这一经典文本内部，得以纵览千年历史，提升国学素养，体味人生智慧。	{"titles": ["第一课 导论与三家分晋", None, "第一课 导论与三家分晋-..."]}
0	C_584313	《资治通鉴》导读	[历史学,中国语言文学]		通过老师导读，同学们可深入这一经典文本内部，得以纵览千年历史，提升国学素养，体味人生智慧。	{"titles": ["第二课 战国前期的政治", "魏文侯治国", "魏文侯治国"], ...}
...
3780	C_2329163	创意写作实践教程	[]	基础写作知识，传播学知识，美学知识等。	《创意写作案例实践教程》是新文科建设中的重要学科，其目的是培养具有复合型知识体系，能够助力文...	{"titles": ["第四章 红色之旅实践写作", "4.16求职简历", "4...."]}

```
: course_info_exploded["resource_id"] = course_info_exploded["resource"].apply(lambda x: x.get("resource_id") if isinstance(x, dict) else None)
course_info_exploded["resource_chapter"] = course_info_exploded["resource"].apply(lambda x: x.get("chapter") if isinstance(x, dict) else None)
```

	id	name	field	prerequisites	about	resource	resource_id	resource_chapter
0	C_584313	《资治通鉴》导读	[历史学,中国语言文学]		通过老师导读，同学们可深入这一经典文本内部，得以纵览千年历史，提升国学素养，体味人生智慧。	{"titles": ["第一课 导论与三家分晋", "导论", "导论"], "resou...	V_849	1.1.1
1	C_584313	《资治通鉴》导读	[历史学,中国语言文学]		通过老师导读，同学们可深入这一经典文本内部，得以纵览千年历史，提升国学素养，体味人生智慧。	{"titles": ["第一课 导论与三家分晋", "智伯的覆亡", "智伯的覆亡"], ...}	V_850	1.2.1
2	C_584313	《资治通鉴》导读	[历史学,中国语言文学]		通过老师导读，同学们可深入这一经典文本内部，得以纵览千年历史，提升国学素养，体味人生智慧。	{"titles": ["第一课 导论与三家分晋", "智伯悲剧的反思", "智伯的覆亡"]}	V_851	1.3.1

	id	resource_id	resource_chapter
0	C_584313	V_849	1.1.1
1	C_584313	V_850	1.2.1
2	C_584313	V_851	1.3.1
3	C_584313	Ex_856	1.4
4	C_584313	V_857	2.1.1
...
271038	C_2329163	V_8630134	3.15
271039	C_2329163	Ex_8638642	3.15.1
271040	C_2329163	V_8630135	3.16
271041	C_2329163	V_8630136	3.17
271042	C_2329163	V_8630137	3.18

271043 rows × 3 columns

Gộp các resouce_id có cùng id khóa học

```

# Group by 'id' and separate Videos and Exercises
grouped = course_info_remove.groupby('id').agg(
    resource_id_video= ('resource_id', lambda x: list(x[x.str.startswith('V')])),
    chapter_video= ('resource_chapter', lambda x: list(x[course_info_remove['resource_id'].str.startswith('V')])),
    resource_id_ex= ('resource_id', lambda x: list(x[x.str.startswith('Ex')])),
    chapter_ex= ('resource_chapter', lambda x: list(x[course_info_remove['resource_id'].str.startswith('Ex')])))
).reset_index()

```

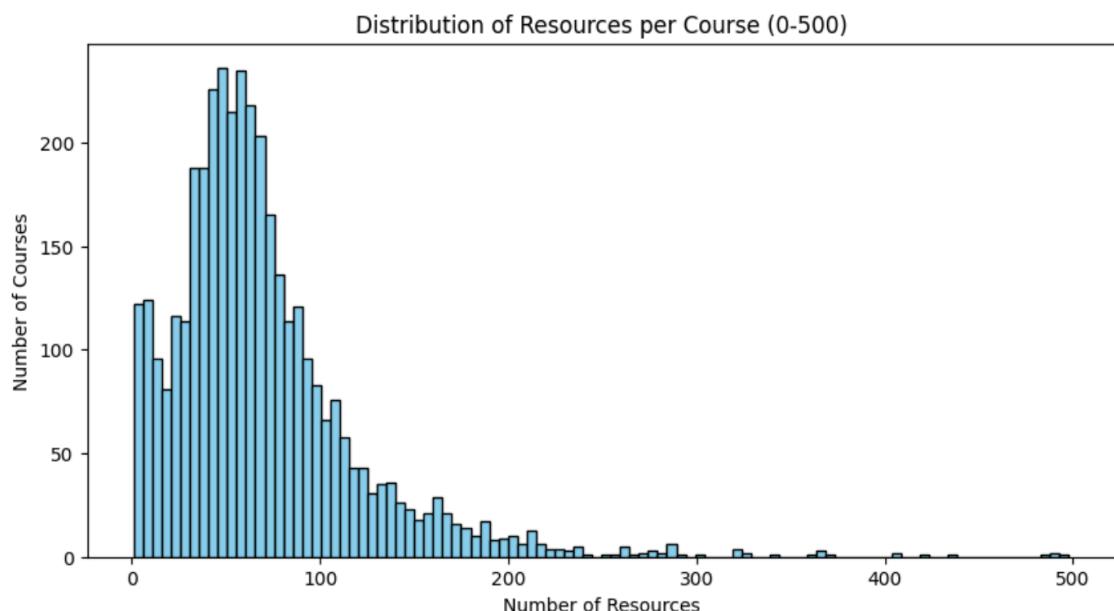
	id	resource_id_video	chapter_video	resource_id_ex	chapter_ex
0	C_1017355	[V_1617852, V_1617854, V_1736570, V_1736573, V...]	[0, 0.1, 1, 1.1, 1.1.1, 1.2.1, 2, 2.1, 3, 3.1,...]	[Ex_1617853, Ex_1617855, Ex_1736572, Ex_173657...]	[0.1, 0.1.1, 1.1, 1.1.1, 1.1.2, 1.2.2, 2.1, 2....]
1	C_1017419	[V_1618247, V_1618249, V_1618250, V_1618252, V...]	[1.2, 2.1, 2.2, 3.1, 3.2, 4.1, 4.2, 5.1, 5.2, ...]	[Ex_1618248, Ex_1618251, Ex_1618254, Ex_161825...]	[1.3, 2.3, 3.3, 4.3, 5.3, 6.3, 7.3, 8.3, 9.3]
2	C_1025064	[V_1622546, V_1622547, V_1622548, V_1622549, V...]	[0, 0.1, 0.2, 0.3, 0.3.1, 0.3.2, 0.4, 0.5, 1, ...]	[Ex_1622556, Ex_1622566, Ex_1622574, Ex_162258...]	[0.6, 1.7, 2.4, 3.2, 4.4, 5.3, 6.4, 7.4, 8.3, ...]
3	C_1025076	[V_1622858, V_1622859, V_1622860, V_1622861, V...]	[0, 0.1, 0.2, 0.3, 0.2, 0.2.1, 0.2.2, 0.2.3, 0...]	[Ex_1622870, Ex_1622875, Ex_1622876, Ex_1622889]	[0.2.6, 1.2, 1.3, 2.11]
4	C_1025079	[V_1622989, V_1622990, V_1622991, V_1622992, V...]	[1.1.1, 1.2.1, 1.3.1, 1.4.1, 2.1.1, 2.2.1, 2.3...]	[Ex_1622993, Ex_1623006, Ex_1623011, Ex_162302...]	[1.5, 4.5, 5.5, 7.5, 8.3]
...
3776	C_955163	[V_1542335, V_1542336, V_1542337, V_1542338, V...]	[0, 0.1, 0.2, 0.3, 1, 2, 2.1, 2.2, 2.3, 2.4, 2...]	[Ex_1576375, Ex_1576376, Ex_1545423, Ex_154542...]	[0.4, 1.1, 2.19, 3.21, 4.43, 5.12, 6.1]
3777	C_956128	[V_1555005, V_1555006, V_1555007, V_1555010, V...]	[0, 0.1, 0.3, 1, 1.1, 1.2, 1.4, 2, 2.1, 2.2, 2...]	[Ex_1555008, Ex_1555014, Ex_1555021, Ex_155502...]	[0.2, 1.3, 2.4, 3.3, 4.3, 5.2, 6.3]
		[V_1555097,	...	[Ex_1555101,	...

```
id_counts = course_info_exploded.groupby('id').size()
id_counts
```

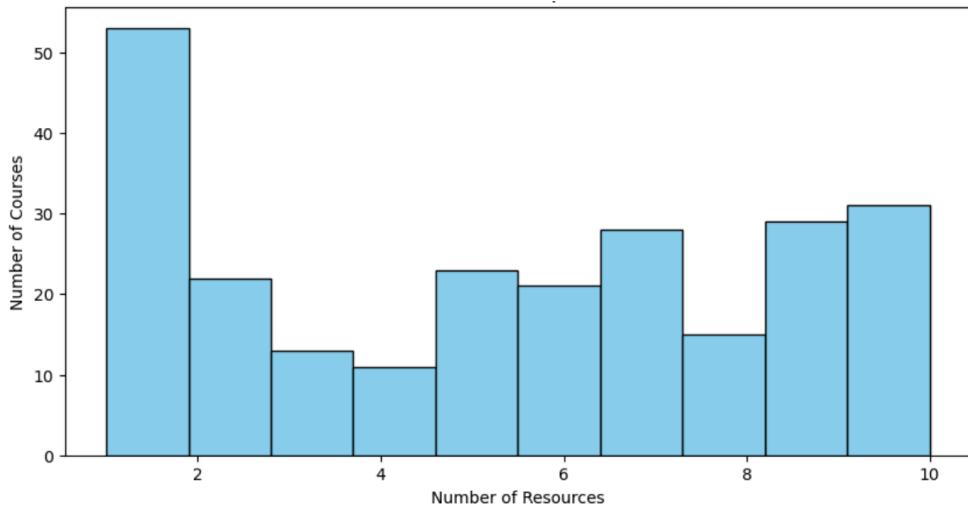
```
id
C_1017355    38
C_1017419    26
C_1025064   103
C_1025076    47
C_1025079    39
...
C_955163    108
C_956128     34
C_956129     96
C_956130     86
C_956450     48
Length: 3781, dtype: int64
```

Vẽ histogram thể hiện số lượng khóa học tương ứng với số resource nằm trong khoảng từ **0 đến 500**.

Biểu đồ phân bố số lượng resource của các khóa học có số lượng từ 0 đến 500 resource



Zoom kỹ vào khoảng từ **0 đến 10 resource** để quan sát chi tiết hơn nhóm khóa học có ít tài nguyên.



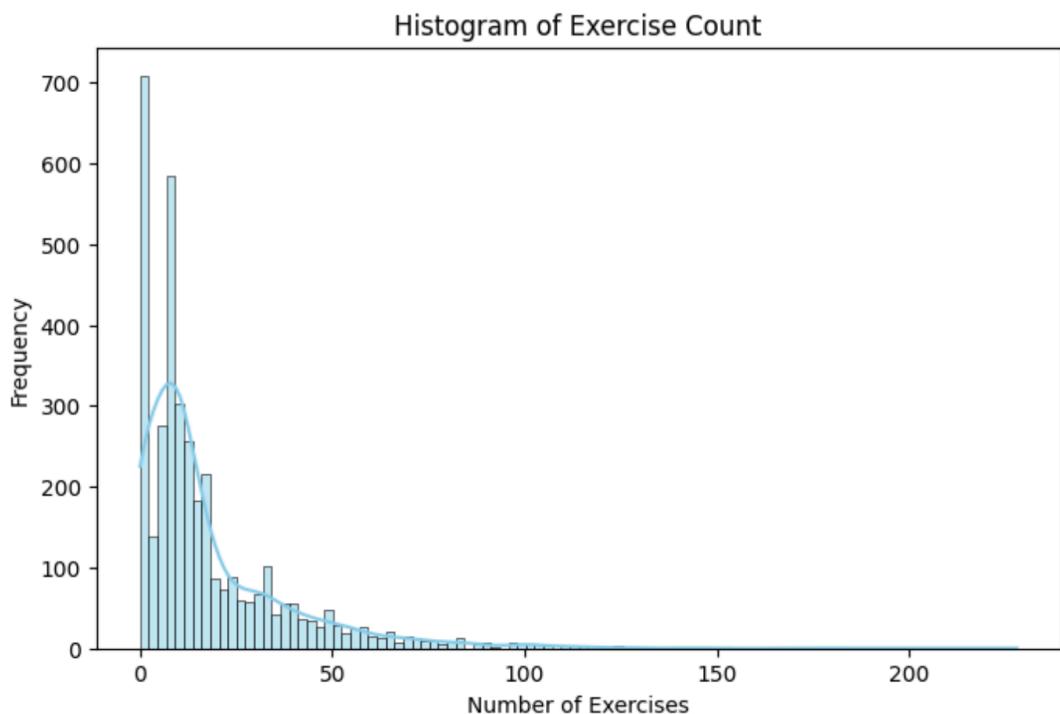
Nhận xét:

- Có những khóa học không có resource nào.
- Sự phân bố cho thấy đa số khóa học có số lượng resource khá thấp, chỉ một phần nhỏ có số lượng lớn resource.
- Điều này có thể ảnh hưởng đến trải nghiệm học tập hoặc kết quả học tập của người học.

5.2.2.6. Lọc những đối tượng không có exercise và video

Tách video và exercise trong chapter.

# Group by 'id' and separate Videos and Exercises				
grouped_course = filtered_courses.groupby('course_id').agg(resource_id_video=('resource_id', lambda x: list(x[x.str.startswith('V')])), chapter_video=('resource_chapter', lambda x: list(x[filtered_courses['resource_id'].str.startswith('V')])), resource_id_ex=('resource_id', lambda x: list(x[x.str.startswith('Ex')])), chapter_ex=('resource_chapter', lambda x: list(x[filtered_courses['resource_id'].str.startswith('Ex')]))).reset_index() grouped_course				
course_id	resource_id_video	chapter_video	resource_id_ex	chapter_ex
0 C_1017355	[V_1617852, V_1617854, V_1736570, V_1736573, V...]	[0, 0.1, 1, 1.1, 1.2, 1, 2, 2.1, 3, 3.1, ...]	[Ex_1617853, Ex_1617855, Ex_1736572, Ex_173657...]	[0.1, 0.1.1, 1.1, 1.1.1, 1.1.2, 1.2.2, 2.1, 2....]
1 C_1017419	[V_1618247, V_1618249, V_1618250, V_1618252, V...]	[1.2, 2.1, 2.2, 3.1, 3.2, 4.1, 4.2, 5.1, 5.2, ...]	[Ex_1618248, Ex_1618251, Ex_1618254, Ex_161825...]	[1.3, 2.3, 3.3, 4.3, 5.3, 6.3, 7.3, 8.3, 9.3]
2 C_1025064	[V_1622546, V_1622547, V_1622548, V_1622549, V...]	[0, 0.1, 0.2, 0.3, 0.3.1, 0.3.2, 0.4, 0.5, 1, ...]	[Ex_1622556, Ex_1622566, Ex_1622574, Ex_162258...]	[0.6, 1.7, 2.4, 3.2, 4.4, 5.3, 6.4, 7.4, 8.3, ...]
3 C_1025076	[V_1622858, V_1622859, V_1622860, V_1622861, V...]	[0, 0.1, 0.2, 0.3, 0.2, 0.2.1, 0.2.2, 0.2.3, 0...]	[Ex_1622870, Ex_1622875, Ex_1622876, Ex_1622889]	[0.2.6, 1.2, 1.3, 2.11]
4 C_1025079	[V_1622989, V_1622990, V_1622991, V_1622992, V...]	[1.1.1, 1.2.1, 1.3.1, 1.4.1, 2.1.1, 2.2.1, 2.3...]	[Ex_1622993, Ex_1623006, Ex_1623011, Ex_162302...]	[1.5, 4.5, 5.5, 7.5, 8.3]
...
2713 C_949542	[V_1505188, V_1505190, V_1505192, V_1505194, V...]	[0, 0.1, 0.2, 1, 1.1, 1.2, 2, 2.1, 2.2, 2.3, 2...]	[Ex_1505189, Ex_1505191, Ex_1505193, Ex_150519...]	[0.1, 0.1.1, 0.2.1, 1.1, 1.1.1, 1.2.1, 2.1, 2....]
2714 C_955163	[V_1542335, V_1542336, V_1542337, V_1542338, V...]	[0, 0.1, 0.2, 0.3, 1, 2, 2.1, 2.2, 2.3, 2.4, 2...]	[Ex_1576375, Ex_1576376, Ex_1545423, Ex_154542...]	[0.4, 1.1, 2.19, 3.21, 4.43, 5.12, 6.1]



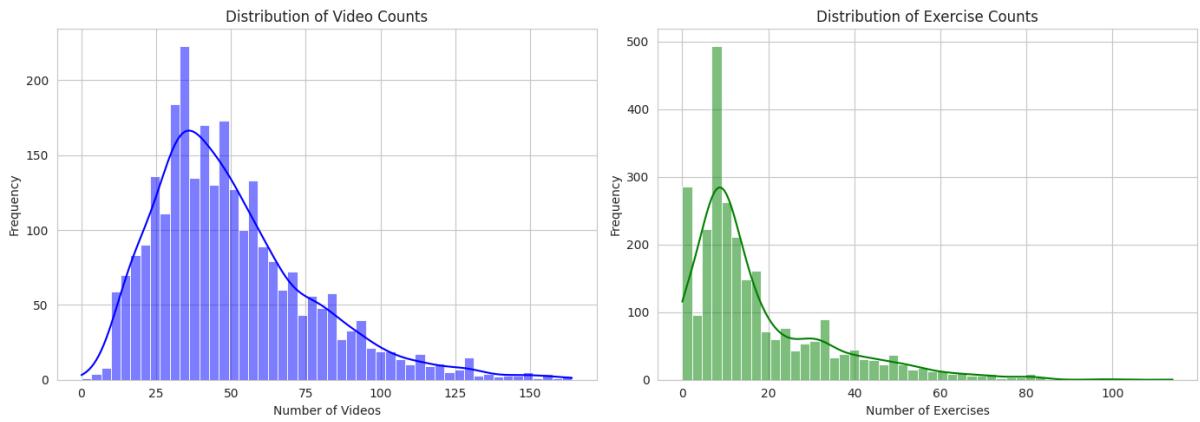
```
# Filter courses where exercise_count is 0
courses_with_no_exercises = grouped[grouped['exercise_count'] == 0]

# Print the results
print(courses_with_no_exercises[['id', 'exercise_count']])
```

	id	exercise_count
6	C_1123814	0
18	C_1328548	0
30	C_1429002	0
47	C_1627979	0
49	C_1628160	0
...
3645	C_947824	0
3657	C_948075	0
3708	C_948248	0
3755	C_948422	0
3770	C_948486	0

[577 rows x 2 columns]

Phân phối số lượng video, exercise của các khóa học.

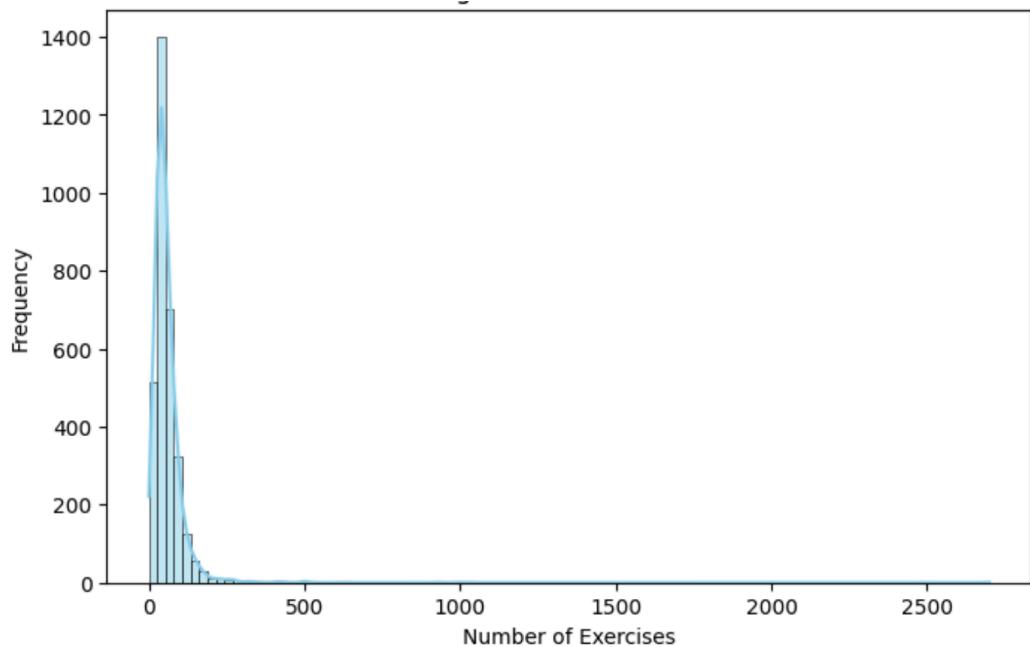


Nhận xét:

- Số lượng video
 - Phân phối tương đối đều.
 - Càng nhiều video thì số lượng khóa học giảm dần.
 - Phân phối dạng phân rã đều (phân phối chuẩn lệch trái nhẹ).
- Số lượng bài tập (exercise):
 - Phân phối lệch phải rõ rệt.
 - Số lượng bài tập phổ biến nhất là 10 bài.
 - Tiếp theo là các khóa học có 0 bài tập.
 - Sau đó số lượng khóa học giảm dần khi số bài tăng.

Đáng chú ý: vẫn tồn tại một số khóa học không có cả bài tập và video, gây ảnh hưởng tới chất lượng học tập và trải nghiệm người dùng.

Có tổng cộng 577 khóa học không có exercise



Xóa những khóa học không có video nhưng chỉ có 1 exercise

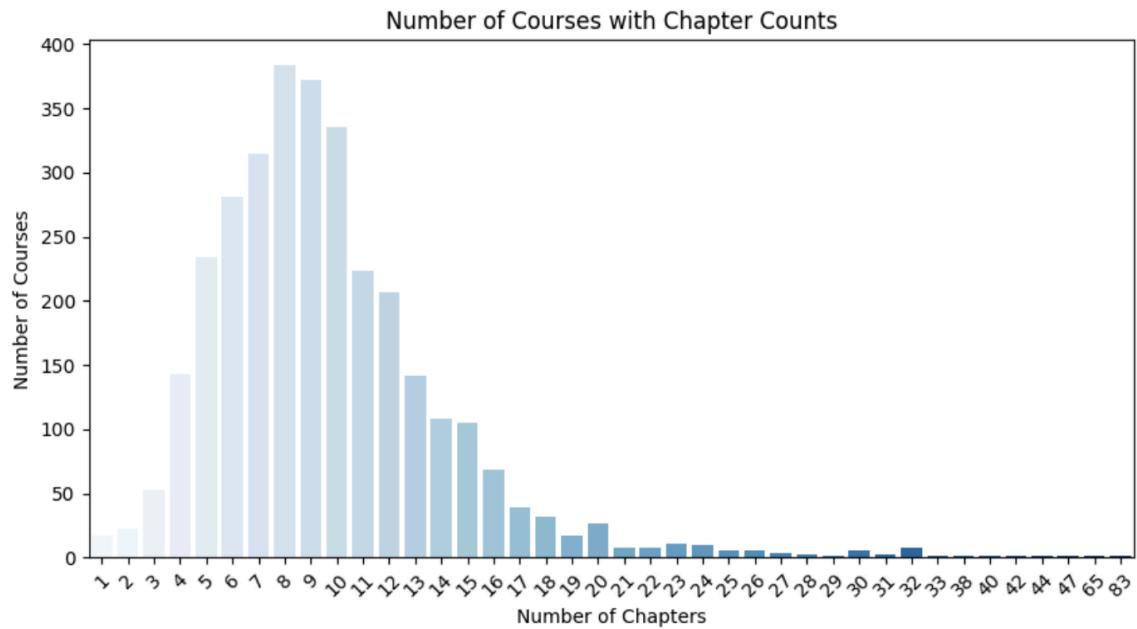
```
# Drop courses where video_count is 0 and exercise_count is 1
grouped = grouped[~((grouped['video_count'] == 0) & (grouped['exercise_count'] == 1))].reset_index(drop=True)

# Print the updated DataFrame
grouped
```

	id	resource_id_video	chapter_video	resource_id_ex	chapter_ex	video_count	exercise_cc
0	C_1017355	[V_1617852, V_1617854, V_1736570, V_1736573, V...]	[0, 0.1, 1, 1.1, 1.1.1, 1.2.1, 2, 2.1, 3, 3.1,...]	[Ex_1617853, Ex_1617855, Ex_1736572, Ex_1736573, ...]	[0.1, 0.1.1, 1.1, 1.1.1, 1.1.2, 1.2.2, 2.1, 2.2.1, ...]	19	19
1	C_1017419	[V_1618247, V_1618249, V_1618250, V_1618252, V...]	[1.2, 2.1, 2.2, 3.1, 3.2, 4.1, 4.2, 5.1, 5.2, ...]	[Ex_1618248, Ex_1618251, Ex_1618254, Ex_1618256, ...]	[1.3, 2.3, 3.3, 4.3, 5.3, 6.3, 7.3, 8.3, 9.3]	17	9
2	C_1025064	[V_1622546, V_1622547, V_1622548, V_1622549, V...]	[0, 0.1, 0.2, 0.3, 0.3.1, 0.3.2, 0.4, 0.5, 1, ...]	[Ex_1622556, Ex_1622566, Ex_1622574, Ex_1622581, ...]	[0.6, 1.7, 2.4, 3.2, 4.4, 5.3, 6.4, 7.4, 8.3, ...]	89	14
3	C_1025076	[V_1622858, V_1622859, V_1622860, V_1622861, V...]	[0, 0.1, 0.2, 0.3, 0.2, 0.2.1, 0.2.2, 0.2.3, 0, ...]	[Ex_1622870, Ex_1622875, Ex_1622876, Ex_1622889]	[0.2.6, 1.2, 1.3, 2.11]	43	4

Còn lại 3201 khóa học

Biểu đồ phân bố số lượng chapter của các khóa học



5.2.2.7. Trích xuất những exercise là bài kiểm tra

1. **Xác định các tài nguyên là bài kiểm tra:** Tạo một hàm để kiểm tra xem tiêu đề của một tài nguyên có chứa các từ khóa liên quan đến bài kiểm tra (như "final exam", "midterm exam", "期中考试", v.v.) hay không. Sử dụng hàm này để lọc ra chỉ những tài nguyên được xác định là bài kiểm tra.

```

def contains_exam_keyword(titles):
    keywords = ['final exam', 'midterm exam', '期中考试', '期末考试', '结课考试']
    return any(
        title and any(kw in title.lower() if isinstance(title, str) else False for kw in keywords)
        for title in titles
    )
# Lọc các dòng thỏa mãn điều kiện
exam_related = df_final[df_final['titles'].apply(contains_exam_keyword)]

# Xem kết quả
exam_related

```

	id	titles	resource_id	chapter
4674	C_676642	[期末考试, None, 期末考试-判断题]	Ex_5458293	11.3
5834	C_676664	[《大国航母与舰载机》期末考试题, None, 一、填空题]	Ex_581330	10.1
5835	C_676664	[《大国航母与舰载机》期末考试题, None, 二、单选题]	Ex_581331	10.2
5836	C_676664	[《大国航母与舰载机》期末考试题, None, 三、多选题]	Ex_581534	10.3
5837	C_676664	[《大国航母与舰载机》期末考试题, None, 四、判断题]	Ex_581535	10.4
...
262457	C_2316362	[期末考试, None, 期末考试-期末考试Part3]	Ex_8530777	12.5
263873	C_2328495	[Step10 Final Exam, 10.1 Guidance: Reviews of ...]	V_8621887	10
265816	C_2333035	[期末考试, None, 课程考试-作业]	Ex_8676612	14.2
270465	C_2338076	[第15章 期末考试与总结, None, 第一部分：基础知识]	Ex_8729369	15.2

2. **Lọc các bài kiểm tra có ID bắt đầu bằng 'Ex_':** Trong số các tài nguyên được xác định là bài kiểm tra, bạn đã lọc tiếp chỉ những tài nguyên có ID bắt đầu bằng 'Ex_'. Điều này có thể nhằm mục đích tập trung vào một loại bài kiểm tra cụ thể.

```

ex_exam_related = exam_related[exam_related['resource_id'].str.startswith('Ex_')]
ex_exam_related

```

	id	titles	resource_id	chapter
4674	C_676642	[期末考试, None, 期末考试-判断题]	Ex_5458293	11.3
5834	C_676664	[《大国航母与舰载机》期末考试题, None, 一、填空题]	Ex_581330	10.1
5835	C_676664	[《大国航母与舰载机》期末考试题, None, 二、单选题]	Ex_581331	10.2
5836	C_676664	[《大国航母与舰载机》期末考试题, None, 三、多选题]	Ex_581534	10.3
5837	C_676664	[《大国航母与舰载机》期末考试题, None, 四、判断题]	Ex_581535	10.4
...
262456	C_2316362	[期末考试, None, 期末考试-期末考试Part2]	Ex_8530776	12.3
262457	C_2316362	[期末考试, None, 期末考试-期末考试Part3]	Ex_8530777	12.5
265816	C_2333035	[期末考试, None, 课程考试-作业]	Ex_8676612	14.2
270465	C_2338076	[第15章 期末考试与总结, None, 第一部分：基础知识]	Ex_8729369	15.2
270541	C_2341259	[期末考试, None, 期末考试-作业]	Ex_8782460	10.2

Có tổng cộng 493 khóa học có bài kiểm tra theo phương pháp này

- 3. Nhóm và đếm số lượng bài kiểm tra cho mỗi khóa học:** Nhóm dữ liệu theo ID khóa học và thu thập danh sách các ID tài nguyên bài kiểm tra (cùng với chương mà chúng thuộc về) cho mỗi khóa học. Sau đó, đếm số lượng các bài kiểm tra cho mỗi khóa học.

```
grouped = ex_exam_related.groupby('id').apply(  
    lambda g: list(zip(g['resource_id'], g['chapter']))  
).reset_index(name='exam_resources')  
  
grouped
```

```
/tmp/ipykernel_13/3125399789.py:1: DeprecationWarning: DataFr  
grouped = ex_exam_related.groupby('id').apply(
```

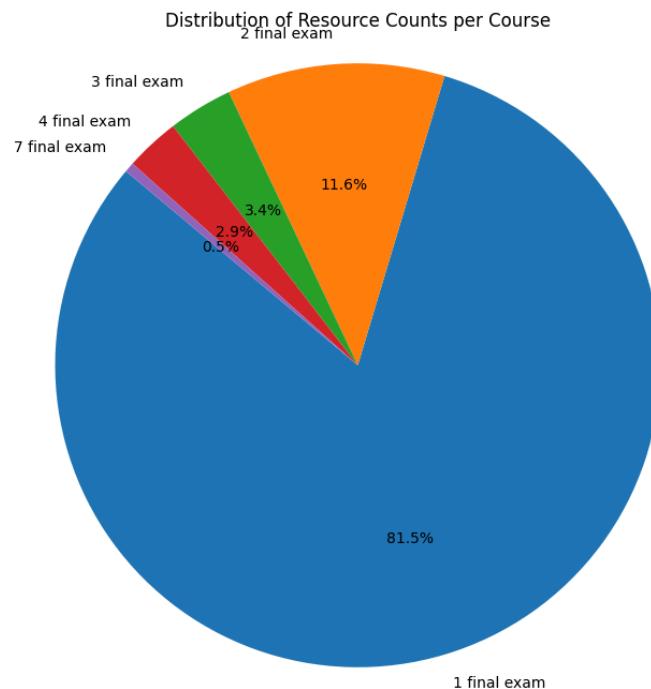
	id	exam_resources
0	C_1073350	[(Ex_1650729, 3), (Ex_1650730, 3.1)]
1	C_1714443	[(Ex_3625291, 11.1)]
2	C_1718815	[(Ex_4974697, 17.1)]
3	C_1721804	[(Ex_3742687, 3), (Ex_3742688, 3.1)]
4	C_1724283	[(Ex_3800298, 6.1)]
...
373	C_948114	[(Ex_1467099, 5), (Ex_1573116, 5.1), (Ex_15731...
374	C_948123	[(Ex_1467720, 7.1)]
375	C_948128	[(Ex_6318041, 4.1), (Ex_6318076, 10.1)]
376	C_948140	[(Ex_1469502, 9.1)]

- 4. Vẽ biểu đồ phân phối:** Trực quan hóa phân phối số lượng bài kiểm tra trên mỗi khóa học bằng cách sử dụng biểu đồ hình tròn (pie chart). Biểu đồ này cho thấy tỷ lệ phần trăm các khóa học có số lượng bài kiểm tra khác nhau.

```

# Bước 1: Đếm số lượng khóa học theo số lượng bài kiểm tra
counts = grouped['resource_count'].value_counts().sort_index()
# Bước 2: Vẽ pie chart
plt.figure(figsize=(8, 8))
plt.pie(
    counts,
    labels=[f"{i} final exam" for i in counts.index],
    autopct='%.1f%%',
    startangle=140
)
plt.title('Distribution of Resource Counts per Course')
plt.axis('equal') # Đảm bảo hình tròn
plt.show()

```



Nhận xét:

- Phần lớn các khóa học có đúng 1 bài kiểm tra → chiếm tỷ lệ cao nhất, có thể là bài cuối kỳ hoặc tổng kết.
- Khóa học có 2 bài kiểm tra đúng thứ hai → thường có thể là bài giữa kỳ và cuối kỳ.
- Các khóa học có 3 đến 4 bài kiểm tra chiếm tỷ lệ trung bình (khoảng 2.9% – 3.5%) → thể hiện sự đánh giá thường xuyên hơn.
- Khóa học có 7 bài kiểm tra là thấp nhất → rất ít khóa học có số lượng bài kiểm tra cao như vậy.
- Đa số khóa học có ít bài kiểm tra, rất hiếm có nhiều bài

5.2.2.2.8. Trích xuất trường field trong phần giới thiệu để fillna

1. Xử lý dữ liệu về khóa học: Tạo một trường văn bản mới ('text') trong dữ liệu khóa học, kết hợp tên và mô tả khóa học để có một đoạn văn bản đầy đủ hơn cho mỗi khóa học.

```
course['text'] = '课程: ' + course['name'] + ' . 课程描述: ' + course['about']
```

	id	name	field	prerequisites	about	resource	text
0	C_584313	《资治通鉴》导读	[历史学, 中国语言文学]		通过老师导读, 同学们可深入这一经典文本内部, 得以纵览千年历史, 提升国学素养, 体味人生智慧。	[{'titles': ['第一课 导论与三家分晋', '导论', '导论'], 'resource_id': ...}]	课程: 《资治通鉴》导读. 课程描述: 通过老师导读, 同学们可深入这一经典文本内部, 得以纵览...
1	C_584329	微积分——极限理论与一元函数	[应用经济学, 数学, 物理学, 理论经济学]		本课程是理工科的一门数学基础课, 系统、全面地介绍了一元函数微积分学。课程既保持了数学的严谨和...	[{'titles': ['序言', '序言', '序言'], 'resource_id': ...}]	课程: 微积分——极限理论与一元函数. 课程描述: 本课程是理工科的一门数学基础课, 系统、全...
2	C_584381	新闻摄影	[艺术学, 新闻传播学]		掌握基本的摄影技能, 了解图片新闻的工作方式, 训练对生活的观察和热爱, 发展对图像的审美和批评能...	[{'titles': ['第一章 绪论', '第一章 引言1', '引言1'], 'resource_id': ...}]	课程: 新闻摄影. 课程描述: 掌握基本的摄影技能, 了解图片新闻的工作方式, 训练对生活的观察...
3	C_597208	数据挖掘: 理论与算法	[计算机科学与技术]		最有趣的理论+最有用的算法=不得不学的数据科学。	[{'titles': ['走进数据科学: 博大精深, 美不胜收', '整装待发', 'Video'], 'resource_id': ...}]	课程: 数据挖掘: 理论与算法. 课程描述: 最有趣的理论+最有用的算法=不得不学

2. Lọc khóa học dựa trên dữ liệu người dùng: Lọc dữ liệu khóa học ban đầu để chỉ giữ lại những khóa học có 'course_id' nằm trong danh sách các khóa học mà người dùng đã tương tác. Điều này giúp tập trung vào các khóa học có liên quan đến hoạt động của người dùng.

```
limit_course = course[course["course_id"].isin(course_ids)]  
limit_course
```

	course_id	name	field	prerequisites	about	resource	text
48	C_674968	巴蜀文化	[中国语言文学, 民族学]		巴蜀文化是中华文明绽放于西南大地的灿烂之花。本课程将从考古、历史、文学、宗教、哲学、艺术、地...	[{'titles': ['第一章:导论——巴蜀文化的悠久历程与风格特色', '1.1...']}]	课程: 巴蜀文化. 课程描述: 巴蜀文化是中华文明绽放于西南大地的灿烂之花。本课程将从考古、...
49	C_674971	宝玉石鉴赏	[]		【国家精品课】宝玉石, 既是自然美的精华, 也是财富身份的象征。《宝玉石鉴赏》让你了解宝石的自然...	[{'titles': ['第一讲 序概', '1.1 宝玉石的基本概念、属性和种类', ...]}]	课程: 宝玉石鉴赏. 课程描述: 【国家精品课】宝玉石, 既是自然美的精华, 也是财富身份的象征...
51	C_676642	创办新企业	[工商管理]		《创办新企业》课程由清华科技园和清华大学经济管理学院联合开...	[{'titles': ['第一章：《创办新企业》-创业者的梦...']}]	课程: 创办新企业. 课程描述: 《创办新企业》课程由清华科技...

3. Tạo DataFrame: Tạo một DataFrame mới (df) chỉ chứa các cột 'course_id', 'field' và 'text' từ dữ liệu khóa học đã lọc. Đây là DataFrame chính mà bạn sẽ làm việc tiếp theo.

```
df = limit_course[['course_id', 'field', 'text']]
df
```

	course_id	field	text
48	C_674968	[中国语言文学, 民族学]	课程: 巴蜀文化. 课程描述: 巴蜀文化是中华文明绽放于西南大地的灿烂之花。本课程将从考古、...
49	C_674971		课程: 宝玉石鉴赏. 课程描述: 【国家精品课】宝玉石，既是自然美的精华，也是财富身份的象征...
51	C_676642	[工商管理]	课程: 创办新企业. 课程描述: 《创办新企业》课程由清华科技园和清华大学经济管理学院联合开...
65	C_676664		课程: 大国航母与舰载机. 课程描述: 本课程将带您走进大国航母与舰载机的世界，为您揭开世界...
79	C_676705	[法学]	课程: 民法与生活. 课程描述: 本课程是一门面向非法学专业学生及社会人士修读的法律类课程。...
...
3745	C_2342496		课程: 推广学. 课程描述: 农业推广人为本，传播沟通是基础。行为观念想改变，教育咨询是关键...
3747	C_2342499		课程: 热流体工程. 课程描述: 奋斗路上，做对选择与锻炼能力同等重要。\\n也许你对掌握基础...
3748	C_2342500		课程: 新闻传播学. 课程描述: 新闻传播学是一门综合性的学科，新闻传播学的研究对象是新闻...

4. Phân loại dữ liệu thành đã gán nhãn và chưa gán nhãn:

Chia DataFrame (df) thành hai phần:

- labeled_df: Chứa các hàng mà cột 'field' (lĩnh vực) có chứa ít nhất một giá trị (có nghĩa là đã có thông tin về lĩnh vực).
- unlabeled_df: Chứa các hàng mà cột 'field' rỗng (chưa có thông tin về lĩnh vực).

```
# Separate labeled and unlabeled
labeled_df = df[df['field'].apply(lambda x: len(x) > 0)]
unlabeled_df = df[df['field'].apply(lambda x: len(x) == 0)]
labeled_df
```

	course_id	field	text
48	C_674968	[中国语言文学, 民族学]	课程: 巴蜀文化. 课程描述: 巴蜀文化是中华文明绽放于西南大地的灿烂之花。本课程将从考古、...
51	C_676642	[工商管理]	课程: 创办新企业. 课程描述: 《创办新企业》课程由清华科技园和清华大学经济管理学院联合开...
79	C_676705	[法学]	课程: 民法与生活. 课程描述: 本课程是一门面向非法学专业学生及社会人士修读的法律类课程。...
119	C_677061	[中国语言文学, 民族学]	课程: 中国少数民族神话赏析. 课程描述: 在中华民族文明史上，55个少数民族神话是一笔珍贵...
124	C_677087	[中国语言文学]	课程: 中国哲学经典著作导读. 课程描述: 这是一个全球化、信息化、快餐化的时代；全球化拷问...
...

5. Chuẩn bị cho việc gán nhãn tự động:

- Sử dụng các lĩnh vực duy nhất từ dữ liệu đã gán nhãn làm "ứng viên nhãn" cho mô hình phân loại.
- Kiểm tra các dòng trong unlabeled_df có trường 'text' bị thiếu dữ liệu.

6. Sử dụng mô hình phân loại Zero-Shot: Thiết lập một mô hình phân loại Zero-Shot (một loại mô hình có thể phân loại văn bản mà không cần được huấn luyện cụ thể trên các nhãn đó) để dự đoán các lĩnh vực cho các khóa học chưa được gán nhãn.

```

from transformers import pipeline
import pandas as pd

# Setup classifier
classifier = pipeline("zero-shot-classification", model="valhalla/distilbart-mnli-12-1")

# Make sure all fields are actual lists
labeled_df['field'] = labeled_df['field'].apply(lambda x: eval(x) if isinstance(x, str) else x)

# Candidate labels = all unique labels from existing 'field'
all_labels = unique_fields

# Predict top 2 labels for each
def predict_top2_labels(text):
    result = classifier(text, all_labels, multi_label=True)
    print(result['labels'][::2])
    return result['labels'][::2]

Downloading:  0%|          | 0.00/1.39k [00:00<?, ?B/s]
Downloading:  0%|          | 0.00/890M [00:00<?, ?B/s]
Downloading:  0%|          | 0.00/899k [00:00<?, ?B/s]
Downloading:  0%|          | 0.00/456k [00:00<?, ?B/s]
Downloading:  0%|          | 0.00/772 [00:00<?, ?B/s]
Downloading:  0%|          | 0.00/26.0 [00:00<?, ?B/s]
/opt/conda/lib/python3.7/site-packages/ipykernel_launcher.py:8: SettingWithCopyWarning:
A value is trying to be set on a copy of a slice from a DataFrame.
Try using .loc[row_indexer,col_indexer] = value instead

```

7. Áp dụng mô hình và lưu kết quả:

- Áp dụng mô hình phân loại Zero-Shot cho trường 'text' của các khóa học trong mẫu này.
- Đối với mỗi khóa học trong mẫu, mô hình đã dự đoán hai nhãn (lĩnh vực) có khả năng cao nhất.
- Các nhãn được dự đoán này đã được lưu vào một cột mới ('predicted_field') trong unlabeled_df_sample.

```

# Apply prediction
unlabeled_df_sample['predicted_field'] = unlabeled_df_sample['text'].apply(predict_top2_labels)

['交通运输工程', '林业工程']
['机械工程', '交通运输工程']
['冶金工程', '交通运输工程']
['轻工技术与工程', '交通运输工程']
['交通运输工程', '管理科学与工程']
['数学', '交通运输工程']
['土木工程', '交通运输工程']
['艺术学', '交通运输工程']
['轻工技术与工程', '交通运输工程']
['交通运输工程', '土木工程']
['系统科学', '控制科学与工程']
['中国语言文学', '矿业工程']
['交通运输工程', '食品科学与工程']
['交通运输工程', '土木工程']
['中国语言文学', '矿业工程']
['基础中医学', '轻工技术与工程']
['艺术学', '土木工程']
['交通运输工程', '矿业工程']
['土木工程', '轻工技术与工程']
['艺术学', '民族学']
['林业工程', '动力工程及工程热物理']
['土木工程', '矿业工程']
['交通运输工程', '矿业工程']
['土木工程', '交通运输工程']
['交通运输工程', '光学工程']
['矿业工程', '水利工程']
['交通运输工程', '机械工程']
['管理科学与工程', '交通运输工程']

```

course-field.csv

Tên cột	Mô tả	Kiểu dữ liệu
course_id	ID của khóa học	string
field	Lĩnh vực của khóa học	list[string]
text	Giới thiệu của khóa học để model có thể dự đoán	list[string/float]
predicted_field	Lĩnh vực được dự đoán	list[string]

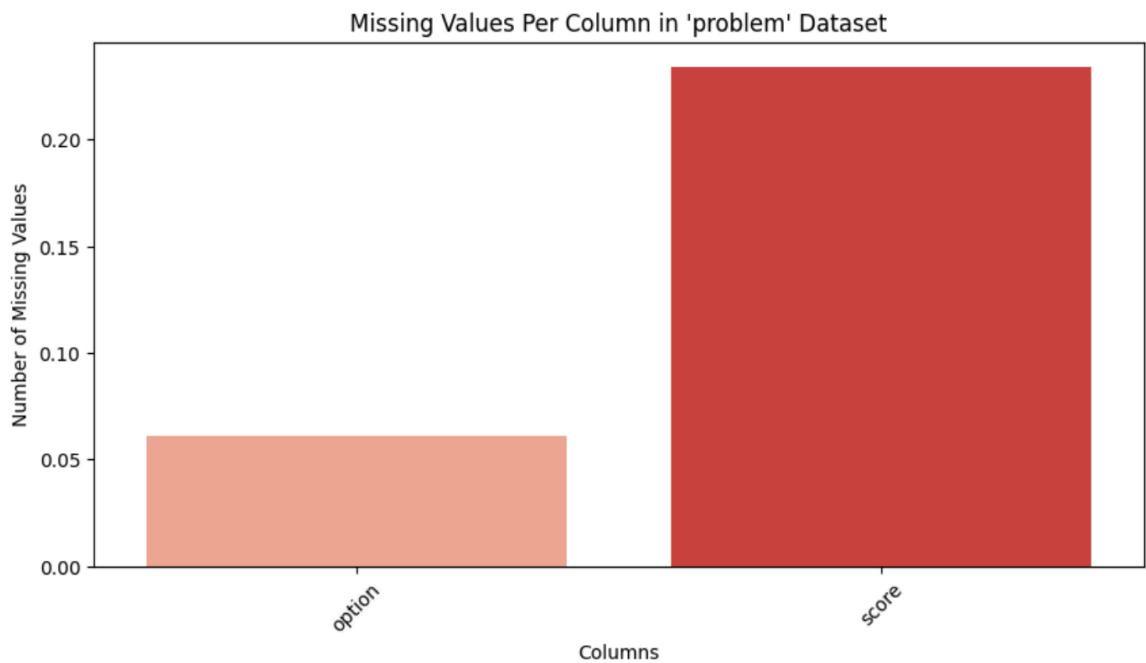
5.2.2.2.9. Kết luận các trường được trích xuất

Tên cột	Mô tả	Kiểu dữ liệu
course_id	ID của khóa học	string
resource_id_video	Danh sách ID video của khóa học	list[string]
chapter_video	Danh sách chương chứa các video tương ứng	list[string/float]
resource_id_ex	Danh sách ID bài tập (exercise) của khóa học	list[string]
chapter_ex	Danh sách chương chứa bài tập tương ứng	list[string/float]
video_count	Tổng số video trong khóa học	integer
exercise_count	Tổng số bài tập trong khóa học	integer
exam_count	Tổng số lượng bài kiểm tra	interger
chapter	Danh sách tất cả chương liên quan tới video và bài tập	list[string]
new_chapter	Danh sách chương được gom nhóm/thông nhất (dạng đơn giản hơn)	list[string or int]
chapter_count	Tổng số chương của khóa học (theo new_chapter)	integer

5.2.2.3. Xử lý bộ dữ liệu entities/problem.json

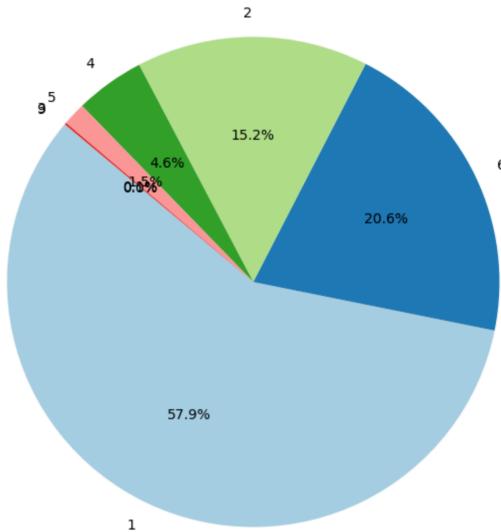
5.2.2.3.1. Phân phối các dạng câu hỏi trong file problem

Số lượng các giá trị bị thiếu



Phần trăm số lượng các câu hỏi xuất hiện trong problem

- 1 → 单选题 → Câu hỏi trắc nghiệm (chỉ chọn một đáp án)
- 2 → 多选题 → Câu hỏi trắc nghiệm (chọn nhiều đáp án)
- 3 → 投票题 → Câu hỏi bình chọn/thăm dò ý kiến
- 4 → 填空题 → Câu hỏi điền vào chỗ trống
- 5 → 主观题 → Câu hỏi tự luận
- 6 → 判断题 → Câu hỏi đúng/sai
- 9 → 编程题 → Câu hỏi lập trình



```
missing_score_rows['typetext'].value_counts()
```

```
typetext
单选题    345258
判断题    108995
多选题    86233
填空题    26271
主观题    6751
投票题    801
编程题    63
Name: count, dtype: int64
```

1. 单选题 (Chọn một) – 345,258 câu
2. 判断题 (Đúng/Sai) – 108,995 câu
3. 多选题 (Chọn nhiều) – 86,233 câu
4. 填空题 (Điền vào chỗ trống) – 26,271 câu
5. 主观题 (Tự luận) – 6,751 câu
6. 投票题 (Bình chọn) – 801 câu
7. 编程题 (Lập trình) – 63 câu

Nhận xét

填空题 (Điền vào chỗ trống) & 主观题 (Tự luận) & 编程题 (Lập trình)

- Đây là loại câu hỏi phổ biến nhất vì nó dễ tự động chấm điểm. 填空题 (Điền vào chỗ trống) & 主观题 (Tự luận) & 编程题 (Lập trình)
- Những loại câu hỏi này có thể không luôn có điểm số cố định, vì đáp án có thể do người chấm hoặc hệ thống đánh giá.
- Do đó, có thể có nhiều giá trị NaN trong cột score.

投票题 (Bình chọn)

- Đây là loại câu hỏi mang tính khảo sát, không yêu cầu chấm điểm, nên gần như chắc chắn cột score sẽ bị thiếu dữ liệu.

单选题 (Chọn một), 多选题 (Chọn nhiều), 判断题 (Đúng/Sai)

- Những loại câu hỏi này thường có đáp án đúng/sai rõ ràng, nên có thể có ít hoặc không có giá trị NaN trong score.

5.2.2.3.2. Thông kê các dạng câu hỏi trong file problem

a. Câu hỏi loại 1

Phân bố số điểm cho câu hỏi loại 1 rất đa dạng

```
In [56]: problem[problem['type'] == 1]['score'].value_counts()
```

Out[56]:

score	
1.0	1013339
2.0	22369
5.0	17196
0.5	6310
4.0	5355
10.0	4106
3.0	3655
1.5	1338
8.0	687
20.0	504
6.0	466
12.5	282
2.5	216
7.0	156
0.3	145
0.8	133
15.0	114
9.0	66
25.0	58

```

]: # Tổng điểm của các bài tập có type = 1
problem[problem['type'] == 1]['score'].sum()

# Trung bình điểm
problem[problem['type'] == 1]['score'].mean()

# Thống kê chi tiết
problem[problem['type'] == 1]['score'].describe()

```

']:
count 1.076693e+06
mean 1.165604e+00
std 1.063132e+00
min 0.000000e+00
25% 1.000000e+00
50% 1.000000e+00
75% 1.000000e+00
max 1.000000e+02
Name: score, dtype: float64

Đa số các câu hỏi trắc nghiệm có 1 đáp án thường có điểm là 1

b. Câu hỏi loại 2

Phân phối điểm của câu hỏi loại 2 (trắc nghiệm nhiều đáp án) thường là 2.

```

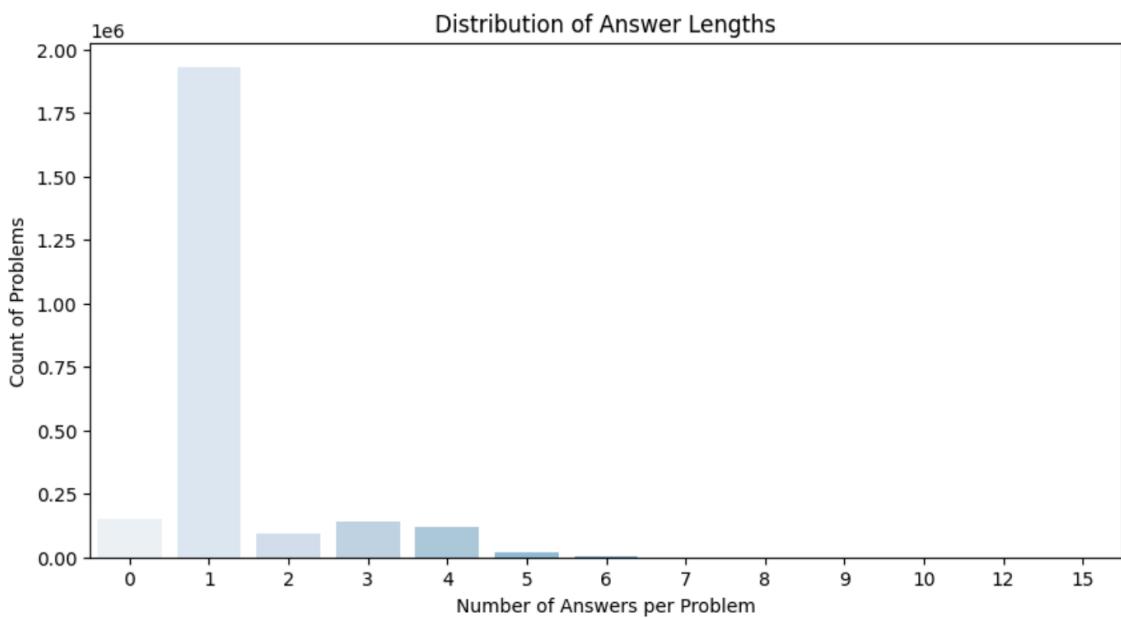
]: problem[problem['type'] == 2]['score'].describe()

```

']:
count 287339.000000
mean 2.214504
std 1.575641
min 0.000000
25% 2.000000
50% 2.000000
75% 2.000000
max 60.000000
Name: score, dtype: float64

```
problem[problem['type'] == 2]['score'].value_counts()
```

```
: score
2.0      256521
1.0      12518
10.0     5044
3.0      3140
0.5      2904
5.0      2294
4.0      1367
8.0      1029
6.0      950
1.5      435
20.0     420
15.0     309
2.5      109
3.5      86
25.0     62
7.0      56
30.0     32
50.0     16
0.0      11
```



Đa số câu hỏi loại 2 có điểm là 2.

c. **Câu hỏi loại 3**

```
: problem[problem['type'] == 3].describe()
```

```
:
```

	problem_id	score	type	answer_length
count	2.869000e+03	2068.000000	2869.0	2869.0
mean	5.330820e+06	0.988878	3.0	0.0
std	2.250823e+06	5.378141	0.0	0.0
min	4.874600e+04	0.000000	3.0	0.0
25%	3.172894e+06	0.000000	3.0	0.0
50%	6.008492e+06	1.000000	3.0	0.0
75%	7.489840e+06	1.000000	3.0	0.0
max	8.327356e+06	100.000000	3.0	0.0

	problem_id	title	content	option	answer	score	type	typetext	location	context_id	ex
16351	48746	请完成下面投票题目	提到混合教学，您的态度是什么？	{"A": "我不太想尝试", "B": "我想尝试，但怕工具太复杂，我学不会", "C": "...."}	[]	5.0	3	投票题	1.1	[34099]	Ex
274873	904342	2.2.2 课后习题	给我国第二艘航母，即第一艘国产航母，起个名字吧。	{"A": "山东号", "B": "广东号", "C": "天津号", "D": "河北号..."}	[]	3.0	3	投票题	2.2.2	[8015136]	Ex
287544	931458	2.2.2 课后习题	给我国第二艘航母，即第一艘国产	{"A": "山东号", "B": "广东号", "C": "天津号..."}	[]	3.0	3	投票题	2.2.2	[8015136]	Ex

Đa số câu hỏi loại 3 có điểm số là 1. Có giá trị cao nhất là 100.

d. Câu hỏi loại 4

```
: empty_answers[problem['type'] == 4]['answer'].info()
```

```
<class 'pandas.core.series.Series'>
Index: 111956 entries, 505 to 2451908
Series name: answer
Non-Null Count Dtype
-----
111956 non-null object
dtypes: object(1)
memory usage: 1.7+ MB
```

```
<ipython-input-88-f831c65a9698>:1: UserWarning: Boolean Series key will be reindexed to match DataFrame index.
```

```
empty_answers[problem['type'] == 4]['answer'].info()
```

Tất cả câu trả lời của type 4 đều không có giá trị là Null

```
# Thống kê chi tiết
problem[problem['type'] == 4]['score'].describe()
```

```
count    85685.000000
mean      1.875439
std       2.004266
min      0.000000
25%     1.000000
50%     1.000000
75%     2.000000
max     60.000000
Name: score, dtype: float64
```

Đa số số điểm trả lời được khi trả lời câu hỏi loại 4 (điền chữ) là 2 điểm

e. Câu hỏi loại 5

Đa số câu trả lời 5 có số điểm là 10

```
: problem[problem['type'] == 5].describe()
```

	problem_id	score	type	answer_length
count	3.740900e+04	30658.000000	37409.0	37409.0
mean	3.836608e+06	10.065748	5.0	0.0
std	2.614419e+06	6.650168	0.0	0.0
min	4.182000e+03	0.000000	5.0	0.0
25%	1.486077e+06	10.000000	5.0	0.0
50%	3.243127e+06	10.000000	5.0	0.0
75%	6.057590e+06	10.000000	5.0	0.0
max	8.430611e+06	100.000000	5.0	0.0

```
: problem.loc[(problem['type'] == 5), 'score'] = 10
```

f. Câu hỏi loại 6

Câu trả lời đúng sai (loại 6) đa số có số điểm là 1.

```
: problem[problem['type'] == 6].describe()
```

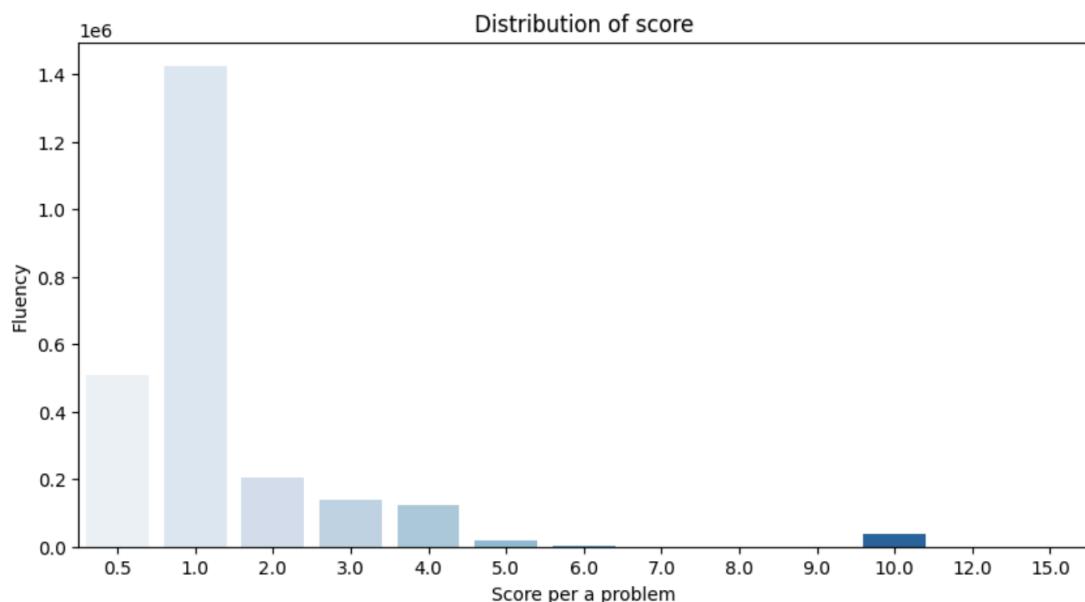
	problem_id	score	type	answer_length
count	5.063840e+05	397389.000000	506384.0	506384.0
mean	4.261787e+06	1.170451	6.0	1.0
std	2.734980e+06	1.105111	0.0	0.0
min	2.952000e+03	0.000000	6.0	1.0
25%	1.622895e+06	1.000000	6.0	1.0
50%	3.452984e+06	1.000000	6.0	1.0
75%	7.051726e+06	1.000000	6.0	1.0
max	8.428239e+06	50.000000	6.0	1.0

g. Câu hỏi loại 9

```
[]: problem[problem['type'] == 9].describe()
```

```
[]:
```

	problem_id	score	type	answer_length
count	2.810000e+02	218.000000	281.0	281.0
mean	5.643465e+06	44.678899	9.0	0.0
std	1.575529e+06	43.901205	0.0	0.0
min	3.530271e+06	10.000000	9.0	0.0
25%	3.591910e+06	10.000000	9.0	0.0
50%	5.862875e+06	10.000000	9.0	0.0
75%	7.213920e+06	100.000000	9.0	0.0
max	8.157310e+06	100.000000	9.0	0.0



5.2.2.4. Xử lý bộ dữ liệu relations/user-problem.json

Vì user-problem.json có dung lượng lớn nên nhóm tách thành 10 file. Và lặp lại việc xử lý cho các file user-problem-{i}.json

```

input_file = "./relations/user-problem.json"
num_parts = 10

# Count total lines
with open(input_file, "r", encoding="utf-8") as f:
    total_lines = sum(1 for _ in f)

lines_per_file = total_lines // num_parts
extra_lines = total_lines % num_parts # Remaining lines to be added to part_10.json

print(f"Total lines: {total_lines}")
print(f"Lines per file: {lines_per_file}, Extra lines (to part 10): {extra_lines}")

# Read and write files in chunks
with open(input_file, "r", encoding="utf-8") as f:
    for i in range(num_parts):
        part_filename = f"part_{i+1}.json"

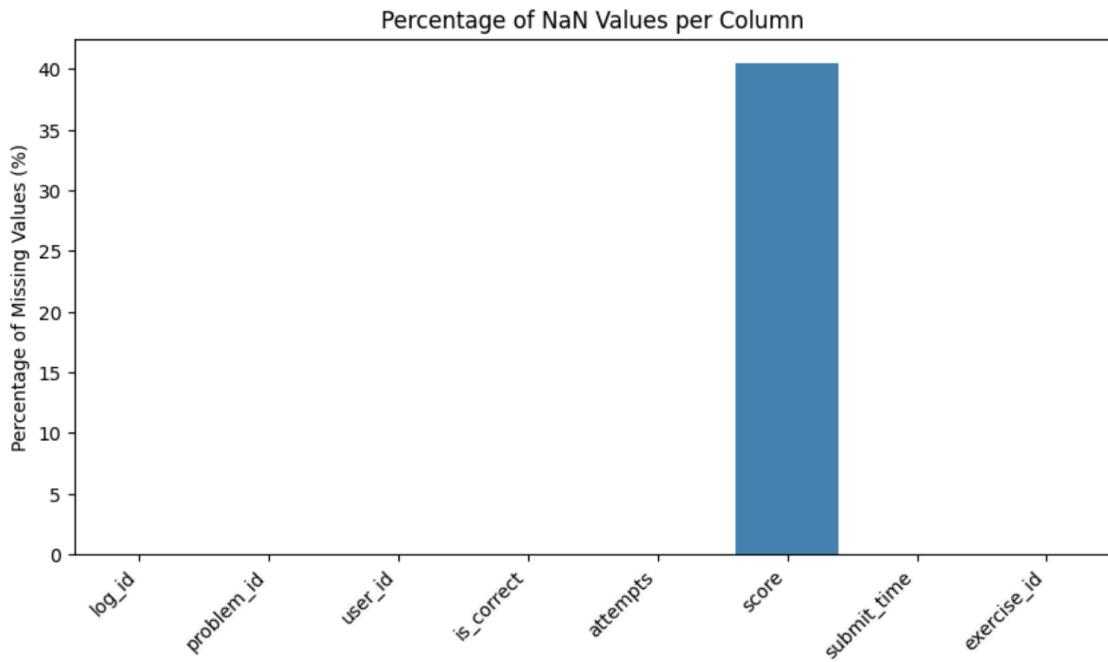
        # Add extra lines only to part_10.json
        lines_to_write = lines_per_file + (extra_lines if i == num_parts - 1 else 0)

        with open(part_filename, "w", encoding="utf-8") as part_file:
            for _ in range(lines_to_write):
                line = f.readline()
                if not line:
                    break
                part_file.write(line)

print("Splitting complete! ✅")

```

Nhóm xử lý file user-problem-1.json và các file khác tương tự.



Nhận xét: số lượng score (điểm của bài tập) bị thiếu rất nhiều

Để xử lý nhóm gộp với file problem.csv

5.2.2.4.1. Gộp dữ liệu user-problem và problem

Kết hợp (merge) DataFrame user_problem với ex_pm dựa trên cột 'problem_id'. Việc kết hợp này sử dụng phương pháp left để giữ lại tất cả các hàng từ user_problem và thêm thông tin 'exercise_id' tương ứng.

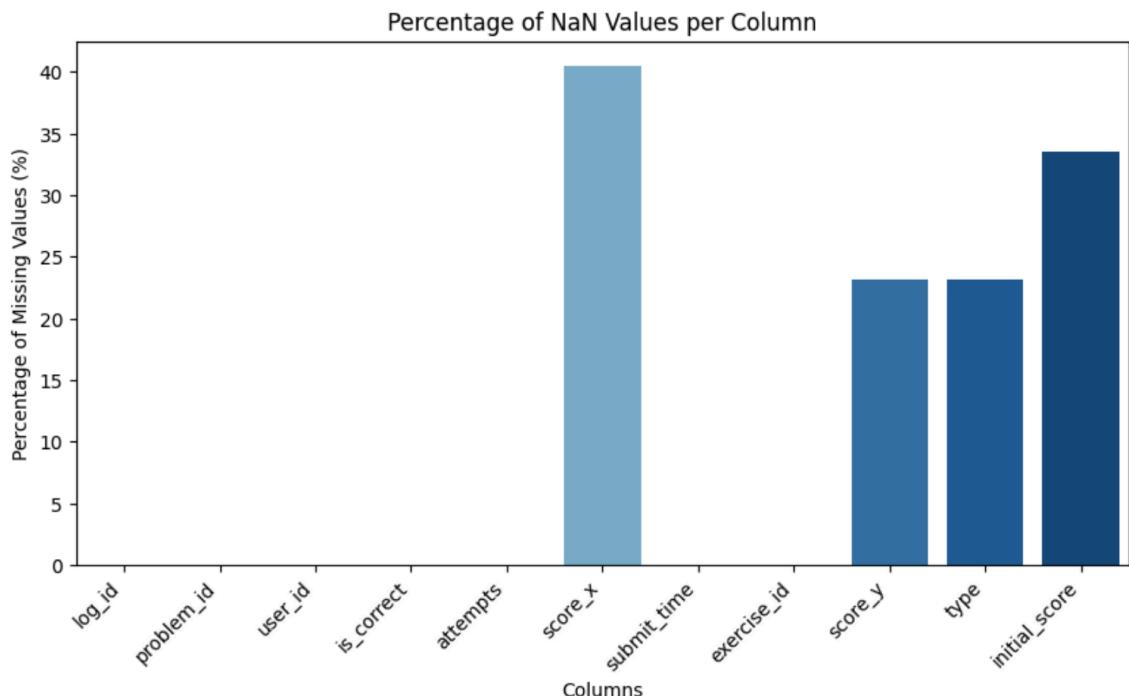
```
# Merge user_problem_course with problem DataFrame
user_problem_course = user_problem_course.merge(
    problem.add_suffix('_pm_info'), # Add suffix to all columns
    left_on="problem_id",
    right_on="problem_id_pm_info", # Keep original problem_id
    how="left"
)

# Drop the duplicate problem_id column from problem (if necessary)
user_problem_course = user_problem_course.drop(columns=["problem_id_pm_info"])

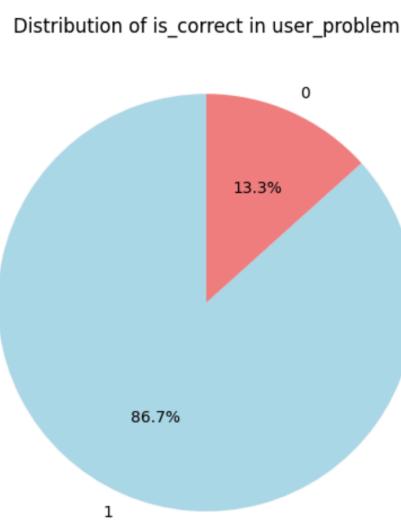
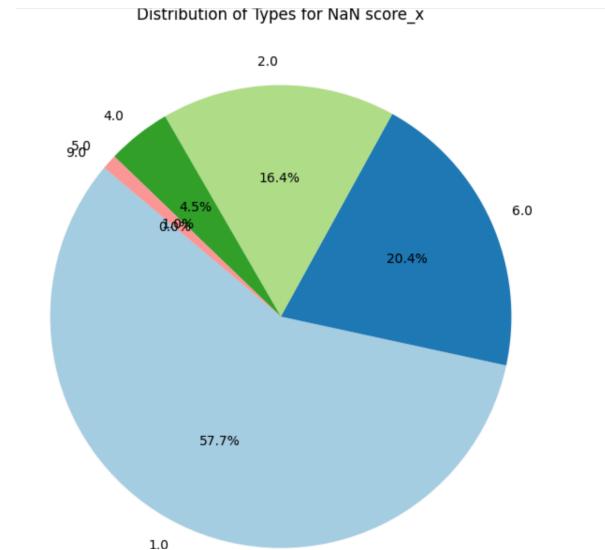
# Display the updated DataFrame
user_problem_course
```

/usr/local/lib/python3.10/dist-packages/pandas/io/format.py:1458: RuntimeWarning: invalid value encountered
has_large_values = (abs_vals > 1e6).any()
/usr/local/lib/python3.10/dist-packages/pandas/io/format.py:1459: RuntimeWarning: invalid value encountered
has_small_values = ((abs_vals < 10 ** (-self.digits)) & (abs_vals > 0)).any()
/usr/local/lib/python3.10/dist-packages/pandas/io/format.py:1459: RuntimeWarning: invalid value encountered
has_small_values = ((abs_vals < 10 ** (-self.digits)) & (abs_vals > 0)).any()

	problem_id	user_id	is_correct	attempts	score	submit_time	exercise_id	course_id	title_pm_info
0	Pm_6906522	U_10000	0	1	NaN	2020-10-27 10:11:56	Ex_7007033	C_2033958	第八章习题
1	Pm_6906523	U_10000	0	1	NaN	2020-10-27 10:12:13	Ex_7007033	C_2033958	第八章习题

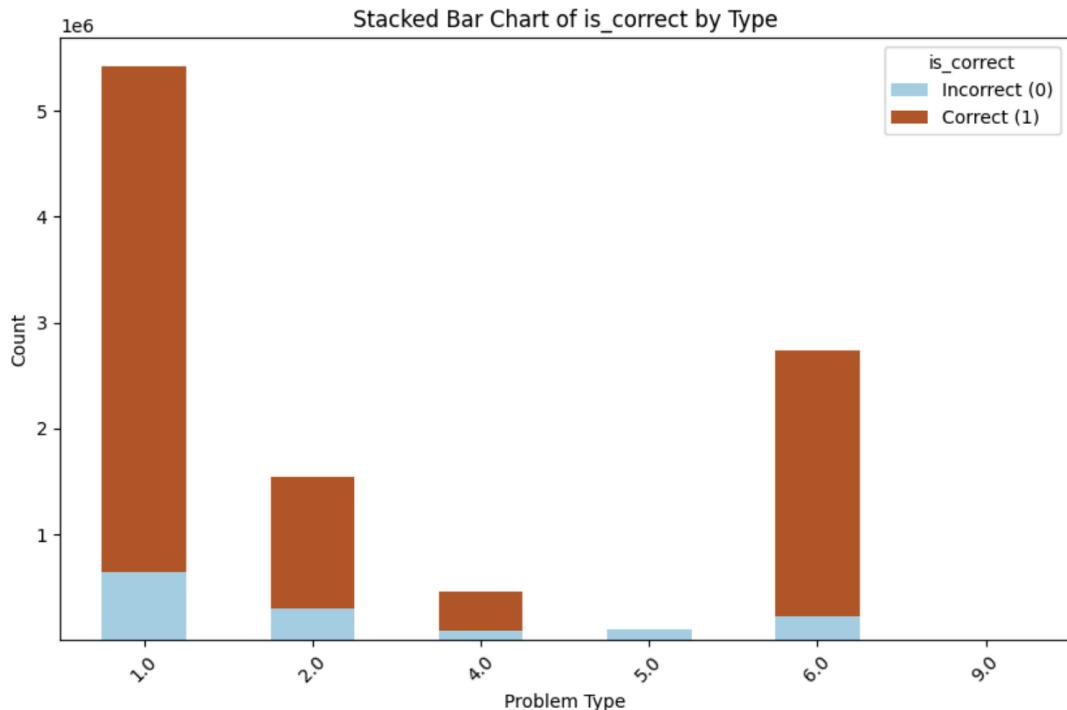


Sau khi merge thì số lượng score trong problem ít NaN hơn nên sẽ điền score_x (score ban đầu của user) dựa trên score_y



- Các giá trị NaN (khuyết) chủ yếu xuất hiện ở các câu hỏi loại 1, chiếm khoảng 57% tổng số các câu hỏi bị thiếu dữ liệu. Điều này cho thấy loại 1 có thể là dạng câu hỏi đặc biệt hoặc có cấu trúc khác biệt khiến dữ liệu không được ghi nhận đầy đủ.
- Các loại câu hỏi có tỷ lệ thiếu dữ liệu tiếp theo là loại 2 và loại 6, điều này đặt ra giả thuyết rằng một số loại bài tập có cấu trúc không đồng nhất hoặc chưa được người học tương tác đủ để sinh dữ liệu.
- Tỷ lệ người dùng trả lời đúng chiếm đa số, cho thấy bài tập có thể chưa đủ độ khó, hoặc người học có hiểu biết tốt về nội dung.

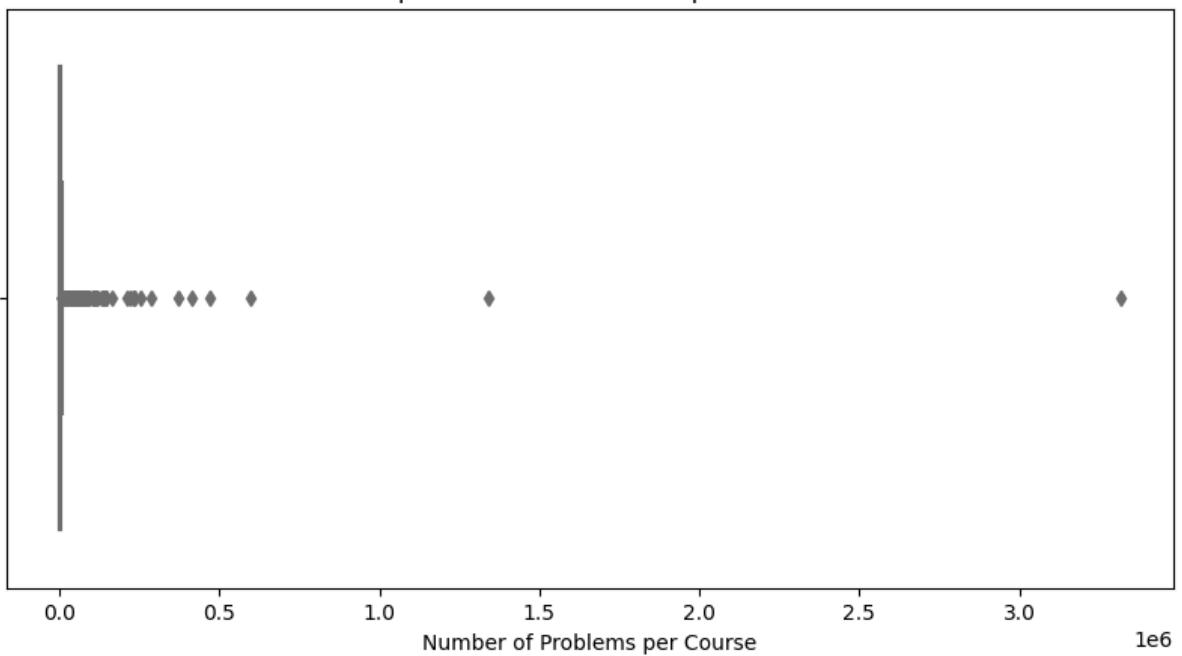
- Các câu trả lời sai ít hơn và không chiếm ưu thế, phản ánh hiệu quả học tập hoặc chất lượng tài liệu giảng dạy khá cao.



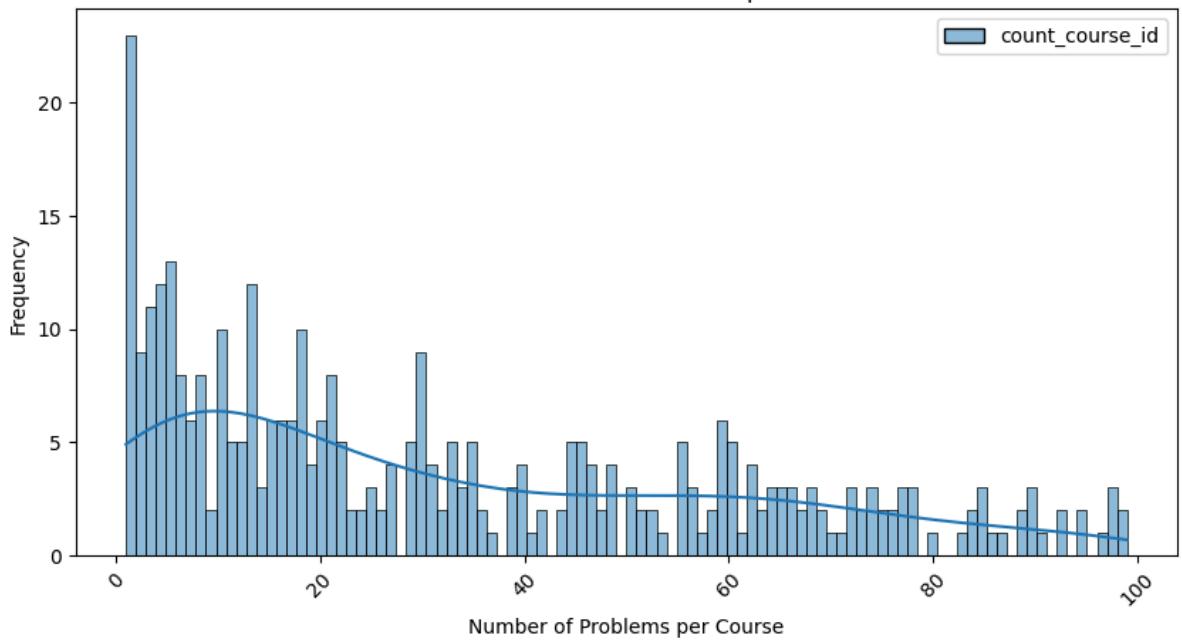
- Loại câu hỏi 1 có số lượng câu trả lời đúng nhiều nhất, cho thấy đây là loại câu hỏi phổ biến hoặc có độ khó phù hợp với người học.
- Loại câu hỏi 6 và loại 2 lần lượt đứng sau, về số lượng câu trả lời đúng. Điều này có thể phản ánh:
 - Các loại này xuất hiện với tần suất ít hơn,
 - Hoặc chúng có mức độ thử thách cao hơn.
- Tất cả câu trả lời cho loại 5 đều được tính là không đúng, có thể do:
 - Cấu trúc đặc biệt của loại 5 (ví dụ: bài kiểm tra, khảo sát không chấm điểm),
 - Hoặc lỗi trong thu thập / xử lý dữ liệu.

Tiếp theo, kết hợp các problem lại để phân tích.

Boxplot of Problem Counts per Course



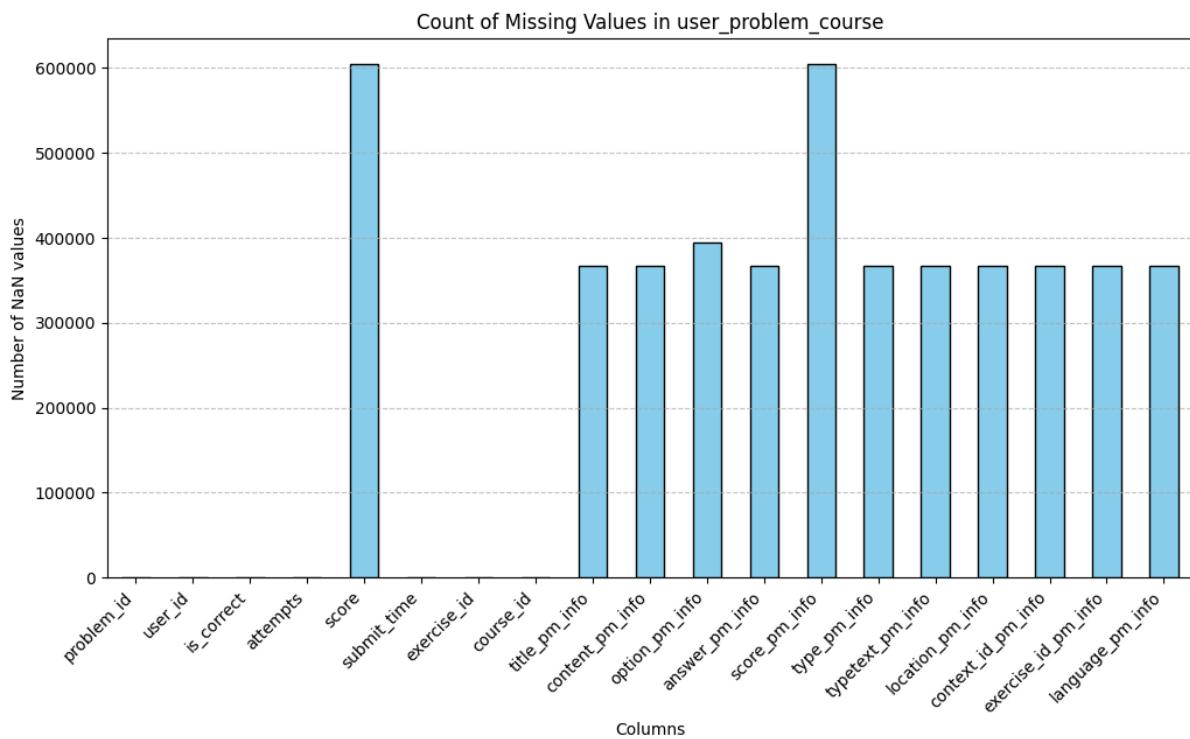
Distribution of Problem Counts per Course



Nhận xét:

- Phần lớn các khóa học có số lượng bài tập trong khoảng từ 1 đến 20.
- Phân phối có xu hướng giảm dần khi số lượng bài tập tăng lên, tức là: Càng nhiều bài tập, thì số khóa học có số lượng đó càng ít.
- Điều này phản ánh rằng phần lớn khóa học được thiết kế với khối lượng bài tập vừa phải, phù hợp với khả năng và thời gian của người học trung bình.

5.2.2.4.2. Điền giá trị từ bảng problem



1. Đọc và chuẩn bị dữ liệu ban đầu:

- Đọc file CSV "user_problem_final.csv" vào DataFrame user_problem.
- Đếm số lượng "context_id" trong cột "context_id_pm_info" và lưu vào cột mới "num_context_ids". Các giá trị rỗng hoặc "[]" được tính là 0.
- Chuyển đổi cột "language_pm_info" thành dạng nhị phân (1 cho 'English', 0 cho 'Chinese' hoặc NaN) và lưu vào cột mới "language_binary".
- Xóa các cột gốc "context_id_pm_info", "language_pm_info", và "user_enroll_time".
- Sắp xếp DataFrame user_problem theo "user_id", "course_id", "exercise_id", và "problem_id".
- Hiển thị các thống kê mô tả cho DataFrame user_problem.
- Tạo biểu đồ tròn hiển thị phân phối của cột "is_correct".

2. Nhóm dữ liệu và tính toán thống kê cấp bài tập (exercise):

- Nhóm DataFrame user_problem theo "user_id", "course_id", và "exercise_id".
- Tính toán các giá trị tổng, đếm, trung bình và danh sách các giá trị cho các cột "is_correct", "attempts", "score", "score_pm_info", "submit_date",

"submit_clock_time", "duration_days", "num_context_ids", và "language_binary". Kết quả được lưu vào DataFrame grouped_lists.

- Làm phẳng tên cột của grouped_lists để dễ truy cập.
- Kết hợp cột ngày ("submit_date_min", "submit_date_max") và giờ ("submit_clock_time_min", "submit_clock_time_max") để tạo đối tượng datetime.
- Tính toán sự khác biệt thời gian giữa thời gian nộp bài sớm nhất và muộn nhất trong mỗi bài tập theo giờ và lưu vào cột "submit_time_diff_hours".
- Trích xuất các giờ nộp bài duy nhất từ danh sách thời gian nộp bài ("submit_clock_time_list") và lưu vào cột "submit_hours_unique".
- Tính toán phần trăm trả lời đúng ("percentage_correct") dựa trên tổng số câu trả lời đúng và tổng số câu trả lời.
- Tính toán phần trăm điểm số ("percentage_score") dựa trên tổng điểm đạt được và tổng điểm tối đa có thể có. Nếu tổng điểm tối đa là 0, sử dụng phần trăm trả lời đúng thay thế.

3. Kết hợp thông tin về bài tập (exercise) và câu hỏi (problem):

- Đọc file văn bản "exercise-problem.txt" (có vẻ như là mối quan hệ giữa bài tập và câu hỏi) vào DataFrame exercise_problem.
- Lọc exercise_problem để chỉ giữ lại các bài tập có trong grouped_lists.
- Tính toán điểm số tối đa cho mỗi câu hỏi từ DataFrame user_problem và lưu vào DataFrame problem_score.
- Kết hợp filtered_exercise_problem với problem_score dựa trên "problem_id".
- Đọc file JSON "problem.json" vào DataFrame problem_info, chỉ giữ lại các cột "problem_id", "score", và "type".
- Định dạng lại cột "problem_id" trong problem_info thêm tiền tố "Pm_".
- Kết hợp DataFrame hiện tại với problem_info dựa trên "problem_id".
- Điền các giá trị thiếu trong cột "score_pm_info" bằng các giá trị từ cột "score".
- Điền các giá trị thiếu còn lại trong cột "score_pm_info" và "score" dựa trên giá trị của cột "type". Gán giá trị điểm số mặc định cho các loại câu hỏi khác nhau.

- Điền tất cả các giá trị NaN còn lại trong DataFrame filtered_exercise_problem_score bằng 1.

- Hiển thị các thông kê mô tả cho DataFrame filtered_exercise_problem_score.

4. Tính toán thống kê hoàn thành bài tập:

- Nhóm filtered_exercise_problem_score theo "exercise_id" và tính toán số lượng câu hỏi ("problem_count") và tổng điểm tối đa ("problem_sum") cho mỗi bài tập. Lưu kết quả vào aggregated_scores.
- Đổi tên cột "exercise_id_" trong grouped_lists thành "exercise_id".
- Kết hợp grouped_lists với aggregated_scores dựa trên "exercise_id".
- Tạo cột mới "is_completed" cho biết bài tập đã hoàn thành hay chưa (1 nếu số câu trả lời đúng bằng hoặc lớn hơn tổng số câu hỏi trong bài tập, 0 ngược lại).
- Tính toán "percentage_completed" (số câu trả lời đúng / tổng số câu hỏi).
- Tính toán "percentage_correct_completed" (tổng số câu trả lời đúng / tổng số câu hỏi).
- Tính toán "percentage_score_completed" (tổng điểm đạt được / tổng điểm tối đa có thể). Nếu tổng điểm tối đa bằng 0, sử dụng "percentage_correct_completed".
- Giới hạn giá trị của "percentage_score_completed" không vượt quá 1.
- Lưu DataFrame grouped_lists cuối cùng vào file CSV "grouped_lists.csv".
- Xóa các cột danh sách gốc ("submit_date_list", "attempts_list", "submit_clock_time_list") từ grouped_lists.

5. Lọc course_detail_problem: Lọc DataFrame course_detail_problem chỉ giữ lại các hàng mà course_id của chúng có trong DataFrame course_limit. Kết quả được lưu vào course_detail_problem_filtered.

6. Tải thông tin người dùng: Tải tệp user_course_final.csv vào DataFrame user_info.

7. Ghép thông tin thời gian đăng ký: Ghép cột user_enroll_time từ user_info vào course_detail_problem_filtered dựa trên user_id và course_id.

8. Kiểm tra lại giá trị thiếu sau khi ghép: Tính lại phần trăm giá trị thiếu cho mỗi cột trong course_detail_problem_filtered và trực quan hóa kết quả.

9. **Tách cột submit_time:** Tách cột submit_time thành hai cột mới: submit_date và submit_clock_time.
10. **Tính toán thời gian hoàn thành (duration_days):** Định nghĩa một hàm để tính số ngày giữa hai ngày và áp dụng hàm này để tạo cột duration_days dựa trên user_enroll_time và submit_date.
11. **Hiển thị tất cả các cột:** Cài đặt tùy chọn hiển thị của pandas để xem tất cả các cột trong DataFrame.
12. **Mô tả dữ liệu khi is_correct là 0:** Hiển thị thống kê mô tả cho DataFrame course_detail_problem_filtered khi cột is_correct có giá trị là 0.
13. **Trực quan hóa phân phối duration_days:** Tạo biểu đồ histogram để xem phân phối của cột duration_days.
14. **Kiểm tra các trường hợp thiểu giá trị đặc biệt:** Lọc và kiểm tra các hàng mà cột score là NaN nhưng score_pm_info thì không, và xem phân phối của type_pm_info trong các trường hợp này bằng biểu đồ hình tròn.
15. **Điền giá trị thiếu cho score:**
 - Điền giá trị cho cột score bằng giá trị của score_pm_info khi score là NaN, score_pm_info không là NaN, và is_correct là 1.
 - Điền giá trị 0 cho cột score khi score là NaN, score_pm_info không là NaN, và is_correct là 0.
16. **Kiểm tra lại thông tin DataFrame và giá trị thiếu:** Hiển thị thông tin về DataFrame và tính toán lại phần trăm giá trị thiếu sau khi điền khuyết.
17. **Kiểm tra các trường hợp thiểu giá trị khác:** Lọc và kiểm tra các hàng mà cột score không là NaN nhưng score_pm_info là NaN, và xem phân phối của type_pm_info (bao gồm cả NaN) trong các trường hợp này bằng biểu đồ cột chồng.
18. **Tìm điểm tối đa cho mỗi vấn đề:** Tìm điểm cao nhất (score) cho mỗi problem_id và lưu kết quả vào DataFrame max_scores.
19. **Cập nhật score_pm_info dựa trên điểm tối đa:**
 - Ghép điểm tối đa (max_score) cho mỗi problem_id vào course_detail_problem_filtered.

- Thay thế giá trị trong cột score_pm_info bằng max_score nếu score_pm_info hiện tại nhỏ hơn max_score.
- Xóa cột max_score sau khi sử dụng.

20. **Kiểm tra lại giá trị thiếu sau khi cập nhật score_pm_info:** Tính toán lại phần trăm giá trị thiếu và trực quan hóa kết quả.

21. **Điền giá trị thiếu cho score (lần 2):** Điền giá trị cho cột score dựa trên giá trị của score_pm_info (đã được cập nhật) và cột is_correct trong các trường hợp score ban đầu là NaN.

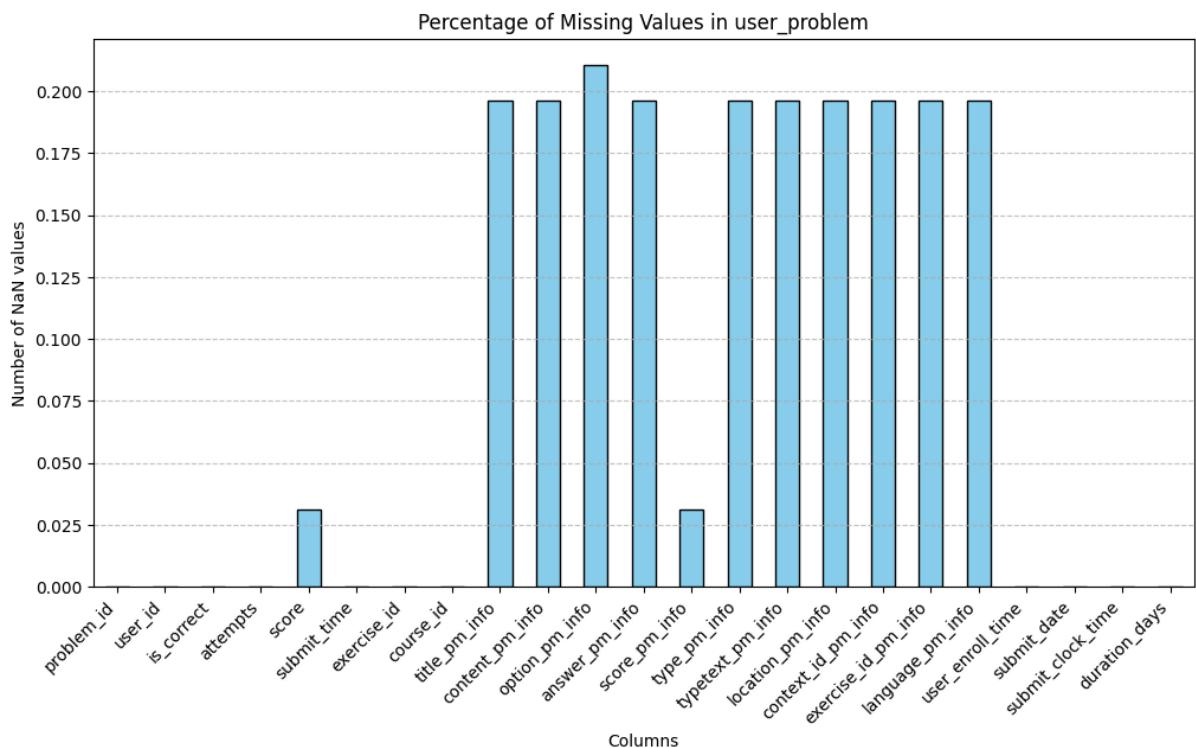
22. **Điền giá trị thiếu cho score và score_pm_info khi cả hai đều NaN:**

- Nếu cả score và score_pm_info đều là NaN và is_correct là 1, điền 1 cho cả hai cột.
- Nếu cả score và score_pm_info đều là NaN và is_correct là 0, điền 0 cho score và 1 cho score_pm_info.

23. **Kiểm tra lại giá trị thiếu lần cuối:** Tính toán lại phần trăm giá trị thiếu và trực quan hóa kết quả với chú thích phần trăm trên các cột.

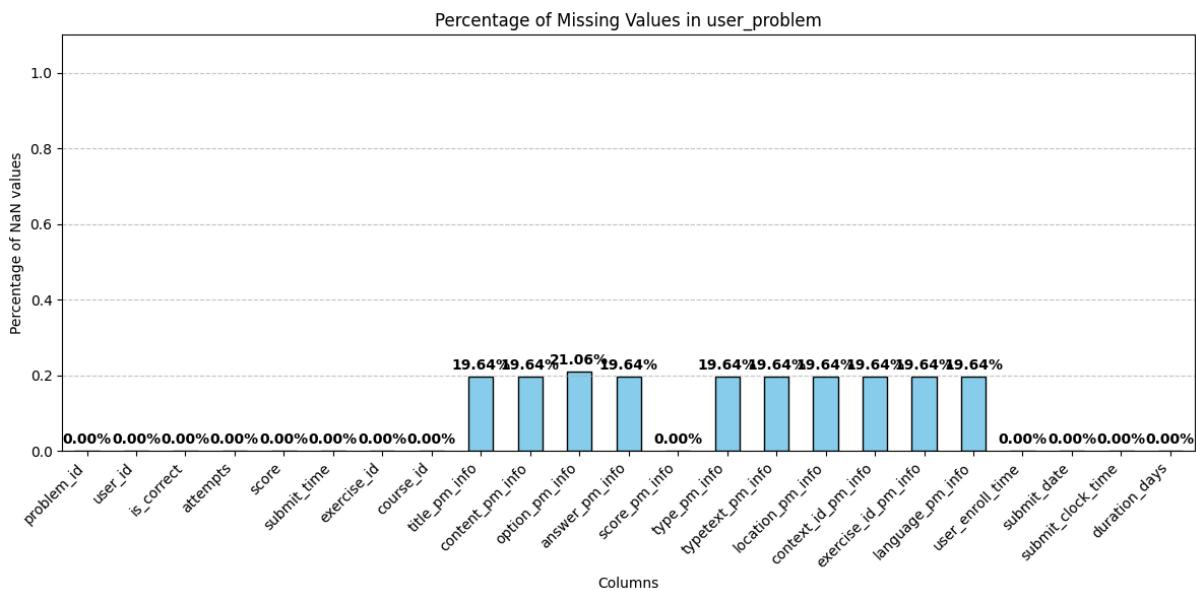
24. **Xóa các cột không cần thiết:** Xóa một số cột khỏi DataFrame course_detail_problem_filtered.

25. **Cập nhật lại score_pm_info dựa trên điểm tối đa (lần 2):** Lặp lại bước tìm điểm tối đa và cập nhật score_pm_info nếu nó nhỏ hơn điểm tối đa cho cùng problem_id.

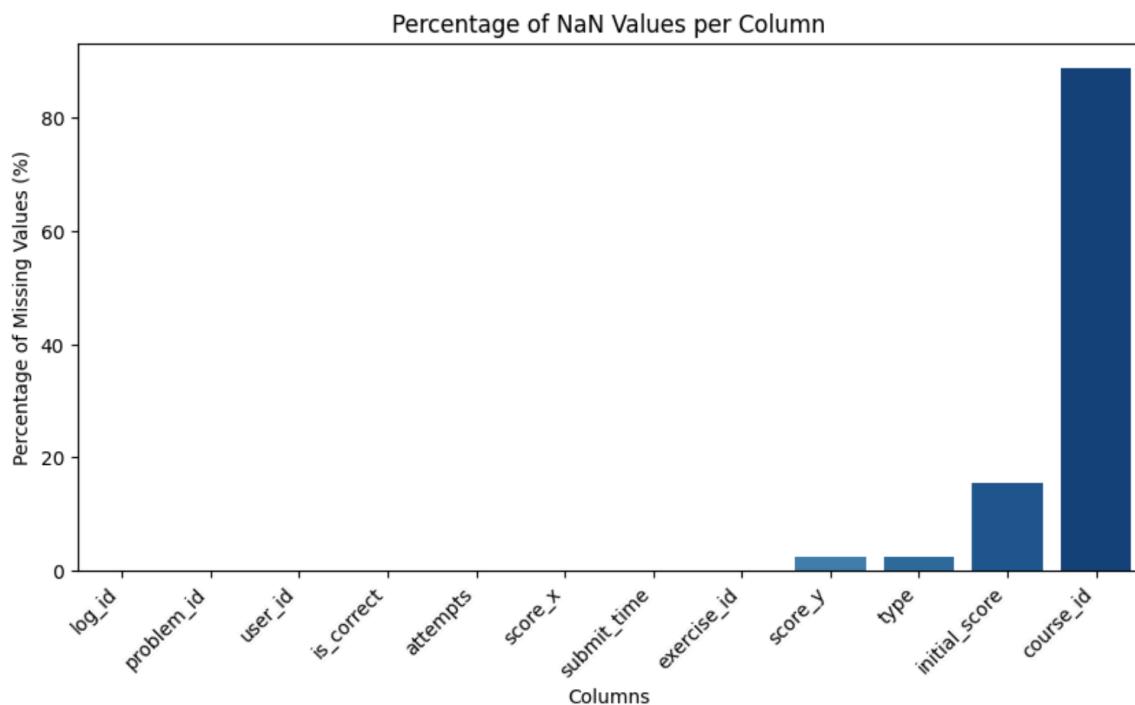


Kết quả sau khi xử lý không còn score của problem nào bị NaN.

Các trường bị thiếu có thể không cần trong việc huấn luyện model dự đoán kết quả học tập.



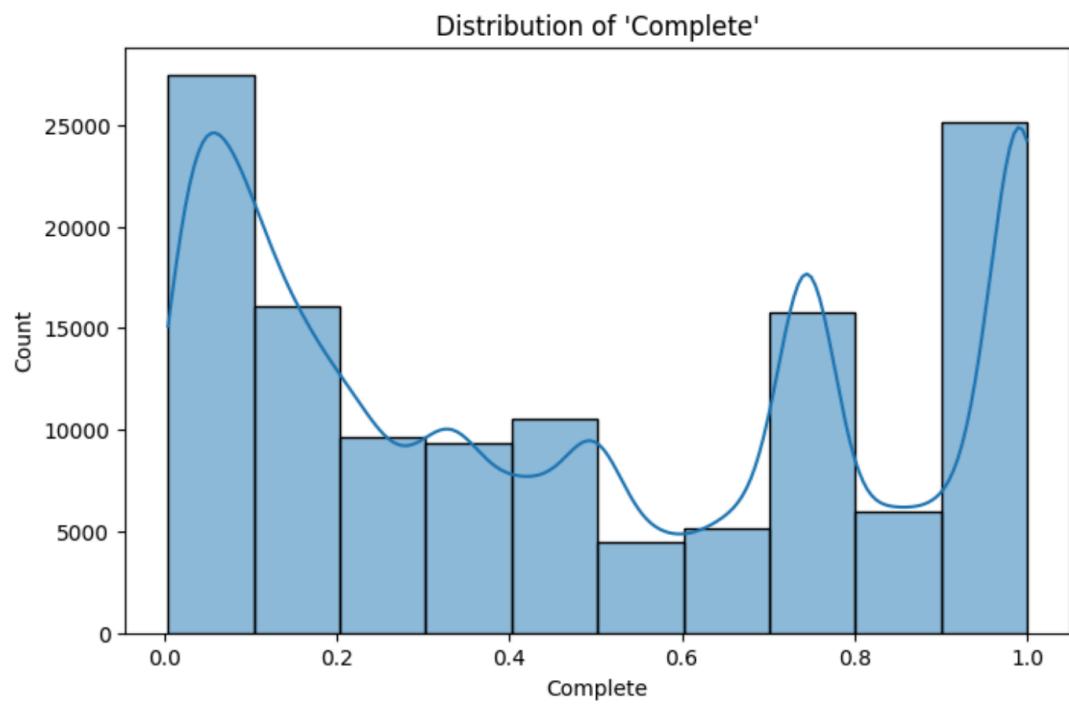
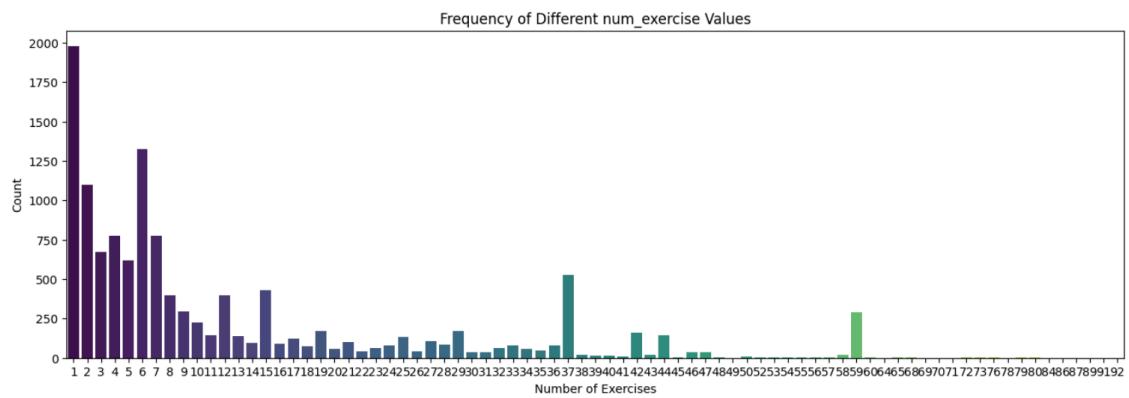
5.2.2.4.3. Gộp dữ liệu user-problem với exercise và course đã được xử lý

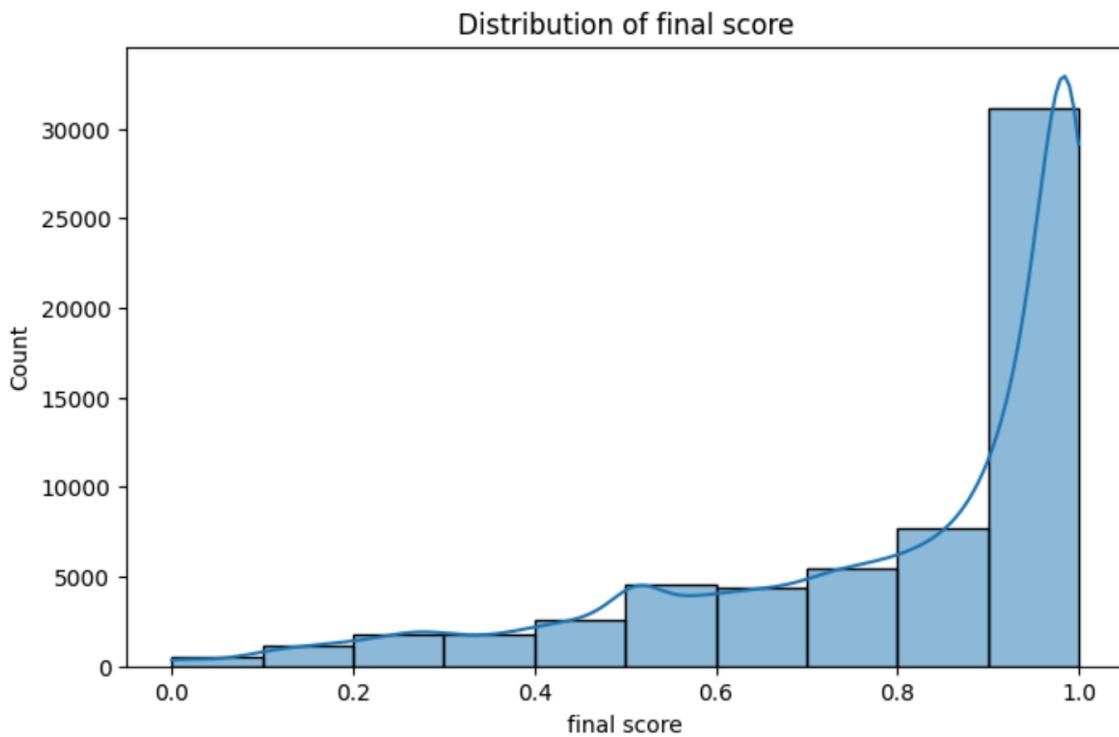


5.2.2.4.3. Gom nhóm theo user_id và course_id

```
] :  
    user_course_summary = user_course_summary.groupby(['user_id', 'course_id']).agg({  
        'exercise_id' : list,  
        'problem_id': list,  
        'is_correct': list,  
        'attempts': list,  
        'score_x': list,  
        'score_y': list,  
        'submit_time': list  
    }).reset_index()  
    user_course_summary
```

Số lượng bài tập của user





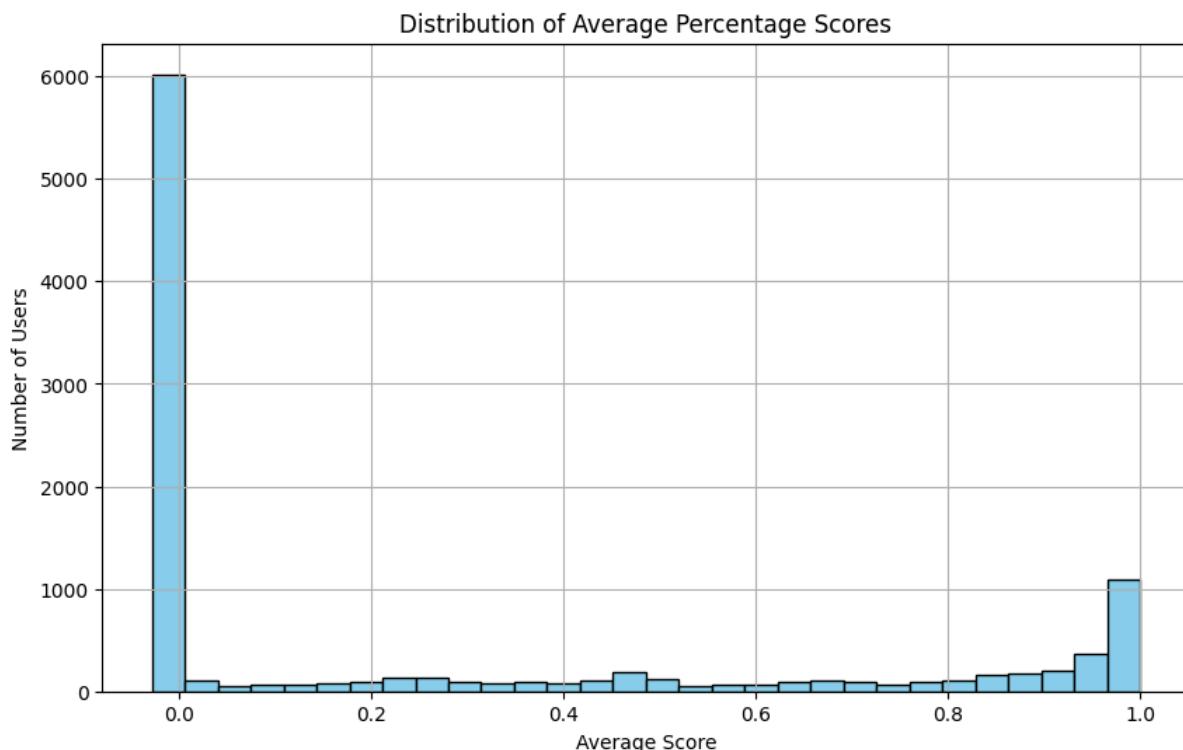
5.2.2.4.5. Phân loại những exercise là bài tập hay bài kiểm tra và tính điểm

Exercise là bài tập

1. Tải dữ liệu người dùng và bài tập: Tải tập dữ liệu grouped_lists.csv vào một DataFrame có tên user_exercise.
 - Đổi tên cột: đổi tên các cột user_id_ và course_id_ trong DataFrame user_exercise thành user_id và course_id.
 - Tải dữ liệu khóa học và bài kiểm tra: Tải tập dữ liệu final_course_exam.csv vào một DataFrame có tên course_exam.
 - Loại bỏ cột không cần thiết: Loại bỏ cột 'Unnamed: 0' khỏi DataFrame course_exam.
 - Đổi tên cột: Đổi tên cột 'id' trong DataFrame course_exam thành course_id.
2. Kết hợp dữ liệu người dùng và bài kiểm tra: Kết hợp DataFrame user_exercise (chỉ giữ lại user_id và course_id) với DataFrame course_exam dựa trên cột course_id. Kết quả được lưu vào DataFrame user_exam.
3. Xử lý các khóa học không có bài kiểm tra cuối kỳ: Loại bỏ các hàng khỏi DataFrame user_exam nơi cột 'exam_resources' (danh sách các bài tập kiểm tra cuối kỳ) bị thiếu (NaN).
 - Loại bỏ các hàng trùng lặp: Loại bỏ các hàng trùng lặp khỏi DataFrame user_exam.
 - Thiết lập lại chỉ mục: Thiết lập lại chỉ mục của DataFrame user_exam.

- Kiểm tra dữ liệu cụ thể: Kiểm tra các hàng trong DataFrame user_exam cho một user_id và course_id cụ thể.
 - Đếm giá trị thiếu: Tính toán và in ra số lượng và tỷ lệ phần trăm các giá trị thiếu trong cột 'exam_resources' của DataFrame user_exam.
 - Tách danh sách bài tập kiểm tra: Xử lý cột 'exam_resources' trong DataFrame user_exam. Chuyển chuỗi đại diện cho danh sách thành danh sách Python và sau đó tách mỗi mục trong danh sách thành một hàng riêng biệt, kết quả là DataFrame exploded. Cột mới được đặt tên là exercise_id.
4. Kết hợp điểm số bài tập: Kết hợp DataFrame exploded với các cột user_id, course_id, exercise_id và percentage_score_completed từ DataFrame user_exercise dựa trên các cột chung. Kết quả được lưu vào DataFrame merged.
- Kiểm tra dữ liệu kết hợp cụ thể: Kiểm tra các hàng trong DataFrame merged cho một user_id và course_id cụ thể.
 - Nhóm điểm số: Nhóm DataFrame merged theo user_id và course_id, tập hợp các giá trị 'percentage_score_completed' thành một danh sách cho mỗi nhóm. Kết quả được lưu vào DataFrame grouped_scores.
 - Kiểm tra dữ liệu được nhóm: Hiển thị các hàng đầu tiên và một mục cụ thể từ DataFrame grouped_scores.
 - Kết hợp điểm số được nhóm trở lại: Kết hợp DataFrame user_exam với DataFrame grouped_scores dựa trên user_id và course_id. Kết quả được lưu vào DataFrame user_exam_with_scores.
 - Xử lý an toàn các danh sách (nếu được lưu trữ dưới dạng chuỗi): Xảm bảo rằng các cột 'percentage_score_completed' và 'exam_resources' là danh sách (chuyển đổi từ chuỗi nếu cần).
5. **Tính toán điểm cuối cùng:** Định nghĩa một hàm để tính điểm cuối cùng cho mỗi hàng. Hàm này kiểm tra xem có danh sách 'exam_resources' hợp lệ hay không.
- Nếu có, nó tính điểm trung bình của các 'percentage_score_completed' tương ứng (thay thế NaN bằng 0). Nếu không có 'exam_resources' hợp lệ hoặc danh sách 'exam_resources' trống, nó trả về NaN. Áp dụng hàm này vào DataFrame user_exam_with_scores để tạo cột 'final_score'.
6. Kiểm tra giá trị thiếu trong điểm cuối cùng: Tính toán và in ra số lượng và tỷ lệ phần trăm các giá trị thiếu trong cột 'final_score'.
7. Kiểm tra phân phối điểm cuối cùng: Tính toán và in ra số lần xuất hiện của các giá trị khác nhau trong cột 'final_score'.
8. Trực quan hóa phân phối điểm: Tạo một biểu đồ tần suất của cột 'final_score' để trực quan hóa phân phối điểm trung bình.

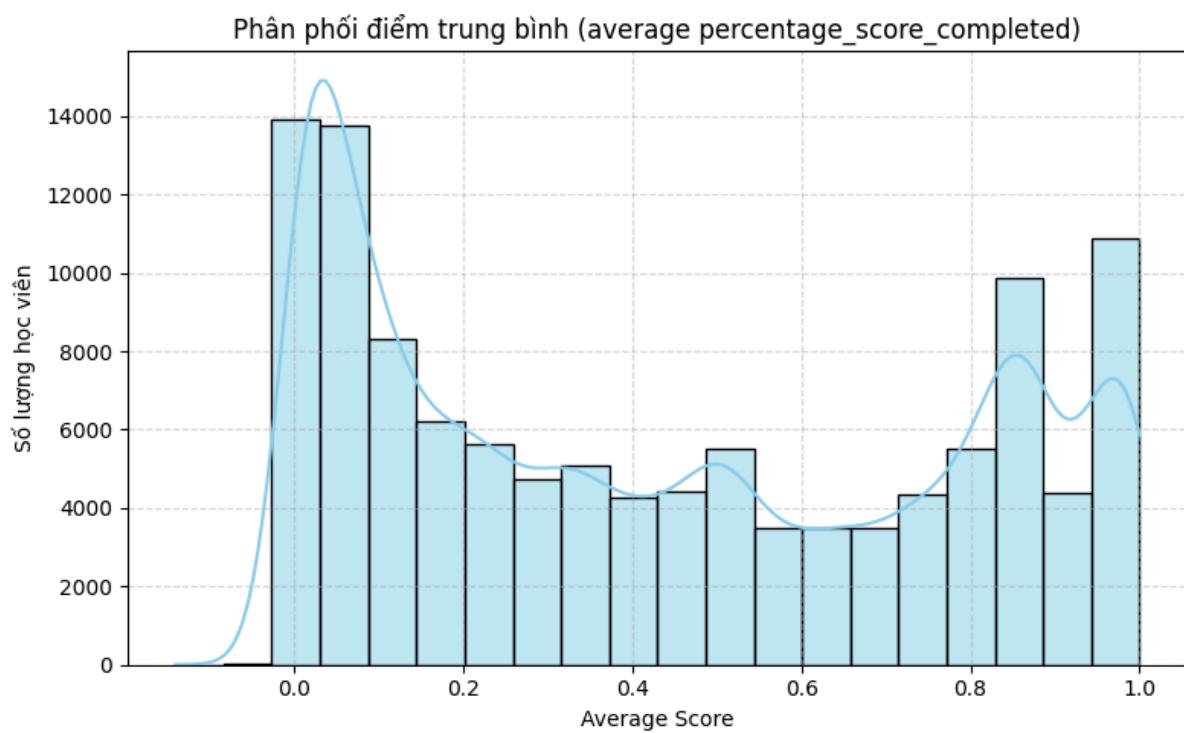
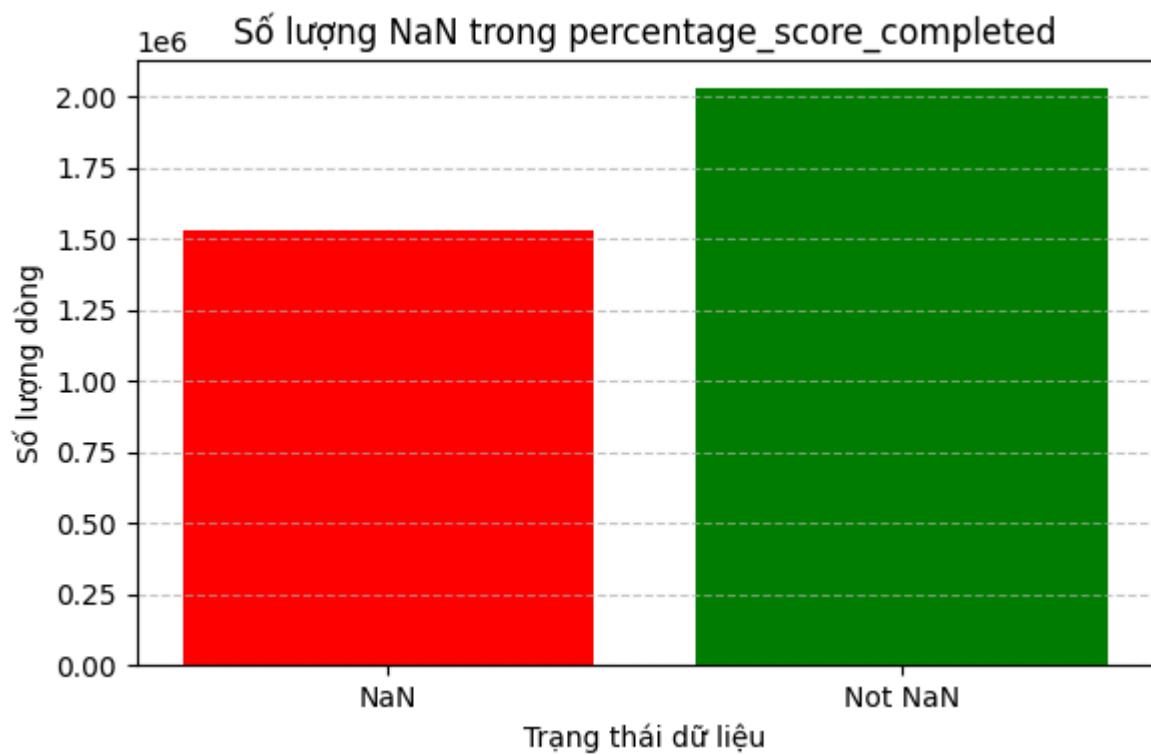
9. Lưu kết quả điểm người dùng: Lưu DataFrame user_exam_with_scores vào tệp CSV có tên user_score.csv.
10. Xác định các bài tập không phải bài kiểm tra cuối kỳ: Xử lý DataFrame course_exam để tạo một danh sách các exercise_id tương ứng với các bài kiểm tra cuối kỳ. Sau đó, lọc DataFrame user_exercise để giữ lại chỉ những hàng mà exercise_id của chúng không nằm trong danh sách các bài tập kiểm tra cuối kỳ. Kết quả được lưu vào DataFrame filtered_user_exercise.



Exercise là bài tập

1. **Đọc dữ liệu ban đầu:** Bạn đã đọc hai tệp CSV: course_resource_new_id.csv và final_course_exam.csv vào các DataFrame Pandas tương ứng là course_resource và course_with_exam. Đọc tệp grouped_lists.csv vào DataFrame user_exercise.
2. **Lọc các bài tập (exercises):** Từ DataFrame course_resource, lọc ra các hàng mà cột resource_id bắt đầu bằng 'Ex_' và lưu kết quả vào DataFrame mới course_resource_ex.
3. **Xử lý dữ liệu kỳ thi:**
 - Chuyển đổi cột exam_resources trong DataFrame course_with_exam từ chuỗi biểu diễn danh sách sang danh sách thực tế bằng cách sử dụng ast.literal_eval.
 - Bạn đã "bung" (explode) DataFrame course_with_exam dựa trên cột exam_resources để mỗi hàng chứa một ID tài nguyên kỳ thi.

4. **Loại bỏ các bài tập cuối khóa:** Loại bỏ các bài tập là kỳ thi cuối khóa khỏi DataFrame course_resource_ex bằng cách kiểm tra xem resource_id có nằm trong danh sách exam_resources của course_with_exam hay không. Kết quả được lưu vào course_resource_ex_no_final.
5. **Chuẩn bị dữ liệu người dùng và bài tập:**
 - Tạo một DataFrame mới user_exercise_course chỉ chứa các cột user_id và course_id từ user_exercise và loại bỏ các hàng trùng lặp.
 - Đổi tên các cột trong user_exercise_course và course_resource_ex_no_final để thống nhất tên cột trước khi gộp.
6. **Gộp dữ liệu người dùng, khóa học và bài tập:** Gộp DataFrame user_exercise_course với course_resource_ex_no_final dựa trên cột course_id để liên kết mỗi người dùng với tất cả các bài tập trong khóa học mà họ đã tham gia. Kết quả được lưu vào user_course_exercise.
7. **Thêm điểm hoàn thành:** Gộp DataFrame user_course_exercise với DataFrame user_exercise ban đầu (sau khi đổi tên cột) dựa trên course_id, user_id và exercise_id để thêm cột percentage_score_completed vào user_course_exercise. Sử dụng gộp trái (how='left') để giữ lại tất cả các hàng từ user_course_exercise.
8. **Xử lý giá trị thiếu (NaN) và trực quan hóa:**
 - Tính toán và trực quan hóa số lượng giá trị thiếu (NaN) và không thiếu trong cột percentage_score_completed bằng biểu đồ cột.
 - Điền giá trị 0 vào tất cả các giá trị thiếu (NaN) trong cột percentage_score_completed.
9. **Nhóm và tính điểm trung bình:**
 - Nhóm DataFrame user_course_exercise theo user_id và course_id.
 - Đổi với mỗi nhóm, bạn đã tổng hợp danh sách exercise_id và danh sách percentage_score_completed.
 - Tính điểm trung bình (avg_score) cho mỗi người dùng trong mỗi khóa học dựa trên danh sách percentage_score_completed. Kết quả được lưu vào DataFrame grouped_df.
10. **Trực quan hóa phân phối điểm trung bình:** Vẽ biểu đồ histogram với đường KDE (ước lượng mật độ hạt nhân) để hiển thị phân phối của avg_score.



5.2.2.4.6. Chia thành các phase cho quá trình huấn luyện

1. **Tải dữ liệu:** Bạn đã đọc dữ liệu từ tệp CSV có tên [/kaggle/input/score-final-exam/exercise_without_final_exam.csv](#) vào một DataFrame của pandas.
2. **Xóa cột không cần thiết:** Loại bỏ một số cột khỏi DataFrame ban đầu.

3. **Đổi tên cột:** Đổi tên nhiều cột trong DataFrame để dễ hiểu hơn.
4. **Phân chia dữ liệu theo gai đoạn:** Lọc dữ liệu thành bốn gai đoạn dựa trên giá trị của cột 'exercise_date_from_enroll':
 - Gai đoạn 1: exercise_date_from_enroll nhỏ hơn hoặc bằng 14.
 - Gai đoạn 2: exercise_date_from_enroll lớn hơn 14 và nhỏ hơn hoặc bằng 28.
 - Gai đoạn 3: exercise_date_from_enroll lớn hơn 28 và nhỏ hơn hoặc bằng 42.
 - Gai đoạn 4: exercise_date_from_enroll lớn hơn 42 và nhỏ hơn hoặc bằng 56.
5. **Xử lý cột 'exercise_hours':** Đổi với mỗi gai đoạn, chuyển đổi cột 'exercise_hours' từ chuỗi sang danh sách (list) bằng cách sử dụng ast.literal_eval.
6. **Nhóm dữ liệu và tổng hợp:** Đổi với mỗi gai đoạn, nhóm dữ liệu theo 'user_id' và 'course_id' và thực hiện các phép tổng hợp (như đếm, tính tổng, trung bình, min, max, độ lệch chuẩn) trên nhiều cột khác nhau. Tạo một danh sách tổng hợp các giờ trong cột 'exercise_hours' cho mỗi nhóm.
7. **Làm phẳng tên cột:** Sau khi nhóm dữ liệu, làm phẳng tên các cột đa cấp được tạo ra từ phép tổng hợp.
8. **Đổi tên cột tổng hợp 'exercise_hours':** Đổi tên cột tổng hợp của 'exercise_hours' cho rõ ràng hơn.
9. **Tính Entropy:** Tính giá trị entropy dựa trên danh sách giờ trong cột 'exercise_hours' cho mỗi nhóm, sử dụng hàm compute_entropy_from_hour_bins.
10. **Loại bỏ cột 'exercise_hours':** Bạn đã loại bỏ cột 'exercise_hours' sau khi tính toán entropy.

5.2.2.4.7. Tổng kết các đặc trưng

Tên trường	Mô tả	Miền giá trị
user_id	ID người dùng (học viên)	Chuỗi ký tự, ví dụ: U_10033064
course_id	ID khóa học	Chuỗi ký tự, ví dụ: C_1822804
exercise_id_count	Số lượng bài tập mà học viên đã tương tác	Số nguyên ≥ 0

exercise_correct_sum	Tổng số bài tập học viên làm đúng	Số nguyên ≥ 0
exercise_correct_mean	Tỷ lệ đúng trung bình (trên mỗi bài tập)	0 – 1
exercise_num_problem_sum	Tổng số câu hỏi trong các bài tập	Số nguyên ≥ 0
exercise_num_problem_mean	Số câu hỏi trung bình mỗi bài tập	≥ 0
exercise_attempts_sum_sum	Tổng số lượt làm bài (mọi lần)	Số nguyên ≥ 0
exercise_attempts_sum_mean	Trung bình số lượt làm bài mỗi bài tập	≥ 0
exercise_attempts_mean_mean	Trung bình số lượt làm mỗi câu hỏi	≥ 0
exercise_date_from_enrol1_min	Ngày làm bài sớm nhất (tính từ ngày đăng ký)	Số nguyên ≥ 0 (đơn vị: ngày)
exercise_date_from_enrol1_mean	Trung bình số ngày kể từ khi đăng ký đến khi làm bài	≥ 0
exercise_date_from_enrol1_max	Ngày làm bài trễ nhất (tính từ ngày đăng ký)	≥ 0
exercise_context_sum	Tổng số "bối cảnh" bài tập (ví dụ: theo chương hoặc chủ đề)	Số thực ≥ 0
exercise_context_mean	Giá trị trung bình "bối cảnh"	≥ 0
exercise_language_binary_mean	Trung bình nhị phân về ngôn ngữ (ví dụ: 0 = không hỗ trợ, 1 = có)	0 hoặc 1
exercise_diff_sum	Tổng độ khó của các bài tập	≥ 0
exercise_diff_mean	Trung bình độ khó của các bài tập	≥ 0
exercise_diff_min	Độ khó nhỏ nhất trong các bài tập	≥ 0
exercise_diff_max	Độ khó cao nhất trong các bài tập	≥ 0

exercise_perc_goal_correct_sum	Tổng phần trăm đúng theo mục tiêu của bài tập	0 – 1
exercise_perc_goal_correct_mean	Trung bình phần trăm đúng theo mục tiêu	0 – 1
exercise_perc_goal_score_sum	Tổng điểm theo mục tiêu bài tập	0 – 1
exercise_perc_goal_score_mean	Trung bình điểm theo mục tiêu	0 – 1
exercise_perc_real_completed_sum	Tổng phần trăm hoàn thành thực tế	0 – 1
exercise_perc_real_completed_mean	Trung bình phần trăm hoàn thành thực tế	0 – 1
exercise_perc_real_completed_std	Độ lệch chuẩn phần trăm hoàn thành thực tế	≥ 0
exercise_perc_real_correct_sum	Tổng phần trăm làm đúng thực tế	0 – 1
exercise_perc_real_correct_mean	Trung bình phần trăm làm đúng thực tế	0 – 1
exercise_perc_real_correct_std	Độ lệch chuẩn phần trăm làm đúng thực tế	≥ 0
exercise_perc_real_score_sum	Tổng điểm thực tế đạt được	0 – 1
exercise_perc_real_score_mean	Trung bình điểm thực tế đạt được	0 – 1
exercise_perc_real_score_std	Độ lệch chuẩn điểm thực tế	≥ 0
exercise_hour_entropy	Độ phân tán thời gian làm bài (entropy theo giờ)	Số thực, thường từ 0 đến ~2

5.2.2.5. Xử lý bộ dữ liệu relation/user-video.json

5.2.2.5.1. Thông kê các trường dữ liệu

Sử dụng hàm shape để xem kích thước của một DataFrame (số hàng, số cột).

Sử dụng hàm schema để cung cấp một bản tóm tắt ngắn gọn của DataFrame (tên cột, kiểu dữ liệu, số lượng giá trị không bị thiếu).

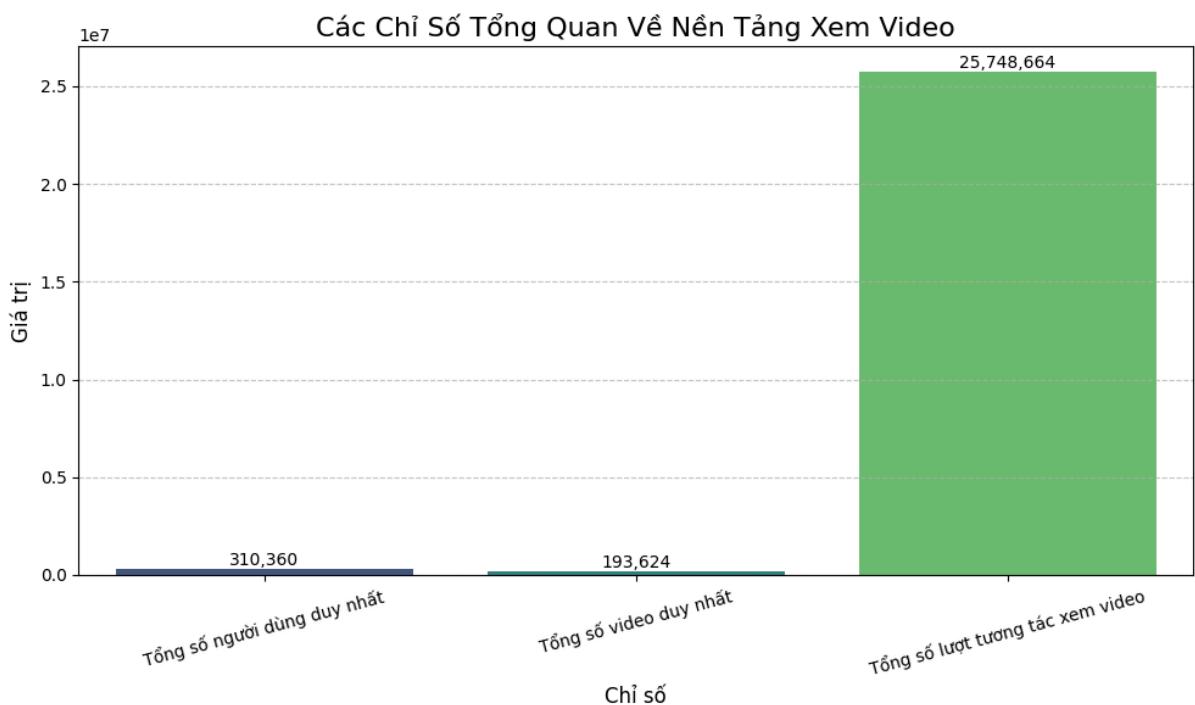
Khảo sát về các cột trong bộ dữ liệu.

```
Schema([('user_id', String), ('video_id', String), ('start_point', Float64), ('end_point', Float64), ('playback_speed', Float64), ('timestamp', Int64), ('datetime', Datetime(time_unit='us', time_zone=None))])  
shape: (9, 8)
```

statistic	user_id	video_id	start_point	end_point	playback_speed	timestamp	datetime
count	25748664	25748664	2.5748664e7	2.5748664e7	2.5748664e7	2.5748664e7	25748664
null_count	0	0	0.0	0.0	0.0	0.0	0
mean	null	null	448.052778	466.846236	1.177257	1.5998e9	2020-09-11 00:47:39.731743
std	null	null	631.256707	2372.182958	0.366499	6.1924e6	null
min	U_10001181	V_1000004	0.0	-2.0239e6	0.5	1.0098e9	2001-12-31 16:19:57
25%	null	null	144.7	170.0	1.0	1.5989e9	2020-09-01 04:18:09
50%	null	null	332.0	356.0	1.0	1.6017e9	2020-10-02 20:02:16
75%	null	null	585.0	606.0	1.0	1.6036e9	2020-10-24 23:42:47
max	U_9999820	V_999993	69209.6	48921.0	2.0	3.4855e9	2080-06-13

5.2.2.5.2. Phân tích Tổng quan về quy mô

Đếm tổng số lượng user_id và video_id riêng biệt, cùng với tổng số hàng (luợt tương tác segment) trong DataFrame đã làm phẳng.

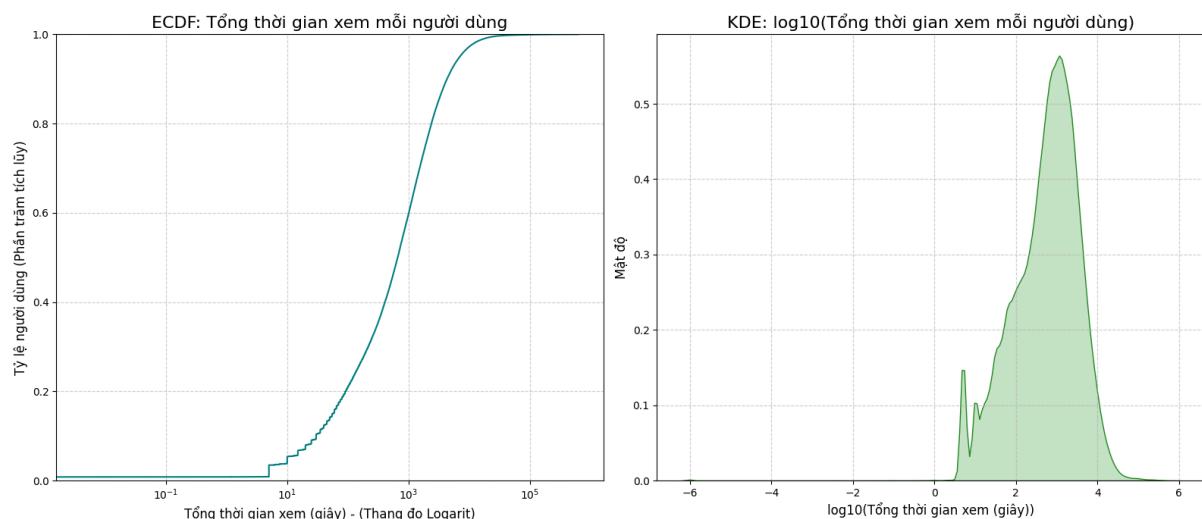


Nhận xét kết quả:

- Nền tảng có một cơ sở người dùng rất lớn với hơn 310 nghìn học viên.
- Số lượng video cung cấp cũng rất đa dạng, gần 200 nghìn video khác nhau.
- Tổng số lượt tương tác xem video (segment) là cực kỳ lớn, cho thấy mức độ hoạt động và sử dụng nội dung video cao trên toàn hệ thống. Điều này ngũ ý rằng mỗi học viên trung bình xem và tương tác với nhiều đoạn video.

5.2.2.5.3. Tổng thời gian xem mỗi người dùng

Tính toán thời lượng của từng đoạn xem (end_point - start_point) thành cột segment_duration_seconds, sau đó nhóm dữ liệu theo user_id và tính tổng thời gian xem cho mỗi người dùng, sắp xếp giảm dần.

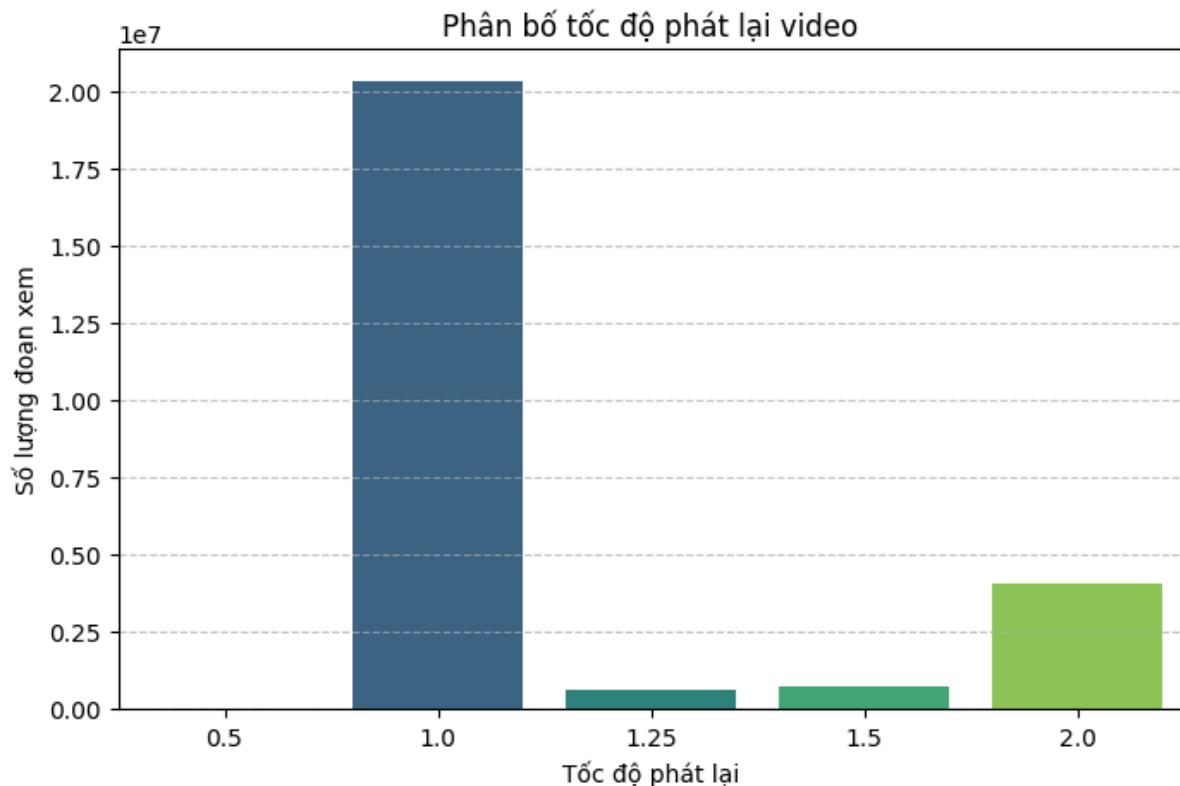


Nhận xét kết quả:

- Có sự chênh lệch rất lớn về tổng thời gian xem giữa các học viên. Top 5 người dùng hàng đầu có tổng thời gian xem lên tới hàng trăm nghìn giây (tương đương hàng trăm giờ).
- Điều này cho thấy có một nhóm học viên cốt lõi cực kỳ tích cực và dành một lượng lớn thời gian tương tác với nội dung video.
- Việc xác định được những "người dùng siêu tích cực" này rất quan trọng để hiểu hành vi và có thể là cơ sở cho các chương trình khen thưởng hoặc nghiên cứu sâu hơn.

5.2.2.5.4. Phân bố tốc độ phát lại video

Nhóm dữ liệu theo playback_speed và đếm số lượng đoạn xem (num_segments) cho mỗi tốc độ phát lại.



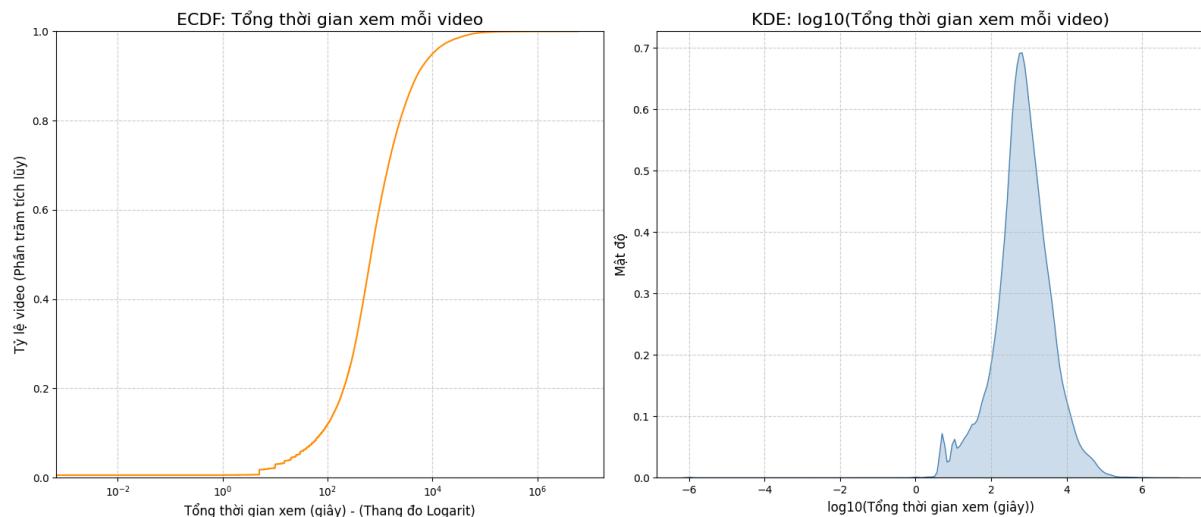
Nhận xét kết quả:

- Tốc độ 1.0 (tốc độ bình thường) là chế độ xem áp đảo với hơn 20 triệu đoạn. Điều này cho thấy đa số học viên ưa thích hoặc cảm thấy thoải mái nhất khi xem video ở tốc độ gốc.

- Tốc độ 2.0 (nhanh gấp đôi) cũng rất phổ biến, với hơn 4 triệu đoạn. Điều này chỉ ra rằng một lượng đáng kể học viên sử dụng tốc độ nhanh hơn để tiết kiệm thời gian, ôn tập nhanh, hoặc họ đã quen với nội dung.
- Các tốc độ khác như 1.25, 1.5, và 0.5 ít được sử dụng hơn, nhưng vẫn có một số lượng đáng kể tương tác, cho thấy sự linh hoạt trong thói quen học tập của người dùng.
- Việc hiểu phân bố tốc độ giúp tối ưu hóa nội dung video và có thể gợi ý các tính năng hỗ trợ người dùng tốt hơn.

5.2.2.5.5. Tổng thời gian xem mỗi video

nhóm dữ liệu theo video_id và tính tổng thời gian xem của mỗi video, sắp xếp giảm dần.



Nhận xét kết quả:

- Giống như người dùng, có những video cực kỳ phổ biến được xem với tổng thời lượng rất lớn (ví dụ: V_1395633 có hơn 6 triệu giây xem).
- Các video này có thể là nội dung cốt lõi, phần giới thiệu quan trọng, hoặc các bài giảng hấp dẫn nhất, thu hút lượng lớn người xem và tương tác.
- Việc xác định các video "hot" này giúp đội ngũ phát triển nội dung hiểu được loại tài liệu nào đang thu hút và giữ chân học viên hiệu quả nhất.

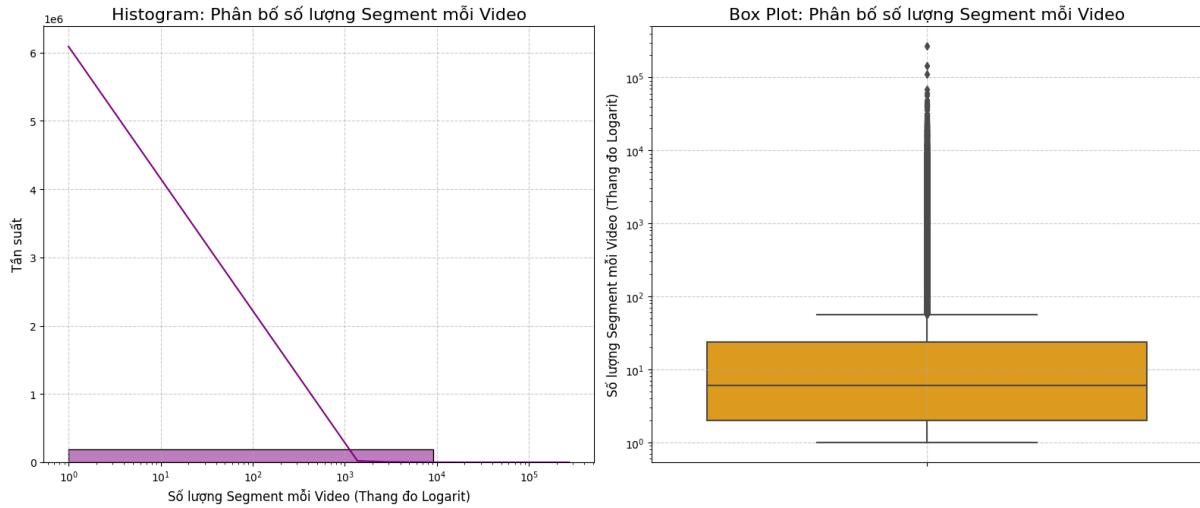
5.2.2.5.6. Số lượng segment trung bình mỗi video

Nhóm dữ liệu theo video_id, đếm số lượng segment cho mỗi video, sau đó tính trung bình các số đếm này.

Nhận xét kết quả:

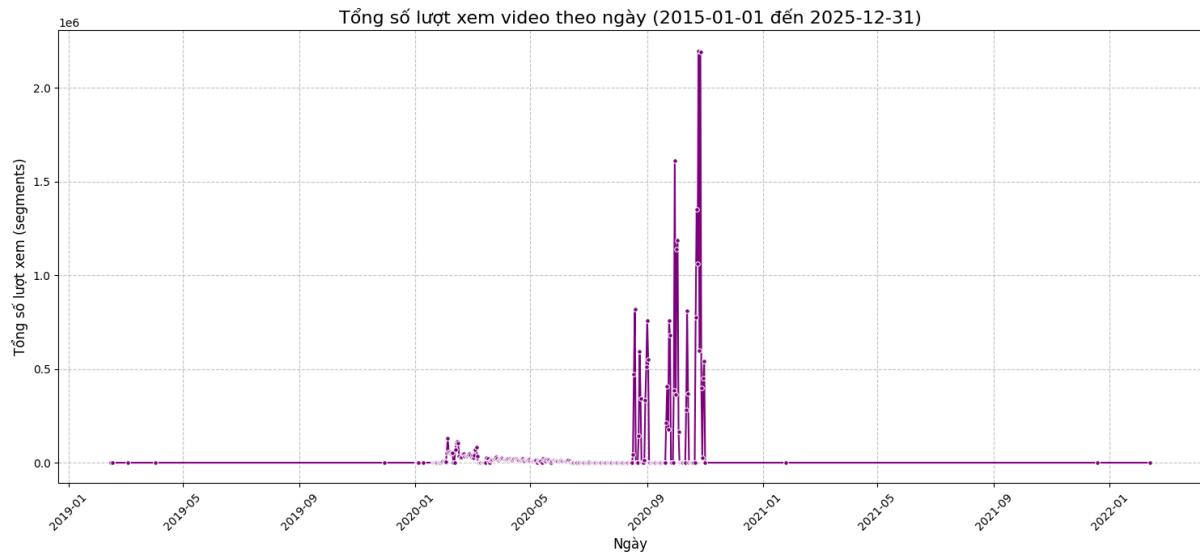
- Trung bình mỗi video có khoảng 132.98 đoạn xem (segments).

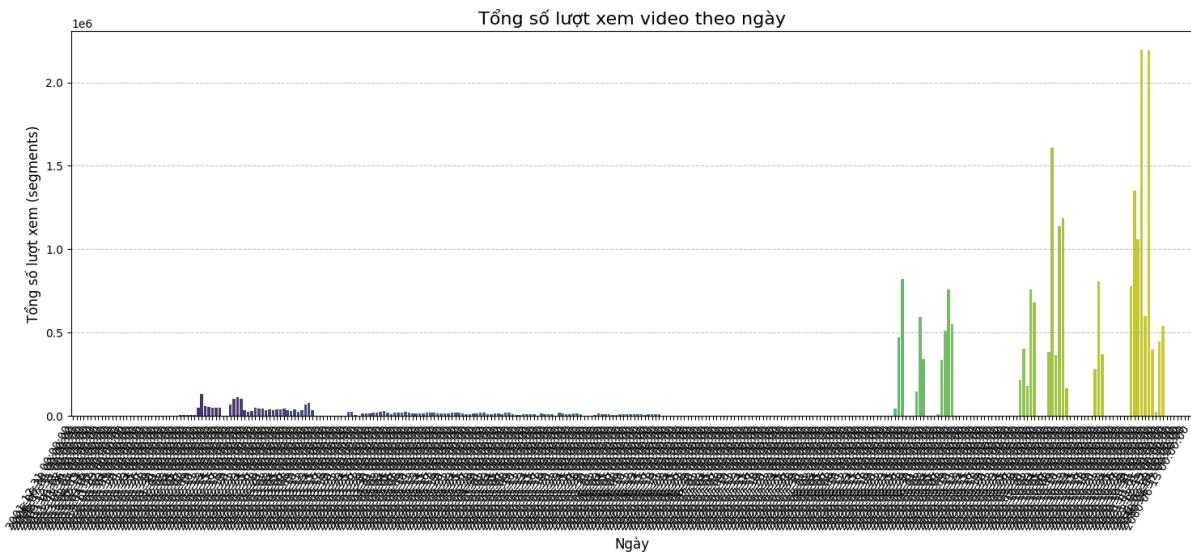
- Điều này cho thấy một video duy nhất thường được xem và tương tác với nhiều đoạn nhỏ. Điều này có thể xảy ra do:
- Học viên tua đi tua lại một phần video.
- Video được chia thành các phần nhỏ logic.
- Học viên dừng và tiếp tục xem nhiều lần.
- Con số này ngũ ý rằng các video không chỉ được xem từ đầu đến cuối mà còn được tương tác một cách không liên tục, với các lần tạm dừng, tua lại hoặc tua tới để tập trung vào các phần cụ thể.



5.2.2.5.7. Tổng số lượt xem theo ngày

Trích xuất ngày từ cột datetime, sau đó nhóm dữ liệu theo ngày và tính tổng số lượt xem (segments) cho mỗi ngày, sắp xếp theo ngày.





Nhận xét kết quả:

- Các ngày được hiển thị có tổng lượt xem tương đối thấp . Điều này có thể cho thấy:
- Đây chỉ là các điểm dữ liệu rất cũ hoặc khởi đầu của nền tảng, nơi hoạt động còn hạn chế.
- Hoạt động học tập chưa tập trung vào những ngày cụ thể này.
- Để có cái nhìn đầy đủ hơn về xu hướng theo thời gian, cần phân tích dữ liệu cho toàn bộ khoảng thời gian, không chỉ top 5, và có thể vẽ biểu đồ theo chuỗi thời gian để thấy được đỉnh điểm và thung lũng của hoạt động.

5.2.2.5.8. Top 10 video được xem nhiều segment nhất

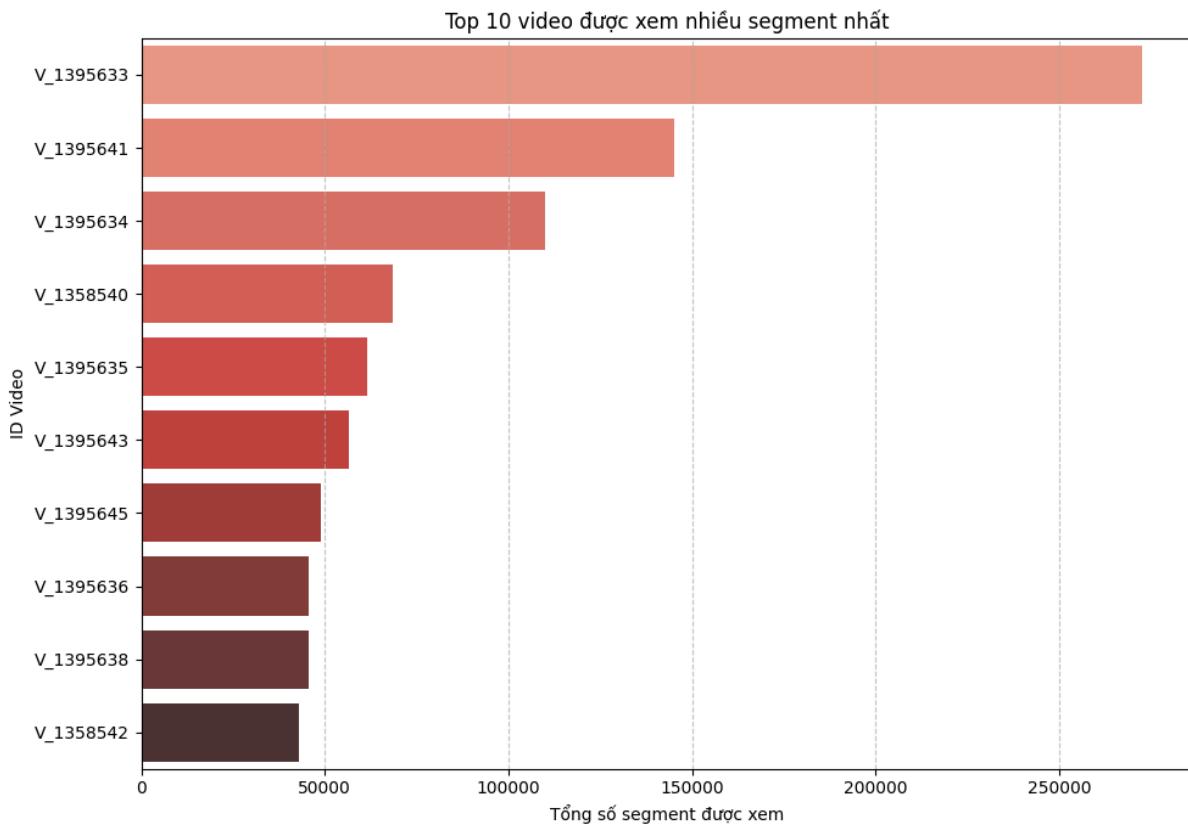
Nhóm dữ liệu theo video_id, đếm tổng số segment được xem cho mỗi video và hiển thị top 10 video có số segment cao nhất.

Nhận xét kết quả:

V_1395633 là video nổi bật nhất với hơn 272 nghìn segment được xem, tiếp theo là V_1395641 với hơn 145 nghìn.

Điều này xác nhận lại nhận định từ mục "Tổng thời gian xem mỗi video" rằng có những video cực kỳ phổ biến.

Danh sách này rất hữu ích để nhận diện các nội dung "hit" của nền tảng. Các nhà phát triển có thể nghiên cứu các video này để hiểu yếu tố thành công và nhân rộng chúng cho các nội dung khác, hoặc đảm bảo rằng các video này luôn có sẵn và được tối ưu hóa.



5.2.2.5.9. Lọc dữ liệu với các file cần thiết

- Đọc file video-ccid, chỉ giữ lại ccid hợp lệ trong video.json

video_id	ccid	
---	---	
str	str	
V_234845	0000363DB5B6E0869C33DC59013074...	
V_234876	0000363DB5B6E0869C33DC59013074...	
V_234907	0000363DB5B6E0869C33DC59013074...	
V_293392	0000363DB5B6E0869C33DC59013074...	
V_293445	0000363DB5B6E0869C33DC59013074...	

- Mapping bảng user-video có video_id hợp lệ (nghĩa là có trong ccid của video)

seq	user_id	filtered_seq	user_videos_id	ccid
list[struct[2]]	str	list[struct[2]]	list[str]	list[str]
[{"V_1358540", [{"59.747.64.747.1.0,1582184174}, {"69.795.74.795.1.0,1582184184}, ... {"584.751.589.751.1.0,1582186212}]), {"V_1358542", [{"124.588.129.588.1.0,1582186455}, {"194.608.199.608.1.0,1582186525}, ... {"579.763.584.763.1.0,1582186910}]), {"V_1358771", [{"38.9.61.5.1.5,1587480094}]]]	"U_197"	[{"V_1358540", [{"59.747.64.747.1.0,1582184174}, {"69.795.74.795.1.0,1582184184}, ... {"584.751.589.751.1.0,1582186212}]), {"V_1358542", [{"124.588.129.588.1.0,1582186455}, {"194.608.199.608.1.0,1582186525}, ... {"579.763.584.763.1.0,1582186910}]), {"V_1358782", [{"350.2.395.3.1.5,1587530324}, {"492.6.95.4.1.5,1587530419}]]]	["V_1358540", "V_1358542", "V_1358782"]	["30BB198AD98E3C0E9C33DC5901307", "E62F36210A8CAD489C33DC5901307", ... "C21C82D66075DBD69C33DC5901307
[{"V_1358540", [{"4.501.598.978027.1.0,1582990570}, {"4.75.19.75.1.0,1582991256}]), {"V_1358542", [{"4.501.204.501.1.0,1582991286}, {"209.502.374.501.1.0,1582991491}, {"384.501.591.916992.1.0,1582991666}]), ... {"V_135854", [{"4.251.405.42099.1.0,1583483786}]]]	"U_514"	[{"V_1358540", [{"4.501.598.978027.1.0,1582990570}, {"4.75.19.75.1.0,1582991256}]), {"V_1358542", [{"4.501.204.501.1.0,1582991286}, {"209.502.374.501.1.0,1582991491}, {"384.501.591.916992.1.0,1582991666}]), ... {"V_1358637", [{"4.001.9.001.1.0,1586081587}, {"59.751.64.751.1.0,1586081796}]]]	["V_1358540", "V_1358542", "V_1358637"]	["30BB198AD98E3C0E9C33DC5901307", "E62F36210A8CAD489C33DC5901307", ... "BC20A1D29F5417AA9C33DC5901307"]

- Trong user_df bỏ các course_id không nằm trong course giới hạn, bỏ enroll_time tương ứng của nó
- Từ ccid tạo cột course_watched_video là course id của video user xem, ánh xạ course id này qua bảng user_df sẽ có được cột enroll_time, tuy nhiên vì course giới hạn nên sẽ có các enroll_time bị null, ta sẽ bỏ các giá trị null này cũng như course tương ứng của nó

course_of_watched_video	enroll_time
list[str]	list[str]
["1761386", "1761386", ... "1761386"]	[null, null, ... null]
["697791", "697791", ... "697791"]	["2019-12-16 18:02:52", "2019-12-16 18:02:52", ... "2019-12-16 18:02:52"]

- Từ đây ta đã có user-video được lọc đi các khóa học, chỉ còn các khóa học bị giới hạn và thông tin liên quan

5.2.2.6. Xử lý bộ dữ liệu entities/reply.json

5.2.2.6.1. Thông kê các trường dữ liệu

Sử dụng hàm shape để xem kích thước của một DataFrame (số hàng, số cột).

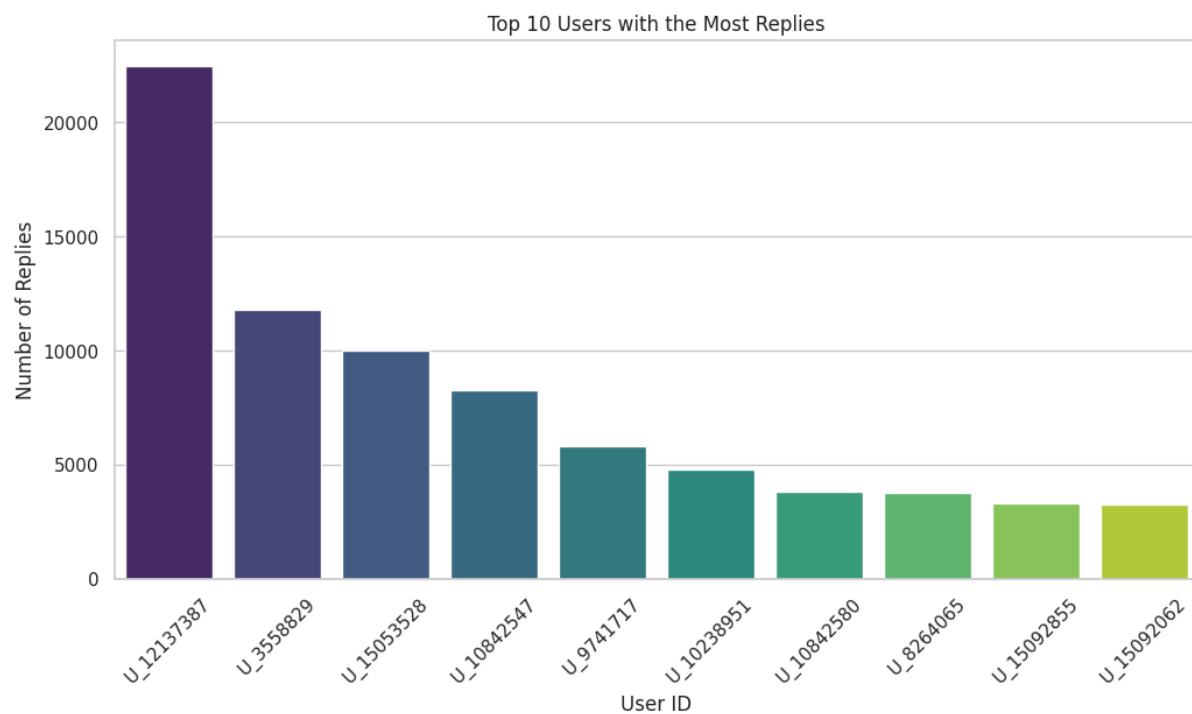
Sử dụng hàm info() để cung cấp một bản tóm tắt ngắn gọn của DataFrame (tên cột, kiểu dữ liệu, số lượng giá trị không bị thiếu).

Khảo sát về các cột trong bộ dữ liệu.

```
[ ] reply_df = reply_df.to_pandas()
      reply_df.info()

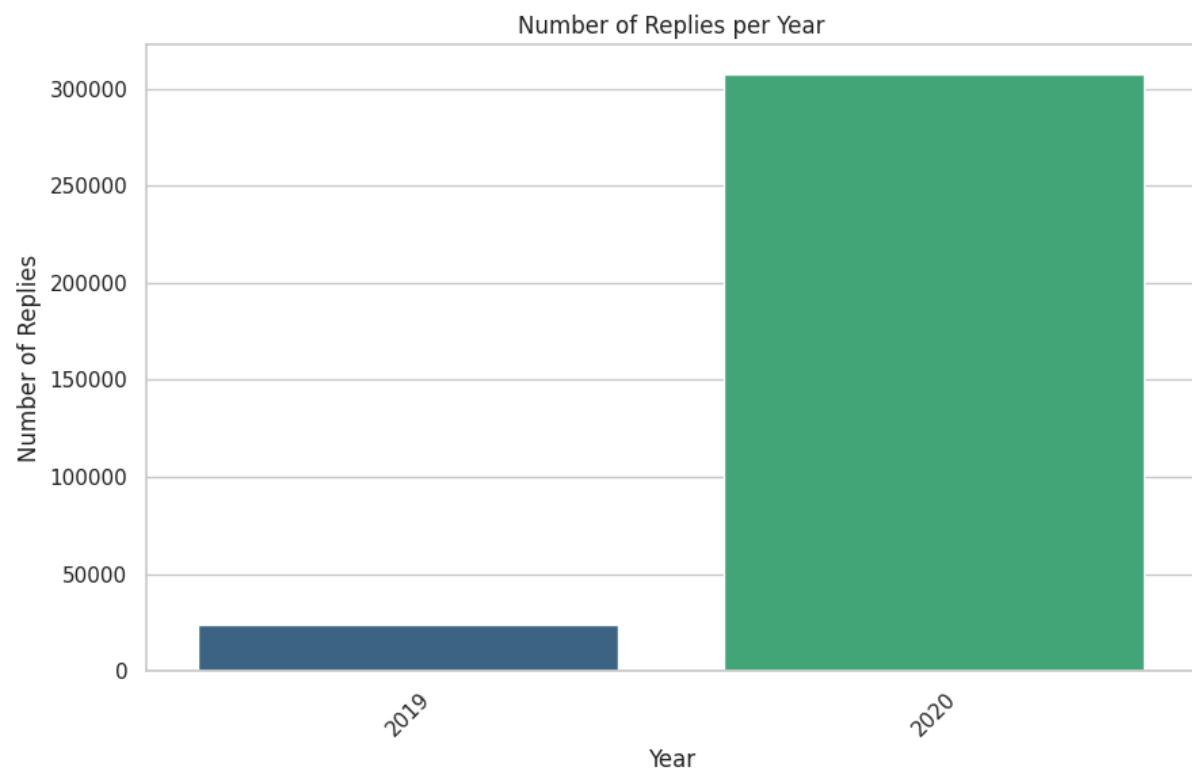
→ <class 'pandas.core.frame.DataFrame'>
RangeIndex: 331011 entries, 0 to 331010
Data columns (total 4 columns):
 #   Column      Non-Null Count  Dtype  
--- 
 0   id          331011 non-null   object 
 1   user_id     331011 non-null   object 
 2   text         331011 non-null   object 
 3   create_time 331011 non-null   object 
dtypes: object(4)
memory usage: 10.1+ MB
```

5.2.2.6.2. Thống kê số lượng reply theo người dùng



Việc xác định số lượng reply của từng người giúp xác định sự tích cực của học viên. Biểu đồ cho thấy các học viên tích cực nhất trong việc thảo luận về các vấn đề trong các khóa học

5.2.2.6.3. Thống kê reply theo thời gian



Biểu đồ thể hiện xu hướng hoạt động của học viên theo từng năm

Biểu đồ cho thấy một bức tranh rất tích cực về sự phát triển của cộng đồng trong năm 2020. Tuy nhiên, để đưa ra những nhận định chính xác và đầy đủ hơn, cần phải tiến hành phân tích sâu hơn dựa trên các dữ liệu chi tiết.

5.2.2.6.4. Gộp dữ liệu reply và với các file cần thiết

Mục tiêu: Trích xuất và liên kết thông tin từ dữ liệu reply.json với:

- User (người tạo reply)
- Comment (mà reply phản hồi lại)
- Course (khóa học chứa comment gốc)

a. Liên kết với comment-reply.txt

Xác định mỗi reply là phản hồi của comment nào.

Đây là bước quan trọng để truy ngược từ reply → comment → course.

Nhóm tiến hành gộp dữ liệu file reply và comment_reply để lấy được comment_id của reply. Mỗi reply là 1 phản hồi của 1 comment nên cần xác định các reply thuộc comment nào

```
: reply_data = reply_pd.merge(comment_reply, on="reply_id", how="left")
reply_data.head(10)
```

	reply_id	user_id	text	create_time	comment_id
0	Rp_1	U_10030806	测试回复	2019-08-05 12:55:54	Cm_1
1	Rp_2	U_10031397	赞	2019-08-09 16:39:06	Cm_12
2	Rp_3	U_10031531	好喜欢	2019-08-10 22:39:35	Cm_24
3	Rp_4	U_10031508	你也好棒	2019-08-12 14:43:34	Cm_43
4	Rp_5	U_10031508	嗯对	2019-08-12 14:44:51	Cm_44
5	Rp_6	U_10031508	人工智能是	2019-08-12 14:47:58	Cm_45
6	Rp_7	U_10031536	我的观点就是，你说啥就时啥	2019-08-13 09:41:32	Cm_60
7	Rp_8	U_10031536	我的观点就是，你说啥就时啥	2019-08-13 09:41:42	Cm_60
8	Rp_9	U_10031536	我的观点就是，你说啥就时啥	2019-08-13 09:41:53	Cm_59
9	Rp_10	U_10031536	我的观点就是，你说啥就时啥	2019-08-13 09:41:59	Cm_59

b. Liên kết với course-comment.txt

Xác định **mỗi comment** thuộc về **khóa học nào**.

Đây là cách gián tiếp gán course_id cho mỗi reply.

Nhóm tiến hành gộp file dữ liệu với course_comment để lấy thông tin về course mà comment đó được đăng.

```
: reply_data = reply_data.merge(course_comment, on="comment_id", how="left")
reply_data.head(10)
```

	reply_id	user_id	text	create_time	comment_id	course_id
0	Rp_1	U_10030806	测试回复	2019-08-05 12:55:54	Cm_1	NaN
1	Rp_2	U_10031397	赞	2019-08-09 16:39:06	Cm_12	NaN
2	Rp_3	U_10031531	好喜欢	2019-08-10 22:39:35	Cm_24	NaN
3	Rp_4	U_10031508	你也好棒	2019-08-12 14:43:34	Cm_43	NaN
4	Rp_5	U_10031508	嘎对	2019-08-12 14:44:51	Cm_44	NaN
5	Rp_6	U_10031508	人工智能是	2019-08-12 14:47:58	Cm_45	NaN
6	Rp_7	U_10031536	我的观点就是，你说啥就时啥	2019-08-13 09:41:32	Cm_60	NaN
7	Rp_8	U_10031536	我的观点就是，你说啥就时啥	2019-08-13 09:41:42	Cm_60	NaN
8	Rp_9	U_10031536	我的观点就是，你说啥就时啥	2019-08-13 09:41:53	Cm_59	NaN
9	Rp_10	U_10031536	我的观点就是，你说啥就时啥	2019-08-13 09:41:59	Cm_59	NaN

~55.6% replies không xác định được course_id.

Điều này cho thấy dữ liệu course-comment.txt **không bao phủ hết tất cả các comment** hoặc có những reply không gắn với comment nào.

```
: reply_data['course_id'].isnull().sum()
```

```
: 184194
```

5.2.2.7. Xử lý bộ dữ liệu entities/comment.json

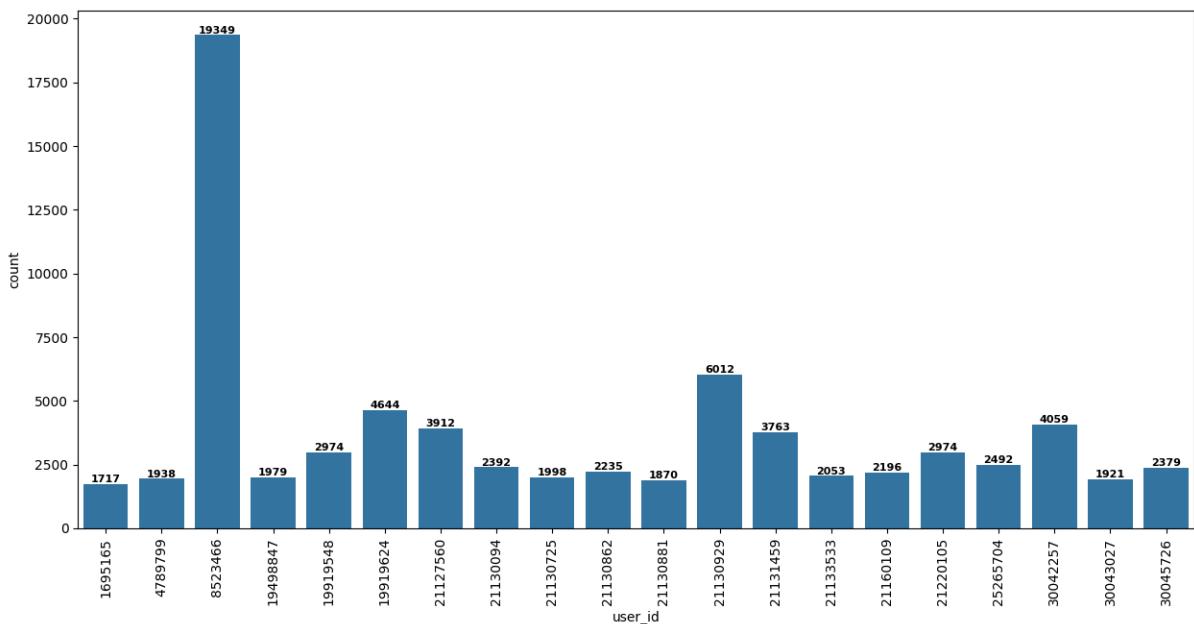
5.2.2.7.1. Thống kê các trường dữ liệu

Bảng dữ liệu có tất cả 5 trường, với tổng cộng 8395141 bản ghi. Năm trường bao gồm user_id có kiểu int64, các trường khác có kiểu object là id, text, resource_id và create_time. Mỗi bản ghi bao gồm thông tin của một câu bình luận của học sinh về một khoá học nhất định.

5.2.2.7.2. Thống kê số lượng comment theo người dùng

Dữ liệu được nhóm theo **user_id** và đếm số lượng comment của mỗi người.

Top 20 người có số lượng comment nhiều nhất:

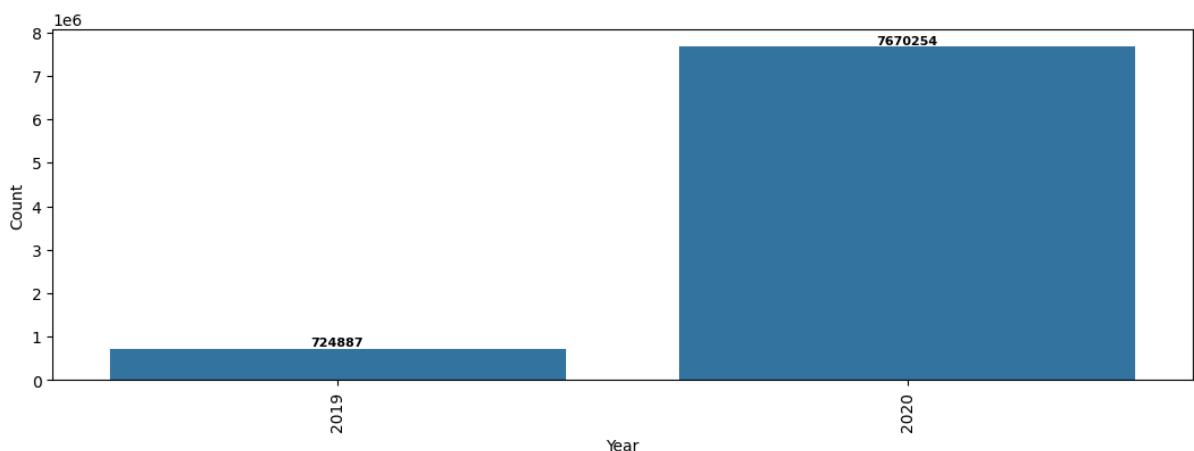


Nhận xét:

- Biểu đồ thể hiện rõ **sự phân bổ không đồng đều**, khi một số người dùng bình luận rất nhiều trong khi phần lớn còn lại ít hơn.
- Sự chênh lệch số lượng bình luận giữa các user trong top 20 là đáng kể.
- Những người dùng có số comment cao có thể là **người học tích cực, admin, hoặc có thể là bot**.
- Cần kiểm tra thêm thông tin của các user_id này để xác định vai trò thực sự trong hệ thống.

5.2.2.7.3. Thông kê thời gian bình luận theo năm

Tách năm từ cột **create_time** và nhóm lại để đếm số lượng comment mỗi năm.



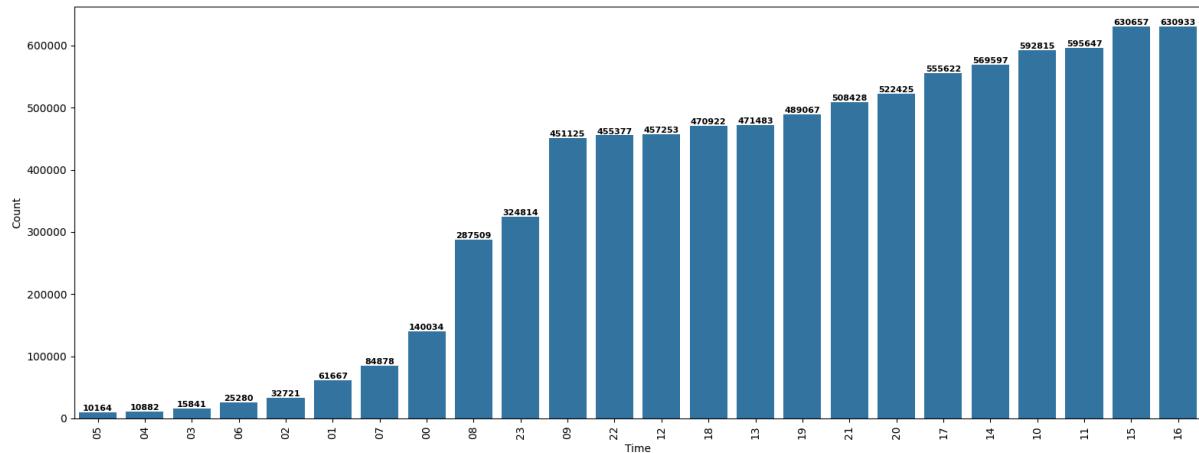
Nhận xét:

- Năm 2020 có số lượng bình luận cao nhất.

- Điều này có thể phản ánh **sự gia tăng người dùng hoặc hoạt động học tập cao nhất trong năm 2020**.
- Cần xem xét các yếu tố khác như chiến dịch marketing, giai đoạn giãn cách xã hội (Covid-19), hoặc việc phát hành các khóa học mới trong năm đó.

5.2.2.7.4. Thống kê thời gian bình luận theo giờ trong ngày

Trích xuất **giờ trong ngày** từ trường `create_time`. Nhóm theo giờ và đếm số lượng comment theo từng mốc thời gian.



Nhận xét:

- Hoạt động bình luận **cao nhất từ 15h đến 16h**, kể đến là **buổi chiều và tối**.
- Thời gian này phản ánh **hành vi học tập hoặc truy cập của người dùng thường diễn ra sau giờ hành chính hoặc giờ học chính khóa**.
- Có thể cân nhắc thời gian này khi lên lịch các thông báo hoặc hoạt động tương tác để **tối ưu hiệu quả truyền thông**.

5.2.2.7.5. Làm sạch dữ liệu

Loại bỏ các dòng có giá trị null ở trường `resource_id` (mã tài nguyên).

Chuẩn bị gộp và xử lý dữ liệu trùng lặp (đặt tên trường là `duplicate_count`, nhưng chưa rõ thực hiện gộp cụ thể ra sao).

Nhận xét:

- Việc loại bỏ dòng null ở `resource_id` là cần thiết vì những comment không liên kết đến tài nguyên sẽ **khó phân tích về mặt ngữ cảnh hoặc chất lượng nội dung**.
- Tuy nhiên, bạn **chưa thực hiện gộp dòng trùng lặp thực sự**, cần bổ sung thao tác `drop_duplicates()` kèm theo điều kiện thích hợp nếu muốn loại trừ các comment bị lặp.

5.2.2.7.6. Gộp dữ liệu comment và với các file cần thiết

Mục tiêu: Trích xuất và liên kết thông tin từ dữ liệu reply.json với:

- User (người tạo reply)
- Course (khóa học chứa comment gốc)

a. Liên kết với course

```
comment_data = course_comment.merge(comment_pd, on="comment_id",
how="right")
```

Mục đích: Liên kết comment với khóa học.

Kết quả: Có 1.78 triệu comment không tìm thấy course_id tương ứng (NaN).

Điều này có thể do:

- Một số comment không thuộc khóa học nào (ví dụ: bình luận trong forum chung).
- Quan hệ course-comment.txt không đầy đủ.

b. Làm sạch và format user_id

```
comment_user['user_id'] = comment_user['user_id'].apply(lambda x: f'U_{x}')
```

Format lại để đồng bộ với user_df, vốn có dạng "U_123".

c. Tích hợp thêm thông tin đăng ký khóa học

Map giữa course_id và enroll_time cho từng user:

```
comment_user["enroll_time"] = comment_user.apply(
    lambda row: row["course_enroll_map"].get(row["course_id_clean"], None)
    if isinstance(row["course_enroll_map"], dict) else None, axis=1
)
```

Merge với comment_user:

```
comment_user = comment_user.merge(user[['user_id', 'course_enroll_map']],
on='user_id', how='left')
```

Mục tiêu: Lấy thời điểm đăng ký khóa học của user để phân tích hành vi thời gian (ví dụ: bình luận sớm hay muộn so với đăng ký).

Tuy nhiên, **275,655 bản ghi** không tìm được course_enroll_map, có thể do:

- User không có thông tin khóa học trong bảng user.
 - Comment là của guest user hoặc user không rõ danh tính.

5.2.3. Trích xuất đặc trưng

5.2.3.1. Trích xuất đặc trưng từ dữ liệu comment và reply

5.2.3.1.1. Comment

Mục tiêu: Nhóm thực hiện trích xuất 3 đặc trưng chính từ dữ liệu bình luận (comment) của người dùng trong mỗi khóa học, theo từng giai đoạn (phase) thời gian kể từ khi người dùng đăng ký khóa học (enroll time):

- **Số lượng bình luận** trong từng phase.
 - **Tổng số** từ trong tất cả các bình luận trong từng phase.
 - **Mức độ phân tán thời gian bình luận (entropy)** trong từng phase, đo bằng Shannon entropy dựa trên phân phối bình luận trong các khoảng thời gian nhỏ (mỗi khoảng 30 phút trong ngày).

Các giai đoạn (phase) được định nghĩa dựa trên khoảng cách ngày giữa thời gian đăng ký khóa học và thời gian tạo comment (create time):

- Phase 1: từ 0 đến 14 ngày.
 - Phase 2: từ 15 đến 28 ngày.
 - Phase 3: từ 29 đến 42 ngày.
 - Phase 4: từ 43 đến 56 ngày.

a. Tiễn xứ lý

Loại bỏ bản ghi thiếu enroll time (~4.18% dữ liệu bị loại).

Chuyển cột create_time, enroll_time sang kiểu datetime để dễ xử lý thời gian.

```
comment['create_time'] = pd.to_datetime(comment['create_time'], errors='coerce')
comment['enroll_time'] = pd.to_datetime(comment['enroll_time'], errors='coerce')
```

Tính số ngày trôi qua kể từ enroll time đến create time (days since enroll).

```
comment['days since enroll'] = (comment['create time'] -
```

```
comment['enroll_time']).dt.days
```

Lọc dữ liệu để chỉ giữ lại những comment được tạo ra sau thời điểm đăng ký khóa học (`create_time > enroll_time`), loại bỏ những comment tạo ra trước hoặc cùng ngày đăng ký khóa học (do có thể là dữ liệu lỗi hoặc không hợp lệ):

```
comment = comment[comment['create_time'] > comment['enroll_time']]
```

Xử lý đếm số từ trong comment:

- Định nghĩa hàm `count_words` để đếm số từ trong nội dung comment.
 - Đối với tiếng Trung, Nhật, Hàn (CJK): đếm số ký tự sau khi loại bỏ dấu câu đặc trưng.
 - Đối với tiếng Anh hoặc các ngôn ngữ dùng khoảng trắng phân tách từ: đếm số từ dựa trên biểu thức chính quy.
- Tính cột mới `num_words_rep` chứa số từ tương ứng cho từng comment.

```
def count_words(text):  
    text = str(text).strip() # Đảm bảo dữ liệu không bị lỗi None hoặc NaN  
  
    # Nếu chứa ký tự CJK (tiếng Trung, Nhật, Hàn)  
    if re.search(r'[\u4e00-\u9fff\u3040-\u30ff\uac00-\ud7af]', text):  
        # Loại bỏ dấu câu trước khi đếm ký tự  
        text = re.sub(r'[。！？、—「」（）《》◊～]', '', text)  
        return len(text) # Đếm số ký tự còn lại  
  
    else:  
        # Đếm số từ tiếng Anh  
        return len(re.findall(r'\b\w+\b', text))  
  
    # Áp dụng vào DataFrame  
comment['num_words_rep'] = comment['text'].apply(count_words)
```

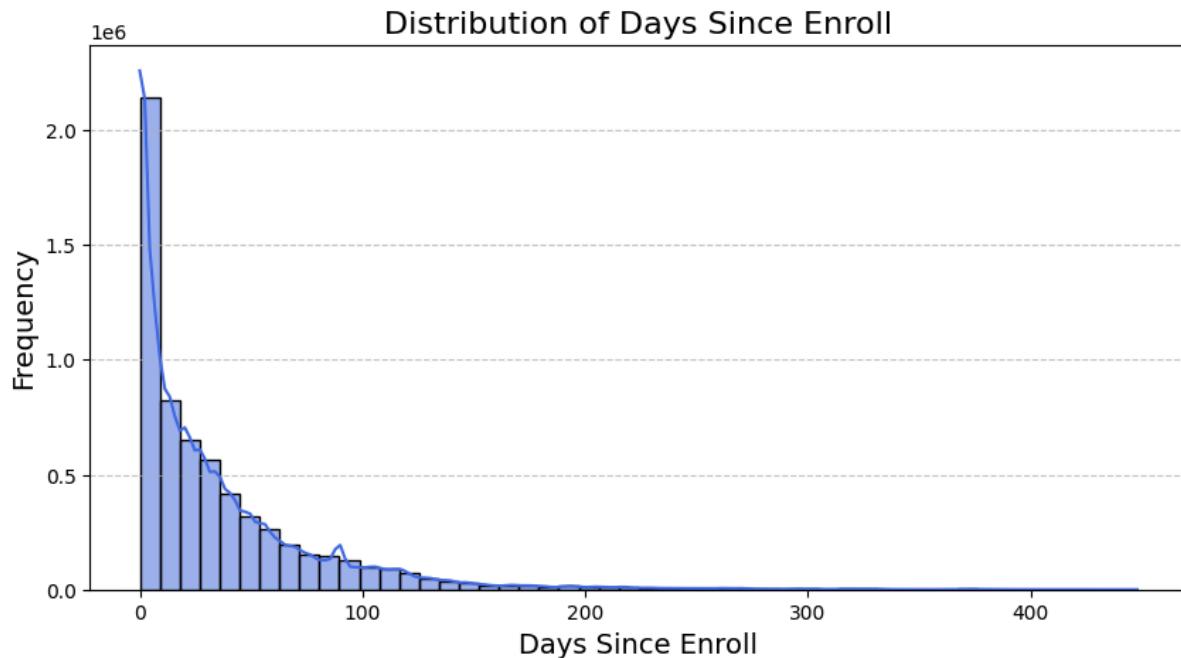
Bộ dữ liệu sau khi đã thực hiện xử lý và gộp các cột cần thiết:

```

<class 'pandas.core.frame.DataFrame'>
Index: 6331256 entries, 0 to 6608579
Data columns (total 8 columns):
 #   Column           Dtype  
--- 
 0   comment_id       object  
 1   user_id          object  
 2   course_id        object  
 3   text              object  
 4   create_time       datetime64[ns]
 5   enroll_time       datetime64[ns]
 6   days_since_enroll int64   
 7   num_words_rep    int64   
dtypes: datetime64[ns](2), int64(2), object(4)
memory usage: 434.7+ MB

```

Phân phối thời gian bình luận kể từ ngày đăng ký khóa học:



b. Trích xuất đặc trưng theo từng phase

1. Tổng số bình luận và tổng số từ trong giai đoạn i

Đặc trưng số lượng comment và tổng từ là các chỉ số trực quan, phản ánh mức độ hoạt động và tương tác của người dùng trong từng giai đoạn.

Với mỗi user-course, nhóm tính tổng số bình luận và tổng số từ trong từng phase, sử dụng groupby(['user_id', 'course_id']).

Các bảng kết quả cho từng phase được tạo riêng biệt, ví dụ result_phasei bao gồm:

- comment_count_phasei: số lượng comment trong phase i.
- total_words_phasei: tổng số từ trong phase i.

```
# Tính tổng số chữ phản hồi của user trong phase 1
word_count_phase1 = Phase1_cmt.groupby(['user_id', 'course_id'])['num_words_rep'].sum().reset_index(name='total_words_phase1')

# Đếm số lượng phản hồi trong phase 1
comment_count_phase1 = Phase1_cmt.groupby(['user_id', 'course_id']).size().reset_index(name='comment_count_phase1')

result_phase1 = pd.merge(comment_count_phase1, word_count_phase1, on=['user_id', 'course_id'], how='outer')
```

2. Mức độ phân tán thời gian comment (entropy) trong giai đoạn i

Đặc trưng entropy đo mức độ phân tán bình luận trong ngày, có ý nghĩa trong việc đánh giá tính đều đặn hay tập trung về thời gian bình luận (ví dụ, người dùng chỉ comment vào một khung giờ cố định hay trải đều suốt ngày).

Mỗi comment được phân vào 48 khoảng thời gian trong ngày (mỗi khoảng 30 phút) theo create_time.

Với mỗi user-course trong từng phase, tính phân phối xác suất comment rơi vào các khoảng thời gian này.

Tính entropy theo Shannon entropy từ phân phối này:

$$H = - \sum_i p_i \log p_i$$

Trong đó p_i là xác suất comment trong khoảng thời gian i.

Entropy càng cao, bình luận càng phân tán đều theo thời gian trong ngày.

Entropy bằng 0 nếu chỉ có 1 hoặc 0 khoảng thời gian có comment.

```
# Bước 1: Sắp xếp dữ liệu theo user_id, course_id, và create_time
Phase1_cmt = Phase1_cmt.sort_values(by=['user_id', 'course_id', 'create_time'])

# Bước 2: Xác định khoảng thời gian (bins)
def assign_time_bin(timestamp):
    """
    Chia ngày thành 48 khoảng thời gian (mỗi khoảng 30 phút) và gán bin tương ứng.
    
```

Chia ngày thành 48 khoảng thời gian (mỗi khoảng 30 phút) và gán bin tương ứng.

"""

```
return (timestamp.hour * 60 + timestamp.minute) // 30
```

```
Phase1_cmt['time_bin'] = Phase1_cmt['create_time'].apply(assign_time_bin)
```

Bước 3: Đếm số lượng comment trong mỗi khoảng thời gian

```
time_bin_counts1 = Phase1_cmt.groupby(['user_id', 'course_id', 'time_bin']).size().reset_index(name='count')
```

Bước 4: Tính tổng số comment của mỗi user trong từng khóa học

```
total_comments1 = Phase1_cmt.groupby(['user_id', 'course_id']).size().reset_index(name='total_count')
```

Gộp dữ liệu để tính xác suất

```
time_bin_counts1 = time_bin_counts1.merge(total_comments1, on=['user_id', 'course_id'], how='left')
```

Bước 5: Tính xác suất xuất hiện của mỗi bin

```
time_bin_counts1['probability'] = time_bin_counts1['count'] / time_bin_counts1['total_count']
```

Bước 6: Tính entropy theo công thức Shannon entropy

```
def calculate_entropy(probabilities):
```

"""

Tính entropy theo công thức Shannon entropy.

Loại bỏ giá trị xác suất bằng 0 để tránh lỗi log(0).

"""

```
probabilities = probabilities[probabilities > 0] # Loại bỏ giá trị 0
```

```
return -np.sum(probabilities * np.log(probabilities)) if len(probabilities) > 1 else 0
```

```
entropy_values_phase1 = (
```

```
    time_bin_counts1.groupby(['user_id', 'course_id'])['probability']
```

```
    .apply(lambda x: calculate_entropy(x.values))
```

```
    .reset_index(name='entropy_time_comment_phase1')
```

```
)
```

3. Kết hợp các đặc trưng

Các đặc trưng tổng số bình luận, tổng từ, và entropy được ghép lại thành bảng đặc trưng cho từng phase.

Với các user-course không có comment trong phase đó, giá trị entropy được điền 0 (mặc định).

5.2.3.1.2. Reply

Mục tiêu: Nhóm thực hiện trích xuất 3 đặc trưng chính từ dữ liệu phản hồi (reply) của người dùng trong mỗi khóa học, theo từng giai đoạn (phase) thời gian kể từ khi người dùng đăng ký khóa học (enroll_time):

- **Số lượng phản hồi** trong từng phase.
- **Tổng số từ** trong tất cả các phản hồi trong từng phase.
- **Mức độ phân tán thời gian phản hồi (entropy)** trong từng phase, đo bằng Shannon entropy dựa trên phân phối phản hồi trong các khoảng thời gian nhỏ (mỗi khoảng 30 phút trong ngày).

Các giai đoạn (phase) được định nghĩa dựa trên khoảng cách ngày giữa thời gian đăng ký khóa học và thời gian tạo reply (create_time):

- Phase 1: từ 0 đến 14 ngày.
- Phase 2: từ 15 đến 28 ngày.
- Phase 3: từ 29 đến 42 ngày.
- Phase 4: từ 43 đến 56 ngày.

a. Tiết xử lý

Loại bỏ các bản ghi không có enroll_time (23.19% bị loại):

```
reply_user.dropna(subset=['enroll_time'], inplace=True)
```

Chuyển đổi cột thời gian sang định dạng datetime để dễ tính toán:

```
reply['create_time'] = pd.to_datetime(reply['create_time'], errors='coerce')
reply['enroll_time'] = pd.to_datetime(reply['enroll_time'], errors='coerce')
```

Tính khoảng thời gian days_since_enroll = create_time - enroll_time (số ngày từ lúc đăng ký đến khi phản hồi):

```
reply['days_since_enroll'] = (reply['create_time'] - reply['enroll_time']).dt.days
```

Đếm số từ/ ký tự trong nội dung phản hồi num_words_rep, xử lý đặc biệt với ngôn ngữ chứa ký tự CJK (Trung-Nhật-Hàn) đếm theo ký tự, còn tiếng Anh đếm theo từ:

```
def count_words(text):
    text = str(text).strip() # Đảm bảo dữ liệu không bị lỗi None hoặc NaN
    # Nếu chứa ký tự CJK (tiếng Trung, Nhật, Hàn)
    if re.search(r'[\u4e00-\u9fff\u3040-\u30ff\uac00-\ud7af]', text):
        # Loại bỏ dấu câu trước khi đếm ký tự
        text = re.sub(r'[。！？、—「」『』〔〕《》〈〉～]', " ", text)
        return len(text) # Đếm số ký tự còn lại
    else:
        # Đếm số từ tiếng Anh
        return len(re.findall(r'\b\w+\b', text))

# Áp dụng vào DataFrame
reply['num_words_rep'] = reply['text'].apply(count_words)
```

b. Trích xuất đặc trưng theo từng phase

1. Tổng số phản hồi và tổng số từ trong giai đoạn i

Đặc trưng số lượng reply và tổng từ là các chỉ số trực quan, phản ánh mức độ hoạt động và tương tác của người dùng trong từng giai đoạn.

Với mỗi user-course, nhóm tính tổng số phản hồi và tổng số từ trong từng phase, sử dụng groupby(['user_id', 'course_id']).

Các bảng kết quả cho từng phase được tạo riêng biệt, ví dụ result_phasei bao gồm:

- reply_count_phasei: số lượng comment trong phase i.
- total_words_phasei: tổng số từ trong phase i.

2. Mức độ phân tán thời gian reply (entropy) trong giai đoạn i

Đặc trưng entropy đo mức độ phân tán phản hồi trong ngày, có ý nghĩa trong việc đánh giá tính đều đặn hay tập trung về thời gian phản hồi (ví dụ, người dùng chỉ phản hồi vào một khung giờ cố định hay trải đều suốt ngày).

Mỗi phản hồi được phân vào 48 khoảng thời gian trong ngày (mỗi khoảng 30 phút) theo `create_time`.

Với mỗi user-course trong từng phase, tính phân phối xác suất phản hồi rơi vào các khoảng thời gian này.

Tính entropy theo Shannon entropy từ phân phối này:

$$H = - \sum_i p_i \log p_i$$

Trong đó p_i là xác suất phản hồi trong khoảng thời gian i.

Entropy càng cao, bình luận càng phân tán đều theo thời gian trong ngày.

Entropy bằng 0 nếu chỉ có 1 hoặc 0 khoảng thời gian có phản hồi.

3. Kết hợp các đặc trưng

Các đặc trưng tổng số phản hồi, tổng từ, và entropy được ghép lại thành bảng đặc trưng cho từng phase.

Với các user-course không có phản hồi trong phase đó, giá trị entropy được điền 0 (mặc định).

5.2.3.1.3. Phân tích cảm xúc từ bình luận (comment) và phản hồi (reply)

Mục tiêu: Nhóm thực hiện trích xuất 3 đặc trưng chính thể hiện phân phối cảm xúc trong các bình luận của người dùng đối với mỗi khóa học, được phân chia theo từng giai đoạn (phase) thời gian kể từ khi người dùng đăng ký khóa học (`enroll_time`):

- **Tổng số lượt bình luận có cảm xúc tích cực (positive) trong từng phase.**
 - **Tổng số lượt bình luận có cảm xúc tiêu cực (negative) trong từng phase.**
 - **Tổng số lượt bình luận có cảm xúc trung tính (neutral) trong từng phase.**
- **Lưu ý:** Bình luận (comment) và phản hồi (reply) được xử lý gộp chung, vì cả hai loại dữ liệu đều có giá trị tương đương trong phân tích cảm xúc và phản ánh trực tiếp thái độ của người dùng trong quá trình tương tác với khóa học.

Mỗi đặc trưng trên được tính riêng biệt theo từng giai đoạn thời gian (phase), được định nghĩa dựa trên khoảng cách ngày giữa thời điểm người dùng đăng ký khóa học (`enroll_time`) và thời điểm bình luận được tạo (`create_time`) như sau:

- Phase 1: từ 0 đến 14 ngày.
- Phase 2: từ 15 đến 28 ngày.
- Phase 3: từ 29 đến 42 ngày.

- Phase 4: từ 43 đến 56 ngày.

Việc trích xuất các đặc trưng cảm xúc theo từng phase nhằm phục vụ cho các mô hình dự đoán hành vi học tập của người dùng trong các khóa học trực tuyến (MOOC), qua đó phản ánh sự thay đổi trong thái độ cảm xúc của người học theo thời gian kể từ khi tham gia khóa học.

a. Gán nhãn dữ liệu comment

Mục tiêu: xác định cảm xúc (tích cực, trung tính, tiêu cực) cho mỗi bình luận hoặc phản hồi của người học, từ đó phục vụ cho việc trích xuất đặc trưng cảm xúc theo từng phase thời gian và hỗ trợ mô hình dự đoán hành vi học tập.

Vấn đề:

- Tổng số bình luận ban đầu: ~8 triệu.
- Sau khi lọc chỉ lấy các khóa học tiêu biểu: giảm còn 3.5 triệu bình luận.
- Tuy nhiên, dữ liệu vẫn quá lớn để áp dụng trực tiếp các mô hình pre-trained như cardiffnlp/twitter-roberta-base-sentiment do giới hạn bộ nhớ và tốc độ xử lý.

Chiến lược gán nhãn comment:

- **Giai đoạn 1:** Gán nhãn **198630 bình luận đầu tiên** sử dụng mô hình pre-trained twitter-roberta-base-sentiment:
 - Mô hình sử dụng: cardiffnlp/twitter-roberta-base-sentiment.
 - Pipeline Hugging Face text-classification.
 - Mapping cảm xúc:
 - "LABEL_0" → **Negative (0)**
 - "LABEL_1" → **Neutral (1)**
 - "LABEL_2" → **Positive (2)**

```
# Load the pipeline for sentiment analysis
pipe = pipeline("text-classification", model="cardiffnlp/twitter-roberta-base-sentiment", device=0)

# Get the tokenizer associated with the pipeline
tokenizer = AutoTokenizer.from_pretrained("cardiffnlp/twitter-roberta-base-sentiment")

# Assuming your text data is in a column named 'text'
# Create a new column 'sentiment' with the analysis results
# {"NEGATIVE": 0, "NEUTRAL": 1, "POSITIVE": 2}
label_to_num = {"LABEL_0": 0, "LABEL_1": 1, "LABEL_2": 2}

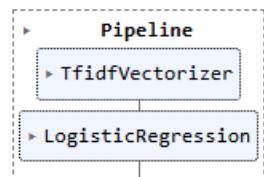
# Áp dụng pipeline và chuyển nhãn thành số
df['sentiment_label'] = df['text'].apply(
    lambda x: label_to_num[pipe(x, truncation=True, max_length=512)[0]['label']])
)
```

- **Giai đoạn 2:** Huấn luyện mô hình truyền thống Logistic Regression dựa trên dữ liệu đã gán nhãn để **gán nhãn phần còn lại (~2.7 triệu bình luận)**.
 - **Quy trình:**

- Đọc dữ liệu đã gán nhãn (comments_labeled.csv) và dữ liệu chưa gán (comments_unlabeled.csv).
- Làm sạch dữ liệu (drop NaN).
- Huấn luyện mô hình TF-IDF + Logistic Regression:

```
# Pipeline: TF-IDF + Softmax (Logistic Regression đa lớp)
pipeline = Pipeline([
    ('tfidf', TfidfVectorizer(analyzer='char_wb', ngram_range=(2,4), max_features=100000)),
    ('clf', LogisticRegression(max_iter=1000, multi_class='multinomial', solver='lbfgs'))
])

pipeline.fit(X_train, y_train)
```



- Đánh giá trên tập kiểm tra:

	precision	recall	f1-score	support
0.0	0.80	0.42	0.55	2038
1.0	0.90	0.96	0.93	26569
2.0	0.90	0.83	0.86	11119
accuracy			0.90	39726
macro avg	0.87	0.74	0.78	39726
weighted avg	0.89	0.90	0.89	39726

➔ Nhận xét: Mô hình hoạt động tốt trên nhãn tích cực và trung tính, nhưng gặp khó khăn với nhãn tiêu cực (ít dữ liệu hơn → mất cân bằng).

- Gán nhãn phần còn lại: comments_unlabeled['sentiment_label'] = pipeline.predict(comments_unlabeled['text'])
- Thống kê dữ liệu sau khi gán nhãn:

```
(3560004, 6)
sentiment_label
1.0    2882936
2.0    633813
0.0    43255
Name: count, dtype: int64
```

b. Gán nhãn dữ liệu reply

Dữ liệu reply ban đầu có số lượng lớn, nhưng sau khi lọc theo tiêu chí phù hợp (giới hạn theo các khóa học và điều kiện nghiên cứu), số lượng reply được rút gọn chỉ còn 46,969 dòng. Với kích thước dữ liệu như vậy, nhóm quyết định sử dụng trực tiếp mô hình pre-trained để gán nhãn cảm xúc mà không cần huấn luyện thêm mô hình phân loại như phần comment.

Mô hình:

- Pre-trained model: cardiffnlp/twitter-roberta-base-sentiment
- Pipeline: HuggingFace transformers.pipeline("text-classification")

c. Phối phối cảm xúc từ bình luận (comment) và phản hồi (reply)

Mục tiêu: Từ dữ liệu comment và reply đã được gán nhãn, nhóm tiến hành tổng hợp phân loại cảm xúc của comment và reply. Sau đó gộp lại để tính tổng số comment và reply và phân phối của các nhãn theo cả comment và reply.

Chuẩn hóa dữ liệu comment để đếm số lượng cảm xúc theo từng người dùng và khóa học.

```
comment.drop(columns=["text", "comment_id", "create_time"], inplace=True)

comment['positive'] = comment['sentiment_label'].apply(lambda x: 1 if x == 2.0 else 0)
comment['negative'] = comment['sentiment_label'].apply(lambda x: 1 if x == 0.0 else 0)
comment['neutral'] = comment['sentiment_label'].apply(lambda x: 1 if x == 1.0 else 0)

comment.drop(columns=['sentiment_label'], inplace=True)
comment = comment.groupby(["course_id", "user_id"]).sum().reset_index()
comment['total_cmt'] = comment['positive'] + comment['negative'] + comment['neutral']
```

Nhận xét:

- Phân loại cảm xúc được thực hiện đúng theo chuẩn "NEGATIVE": 0, "NEUTRAL": 1, "POSITIVE": 2.
- Việc groupby theo course_id và user_id giúp tổng hợp dữ liệu phù hợp cho phân tích theo từng học viên và khóa học.
- total_cmt là tổng số bình luận, phục vụ cho các phân tích tỷ lệ sau này

Thực hiện tương tự như comment, tổng hợp cảm xúc reply theo từng user-course:

```

reply.drop(columns=["text", "comment_id", "create_time", "reply_id"], inplace=True)

reply['positive'] = reply['sentiment_label'].apply(lambda x: 1 if x == 2.0 else 0)
reply['negative'] = reply['sentiment_label'].apply(lambda x: 1 if x == 0.0 else 0)
reply['neutral'] = reply['sentiment_label'].apply(lambda x: 1 if x == 1.0 else 0)

reply.drop(columns=["sentiment_label"], inplace=True)
reply = reply.groupby(["course_id", "user_id"]).sum().reset_index()
reply['total_reply'] = reply['positive'] + reply['negative'] + reply['neutral']

```

Nhận xét:

- Quá trình xử lý giống như phần comment.
- Lưu ý: sentiment_label có 1 giá trị null, nhưng được xử lý an toàn vì Pandas .apply() không gây lỗi cho NaN nếu không khớp điều kiện.

Tạo bảng tổng hợp toàn bộ thông tin cảm xúc từ cả bình luận và phản hồi cho mỗi học viên-khoa học.

```

final_df = pd.merge(comment, reply, on=['user_id', 'course_id'], how='outer', suffixes=('_comment'
final_df.fillna(0, inplace=True)

final_df['positive'] = final_df['positive_comment'] + final_df['positive_reply']
final_df['negative'] = final_df['negative_comment'] + final_df['negative_reply']
final_df['neutral'] = final_df['neutral_comment'] + final_df['neutral_reply']

final_df['total'] = final_df['total_reply'] + final_df['total_cmt']

final_df.drop(columns=[
    'positive_comment', 'negative_comment', 'neutral_comment', 'total_reply',
    'positive_reply', 'negative_reply', 'neutral_reply', 'total_cmt'
], inplace=True)

```

Nhận xét:

- Dùng outer join đảm bảo không bỏ sót bất kỳ dữ liệu nào từ comment hoặc reply.
- fillna(0) rất cần thiết vì có nhiều cặp user-course chỉ xuất hiện trong một phía.
- Tổng hợp positive, negative, neutral, và total là đặc trưng cuối cùng cần để phân tích người học.

5.2.3.2. Trích xuất đặc trưng từ dữ liệu user-video

5.2.3.2.1 Tạo đặc trưng duration_seg và avg_duration_seg

Trích xuất và tính toán thông tin về thời lượng của các phân đoạn video (segments) trong mỗi tương tác xem của người dùng.

- **Tạo cột duration_seg:**

Duyệt qua danh sách các video đã xem bởi mỗi người dùng, sau đó đi sâu vào từng segment trong mỗi video đó. Với mỗi segment, nó tính toán thời lượng thực tế của segment bằng cách lấy giá trị end_point trừ đi start_point, đảm bảo kết quả là một số dương. Kết quả là một cột mới có tên duration_seg, chứa một danh sách lồng ghép các thời lượng segment cho mỗi chuỗi video đã xem.

Ý nghĩa của đặc trưng mới: duration_seg đại diện cho chi tiết thời lượng từng phân đoạn video mà người dùng đã xem. Đặc trưng này cho phép phân tích sâu hơn về độ dài của các phần mà người dùng đã tương tác, giúp hiểu các kiểu xem video theo từng khoảnh khắc.

- **Tạo cột avg_duration_seg:**

Tiếp tục xử lý cột duration_seg vừa tạo. Với mỗi danh sách các thời lượng segment trong duration_seg, nó tính toán giá trị trung bình của các thời lượng đó. Kết quả được lưu vào một cột mới có tên avg_duration_seg.

Ý nghĩa của đặc trưng mới: avg_duration_seg biểu thị thời lượng trung bình của một phân đoạn video mà người dùng đã xem trong một chuỗi video nhất định. Đặc trưng này cung cấp một cái nhìn tổng quan về xu hướng xem của người dùng: liệu họ có thường xem các phân đoạn ngắn hay dài, giúp nhận diện hành vi như tua nhanh, bỏ qua, hoặc xem kỹ.

user_id	filtered_seq	user_videos_id	ccid	duration_seg	avg_duration_seg
"U_112"	[{"V_1395633", [{"130.0,190.0,1.0,1588431144}, {"220.0,250.0,1.0,1588431234}, ... {655.1,692.55,1.25,1588437514}]}], [{"V_1395635", [{"135.0,170.0,1.0,1588438045}]}], [{"V_1395639", [{"100.0,106.25,1.25,1588438980}, {"180.0,188.25,1.25,1588439045}]}]]	list[str]	list[str]	list[list[f64]]	list[f64]
"U_189"	[{"V_6334508", [{"4.0,109.0,1.0,1598832781}, {"119.0,254.0,1.0,1598832896}]}], [{"V_6334516", [{"593.0,598.0,1.0,1598834837}]}], [{"V_6254676", [{"5.0,105.0,1.0,1603849805}]}]]	list[str]	list[str]	list[list[f64]]	list[f64]

5.2.3.2.2. Chuyển đổi và Gộp Dữ liệu theo user_id và course_of_watched_video

- **Làm phẳng DataFrame (explode):**

Đoạn code này làm phẳng DataFrame bằng cách biến đổi các cột chứa danh sách (ngoại trừ user_id) thành các hàng riêng biệt. Điều này có nghĩa là nếu một người

dùng có nhiều tương tác xem video, hoặc một video có nhiều segment, mỗi tương tác/segment sẽ trở thành một hàng riêng biệt trong DataFrame.

- **Gộp lại theo user_id và course_of_watched_video (group_by và agg):**

Sau khi làm phẳng, đoạn code này nhóm các hàng lại với nhau dựa trên hai yếu tố chính: ID của người dùng (user_id) và khóa học của video đã xem (course_of_watched_video). Đối với tất cả các cột còn lại, nó thu thập lại các giá trị tương ứng vào thành các danh sách (list) trong mỗi nhóm.

Ý nghĩa của đặc trung mới/cấu trúc dữ liệu mới:

- Việc nhóm dữ liệu theo user_id và course_of_watched_video cho phép chúng ta tổng hợp tất cả các hành vi và tương tác của một người dùng đối với các video trong một khóa học cụ thể.
- Các cột được gộp lại thành danh sách giúp giữ lại thứ tự và chi tiết của từng tương tác, ví dụ: thứ tự các segment đã xem, thời điểm xem, tốc độ phát lại, v.v., cho phép phân tích hành vi trong ngữ cảnh của một khóa học.
- Cấu trúc mới này tạo ra một khung dữ liệu tổng quan hơn, lý tưởng cho việc phân tích các mẫu hành vi của người dùng trong một khóa học cụ thể, ví dụ: mức độ hoàn thành khóa học, các video phổ biến nhất trong khóa học, hoặc các vấn đề mà người học gặp phải trong quá trình học.

5.2.3.2.3. Tạo Đặc Trung video_watch_count

- Tính toán **video_watch_count**: Xử lý cột ccid (ccid của các video hoặc các thành phần nội dung) trong DataFrame đã được nhóm theo người dùng và khóa học. Nó lấy danh sách các ccid trong mỗi nhóm, sau đó xác định các giá trị ccid duy nhất và cuối cùng đếm số lượng các giá trị duy nhất đó. Kết quả được lưu vào một cột mới có tên video_watch_count.
- Ý nghĩa của đặc trung mới: video_watch_count cho biết mức độ bao phủ nội dung của người dùng trong một khóa học cụ thể. Đặc trung này giúp chúng ta hiểu liệu người dùng có xem nhiều video khác nhau trong một khóa học hay họ chỉ tập trung vào một vài video nhất định. Nó là một chỉ số quan trọng để đánh giá sự khám phá và mức độ tham gia của người học vào toàn bộ nội dung của khóa học.

eg	segments_list	start_points	end_points	speed	watch_time_seg	video_length	enroll_time	video_watch_count
l	list[list[struct[4]]]	list[list[f64]]	list[list[f64]]	list[list[f64]]	list[list[f64]]	list[f64]	list[str]	u32
)]	[[[15.1,113.2,1.0,1598235665), {120.4,150.4,1.0,1598237633}]]	[[15.1, 120.4]]	[[113.2, 150.4]]	[[1.0, 1.0]]	[[98.1, 30.0]]	[474.584]	["2020-08-12 08:42:38"]	1
	[[{9.251,14.251,1.0,1581503242}, {19.251,44.251,1.0,1581503252}, {88.8,599.0,2.0,1581504249}], [{30.1,400.8,2.0,1581504561}, {364.5,464.6,2.0,1581504915}, ... {584.6,591.9,2.0,1581507019}], ... [{226.0,236.0,2.0,1586918953}]]	[[9.251, 19.251, 88.8], [30.1, 364.5, ... 584.6], ... [226.0]]	[[14.251, 44.251, 599.0], [400.8, 464.6, ... 591.9], ... [236.0]]	[[1.0, 1.0, 2.0], [2.0, ..., 2.0], ..., [2.0]]	[[5.0, 25.0, 255.1], [185.35, 50.05, ... 3.65], ..., [5.0]]	[593.84, 587.48, ... 487.44]	["2020-02-09 21:46:04", "2020-02-09 21:46:04", ... "2020-02-09 21:46:04"]	6

5.2.3.2.4. Tạo Đặc Trung total_videos_for_user và video_watched_percentage

- Tải và chuẩn bị course_info_limit: Tải dữ liệu course_info_limit.parquet, một tệp chứa thông tin về các khóa học, bao gồm ID khóa học (clean_course_id) và danh sách các video (ccids) trong mỗi khóa học. Sau đó, nó chuyển đổi dữ liệu này thành một từ điển (course_video_count) nơi khóa là ID khóa học và giá trị là tổng số video có trong khóa học đó.
- Tạo cột unique_course: Tạo một bản sao của cột course_of_watched_video và đặt tên là unique_course. Đây là một bước đơn giản để đảm bảo tên cột rõ ràng cho việc sử dụng sau này, đặc biệt khi course_of_watched_video đã là một giá trị chuỗi duy nhất cho mỗi hàng (do bước nhóm trước đó).
- Tính toán total_videos_for_user: Sử dụng từ điển course_video_count đã tạo để tra cứu và gán tổng số video có trong khóa học tương ứng cho mỗi hàng trong DataFrame. Hàm total_video_count được định nghĩa để thực hiện việc tra cứu này, đảm bảo rằng nếu một khóa học không tìm thấy trong từ điển (ví dụ: dữ liệu không khớp), nó sẽ trả về 0. Kết quả được lưu vào cột mới total_videos_for_user.

Ý nghĩa của đặc trưng mới: total_videos_for_user cho biết quy mô tổng thể của khóa học mà người dùng đang tương tác. Nó là một yếu tố quan trọng để chuẩn hóa hoặc so sánh hành vi xem giữa các khóa học có độ dài hoặc số lượng nội dung khác nhau.

- Tính toán video_watched_percentage: Tính toán tỷ lệ phần trăm video đã xem trong một khóa học bằng cách lấy video_watch_count (số video duy nhất mà người dùng đã xem trong khóa học) chia cho total_videos_for_user (tổng số video có trong khóa học đó), sau đó nhân với 100.

Ý nghĩa của đặc trưng mới: video_watched_percentage là một chỉ số hiệu quả để đo lường mức độ hoàn thành hoặc cam kết của người dùng đối với một khóa học cụ thể. Giá trị này nằm trong khoảng từ 0 đến 100, cung cấp một cái nhìn

chuẩn hóa về tiến độ của người học, không phụ thuộc vào tổng số video của khóa học. Đặc trưng này rất hữu ích cho việc phân đoạn người dùng, đánh giá mức độ tương tác và dự đoán sự hoàn thành khóa học.

speed	watch_time_seg	video_length	enroll_time	video_watch_count	unique_course	total_videos_for_user	video_watched_percentage
list[[list[f64]]]	list[[list[f64]]]	list[f64]	list[str]	u32	str	i64	f64
[[1.0, 1.0]]	[[98.1, 30.0]]	[474.584]	["2020-08-12 08:42:38"]	1	"1829948"	38	2.631579
[[1.0, 1.0, 2.0], [2.0, 2.0, ... 2.0], ... [2.0]]	[[5.0, 25.0, 255.1], [185.35, 50.05, ... 3.65], ... [5.0]]	[593.84, 587.48, ... 487.44]	["2020-02-09 21:46:04", "2020-02-09 21:46:04", ... "2020-02-09 21:46:04"]	6	"697791"	114	5.263158
[[1.0], [1.0, 1.0, ... 1.0], ... [1.0, 1.0, 1.0]]	[[262.0], [5.0, 70.2, ... 42.9], ... [409.8, 36.7, 145.0]]	[399.19, 653.17, ... 653.1]	["2020-08-01 08:18:48", "2020-08-01 08:18:48", ... "2020-08-01 08:18:48"]	4	"1741511"	38	10.526316

5.2.3.2.5. Tạo Đặc Trung max_watch_per_video, watch_percentages, và video_percentage_watch_time

- Tính toán max_watch_per_video:

Duyệt qua cột duration_seg (chứa danh sách lồng ghép các thời lượng segment của mỗi video đã xem). Với mỗi danh sách thời lượng segment tương ứng với một video, nó tìm ra thời lượng xem lớn nhất trong số các segment đó. Nếu không có segment nào, giá trị sẽ là 0. Kết quả được lưu vào cột mới max_watch_per_video.

Ý nghĩa của đặc trưng: max_watch_per_video cho biết phân đoạn video dài nhất mà người dùng đã xem trong một video cụ thể. Đặc trưng này giúp nhận diện những khoảnh khắc hoặc phần nội dung thu hút sự chú ý của người dùng nhiều nhất, gợi ý về các điểm quan trọng hoặc hấp dẫn trong video.

- Tính toán watch_percentages:

So sánh max_watch_per_video với video_length (tổng thời lượng của video đó). Với mỗi video, nó tính toán tỷ lệ phần trăm thời gian xem lớn nhất của một segment so với tổng thời lượng video. Nếu video_length là 0, tỷ lệ sẽ là 0 để tránh lỗi chia cho 0. Kết quả được lưu vào cột mới watch_percentages.

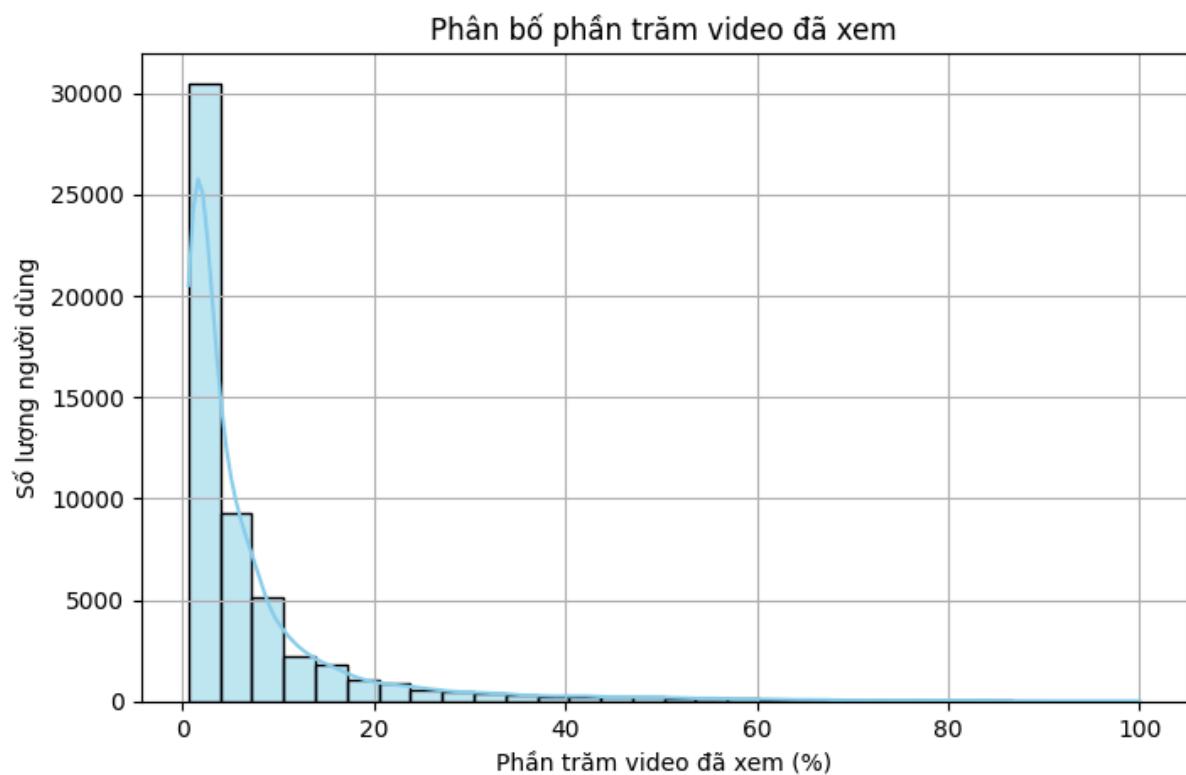
Ý nghĩa của đặc trưng: watch_percentages đại diện cho mức độ bao phủ tối đa của một segment đã xem trên toàn bộ video. Đặc trưng này giúp xác định những video mà người dùng có xu hướng xem một phần đáng kể tại một thời điểm, có thể là dấu hiệu của sự quan tâm đặc biệt hoặc chất lượng nội dung hấp dẫn tại một phân đoạn cụ thể.

- Tính toán video_percentage_watch_time:

Tính toán tỷ lệ phần trăm thời gian xem trung bình của tất cả các video trong một khóa học. Nó lấy trung bình của các giá trị trong cột watch_percentages và giới hạn giá trị tối đa ở 100% để đảm bảo tính hợp lý (vì không thể xem quá 100% video). Kết quả được lưu vào cột mới video_percentage_watch_time.

Ý nghĩa của đặc trưng: video_percentage_watch_time là một chỉ số mạnh mẽ cho thấy mức độ tham gia trung bình của người dùng đối với nội dung video trong một khóa học cụ thể. Nó phản ánh hiệu quả mà người dùng tương tác với các video, cung cấp cái nhìn tổng quan về sự cam kết và mức độ quan tâm của họ đến nội dung học tập.

unt	unique_course	total_videos_for_user	video_watched_percentage	max_watch_per_video	watch_percentages	video_percentage_watch_time
	str	i64	f64	list[f64]	list[f64]	f64
"1829948"	38	2.631579	[98.1]	[20.670735]	20.670735	
"697791"	114	5.263158	[510.2, 370.7, ... 10.0]	[85.915398, 63.10002, ... 2.051535]	26.321141	



5.2.3.2.6. Tạo Đặc Trung video_speed_avg, video_time_between_views_avg, và video_time_between_views_std

- Tính toán video_speed_avg:

Đoạn code này tính toán tốc độ phát lại trung bình của video cho mỗi người dùng trong ngữ cảnh của một khóa học. Nó duyệt qua tất cả các tốc độ phát lại (speed) của từng phân đoạn video mà người dùng đã xem, tính trung bình tốc độ cho từng video, sau đó lấy trung bình của các tốc độ trung bình đó.

Ý nghĩa của đặc trưng mới: `video_speed_avg` phản ánh thói quen tổng thể của người dùng về việc điều chỉnh tốc độ xem video. Giá trị lớn hơn 1.0 cho thấy người dùng có xu hướng xem nhanh hơn bình thường, trong khi giá trị nhỏ hơn 1.0 có thể chỉ ra việc xem chậm lại. Đặc trưng này hữu ích để phân tích sự vội vàng, tập trung, hoặc khó khăn trong việc nắm bắt nội dung của người học.

- Tính toán `video_time_between_views`:

Đoạn code này tính toán khoảng thời gian "nghỉ" hoặc "bỏ qua" giữa các phân đoạn video liên tiếp trong cùng một video. Nó lấy điểm bắt đầu của phân đoạn tiếp theo trừ đi điểm kết thúc của phân đoạn hiện tại. Nếu có nhiều hơn một phân đoạn, nó sẽ tính toán khoảng cách này; nếu chỉ có một phân đoạn, nó gán 0.0. Kết quả là một danh sách lồng ghép các khoảng thời gian gián đoạn cho mỗi video.

- Tính toán `video_time_between_views_avg` và `video_time_between_views_std`:
- Từ cột `video_time_between_views` vừa tạo, đoạn code này tính toán hai đặc trưng quan trọng:
 - `video_time_between_views_avg`: Đây là thời gian gián đoạn trung bình giữa các lần xem liên tiếp trong tất cả các video của một khóa học.
 - `video_time_between_views_std`: Đây là độ lệch chuẩn của thời gian gián đoạn giữa các lần xem liên tiếp, phản ánh sự biến động trong các khoảng nghỉ/bỏ qua của người dùng.
- Ý nghĩa của đặc trưng mới:

`video_time_between_views_avg` giúp hiểu mức độ liên tục trong hành vi xem video của người dùng. Giá trị cao có thể cho thấy người dùng thường xuyên tạm dừng, tua đi một đoạn dài, hoặc quay lại video sau một thời gian.

`video_time_between_views_std` cung cấp thông tin về sự nhất quán trong các khoảng gián đoạn này. Một độ lệch chuẩn cao có thể chỉ ra hành vi xem không đều, với cả những khoảng nghỉ ngắn và rất dài.

Kết hợp hai đặc trưng này giúp phân biệt giữa người dùng xem liền mạch và những người thường xuyên ngắt quãng, có thể là dấu hiệu của sự mất tập trung, tìm kiếm thông tin cụ thể, hoặc gấp khó khăn với nội dung.

jes	video_percentage_watch_time	video_speed_avg	video_time_between_views	video_time_between_views_avg	video_time_between_views_std
f64	f64	list[[f64]]	f64	f64	
20.670735	1.0	[[7.2]]	7.2	0.0	
26.321141	1.888889	[[5.0, 44.549], [-36.3, 69.2, 10.7], ... [0.0]]]	39.8149	50.034622	

5.2.3.2.7. Tạo Đặc Trung video_pause_count, video_pause_avg, và video_pause_std

- Tính toán video_pause_count:
 - Tính toán tổng số lần mà người dùng đã bắt đầu một phân đoạn xem video trong tất cả các video thuộc một khóa học. Về mặt ngữ nghĩa, mỗi lần start_point xuất hiện (tức là mỗi segment trong start_points của mỗi video), nó có thể được coi là một lần "tiếp tục" xem, thường là sau một lần tạm dừng hoặc bỏ qua. Bằng cách đếm tổng số start_point trong các danh sách lồng ghép của cột start_points, chúng ta có thể ước tính tổng số lần "ngắt quãng" (tức là số lần bắt đầu một segment mới sau một khoảng nghỉ).
 - Ý nghĩa của đặc trưng: video_pause_count cung cấp một chỉ số thô về tần suất người dùng tạm dừng hoặc bỏ qua các phần của video. Một số lượng cao có thể chỉ ra người dùng đang học theo nhịp độ riêng, tìm kiếm thông tin cụ thể, hoặc gặp khó khăn và cần dừng lại thường xuyên.
- Tính toán video_pause_avg:
 - Tính toán số lần ngắt quãng trung bình trên mỗi video mà người dùng đã xem trong một khóa học. Nó lấy tổng số start_point (từ video_pause_count) và chia cho số lượng video mà người dùng đã xem trong khóa học đó.
 - Ý nghĩa của đặc trưng: video_pause_avg chuẩn hóa tần suất ngắt quãng theo số lượng video, giúp so sánh hành vi giữa những người dùng xem nhiều video với những người xem ít hơn. Đặc trưng này phản ánh mức độ thường xuyên của hành vi tạm dừng/gián đoạn trong mỗi video.
- Tính toán video_pause_std:
 - Tính toán độ lệch chuẩn của số lần ngắt quãng giữa các video khác nhau trong một khóa học. Nó đo lường mức độ biến động trong số lần tạm dừng từ video này sang video khác.

- Ý nghĩa của đặc trưng: video_pause_std cho biết sự nhất quán trong hành vi tạm dừng của người dùng. Độ lệch chuẩn cao có thể chỉ ra rằng người dùng tạm dừng rất nhiều trong một số video nhưng lại xem liền mạch ở những video khác, có thể do nội dung hoặc độ khó của video. Ngược lại, độ lệch chuẩn thấp gợi ý hành vi tạm dừng nhất quán trên các video.

5.2.3.2.8. Tạo Đặc Trưng video_rewatch_avg và video_rewatch_std

- Tính toán video_rewatch_count:
- Sử dụng một hàm tùy chỉnh rewatch_counts để xác định số lần xem lại trong mỗi video. Hành vi xem lại được định nghĩa là khi điểm bắt đầu của một phân đoạn xem tiếp theo (starts[i]) nhỏ hơn điểm kết thúc của phân đoạn xem trước đó (ends[i - 1]). Điều này ngũ ý rằng người dùng đã tua lùi hoặc quay lại một phần video đã xem. Kết quả là một cột mới video_rewatch_count, chứa danh sách số lần xem lại cho từng video.
- Ý nghĩa của đặc trưng: video_rewatch_count cung cấp thông tin chi tiết về tần suất người dùng quay lại hoặc xem lại các phần cụ thể của video. Chỉ số này giúp nhận diện những video hoặc nội dung mà người dùng cảm thấy cần xem lại, có thể do độ phức tạp, sự hấp dẫn, hoặc nhu cầu cung cấp kiến thức.
- Tính toán video_rewatch_avg và video_rewatch_std:
- Từ cột video_rewatch_count vừa tạo, đoạn code này tiếp tục tính toán hai đặc trưng tổng hợp:
 - video_rewatch_avg: Đây là số lần xem lại trung bình trên mỗi video trong khóa học mà người dùng đã tương tác.
 - video_rewatch_std: Đây là độ lệch chuẩn của số lần xem lại trên mỗi video, phản ánh sự biến động của hành vi xem lại giữa các video.

Ý nghĩa của đặc trưng mới:

- video_rewatch_avg giúp định lượng mức độ trung bình của hành vi xem lại của người dùng. Giá trị cao có thể chỉ ra rằng người dùng đang vặt lộn với nội dung, hoặc đang tìm kiếm sự hiểu biết sâu hơn bằng cách xem lại các điểm chính.
- video_rewatch_std cung cấp cái nhìn về sự nhất quán trong hành vi xem lại. Một độ lệch chuẩn cao có thể gợi ý rằng người dùng chỉ xem lại một số video nhất định (có thể là những video khó hoặc quan trọng), trong khi một độ lệch chuẩn thấp cho thấy hành vi xem lại tương đồng đều trên các video.

<u>_time_between_views_std</u>	<u>video_pause_count</u>	<u>video_pause_avg</u>	<u>video_pause_std</u>	<u>video_rewatch_count</u>	<u>video_rewatch_avg</u>	<u>video_rewatch_std</u>
	i64	f64	f64	list[i64]	f64	f64
	2	2.0	0.0	[0]	0.0	0.0
4622	15	2.5	1.048809	[0, 1, ... 0]	0.166667	0.408248

5.2.3.2.9. Tạo Đặc Trung ent_seg và entropy_time

- Tính toán ent_seg:
 - Sử dụng hàm tùy chỉnh entropy_single để tính toán entropy của thời lượng các phân đoạn xem cho từng video. Entropy là một thước đo từ lý thuyết thông tin, cho biết mức độ "bất ngờ" hay "đa dạng" của một phân bố. Trong ngữ cảnh này, nếu thời lượng các segment trong một video đều nhau (ví dụ: người dùng xem tất cả các phần video với thời gian gần như bằng nhau), entropy sẽ thấp. Ngược lại, nếu thời lượng các segment rất khác nhau (người dùng xem một số phần rất lâu và bỏ qua nhanh các phần khác), entropy sẽ cao.
 - Ý nghĩa của đặc trưng mới: ent_seg (entropy cho từng video) đại diện cho sự đa dạng hoặc không đồng đều trong cách người dùng phân bổ thời gian xem qua các phân đoạn của một video cụ thể. Một giá trị entropy cao có thể chỉ ra rằng người dùng đang tập trung vào một số phần nhất định và bỏ qua những phần khác, trong khi giá trị thấp cho thấy họ xem khá đồng đều toàn bộ video.
- Tính toán entropy_time:
 - Tính toán entropy trung bình của thời gian xem trên tất cả các video mà người dùng đã tương tác trong một khóa học. Nó lấy các giá trị entropy của từng video từ cột ent_seg và sau đó tính giá trị trung bình của chúng.
 - Ý nghĩa của đặc trưng mới: entropy_time cung cấp một cái nhìn tổng quan về mức độ đồng đều trung bình trong hành vi phân bổ thời gian xem video của người dùng xuyên suốt một khóa học. Đặc trưng này có thể giúp phân loại người dùng: những người có entropy_time cao có thể là những người học "chọn lọc", chỉ tập trung vào các phần nội dung mà họ quan tâm nhất, trong khi những người có entropy_time thấp có thể là những người học có kỷ luật, xem tương đối đều đặn mọi phần.

ideo_pause_count	video_pause_avg	video_pause_std	video_rewatch_count	video_rewatch_avg	video_rewatch_std	ent_seg	entropy_time
34	f64	f64	list[f64]	f64	f64	list[f64]	f64
!	2.0	0.0	[0]	0.0	0.0	[0.785248]	0.785248
5	2.5	1.048809	[0, 1, ... 0]	0.166667	0.408248	[0.345558, 1.176227, ... 0.0]	0.851125

5.2.3.3. Trích xuất đặc trưng từ dữ liệu user-course

5.2.3.3.1. Trích xuất đặc trưng sẵn có từ dữ liệu course

Dựa vào dữ liệu của course, nhóm trích xuất được các đặc trưng sau để thực hiện cho bài toán:

Thông tin khóa học (course_*)	course_id	Mã khóa học
	num_prerequisites	Số lượng môn học tiên quyết
	field_x	Lĩnh vực khóa học (optional)
	num_field_x	Số lượng lĩnh vực liên quan

5.2.3.3.2. Tạo đặc trưng start_date, end_date và duration_days.

Nhóm dựa vào thông tin về khóa học trên nền tảng XueTangx để lấy dữ liệu về thời gian bắt đầu và kết thúc của khóa học mà học viên đang theo học

The screenshot shows a course page on the XueTangX platform. At the top, there's a navigation bar with links for Home, Courses, HDI, IVEC, GOC, ICE Institute, and More. On the right side of the header, there are search, language selection (English), and user profile icons. The main content area features a large image of a city skyline at sunset. Overlaid on the image are several text elements: '电子政务' (Electronic Government) in large Chinese characters, 'Offered by Tsinghua University. Instructors: Qingguo Meng, Nan Zhang.', a dropdown menu showing '2025春' (Spring 2025), the enrollment period '2025-01-15 - 2025-07-22', and a note '20178 already enrolled'. In the bottom right corner of the image area, there's a button labeled 'Go to study'.

5.2.3.3.3. Trích xuất đặc trưng từ resource của course.

Sau khi thực hiện quá trình xử lý file course, nhóm trích xuất được các đặc trưng sau:

Nhóm	Tên cột	Ý nghĩa
Tài liệu khóa học (resource_*)	video_count	Tổng số lượng video
	exercise_count	Tổng số lượng bài tập
	chapter_count	Tổng số lượng chương

5.2.3.3.4. Tạo đặc trưng thành phần điểm của khóa học.

Nhóm dựa vào thông tin về khóa học trên nền tảng XueTangx để lấy dữ liệu về thành phần điểm của khóa học, bao gồm các thành phần và các cột sau:

Nhóm	Tên cột	Ý nghĩa
Thành phần điểm (score_*)	assignment	Điểm bài tập
	exam	Điểm bài thi cuối kỳ
	video	Điểm từ xem video
	certificate	Có nhận chứng chỉ hay không (0, 1)

The screenshot shows a student's score summary on the left and a detailed scoring breakdown on the right.

Total score: Score 0 (Total 100 points). A progress bar shows 0 points, with F at 0 points and A at 100 points. A message says "Keep going, you need 60 points to the next level".

Scoring details (Total 100 points):

Category	Points	Proportion	Status	Score
Assignment	40	40%	Completed 0%	0.00
Discussion	10	10%	Completed 0%	0.00
Exam	10	10%	Completed 0%	0.00
Reading	10	10%	Completed 0%	0.00
Video	30	30%	Completed 0%	0.00

Nhóm vẫn duy trì phương pháp tính điểm như trên nền tảng **XuetangX**, vốn sử dụng mô hình đánh giá dựa trên nhiều thành phần để phản ánh toàn diện quá trình học tập

của học viên. Tuy nhiên, để phục vụ mục đích gán nhãn và phân tích dữ liệu học tập, nhóm xác định rõ **3 thành phần điểm chính** như sau:

Thành phần điểm:

Thành phần	Bao gồm	Ghi chú thêm
Assignment	Bài tập trắc nghiệm, tự luận, câu hỏi chương	Gộp cả điểm Discussion (nếu có)
Exam	Chỉ tính điểm Final Exam	Bỏ qua điểm giữa kỳ nếu có
Video	Tỉ lệ xem video, mức độ hoàn thành tài liệu học	Gộp thêm phần Reading (nếu có)

5.2.3.3.5. Trích xuất đặc trưng sẵn có từ dữ liệu user

Sau khi thực hiện quá trình xử lý file user, nhóm trích xuất được các đặc trưng sau:

Nhóm	Tên cột	Ý nghĩa
Thông tin người dùng (user_*)	user_id	ID người dùng
	school	Trường học
	user_enroll_time	Thời gian người dùng đăng ký khóa học

5.2.3.3.6. Tạo đặc trưng user_past_course_count và user_time_since_last_course

Nhóm dựa vào việc phân tích file user-course và khoảng thời gian mà nhóm xét cho bài toán để trích xuất ra 2 đặc trưng sau, gồm:

Nhóm	Tên cột	Ý nghĩa
Thông tin người dùng (user_*)	user_past_course_count	Số lượng khóa học đã đăng ký trước đó

	user_time_since_last_course	Khoảng cách (giờ) từ lần đăng ký hiện tại đến lần gần nhất
	remaining_time	Khoảng thời gian còn lại đến khi khóa học kết thúc

5.2.3.4. Trích xuất đặc trưng từ dữ liệu user-problem

5.2.3.4.1. Tạo đặc trưng liên quan đến số lượng và phạm vi bài tập

- a. **Đặc trưng exercise_id_count_{i}: Số lượng bài tập được làm.**

Cách tạo:

- Dữ liệu được chia thành 4 giai đoạn (Phase 1: ≤14 ngày, Phase 2: 15-28 ngày, Phase 3: 29-42 ngày, Phase 4: 43-56 ngày) dựa trên cột exercise_date_from_enroll.
- Trong mỗi giai đoạn i, nhóm dữ liệu theo user_id và course_id, sau đó đếm số lượng exercise_id duy nhất mà người dùng đã thực hiện. Kết quả được lưu vào cột exercise_id_count_{i}.

Ý nghĩa:

- Đặc trưng này thể hiện số lượng bài tập (exercise) mà người dùng đã thực hiện trong giai đoạn i.
- Giúp đánh giá mức độ tham gia của người dùng vào các bài tập trong từng giai đoạn, từ đó nhận diện xu hướng học tập (ví dụ: người dùng có tích cực làm bài tập ở giai đoạn đầu hay không).

- b. **Đặc trưng exercise_num_problem_sum_{i}: Tổng số câu hỏi trong tất cả bài tập.**

Cách tạo:

- Trong mỗi giai đoạn i, nhóm dữ liệu theo user_id và course_id, sau đó tính tổng số câu hỏi (problem_count) trong tất cả các bài tập mà người dùng đã thực hiện. Kết quả được lưu vào cột exercise_num_problem_sum_{i}.

Ý nghĩa:

- Thể hiện tổng số câu hỏi mà người dùng đã đối mặt trong giai đoạn i.
- Đặc trưng này giúp đánh giá khái lượng bài tập mà người dùng đã tiếp cận, từ đó hiểu được mức độ nỗ lực và phạm vi nội dung đã được bao phủ.

- c. **Đặc trưng exercise_num_problem_mean_{i}: Số câu hỏi trung bình mỗi bài.**

Cách tạo:

- Trong mỗi giai đoạn i, nhóm dữ liệu theo user_id và course_id, sau đó tính giá trị trung bình của số lượng câu hỏi (problem_count) trên mỗi bài tập. Kết quả được lưu vào cột exercise_num_problem_mean_{i}.

Ý nghĩa:

- Biểu thị số câu hỏi trung bình trong mỗi bài tập mà người dùng đã thực hiện trong giai đoạn i.
- Giúp nhận diện mức độ phức tạp trung bình của các bài tập mà người dùng thực hiện, từ đó đánh giá sự thay đổi về độ khó qua các giai đoạn.

- d. **Đặc trưng exercise_context_sum_{i}: Tổng số ngữ cảnh làm bài tập.**

Cách tạo:

- Trong mỗi giai đoạn i, nhóm dữ liệu theo user_id và course_id, sau đó tính tổng số ngữ cảnh (num_context_ids) liên quan đến các bài tập. Kết quả được lưu vào cột exercise_context_sum_{i}.

Ý nghĩa:

- Thể hiện tổng số ngữ cảnh mà người dùng đã tiếp cận trong các bài tập ở giai đoạn i.
 - Phản ánh mức độ đa dạng của nội dung bài tập mà người dùng đã làm, giúp đánh giá phạm vi học tập.
- e. **Đặc trưng exercise_context_mean_{i}: Trung bình số ngữ cảnh mỗi bài tập.**

Cách tạo:

- Trong mỗi giai đoạn i, nhóm dữ liệu theo user_id và course_id, sau đó tính giá trị trung bình của số ngữ cảnh (num_context_ids) trên mỗi bài tập. Kết quả được lưu vào cột exercise_context_mean_{i}.

Ý nghĩa:

- Biểu thị số ngữ cảnh trung bình của mỗi bài tập trong giai đoạn i.
- Giúp đánh giá mức độ đa dạng trung bình của nội dung bài tập, từ đó hiểu được tính chất của các bài tập mà người dùng thực hiện.

5.2.3.4.2. Tạo đặc trưng liên quan đến hiệu suất trả lời đúng

- a. Đặc trưng exercise_correct_sum_{i}: Tổng số câu trả lời đúng.**

Cách tạo:

- Trong mỗi giai đoạn i, nhóm dữ liệu theo user_id và course_id, sau đó tính tổng số câu trả lời đúng (is_correct) trong tất cả các bài tập. Kết quả được lưu vào cột exercise_correct_sum_{i}.

Ý nghĩa:

- Thể hiện tổng số câu trả lời đúng của người dùng trong giai đoạn i.
- Đặc trưng này phản ánh hiệu suất học tập tổng thể của người dùng, giúp đánh giá mức độ hiểu bài và khả năng hoàn thành bài tập chính xác.

- b. Đặc trưng exercise_correct_mean_{i}: Tỷ lệ trả lời đúng trung bình mỗi bài.**

Cách tạo:

- Trong mỗi giai đoạn i, nhóm dữ liệu theo user_id và course_id, sau đó tính giá trị trung bình của tỷ lệ trả lời đúng (is_correct) trên mỗi bài tập. Kết quả được lưu vào cột exercise_correct_mean_{i}.

Ý nghĩa:

- Biểu thị tỷ lệ trả lời đúng trung bình của mỗi bài tập trong giai đoạn i.
- Giúp đánh giá mức độ chính xác trung bình của người dùng trên từng bài tập, từ đó nhận diện các giai đoạn mà người dùng gặp khó khăn hoặc cải thiện.

- c. Đặc trưng exercise_perc_goal_correct_sum_{i}: Tổng tỷ lệ trả lời đúng.**

Cách tạo:

- Trong mỗi giai đoạn i, nhóm dữ liệu theo user_id và course_id, sau đó tính tổng của tỷ lệ trả lời đúng (percentage_correct). Kết quả được lưu vào cột exercise_perc_goal_correct_sum_{i}.

Ý nghĩa:

- Thể hiện tổng tỷ lệ trả lời đúng của các bài tập trong giai đoạn i.
- Phản ánh hiệu suất tổng thể của người dùng trong việc đạt được mục tiêu trả lời đúng, giúp đánh giá sự tiến bộ qua các giai đoạn.

d. Đặc trung exercise_perc_goal_correct_mean_{i}: Trung bình tỷ lệ trả lời đúng.

Cách tạo:

- Trong mỗi giai đoạn i, nhóm dữ liệu theo user_id và course_id, sau đó tính giá trị trung bình của tỷ lệ trả lời đúng (percentage_correct). Kết quả được lưu vào cột exercise_perc_goal_correct_mean_{i}.

Ý nghĩa:

- Biểu thị tỷ lệ trả lời đúng trung bình của mỗi bài tập trong giai đoạn i.
- Giúp đánh giá mức độ chính xác trung bình của người dùng, từ đó nhận diện sự ổn định hoặc thay đổi trong hiệu suất.

e. Đặc trung exercise_perc_real_correct_sum_{i}: Tổng tỷ lệ đúng thực tế.

Cách tạo:

- Trong mỗi giai đoạn i, nhóm dữ liệu theo user_id và course_id, sau đó tính tổng tỷ lệ đúng thực tế (percentage_correct_completed). Kết quả được lưu vào cột exercise_perc_real_correct_sum_{i}.

Ý nghĩa:

- Thể hiện tổng tỷ lệ đúng thực tế của các bài tập trong giai đoạn i.
- Phản ánh hiệu suất thực tế của người dùng trong việc trả lời đúng, giúp đánh giá khả năng học tập thực tế qua các giai đoạn.

f. Đặc trung exercise_perc_real_correct_mean_{i}: Tỷ lệ đúng thực tế trung bình.

Cách tạo:

- Trong mỗi giai đoạn i, nhóm dữ liệu theo user_id và course_id, sau đó tính giá trị trung bình của tỷ lệ đúng thực tế (percentage_correct_completed). Kết quả được lưu vào cột exercise_perc_real_correct_mean_{i}.

Ý nghĩa:

- Biểu thị tỷ lệ đúng thực tế trung bình của mỗi bài tập trong giai đoạn i.
- Giúp đánh giá mức độ chính xác thực tế trung bình của người dùng, từ đó nhận diện sự tiến bộ hoặc khó khăn trong học tập.

- g. Đặc trưng exercise_perc_real_correct_std_{i}: Độ lệch chuẩn tỷ lệ đúng thực tế.**

Cách tạo:

- Trong mỗi giai đoạn i, nhóm dữ liệu theo user_id và course_id, sau đó tính độ lệch chuẩn của tỷ lệ đúng thực tế (percentage_correct_completed). Kết quả được lưu vào cột exercise_perc_real_correct_std_{i}.

Ý nghĩa:

- Thể hiện độ lệch chuẩn của tỷ lệ đúng thực tế trong giai đoạn i.
- Phản ánh sự biến thiên trong hiệu suất trả lời đúng, giúp đánh giá tính ổn định của người dùng trong việc đạt kết quả chính xác.

5.2.3.4.3. Tạo đặc trưng liên quan đến điểm số đạt được

- a. Đặc trưng exercise_perc_goal_score_sum_{i}: Tổng điểm đạt được.**

Cách tạo:

- Trong mỗi giai đoạn i, nhóm dữ liệu theo user_id và course_id, sau đó tính tổng điểm đạt được (percentage_score) trong tất cả các bài tập. Kết quả được lưu vào cột exercise_perc_goal_score_sum_{i}.

Ý nghĩa:

- Thể hiện tổng điểm đạt được của người dùng trong giai đoạn i.
- Phản ánh hiệu suất tổng thể dựa trên điểm số, giúp đánh giá mức độ hoàn thành mục tiêu điểm số qua các giai đoạn.

- b. Đặc trưng exercise_perc_goal_score_mean_{i}: Trung bình điểm đạt được mỗi bài.**

Cách tạo:

- Trong mỗi giai đoạn i, nhóm dữ liệu theo user_id và course_id, sau đó tính giá trị trung bình của điểm đạt được (percentage_score) trên mỗi bài tập. Kết quả được lưu vào cột exercise_perc_goal_score_mean_{i}.

Ý nghĩa:

- Biểu thị điểm trung bình đạt được của mỗi bài tập trong giai đoạn i.
- Giúp đánh giá hiệu suất trung bình của người dùng dựa trên điểm số, từ đó nhận diện sự thay đổi trong chất lượng học tập.

c. Đặc trưng exercise_perc_real_score_sum_{i}: Tổng điểm thực tế.

Cách tạo:

- Trong mỗi giai đoạn i, nhóm dữ liệu theo user_id và course_id, sau đó tính tổng điểm thực tế đạt được (percentage_score_completed). Kết quả được lưu vào cột exercise_perc_real_score_sum_{i}.

Ý nghĩa:

- Thể hiện tổng điểm thực tế đạt được của người dùng trong giai đoạn i.
- Phản ánh hiệu suất thực tế dựa trên điểm số, giúp đánh giá mức độ thành công của người dùng trong việc đạt điểm cao.

d. Đặc trưng exercise_perc_real_score_mean_{i}: Trung bình điểm thực tế.

Cách tạo:

- Trong mỗi giai đoạn i, nhóm dữ liệu theo user_id và course_id, sau đó tính giá trị trung bình của điểm thực tế đạt được (percentage_score_completed). Kết quả được lưu vào cột exercise_perc_real_score_mean_{i}.

Ý nghĩa:

- Biểu thị điểm thực tế trung bình của mỗi bài tập trong giai đoạn i.
- Giúp đánh giá hiệu suất trung bình thực tế của người dùng, từ đó nhận diện sự tiến bộ hoặc suy giảm qua các giai đoạn.

e. Đặc trưng exercise_perc_real_score_std_{i}: Độ lệch chuẩn điểm thực tế.

Cách tạo:

- Trong mỗi giai đoạn i, nhóm dữ liệu theo user_id và course_id, sau đó tính độ lệch chuẩn của điểm thực tế đạt được (percentage_score_completed). Kết quả được lưu vào cột exercise_perc_real_score_std_{i}.

Ý nghĩa:

- Thể hiện độ lệch chuẩn của điểm thực tế trong giai đoạn i.
- Phản ánh sự biến thiên trong hiệu suất điểm số, giúp đánh giá tính ổn định của người dùng trong việc đạt điểm cao.

5.2.3.4.4. Tạo đặc trưng liên quan đến mức độ hoàn thành

- a. **Đặc trưng exercise_perc_real_completed_sum_{i}**: **Tổng tỷ lệ hoàn thành thực tế**.

Cách tạo:

- Trong mỗi giai đoạn i, nhóm dữ liệu theo user_id và course_id, sau đó tính tổng tỷ lệ hoàn thành thực tế (percentage_completed) của các bài tập. Kết quả được lưu vào cột exercise_perc_real_completed_sum_{i}.

Ý nghĩa:

- Thể hiện tổng tỷ lệ hoàn thành thực tế của các bài tập trong giai đoạn i.
- Phản ánh mức độ hoàn thành bài tập của người dùng, giúp đánh giá sự kiên trì và khả năng hoàn thành mục tiêu học tập.

- b. **Đặc trưng exercise_perc_real_completed_mean_{i}**: **Trung bình tỷ lệ hoàn thành thực tế**.

Cách tạo:

- Trong mỗi giai đoạn i, nhóm dữ liệu theo user_id và course_id, sau đó tính giá trị trung bình của tỷ lệ hoàn thành thực tế (percentage_completed). Kết quả được lưu vào cột exercise_perc_real_completed_mean_{i}.

Ý nghĩa:

- Biểu thị tỷ lệ hoàn thành thực tế trung bình của mỗi bài tập trong giai đoạn i.
 - Giúp đánh giá mức độ hoàn thành trung bình của người dùng, từ đó nhận diện các giai đoạn mà người dùng có xu hướng bỏ dở bài tập.
- c. **Đặc trưng exercise_perc_real_completed_std_{i}**: **Độ lệch chuẩn tỷ lệ hoàn thành thực tế**.

Cách tạo:

- Trong mỗi giai đoạn i, nhóm dữ liệu theo user_id và course_id, sau đó tính độ lệch chuẩn của tỷ lệ hoàn thành thực tế (percentage_completed). Kết quả được lưu vào cột exercise_perc_real_completed_std_{i}.

Ý nghĩa:

- Thể hiện độ lệch chuẩn của tỷ lệ hoàn thành thực tế trong giai đoạn i.

- Phản ánh sự biến thiên trong mức độ hoàn thành bài tập, giúp đánh giá tính ổn định của người dùng trong việc hoàn thành bài tập.

5.2.3.4.5. Tạo đặc trưng liên quan đến số lần làm bài

- a. **Đặc trưng exercise_attempts_sum_sum_{i}: Tổng số lần thử.**

Cách tạo:

- Trong mỗi giai đoạn i, nhóm dữ liệu theo user_id và course_id, sau đó tính tổng số lần thử (attempts) của tất cả các bài tập. Kết quả được lưu vào cột exercise_attempts_sum_sum_{i}.

Ý nghĩa:

- Thể hiện tổng số lần thử mà người dùng đã thực hiện để hoàn thành các bài tập trong giai đoạn i.
- Phản ánh mức độ kiên trì và nỗ lực của người dùng, đồng thời có thể chỉ ra các giai đoạn mà người dùng gặp khó khăn (số lần thử cao).

- b. **Đặc trưng exercise_attempts_sum_mean_{i}: Số lần thử trung bình mỗi bài.**

Cách tạo:

- Trong mỗi giai đoạn i, nhóm dữ liệu theo user_id và course_id, sau đó tính giá trị trung bình của tổng số lần thử (attempts) trên mỗi bài tập. Kết quả được lưu vào cột exercise_attempts_sum_mean_{i}.

Ý nghĩa:

- Biểu thị số lần thử trung bình trên mỗi bài tập trong giai đoạn i.
 - Giúp đánh giá mức độ khó khăn trung bình của các bài tập mà người dùng đối mặt, từ đó nhận diện các giai đoạn mà người dùng cần nhiều nỗ lực hơn.
- c. **Đặc trưng exercise_attempts_mean_mean_{i}: Trung bình số lần thử mỗi câu hỏi.**

Cách tạo:

- Trong mỗi giai đoạn i, nhóm dữ liệu theo user_id và course_id, sau đó tính giá trị trung bình của số lần thử trung bình trên mỗi câu hỏi (attempts chia cho số câu hỏi). Kết quả được lưu vào cột exercise_attempts_mean_mean_{i}.

Ý nghĩa:

- Thể hiện trung bình số lần thử trên mỗi câu hỏi trong giai đoạn i.
- Đặc trưng này cung cấp cái nhìn chi tiết về mức độ khó khăn của từng câu hỏi, giúp nhận diện hành vi học tập (ví dụ: người dùng có xu hướng thử nhiều lần do gặp khó khăn không).

5.2.3.4.6. Tạo đặc trưng liên quan đến thời gian làm bài

- a. Đặc trưng exercise_date_from_enroll_min_{i}: Ngày đầu tiên làm bài.**

Cách tạo:

- Trong mỗi giai đoạn i, nhóm dữ liệu theo user_id và course_id, sau đó xác định giá trị nhỏ nhất của exercise_date_from_enroll (số ngày từ ngày đăng ký đến ngày làm bài tập). Kết quả được lưu vào cột exercise_date_from_enroll_min_{i}.

Ý nghĩa:

- Thể hiện ngày đầu tiên mà người dùng bắt đầu làm bài tập trong giai đoạn i (so với ngày đăng ký).
- Giúp đánh giá thời điểm bắt đầu tương tác với bài tập, từ đó nhận diện mức độ tích cực hoặc trì hoãn của người dùng.

- b. Đặc trưng exercise_date_from_enroll_mean_{i}: Ngày trung bình làm bài.**

Cách tạo:

- Trong mỗi giai đoạn i, nhóm dữ liệu theo user_id và course_id, sau đó tính giá trị trung bình của exercise_date_from_enroll. Kết quả được lưu vào cột exercise_date_from_enroll_mean_{i}.

Ý nghĩa:

- Biểu thị ngày trung bình mà người dùng làm bài tập trong giai đoạn i (so với ngày đăng ký).
- Cung cấp thông tin về thời gian trung bình mà người dùng tham gia làm bài tập, giúp phân tích sự đều đặn trong tiến độ học tập.

- c. Đặc trưng exercise_date_from_enroll_max_{i}: Ngày cuối cùng làm bài.**

Cách tạo:

- Trong mỗi giai đoạn i, nhóm dữ liệu theo user_id và course_id, sau đó xác định giá trị lớn nhất của exercise_date_from_enroll. Kết quả được lưu vào cột exercise_date_from_enroll_max_{i}.

Ý nghĩa:

- Thể hiện ngày cuối cùng mà người dùng làm bài tập trong giai đoạn i (so với ngày đăng ký).
- Giúp đánh giá thời gian kết thúc hoạt động làm bài tập, từ đó nhận diện mức độ duy trì tương tác qua các giai đoạn.

d. Đặc trưng exercise_diff_sum_{i}: Tổng thời gian làm bài.

Cách tạo:

- Trong mỗi giai đoạn i, nhóm dữ liệu theo user_id và course_id, sau đó tính tổng thời gian làm bài tập (submit_time_diff_hours) trong tất cả các bài tập. Kết quả được lưu vào cột exercise_diff_sum_{i}.

Ý nghĩa:

- Thể hiện tổng thời gian mà người dùng đã dành để làm bài tập trong giai đoạn i.
- Phản ánh mức độ đầu tư thời gian của người dùng, giúp đánh giá sự nghiêm túc và nỗ lực học tập.

e. Đặc trưng exercise_diff_mean_{i}: Trung bình thời gian làm mỗi bài.

Cách tạo:

- Trong mỗi giai đoạn i, nhóm dữ liệu theo user_id và course_id, sau đó tính giá trị trung bình của thời gian làm bài tập (submit_time_diff_hours) trên mỗi bài tập. Kết quả được lưu vào cột exercise_diff_mean_{i}.

Ý nghĩa:

- Biểu thị thời gian trung bình để hoàn thành mỗi bài tập trong giai đoạn i.
- Giúp đánh giá tốc độ làm bài tập của người dùng, từ đó nhận diện các giai đoạn mà người dùng làm bài nhanh hay chậm.

f. Đặc trưng exercise_diff_min_{i}: Thời gian làm bài ít nhất.

Cách tạo:

- Trong mỗi giai đoạn i, nhóm dữ liệu theo user_id và course_id, sau đó xác định giá trị nhỏ nhất của thời gian làm bài tập (submit_time_diff_hours). Kết quả được lưu vào cột exercise_diff_min_{i}.

Ý nghĩa:

- Thể hiện thời gian làm bài ít nhất trong giai đoạn i.
- Phản ánh khả năng làm bài nhanh nhất của người dùng, giúp nhận diện các bài tập dễ hoặc thời điểm người dùng làm việc hiệu quả.

g. Đặc trưng exercise_diff_max_{i}: Thời gian làm bài nhiều nhất.

Cách tạo:

- Trong mỗi giai đoạn i, nhóm dữ liệu theo user_id và course_id, sau đó xác định giá trị lớn nhất của thời gian làm bài tập (submit_time_diff_hours). Kết quả được lưu vào cột exercise_diff_max_{i}.

Ý nghĩa:

- Thể hiện thời gian làm bài nhiều nhất trong giai đoạn i.
- Phản ánh bài tập khó nhất hoặc thời điểm người dùng gấp khó khăn, giúp đánh giá sự thay đổi về độ khó qua các giai đoạn.

h. Đặc trưng exercise_hour_entropy_{i}: Entropy thời gian làm bài theo giờ.

Cách tạo:

- Trong mỗi giai đoạn i, nhóm dữ liệu theo user_id và course_id, sau đó tổng hợp danh sách giờ làm bài tập (exercise_hours). Sử dụng hàm compute_entropy_from_hour_bins để tính entropy của thời gian làm bài tập theo giờ. Kết quả được lưu vào cột exercise_hour_entropy_{i}.

Ý nghĩa:

- Thể hiện độ phân tán của thời gian làm bài tập (theo giờ) trong ngày ở giai đoạn i.
- Giúp đánh giá thói quen học tập của người dùng (ví dụ: họ làm bài tập tập trung vào một khung giờ hay phân tán trong ngày), từ đó nhận diện các mẫu hành vi thời gian.

5.2.3.4.7. Tạo đặc trưng liên quan đến ngôn ngữ bài tập

Đặc trưng: exercise_language_binary_mean_{i}: Trung bình giá trị ngôn ngữ bài tập (1 = Tiếng Anh, 0 = Tiếng Trung).

Cách tạo:

- Trong mỗi giai đoạn i, nhóm dữ liệu theo user_id và course_id, sau đó tính giá trị trung bình của cột language_binary (1 cho Tiếng Anh, 0 cho Tiếng Trung hoặc NaN). Kết quả được lưu vào cột exercise_language_binary_mean_{i}.

Ý nghĩa:

- Thể hiện tỷ lệ trung bình các bài tập bằng Tiếng Anh trong giai đoạn i.
- Giúp đánh giá sở thích hoặc khả năng ngôn ngữ của người dùng, từ đó nhận diện ảnh hưởng của ngôn ngữ đến hiệu suất học tập.

5.2.3.5. Tổng hợp toàn bộ các đặc trưng của bài toán

Nhóm	Tên cột	Ý nghĩa
Thông tin khóa học (course_*)	course_id	Mã khóa học
	num_prerequisites	Số lượng môn học tiên quyết
	field_x	Lĩnh vực khóa học (optional)
	num_field_x	Số lượng lĩnh vực liên quan
	start_date	Ngày bắt đầu khóa học
	end_date	Ngày kết thúc khóa học
	duration_days	Độ dài của khóa học (ngày)
Tài liệu khóa học (resource_*)	video_count	Tổng số lượng video
	exercise_count	Tổng số lượng bài tập
	chapter_count	Tổng số lượng chương
Thành phần điểm (score_*)	assignment	Điểm bài tập
	exam	Điểm bài thi cuối kỳ
	video	Điểm từ xem video

	certificate	Có nhận chứng chỉ hay không (0, 1)
Thông tin người dùng (user_*)	user_id	ID người dùng
	school	Trường học
	user_enroll_time	Thời gian người dùng đăng ký khóa học
	user_past_course_count	Số lượng khóa học đã đăng ký trước đó
	user_time_since_last_course	Khoảng cách (giờ) từ lần đăng ký hiện tại đến lần gần nhất
Hành vi học tập - Bình luận (comment_*) - Theo từng đợt	comment_count_phase{i}	Số lượng bình luận trong giai đoạn i
	total_words_phase{i}_x	Tổng số từ trong các bình luận trong giai đoạn i
	entropy_time_comment_phase{i}	Mức độ phân tán thời gian của bình luận (entropy) trong giai đoạn i
	total_positive{i}	Tổng bình luận và phản hồi tích cực của học viên trong giai đoạn i
	total_negative1	Tổng bình luận và phản hồi tiêu cực của học viên trong giai đoạn i
	total_neutral1	Tổng bình luận và phản hồi trung lập của học viên trong giai đoạn i
Hành vi học tập - Phản hồi (reply_*) - Theo từng đợt	reply_count_phase{i}	Số lượng phản hồi trong giai đoạn i

	total_words_phase{i}_y	Tổng số từ trong các phản hồi trong giai đoạn i
	entropy_time_reply_phase{i}	Mức độ phân tán thời gian của phản hồi (entropy) trong giai đoạn i
	total_positive{i}	Tổng bình luận và phản hồi tích cực của học viên trong giai đoạn i
	total_negative1	Tổng bình luận và phản hồi tiêu cực của học viên trong giai đoạn i
	total_neutral1	Tổng bình luận và phản hồi trung lập của học viên trong giai đoạn i
Hành vi học tập - Bài tập (exercise_*) - Theo từng đợt	exercise_id_count_{i}	Số lượng bài tập (exercise) được làm trong giai đoạn i
	exercise_correct_sum_{i}	Tổng số câu trả lời đúng trong giai đoạn i
	exercise_correct_mean_{i}	Tỷ lệ trả lời đúng trung bình của mỗi bài tập trong giai đoạn i
	exercise_num_problem_sum_{i}	Tổng số câu hỏi trong tất cả các bài tập trong giai đoạn i
	exercise_num_problem_mean_{i}	Số câu hỏi trung bình mỗi bài trong giai đoạn i
	exercise_attempts_sum_sum_{i}	Tổng số lần thử làm các bài tập trong giai đoạn i
	exercise_attempts_sum_mean_{i}	Số lần thử trung bình mỗi bài trong giai đoạn i
	exercise_attempts_mean_mean_{i}	Trung bình số lần thử mỗi câu hỏi (problem) trong giai đoạn i

	exercise_date_from_enroll_min_{i}	Ngày đầu tiên làm bài tập (so với ngày đăng ký) trong giai đoạn i
	exercise_date_from_enroll_mean_{i}	Ngày trung bình làm bài (so với ngày đăng ký) trong giai đoạn i
	exercise_date_from_enroll_max_{i}	Ngày cuối cùng làm bài (so với ngày đăng ký)
	exercise_context_sum_{i}	Tổng số ngữ cảnh làm bài tập trong giai đoạn i
	exercise_context_mean_{i}	Trung bình số ngữ cảnh làm bài tập trong giai đoạn i
	exercise_language_binary_mean_{i}	Trung bình giá trị ngôn ngữ bài tập (1 = Tiếng Anh, 0 = Tiếng Trung) trong giai đoạn i
	exercise_diff_sum_{i}	Tổng thời gian làm bài tập trong giai đoạn i
	exercise_diff_mean_{i}	Trung bình thời gian làm mỗi bài tập trong giai đoạn i
	exercise_diff_min_{i}	Thời gian làm bài ít nhất trong giai đoạn i
	exercise_diff_max_{i}	Thời gian làm bài nhiều nhất trong giai đoạn i
	exercise_perc_goal_correct_sum_{i}	Tổng của tỷ lệ trả lời đúng giai đoạn i
	exercise_perc_goal_correct_mean_{i}	Trung bình của tỷ lệ trả lời đúng trong giai đoạn i
	exercise_perc_goal_score_sum_{i}	Tổng điểm đạt được trong giai đoạn i

	exercise_perc_goal_score_mean_{i}	Trung bình điểm đạt được của mỗi bài tập trong giai đoạn i
	exercise_perc_real_completed_sum_{i}	Tổng tỷ lệ hoàn thành bài tập thực tế trong giai đoạn i
	exercise_perc_real_completed_mean_{i}	Trung bình tỷ lệ hoàn thành thực tế
	exercise_perc_real_completed_std_{i}	Độ lệch chuẩn của tỷ lệ hoàn thành thực tế
	exercise_perc_real_correct_sum_{i}	Tổng tỷ lệ đúng thực tế
	exercise_perc_real_correct_mean_{i}	Tỷ lệ đúng thực tế trung bình
	exercise_perc_real_correct_std_{i}	Độ lệch chuẩn của tỷ lệ đúng thực tế
	exercise_perc_real_score_sum_{i}	Tổng điểm thực tế đạt được
	exercise_perc_real_score_mean_{i}	Trung bình điểm thực tế
	exercise_perc_real_score_std_{i}	Độ lệch chuẩn điểm thực tế
	exercise_hour_entropy_{i}	Entropy của thời gian làm bài tập theo giờ (phân tán trong ngày)
Hành vi học tập - Video (video_*) - Theo từng đợt	video_watched_count_{i}	Số lượng video được xem trong đợt i (user coi ở giây cuối cùng được coi là hoàn thành)
	video_watched_percentage_{i}	Tỷ lệ video đã xem trên tổng số video của khóa học trong từng đợt

	video_watch_time_{i}	Tổng thời gian user xem video trong đợt i
	video_time_{i}	Tổng thời gian video được chọn để coi đợt i
	video_percentage_watch_{i}	Phần trăm thời lượng user coi trên tổng thời lượng của video
	video_pause_count_{i}	Số lượng ngắt quãng trong tổng số video đợt i
	video_pause_avg_{i}	Trung bình số lần ngắt quãng đợt i
	video_pause_std_{i}	Độ lệch chuẩn số lần ngắt quãng đợt i
	video_rewatch_count_{i}	Số lần xem lại đơn vị video (segment) đợt i
	video_rewatch_avg_{i}	Trung bình số lần xem lại trên mỗi video đợt i
	video_rewatch_std_{i}	Độ lệch chuẩn số lần xem lại trên mỗi video đợt i
	video_time_between_views_avg_{i}	Trung bình thời gian giữa các lần xem video đợt i
	video_speed_avg_{i}	Trung bình tốc độ xem video đợt i
	video_entropy_time_{i}	Entropy thời điểm user coi video đợt i
	video_final_score_percentage_{i}	Phần trăm điểm đặt được vào điểm cuối khóa
Nhãn	label	A, B, C, D, E thể hiện kết quả học tập cuối cùng của học viên

5.2.4. Tổng hợp dữ liệu

Nhóm đã tiến hành hợp nhất dữ liệu từ những đặc trưng đã trích xuất ở phần trước để chuẩn bị cho quá trình phân tích và xây dựng mô hình. Dữ liệu đầu vào bao gồm

thông tin người dùng, khóa học, nhãn kết quả học tập, và hành vi học tập qua nhiều giai đoạn. Cụ thể:

5.2.4.1. Nguồn dữ liệu chính:

User-course-label.csv: Nhãn kết quả học tập (label) theo từng cặp người dùng-khoa học.

User-course-info.csv: Thông tin chi tiết về khóa học và người học.

Course-field.csv: Lĩnh vực chuyên môn của từng khóa học.

Comment{i}.csv, Reply{i}.csv, Problem{i}.csv, Video{i}.csv: Dữ liệu hành vi trong từng giai đoạn học, bao gồm bình luận, phản hồi, làm bài và xem video.

5.2.4.2. Các bước hợp nhất dữ liệu:

Ghép nhãn kết quả: Kết hợp user-course-info với user-course-label dựa trên user_id và course_id để thêm nhãn kết quả học tập vào bảng chính.

```
user_info= user_info.merge(  
    user_problem[['user_id', 'course_id']].drop_duplicates(),  
    on=['user_id', 'course_id'],  
    how='inner'  
)  
user_info
```

	user_id	school	course_id	user_enroll_time	user_past_course_count	user_time_since_last_course	video_
0	U_10000	NaN	C_2033958	2020-10-27	0		0.0
1	U_1000979	云南大学	C_947149	2020-03-03	0		0.0
2	U_1000982	云南大学	C_947149	2020-06-30	0		0.0
3	U_1001176	云南大学	C_947149	2020-03-02	0		0.0
4	U_1001413	昆明理工大学	C_735164	2020-11-26	0		0.0
...
108117	U_99746	河南工学院 哈尔滨	C_674971	2020-05-13	3	768.0	

Cập nhật lĩnh vực khóa học: Bổ sung thông tin lĩnh vực từ Course-field.csv, đặc biệt cho các khóa học chưa có sẵn lĩnh vực.

```

# Assuming user_info and course_field are DataFrames, and you want to update user_info['field_x']
def update_field_x(row):
    if row['num_field_x'] == 0:
        # Find the corresponding field from course_field where course_id matches id
        matching_field = course_field[course_field['course_id'] == row['course_id']]['predicted_field']
        if not matching_field.empty: # Check if the match was found
            return matching_field.iloc[-1] # Get the first matching field value
    return row['field_x'] # Return the original value if the condition is not met

# Apply the function to update 'field_x' in user_info
user_info['field_x'] = user_info.apply(update_field_x, axis=1)

```

Mã hóa lĩnh vực: Mã hóa danh sách lĩnh vực thành số nguyên bằng LabelEncoder, sau đó tính tổng để tạo đặc trưng số hóa (encoded_field_sum).

```

from sklearn.preprocessing import LabelEncoder

set_fields= set(item for sublist in user_info['field_x'] for item in sublist)
print(len(set_fields))
# Flatten the 'field_x' lists and create a LabelEncoder
flat_fields = [item for sublist in user_info['field_x'] for item in sublist]
label_encoder = LabelEncoder()
print(len(flat_fields))

74
208679

# Fit label encoder on all unique categories in the field
label_encoder.fit(flat_fields)

# Encode 'field_x' lists to integer indices
user_info['encoded_field_x'] = user_info['field_x'].apply(lambda x: [label_encoder.transform([i])[0] for i in x])

# Show the encoded field_x
print(user_info[['field_x', 'encoded_field_x']])

      field_x      encoded_field_x
0   [地质资源与地质工程, 轻工技术与工程]      [28, 72]
1   [矿业工程, 交通运输工程]      [62, 4]
2   [矿业工程, 交通运输工程]      [62, 4]
3   [矿业工程, 交通运输工程]      [62, 4]
4   [外国语言文学]      [31]
...
108117      ...
108118      ...

```

```
# Tính tổng từng hàng trong cột `encoded_field_x`
user_info['encoded_field_sum'] = user_info['encoded_field_x'].apply(lambda x: sum(eval(x)) if isinstance(x, str)
```

user_info

exam	type	contain_exam	start_date	end_date	duration_days	remaining_time	encoded_field_x	encoded_field_sum
30.0	1.0	1	2020-09-04	2020-12-31	118.0	65	[28, 72]	100
35.0	1.0	1	2019-12-30	2020-04-19	111.0	47	[62, 4]	66
35.0	1.0	1	2020-04-20	2020-07-31	102.0	31	[62, 4]	66
35.0	1.0	1	2019-12-30	2020-04-19	111.0	48	[62, 4]	66

Trích xuất đặc trưng thời gian: Rút trích năm và tháng từ các mốc thời gian (ngày bắt đầu, kết thúc, đăng ký) để tạo các cột như start_year, user_month, v.v.

```
user_info['user_year'] = user_info['user_enroll_time'].str.split('-').str[0].astype(float)
user_info['user_month'] = user_info['user_enroll_time'].str.split('-').str[1].astype(float)
user_info
```

duration_days	remaining_time	encoded_field_sum	start_year	start_month	end_year	end_month	user_year	user_month
118.0	65	100	2020.0	9.0	2020.0	12.0	2020.0	10.0
111.0	47	66	2019.0	12.0	2020.0	4.0	2020.0	3.0
102.0	31	66	2020.0	4.0	2020.0	7.0	2020.0	6.0
111.0	48	66	2019.0	12.0	2020.0	4.0	2020.0	3.0
110.0	24	31	2020.0	9.0	2020.0	12.0	2020.0	11.0
...
171.0	79	49	2020.0	2.0	2020.0	7.0	2020.0	5.0
115.0	48	105	2020.0	9.0	2020.0	12.0	2020.0	11.0
121.0	73	80	2020.0	9.0	2020.0	12.0	2020.0	10.0
118.0	107	104	2020.0	9.0	2020.0	12.0	2020.0	9.0

Loại bỏ cột không cần thiết: Sau khi xử lý, các cột dư thừa hoặc không còn giá trị phân tích (như field_x, contain_exam) được loại bỏ để tinh gọn dữ liệu.

```

# Define the new column order
new_column_order = [
    'user_id', 'school', 'course_id',
    'encoded_field_sum',
    'start_year', 'start_month',
    'end_year', 'end_month',
    'user_year', 'user_month',
    'video_count', 'exercise_count', 'chapter_count',
    'user_past_course_count', 'user_time_since_last_course',
    'num_prerequisites', 'certificate', 'assignment', 'video', 'exam',
    'duration_days', 'remaining_time',
]

# Apply the new order
user_info = user_info[new_column_order]

```

5.2.4.3. Chia dữ liệu thành các tuần và giai đoạn:

Dữ liệu hành vi người dùng gồm **comment**, **reply**, **problem**, **video** được xử lý riêng cho từng **giai đoạn thời gian** (Phase 1, 2, 3...) dựa trên **remaining_time**.

Phase 1 được tích hợp đầu tiên vào user_info, theo cặp khóa user_id và course_id.

Tích hợp dữ liệu từng loại trong Phase i:

- **Comment:** comment_phase{i}.csv gộp vào, thêm các cột như total_words_phase{i}, total_positive{i}, ...

```

# Rename reply columns (excluding merge keys)
user_train_phase_1 = user_info.merge(comment, on=['user_id', 'course_id'], how='left')
user_train_phase_1

```

duration_days	remaining_time	comment_count_phase1	total_words_phase1	total_positive1	total_negative1	total_neutral
118.0	65	NaN	NaN	NaN	NaN	NaN
111.0	47	NaN	NaN	NaN	NaN	NaN
102.0	31	NaN	NaN	NaN	NaN	NaN
111.0	48	NaN	NaN	NaN	NaN	NaN
110.0	24	NaN	NaN	NaN	NaN	NaN
...
171.0	79	NaN	NaN	NaN	NaN	NaN

- **Reply:** reply_phase{i}.csv gộp vào, thêm các cột tương tự.

```
user_train_phase_1 = user_train_phase_1.merge(reply, on=['user_id', 'course_id'], how='left')
user_train_phase_1
```

count_phase1	total_words_phase1_y	total_positive1_y	total_negative1_y	total_neutral1_y	entropy_time_reply_phase1
NaN	NaN	NaN	NaN	NaN	NaN
NaN	NaN	NaN	NaN	NaN	NaN
NaN	NaN	NaN	NaN	NaN	NaN
NaN	NaN	NaN	NaN	NaN	NaN
NaN	NaN	NaN	NaN	NaN	NaN
...
NaN	NaN	NaN	NaN	NaN	NaN

- **Problem:** user_problem_{i}.csv gộp vào, tất cả các cột được thêm hậu tố _1 để phản ánh giai đoạn.

```
# Rename reply columns (excluding merge keys)
exercise = exercise.rename(columns={
    col: f"{col}_1" for col in exercise.columns if col not in ['user_id', 'course_id']
})
user_train_phase_1 = user_train_phase_1.merge(exercise, on=['user_id', 'course_id'], how='left')
user_train_phase_1
```

is_mean_mean_1	exercise_date_from_enroll_min_1	exercise_date_from_enroll_mean_1	exercise_date_from_enroll_max_1
1.00000	0.0	0.000000	0.0
NaN	NaN	NaN	NaN
1.00000	0.0	0.000000	0.0
NaN	NaN	NaN	NaN
1.05303	5.0	5.977273	7.0
...
1.00000	0.0	4.111111	12.0

- **Video:** Lọc dữ liệu uv_all_phases_final để lấy dữ liệu Phase i (_i), rồi gộp vào theo user_id, course_id.

```
user_train_phase_1 = user_train_phase_1.merge(user_video_phase_1, on=['user_id', 'course_id'], how='left')
user_train_phase_1
```

_pause_std_1	video_rewatch_avg_1	video_rewatch_std_1	video_time_between_views_avg_1	video_time_between_views_std_1
NaN	NaN	NaN	NaN	NaN
NaN	NaN	NaN	NaN	NaN
NaN	NaN	NaN	NaN	NaN
NaN	NaN	NaN	NaN	NaN
NaN	NaN	NaN	NaN	NaN
...
NaN	NaN	NaN	NaN	NaN
NaN	NaN	NaN	NaN	NaN

- **Tổng hợp cột bình luận:** Các cột total_words_phase{i}, total_positive{i}, total_negative{i}, total_neutral{i} được tổng hợp từ cả **comment** và **reply** (sau khi gộp có hậu tố _x, _y), rồi xóa các cột gốc để giảm chiều dữ liệu.

```
# Replace NaN values with 0 before summing
user_train_phase_1['total_words_phase1'] = (user_train_phase_1['total_words_phase1_x'].fillna(0) + user_train_phase_1['total_words_phase1_y'].fillna(0))
user_train_phase_1['total_positive1'] = (user_train_phase_1['total_positive1_x'].fillna(0) + user_train_phase_1['total_positive1_y'].fillna(0))
user_train_phase_1['total_negative1'] = (user_train_phase_1['total_negative1_x'].fillna(0) + user_train_phase_1['total_negative1_y'].fillna(0))
user_train_phase_1['total_neutral1'] = (user_train_phase_1['total_neutral1_x'].fillna(0) + user_train_phase_1['total_neutral1_y'].fillna(0))

# Drop the original reply/comment-related columns
columns_to_drop = [
    'total_words_phase1_x', 'total_words_phase1_y',
    'total_positive1_x', 'total_positive1_y',
    'total_negative1_x', 'total_negative1_y',
    'total_neutral1_x', 'total_neutral1_y',
    'comment_count_phase1', 'reply_count_phase1',
    'entropy_time_reply_phase1'
]

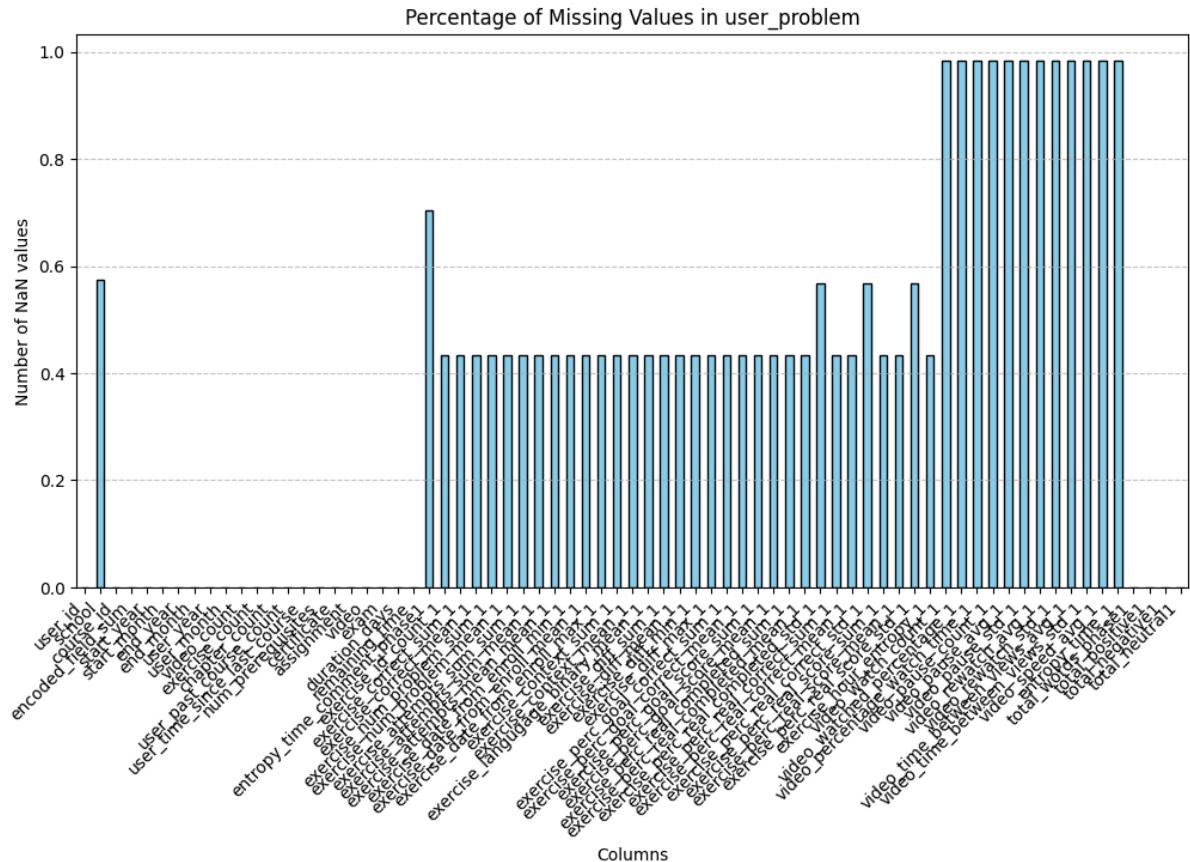
user_train_phase_1 = user_train_phase_1.drop(columns=columns_to_drop)

# Optional: Reset index after the transformation
user_train_phase_1.reset_index(drop=True, inplace=True)
```

5.2.4.4. Xử lý dữ liệu số:

Dữ liệu được khai thác theo đơn vị tuần, do đó: Các giá trị NaN tại các đặc trưng hành vi (ví dụ: làm bài, xem video, thảo luận, v.v.) phản ánh rằng người dùng không có hoạt động nào trong tuần đó. Vì vậy, các giá trị NaN trong trường hợp này không phải do thiếu dữ liệu, mà do không có tương tác thực sự xảy ra.

Phương pháp xử lý: Điền các giá trị NaN bằng 0. Nghĩa là người học không có hoạt động tương ứng trong tuần (không làm bài, không xem video, không thảo luận...), đồng thời giữ được tính toàn vẹn và chính xác của dữ liệu hành vi theo thời gian.



Các cột liên quan đến tương tác bình luận và trả lời (total_words_phase1, total_positive1, v.v.) được điền giá trị 0 cho các giá trị thiếu, dựa trên giả định rằng việc thiếu dữ liệu cho thấy không có tương tác trong Phase 1.

Cột entropy_time_comment_phase1 cũng được điền giá trị 0 cho các giá trị thiếu.

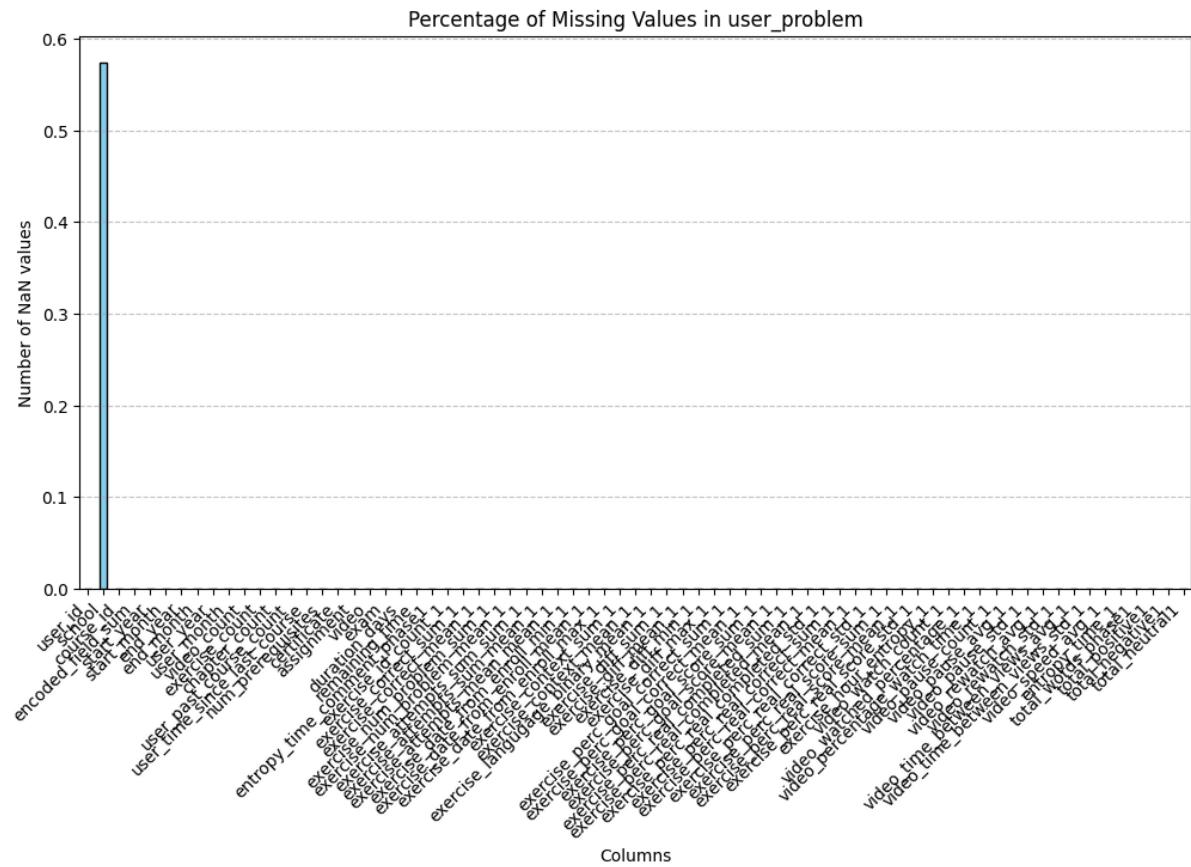
Các cột liên quan đến bài tập (có hậu tố _1) và video (có hậu tố _1) cũng được điền giá trị 0 cho các giá trị thiếu.

```
user_train_phase_1['entropy_time_comment_phase1'] = user_train_phase_1['entropy_time_comment_phase1'].fillna(0)

# List of columns to fill with 0
columns_to_fill_with_zero = [
    'exercise_id_count_1', 'exercise_correct_sum_1', 'exercise_correct_mean_1',
    'exercise_num_problem_sum_1', 'exercise_num_problem_mean_1', 'exercise_attempts_sum_sum_1',
    'exercise_attempts_sum_mean_1', 'exercise_attempts_mean_mean_1', 'exercise_date_from_enroll_min_1',
    'exercise_date_from_enroll_mean_1', 'exercise_date_from_enroll_max_1', 'exercise_context_sum_1',
    'exercise_context_mean_1', 'exercise_language_binary_mean_1', 'exercise_diff_sum_1',
    'exercise_diff_mean_1', 'exercise_diff_min_1', 'exercise_diff_max_1', 'exercise_perc_goal_correct_sum_1',
    'exercise_perc_goal_correct_mean_1', 'exercise_perc_goal_score_sum_1', 'exercise_perc_goal_score_mean_1',
    'exercise_perc_real_completed_sum_1', 'exercise_perc_real_completed_mean_1', 'exercise_perc_real_completed_std',
    'exercise_perc_real_correct_sum_1', 'exercise_perc_real_correct_mean_1', 'exercise_perc_real_correct_std_1',
    'exercise_perc_real_score_sum_1', 'exercise_perc_real_score_mean_1', 'exercise_perc_real_score_std_1',
    'exercise_hour_entropy_1', 'video_watch_count_1', 'video_watched_percentage_1', 'video_percentage_watch_time_1',
    'video_pause_count_1', 'video_pause_avg_1', 'video_pause_std_1', 'video_rewatch_avg_1', 'video_rewatch_std_1',
    'video_time_between_views_avg_1', 'video_time_between_views_std_1', 'video_speed_avg_1', 'entropy_time_1'
]

# Fill the specified columns with 0
user_train_phase_1[columns_to_fill_with_zero] = user_train_phase_1[columns_to_fill_with_zero].fillna(0)
```

Kết quả sau khi xử lý:



5.2.5. Thống kê dữ liệu sau khi tổng hợp

5.2.5.1. Thông tin cơ bản bộ dữ liệu

Kiểm tra thông tin dữ liệu: Sử dụng user_train_phase_1.info() để xem các thông tin cơ bản về DataFrame như số lượng dòng, cột, kiểu dữ liệu của từng cột và số lượng giá trị không thiêú. Để giúp nắm bắt cấu trúc dữ liệu và phát hiện cột nào có giá trị thiêú.

```
user_train_phase_1.info()
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 108122 entries, 0 to 108121
Data columns (total 74 columns):
 #   Column           Non-Null Count  Dtype  
--- 
 0   user_id          108122 non-null   object  
 1   school           46058 non-null   object  
 2   course_id        108122 non-null   object  
 3   encoded_field_sum 108122 non-null   int64  
 4   start_year       108122 non-null   float64 
 5   start_month      108122 non-null   float64 
 6   end_year          108122 non-null   float64 
 7   end_month         108122 non-null   float64 
 8   user_year         108122 non-null   float64 
 9   user_month        108122 non-null   float64 
 10  video_count      108122 non-null   int64  
 11  exercise_count   108122 non-null   int64  
 12  chapter_count    108122 non-null   int64  
 13  user_past_course_count 108122 non-null   int64  
 14  user_time_since_last_course 108122 non-null   float64 
 15  num_prerequisites 108122 non-null   int64  
 16  certificate       108122 non-null   float64 
 17  assignment        108122 non-null   float64 
 18  video             108122 non-null   float64 
 19  exam              108122 non-null   float64 
 20  duration_days     108122 non-null   float64 
 21  remaining_time    108122 non-null   int64  
 22  entropy_time_comment_phase1 108122 non-null   float64 
 23  exercise_id_count_1 108122 non-null   float64 
 24  exercise_correct_sum_1 108122 non-null   float64 
 25  exercise_correct_mean_1 108122 non-null   float64
```

27	exercise_num_problem_mean_1	108122	non-null	float64
28	exercise_attempts_sum_sum_1	108122	non-null	float64
29	exercise_attempts_sum_mean_1	108122	non-null	float64
30	exercise_attempts_mean_mean_1	108122	non-null	float64
31	exercise_date_from_enroll_min_1	108122	non-null	float64
32	exercise_date_from_enroll_mean_1	108122	non-null	float64
33	exercise_date_from_enroll_max_1	108122	non-null	float64
34	exercise_context_sum_1	108122	non-null	float64
35	exercise_context_mean_1	108122	non-null	float64
36	exercise_language_binary_mean_1	108122	non-null	float64
37	exercise_diff_sum_1	108122	non-null	float64
38	exercise_diff_mean_1	108122	non-null	float64
39	exercise_diff_min_1	108122	non-null	float64
40	exercise_diff_max_1	108122	non-null	float64
41	exercise_perc_goal_correct_sum_1	108122	non-null	float64
42	exercise_perc_goal_correct_mean_1	108122	non-null	float64
43	exercise_perc_goal_score_sum_1	108122	non-null	float64
44	exercise_perc_goal_score_mean_1	108122	non-null	float64
45	exercise_perc_real_completed_sum_1	108122	non-null	float64
46	exercise_perc_real_completed_mean_1	108122	non-null	float64
47	exercise_perc_real_completed_std_1	108122	non-null	float64
48	exercise_perc_real_correct_sum_1	108122	non-null	float64
49	exercise_perc_real_correct_mean_1	108122	non-null	float64
50	exercise_perc_real_correct_std_1	108122	non-null	float64
51	exercise_perc_real_score_sum_1	108122	non-null	float64
52	exercise_perc_real_score_mean_1	108122	non-null	float64
53	exercise_perc_real_score_std_1	108122	non-null	float64
54	exercise_hour_entropy_1	108122	non-null	float64
55	video_watch_count_1	108122	non-null	float64
56	video_WATCHED_PERCENTAGE_1	108122	non-null	float64
57	video_PERCENTAGE_WATCH_TIME_1	108122	non-null	float64
58	video_PAUSE_COUNT_1	108122	non-null	float64
59	video_PAUSE_AVG_1	108122	non-null	float64
60	video_PAUSE_STD_1	108122	non-null	float64

```

45 exercise_perc_real_completed_sum_1    108122 non-null   float64
46 exercise_perc_real_completed_mean_1   108122 non-null   float64
47 exercise_perc_real_completed_std_1   108122 non-null   float64
48 exercise_perc_real_correct_sum_1    108122 non-null   float64
49 exercise_perc_real_correct_mean_1   108122 non-null   float64
50 exercise_perc_real_correct_std_1   108122 non-null   float64
51 exercise_perc_real_score_sum_1    108122 non-null   float64
52 exercise_perc_real_score_mean_1   108122 non-null   float64
53 exercise_perc_real_score_std_1   108122 non-null   float64
54 exercise_hour_entropy_1           108122 non-null   float64
55 video_watch_count_1             108122 non-null   float64
56 video_watched_percentage_1     108122 non-null   float64
57 video_percentage_watch_time_1  108122 non-null   float64
58 video_pause_count_1           108122 non-null   float64
59 video_pause_avg_1             108122 non-null   float64
60 video_pause_std_1             108122 non-null   float64
61 video_rewatch_avg_1          108122 non-null   float64
62 video_rewatch_std_1          108122 non-null   float64
63 video_time_between_views_avg_1 108122 non-null   float64
64 video_time_between_views_std_1 108122 non-null   float64
65 video_speed_avg_1            108122 non-null   float64
66 entropy_time_1               108122 non-null   float64
67 total_words_phase1           108122 non-null   float64
68 total_positive1              108122 non-null   float64
69 total_negative1              108122 non-null   float64
70 total_neutral1               108122 non-null   float64
71 total_score                  108122 non-null   float64
72 label                        108122 non-null   object
73 label_encoded                108122 non-null   int64
dtypes: float64(62), int64(8), object(4)
memory usage: 61.0+ MB

```

Phân tích sơ bộ cho thấy đa số các cột dữ liệu thuộc kiểu số (float hoặc int), chỉ có một vài cột là kiểu chuỗi (object), bao gồm: user_id, course_id school, label (biến mục tiêu)

Trong các cột này, chỉ có cột school chứa giá trị khuyết (missing). Để xử lý giá trị khuyết trong cột school, áp dụng phương pháp: Điện giá trị thiếu bằng trường có tần suất xuất hiện cao nhất (mode). Vì school là biến phân loại (categorical), nên sử dụng mode là cách hợp lý và đơn giản. Giúp giữ lại phân bố dữ liệu tự nhiên, tránh làm sai lệch mô hình.

```

# Mapping school
school_count_map = df['school'].value_counts().to_dict()
df['school'] = df['school'].map(school_count_map).fillna(0)

```

5.2.5.2. Trực quan hóa phân bố dữ liệu

- Chọn các cột có kiểu dữ liệu số (float64, int64) bằng select_dtypes.
- Loại bỏ cột 'label_encoded' khỏi danh sách các cột số để không vẽ biểu đồ phân bố của nó cùng với các feature.
- Sử dụng biểu đồ histogram (hist) để xem phân bố của từng feature số. Điều này giúp hiểu về hình dạng phân bố (chuẩn, lệch, đa đỉnh), khoảng giá trị của từng feature và phát hiện các giá trị ngoại lai tiềm năng.

```
import matplotlib.pyplot as plt
import numpy as np

# Select numeric features
numeric_cols = user_train_phase_1.select_dtypes(include=['float64', 'int64']).columns.drop(['label_encoded'])

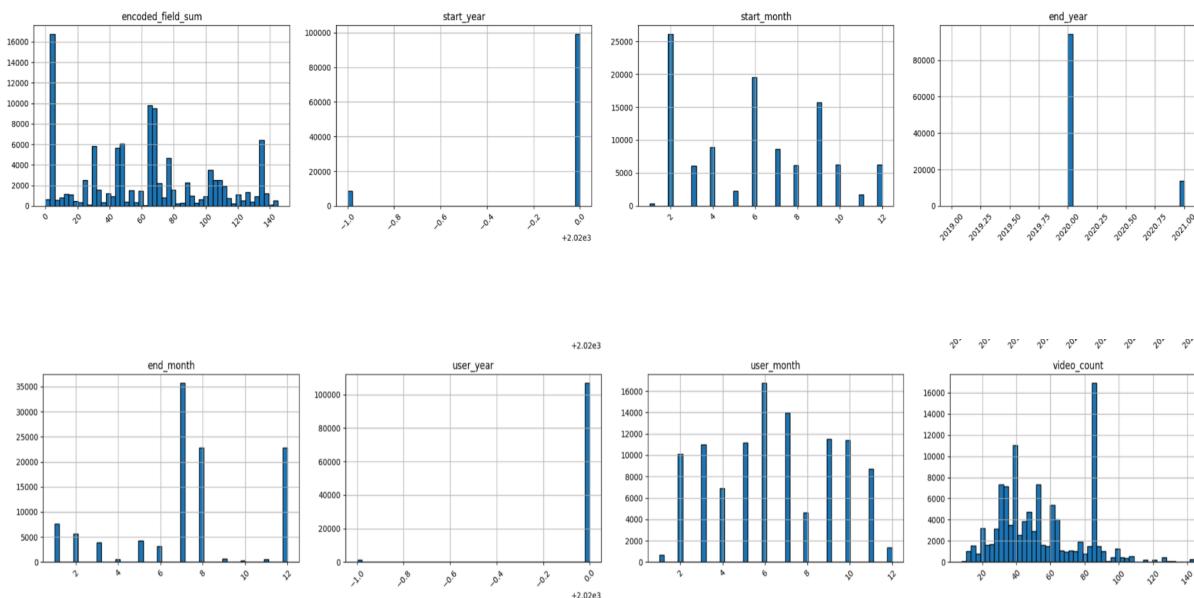
# Number of columns in subplot grid
cols = 4
numeric_cols = user_train_phase_1.select_dtypes(include=['float64', 'int64']).columns.drop(['label_encoded'])
rows = int(np.ceil(len(numeric_cols) / cols))

fig, axes = plt.subplots(rows, cols, figsize=(6 * cols, 4 * rows)) # larger individual plots
axes = axes.flatten()

for i, col in enumerate(numeric_cols):
    user_train_phase_1[col].hist(ax=axes[i], bins=50, edgecolor='k')
    axes[i].set_title(col, fontsize=12)
    axes[i].tick_params(axis='x', labelrotation=45)

# Hide any empty subplots
for j in range(i+1, len(axes)):
    fig.delaxes(axes[j])

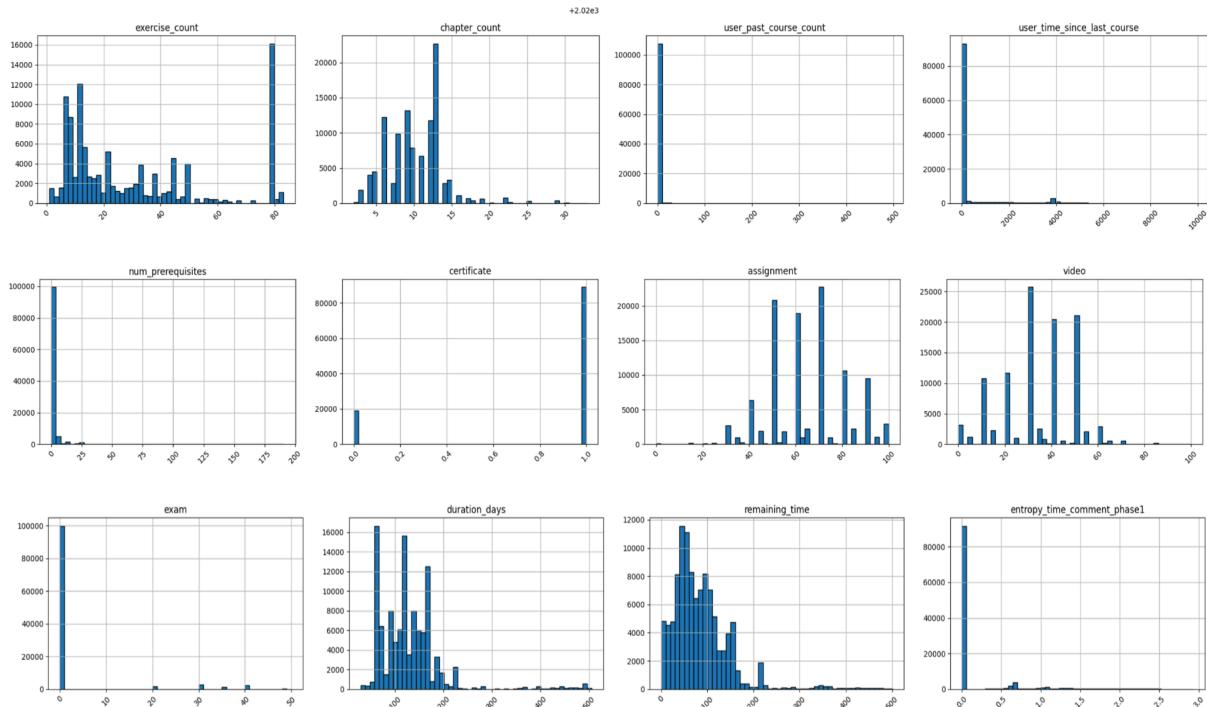
plt.tight_layout()
plt.show()
```



* year: phân phối ở 2 giá trị.

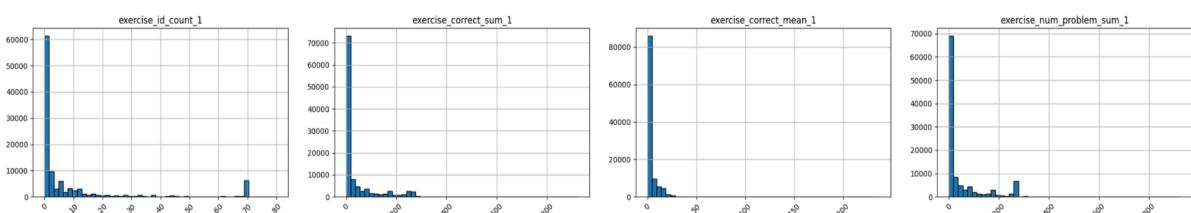
start_moth: phân bố đồng đều, đặc biệt nhiều nhất vào tháng 2

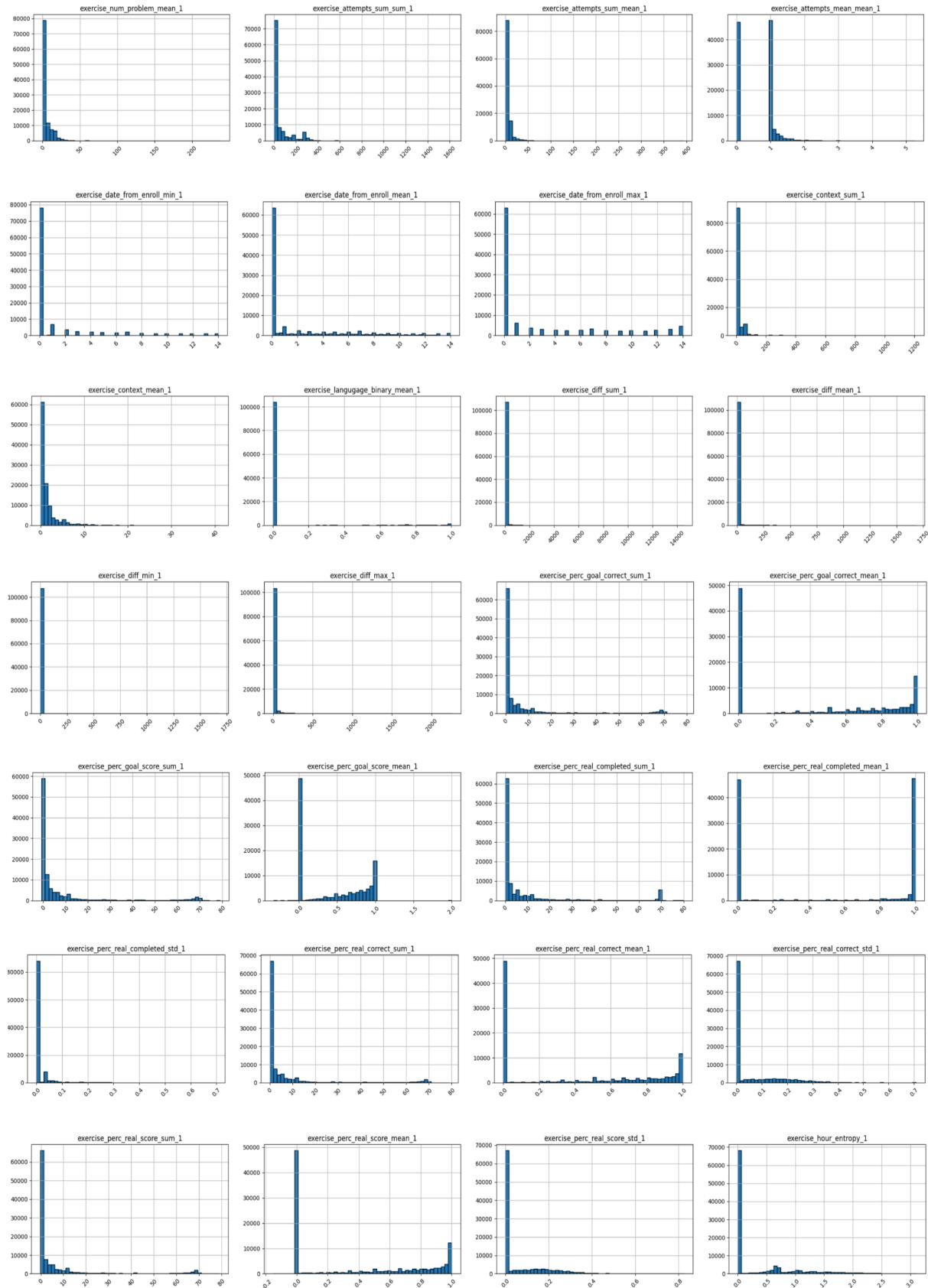
User_month: phân bố giống phân phối chuẩn

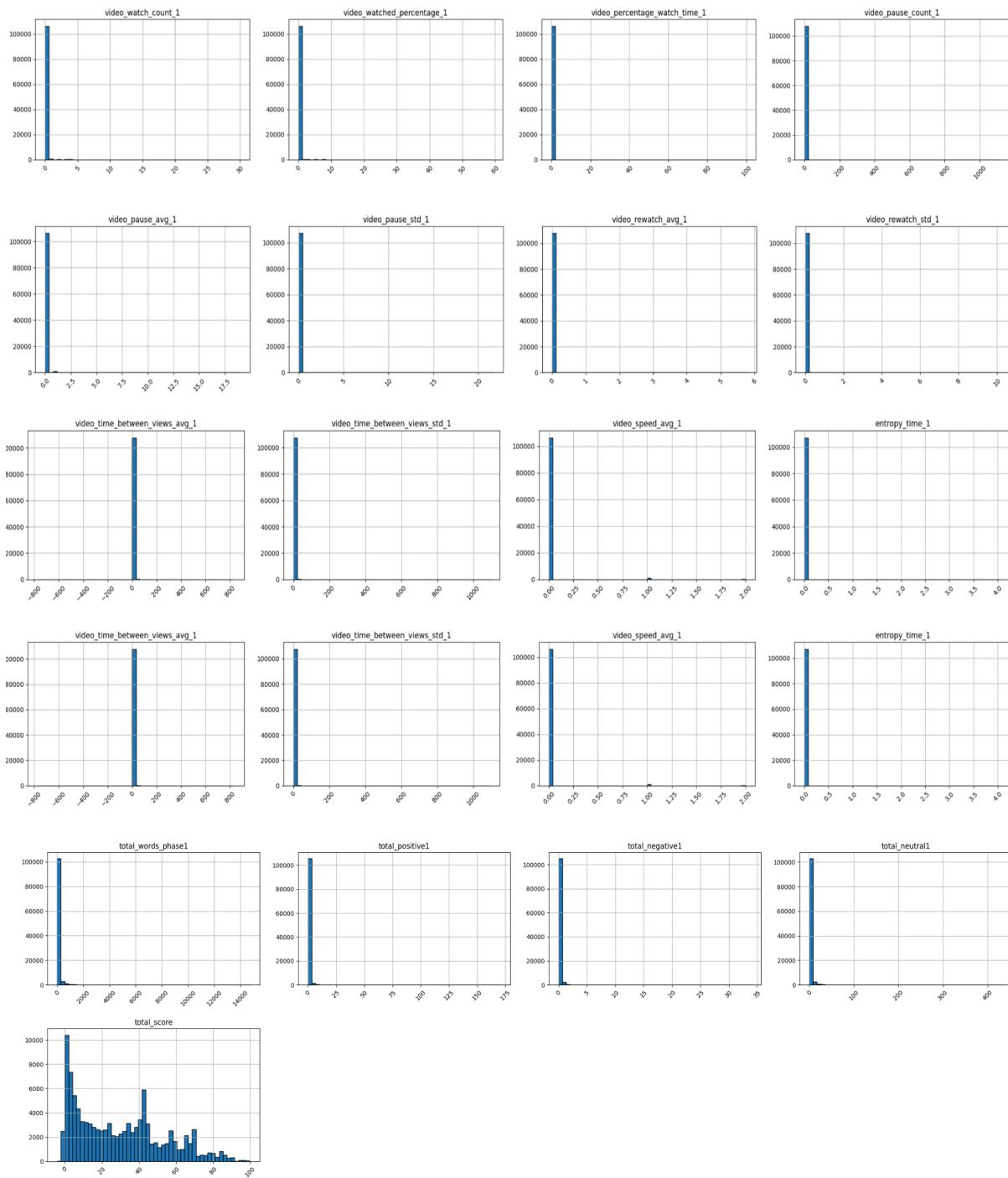


- **Video_count:** Phân bố của số lượng video không hoàn toàn đồng đều. Có nhiều đỉnh cho thấy một số lượng video nhất định xuất hiện thường xuyên hơn. Có một đỉnh đáng chú ý ở khoảng gần 90 video, và các đỉnh khác thấp hơn ở khoảng 20, 40, 60, và 140. Số lượng khóa học có trên 100 video là rất ít.
- **Exercise_count:** Phân bố này lệch phải. Phần lớn các khóa học có số lượng bài tập tương đối ít, tập trung chủ yếu ở khoảng dưới 40. Có một đỉnh cao nhất ở khoảng 10-20 bài tập. Số lượng khóa học giảm dần khi số lượng bài tập tăng lên. Nhưng nhiều nhất là 80.
- **Chapter_count:** Phân bố này cũng lệch phải. Hầu hết các khóa học có số lượng chương ít, với đỉnh cao nhất ở khoảng 5-10 chương. Số lượng khóa học giảm đáng kể khi số lượng chương tăng.
- **User_past_course_count:** Phân bố này lệch phải mạnh. Phần lớn người dùng có số lượng khóa học đã tham gia trong quá khứ rất ít, chủ yếu là 0 hoặc gần 0. Điều này cho thấy có nhiều người dùng mới hoặc người dùng chỉ tham gia một vài khóa học.

- **User_time_since_last_course:** Phân bố này lệch phải cực kỳ mạnh. Đại đa số người dùng có thời gian kể từ khóa học cuối cùng rất ngắn (gần bằng 0), cho thấy họ đang hoạt động tích cực hoặc vừa mới hoàn thành một khóa học.
- **Num_prerequisites:** Phân bố này lệch phải mạnh. Phần lớn các khóa học có rất ít hoặc không có điều kiện tiên quyết (số lượng bằng 0 là cao nhất). Số lượng khóa học giảm nhanh chóng khi số lượng điều kiện tiên quyết tăng lên.
- **Certificate:** Biểu đồ này có vẻ thể hiện một biến phân loại hoặc nhị phân. Có hai đỉnh rõ rệt: một đỉnh rất cao ở giá trị 0 và một đỉnh thấp hơn đáng kể ở giá trị 1. Điều này có thể có nghĩa là phần lớn các khóa học không cấp chứng chỉ (giá trị 0) và một phần nhỏ hơn có cấp chứng chỉ (giá trị 1).
- **Assignment:** Phân bố này có nhiều đỉnh, cho thấy sự tập trung ở một số lượng bài tập nhất định. Các đỉnh đáng chú ý ở khoảng 20, 40, 60, và 80. Có vẻ ít khóa học có số lượng bài tập rất thấp hoặc rất cao.
- **Video:** Đây chính là biểu đồ "Video_count" đã được nhận xét ở trên. Phân bố không đồng đều, có nhiều đỉnh, với đỉnh cao nhất gần 90.
- **Exam:** Phân bố này lệch phải mạnh. Phần lớn các khóa học có rất ít bài kiểm tra, với đỉnh cao nhất ở giá trị 0 hoặc rất gần 0. Số lượng khóa học giảm nhanh khi số lượng bài kiểm tra tăng.
- **Duration_days:** Phân bố này có vẻ đa phương thức với nhiều đỉnh. Có các cụm tập trung quanh các khoảng 30-60 ngày, 90-120 ngày, và có thể một cụm nữa ở khoảng 180-210 ngày. Điều này cho thấy các khóa học thường có một số khoảng thời gian phổ biến.
- **remaining_time:** Phân bố này lệch phải. Phần lớn các trường hợp có thời gian còn lại tương đối ngắn, với đỉnh cao nhất ở gần 0. Số lượng giảm dần khi thời gian còn lại tăng lên.







- Các biểu đồ phân phối của các đặc trưng hành vi học tập (xem video, làm bài tập, thảo luận...) cho thấy tần suất xuất hiện tại giá trị 0 chiếm ưu thế rõ rệt. Điều này phản ánh rằng phần lớn người học không thực hiện hoạt động tương ứng trong nhiều tuần.

- Đối với các đặc trưng liên quan đến hành vi làm bài tập, tuy có sự xuất hiện của các giá trị khác 0 với tần suất cao hơn, nhưng vẫn thấp đáng kể so với tần suất tại 0, và phân phối nhìn chung có xu hướng giảm dần theo giá trị.

Kết luận: Dữ liệu hành vi học tập thể hiện rõ tính chất thưa thớt (sparse), với đa số người học không hoạt động thường xuyên. Điều này cần được lưu ý trong quá trình xử lý và xây dựng mô hình, đặc biệt khi chọn thuật toán phù hợp và chuẩn hóa dữ liệu.

5.2.5.3. Trực quan hóa phân phối dữ liệu

- Chọn một số cặp feature quan tâm hoặc nghi ngờ có mối quan hệ với nhau và với nhãn.
- Sử dụng biểu đồ scatter plot (scatterplot) để trực quan hóa mối quan hệ giữa hai feature, đồng thời sử dụng màu sắc (hue='label') để phân biệt các nhãn khác nhau. Điều này giúp bạn xem liệu có sự phân tách giữa các lớp nhãn dựa trên giá trị của các feature này hay không.

```
import seaborn as sns
import matplotlib.pyplot as plt

# Selected feature pairs (excluding total_score)
selected_pairs = [
    ('exercise_correct_sum_1', 'total_words_phase1'),
    ('exercise_perc_real_score_mean_1', 'exercise_perc_real_correct_mean_1'),
    ('exercise_perc_goal_score_sum_1', 'exercise_perc_goal_correct_sum_1'),
    ('exercise_perc_real_score_sum_1', 'exercise_perc_real_correct_sum_1'),
    ('exercise_hour_entropy_1', 'exercise_context_sum_1'),
    ('entropy_time_comment_phase1', 'exercise_correct_sum_1'),
    ('exercise_id_count_1', 'exercise_num_problem_sum_1'),
    ('assignment', 'exercise_correct_mean_1'),
    ('video', 'exercise_perc_goal_score_mean_1'),
    ('duration_days', 'exercise_perc_real_correct_sum_1'),
    # Interaction or effort relationships
    ('exercise_attempts_sum_sum_1', 'exercise_correct_sum_1'),
    ('exercise_num_problem_sum_1', 'exercise_correct_sum_1'),
    ('exercise_attempts_sum_sum_1', 'exercise_num_problem_sum_1'),

    # Score-performance relationships
    ('exercise_perc_real_score_mean_1', 'exercise_perc_goal_score_mean_1'),
    ('exercise_perc_real_correct_mean_1', 'exercise_correct_mean_1'),
    ('exercise_perc_real_completed_mean_1', 'exercise_perc_real_correct_mean_1'),

    # Engagement vs performance
    ('video_pause_count_1', 'exercise_perc_real_score_mean_1'),
    ('video_watch_count_1', 'exercise_correct_sum_1'),
    ('video_percentage_watch_time_1', 'total_words_phase1'),
```

```

('total_positivel', 'exercise_perc_real_score_mean_1'),
('total_negativel', 'exercise_hour_entropy_1'),

# Diversity of behavior
('entropy_time_comment_phase1', 'video_watch_count_1'),
('entropy_time_1', 'exercise_hour_entropy_1')
]

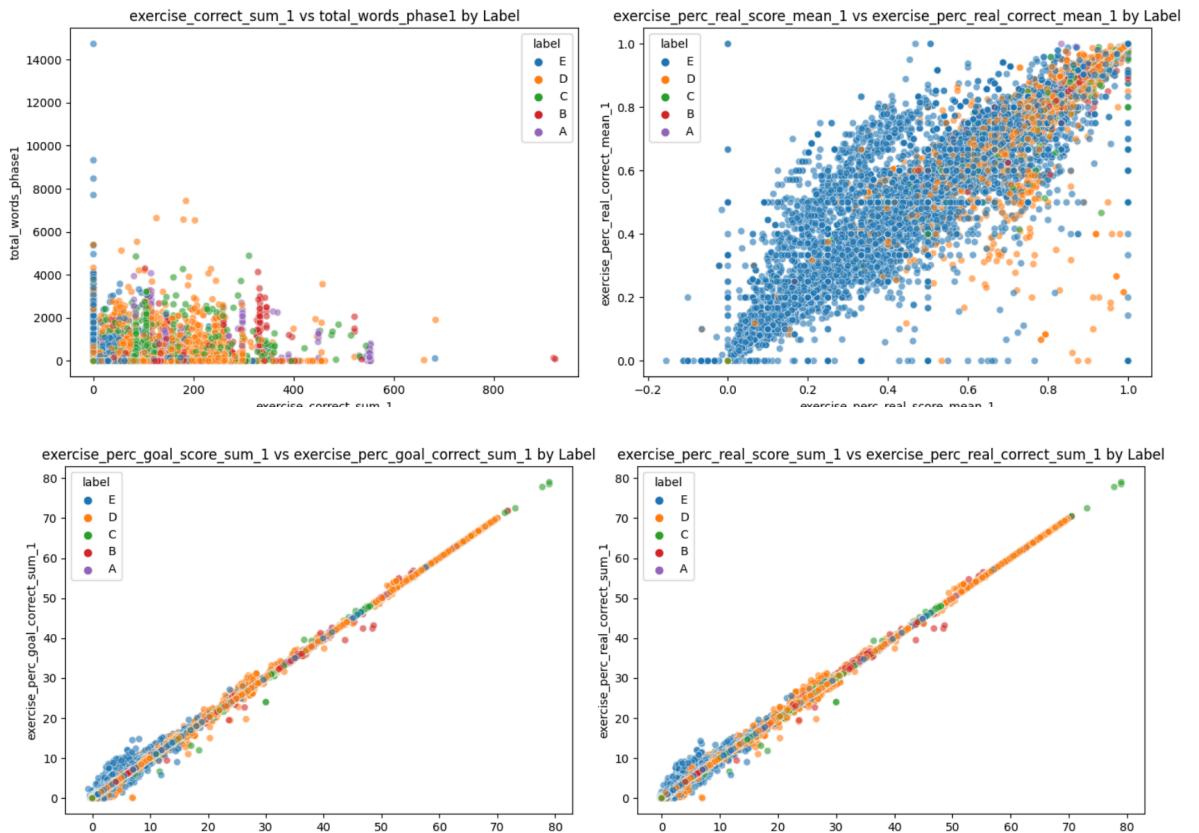
# Plot: 2 per row
n_cols = 2
n_rows = (len(selected_pairs) + 1) // n_cols
fig, axes = plt.subplots(n_rows, n_cols, figsize=(14, 5 * n_rows))
axes = axes.flatten()

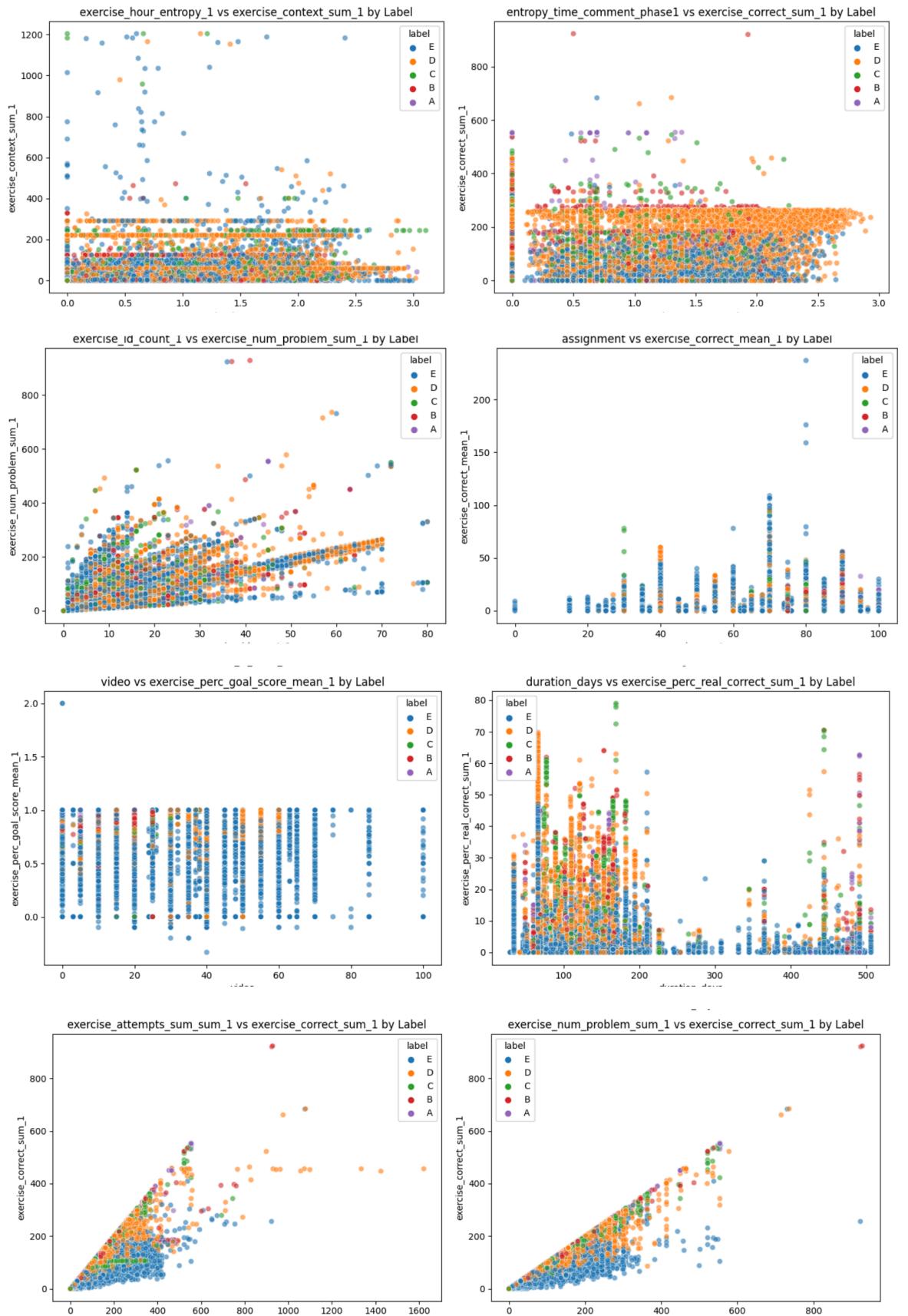
for i, (x_feature, y_feature) in enumerate(selected_pairs):
    sns.scatterplot(
        data=user_train_phase_1,
        x=x_feature,
        y=y_feature,
        hue='label',
        alpha=0.6,
        ax=axes[i]
    )
    axes[i].set_title(f'{x_feature} vs {y_feature} by Label')

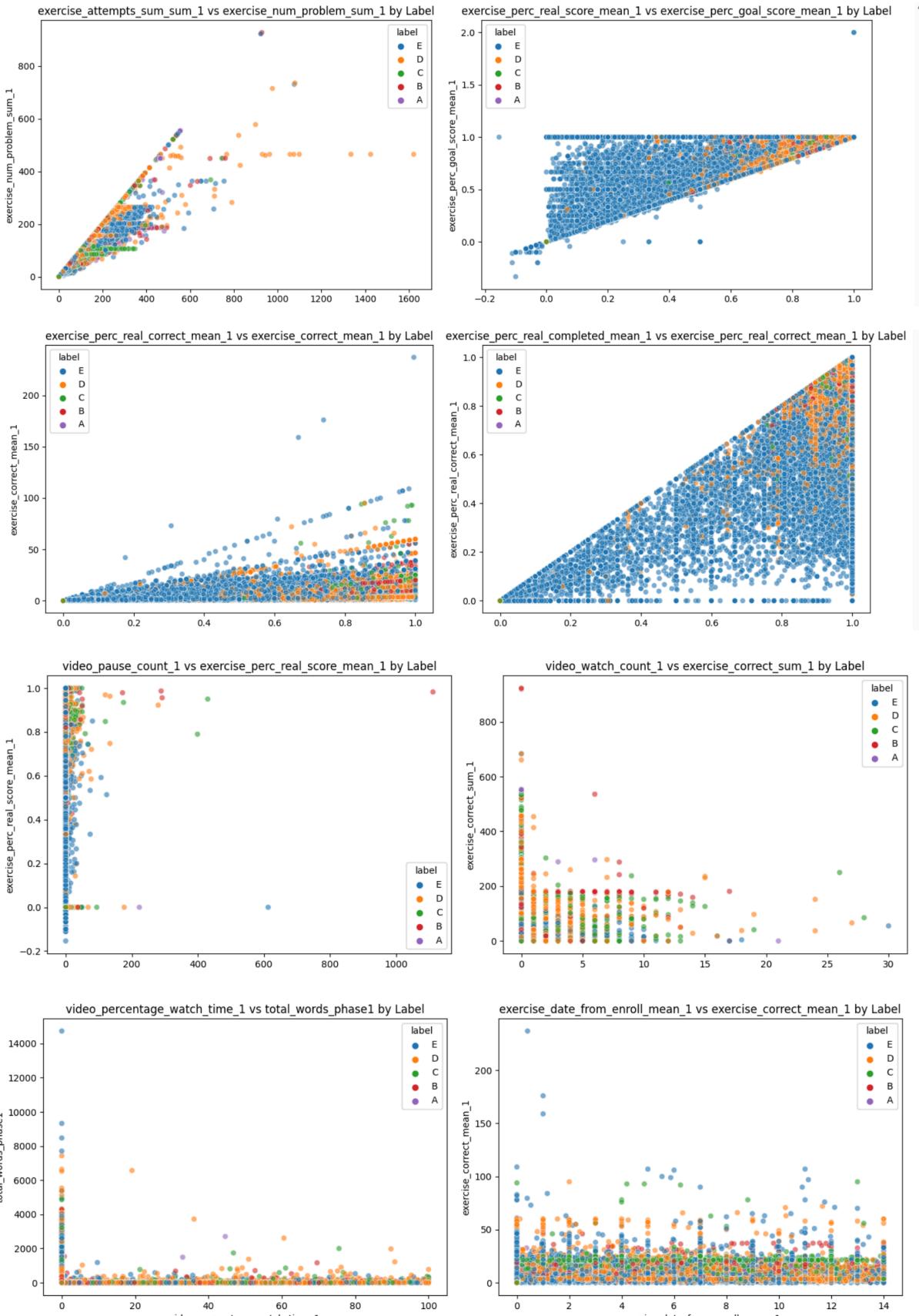
# Hide unused axes if any
for j in range(i + 1, len(axes)):
    fig.delaxes(axes[j])

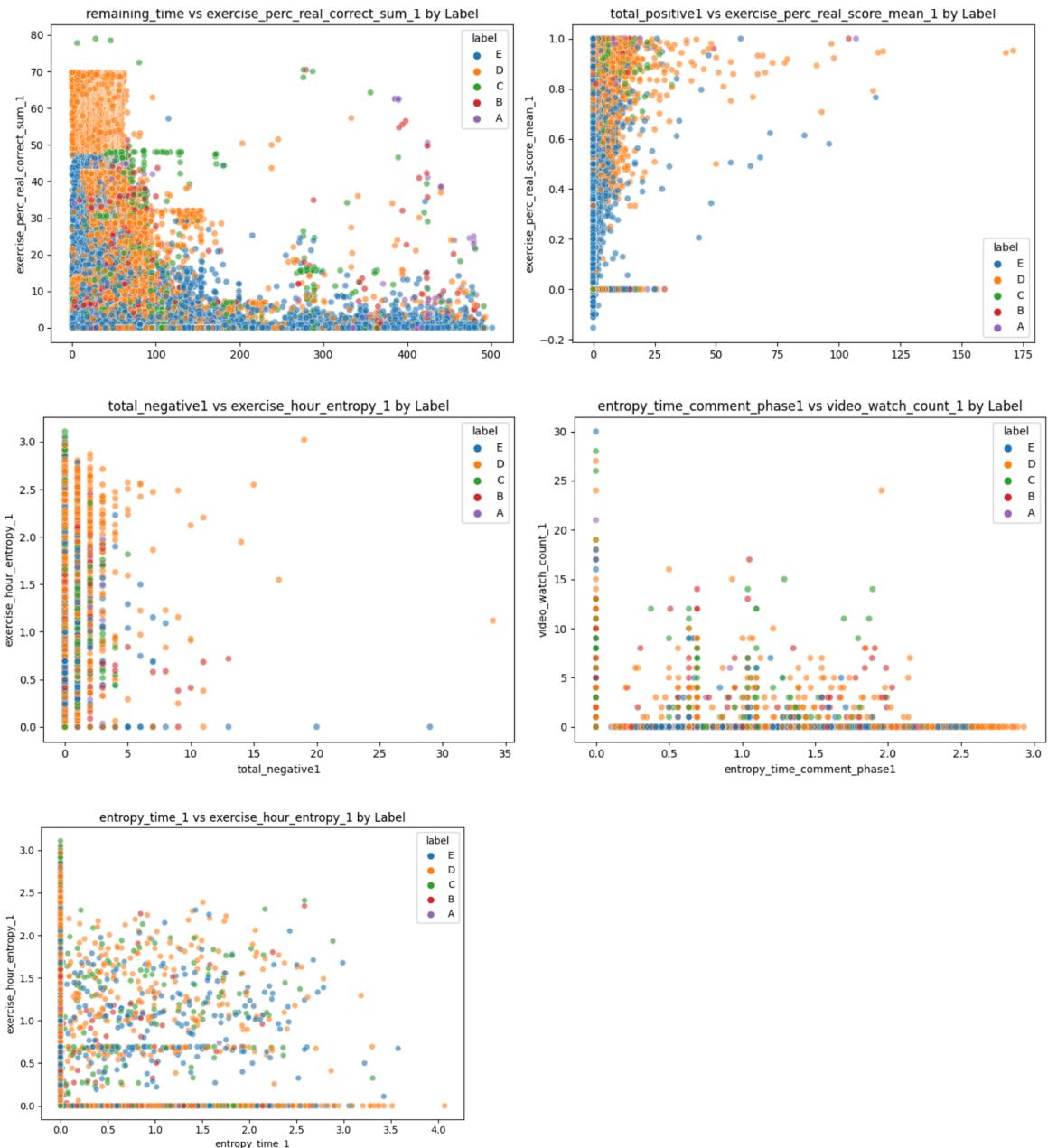
plt.tight_layout()
plt.show()

```









Nhận xét: Người học đạt nhãn A có xu hướng có tỷ lệ làm bài tập đúng cao hơn đáng kể ngay trong Giai đoạn 1 so với người học nhãn E

- Sử dụng biểu đồ box plot (boxplot) để so sánh phân bố của từng feature số giữa các lớp nhãn khác nhau. Để xem liệu giá trị trung bình hoặc phân bố của một feature có khác nhau đáng kể giữa các lớp nhãn hay không, cho thấy feature đó có thể hữu ích cho việc phân loại.

```

numeric_cols = user_train_phase_1.select_dtypes(include=['float64', 'int64']).columns.tolist()

# Exclude 'label_encoded' and 'type'
for col in ['label_encoded', 'type']:
    if col in numeric_cols:
        numeric_cols.remove(col)

# Layout config
n_cols = 2
n_rows = math.ceil(len(numeric_cols) / n_cols)

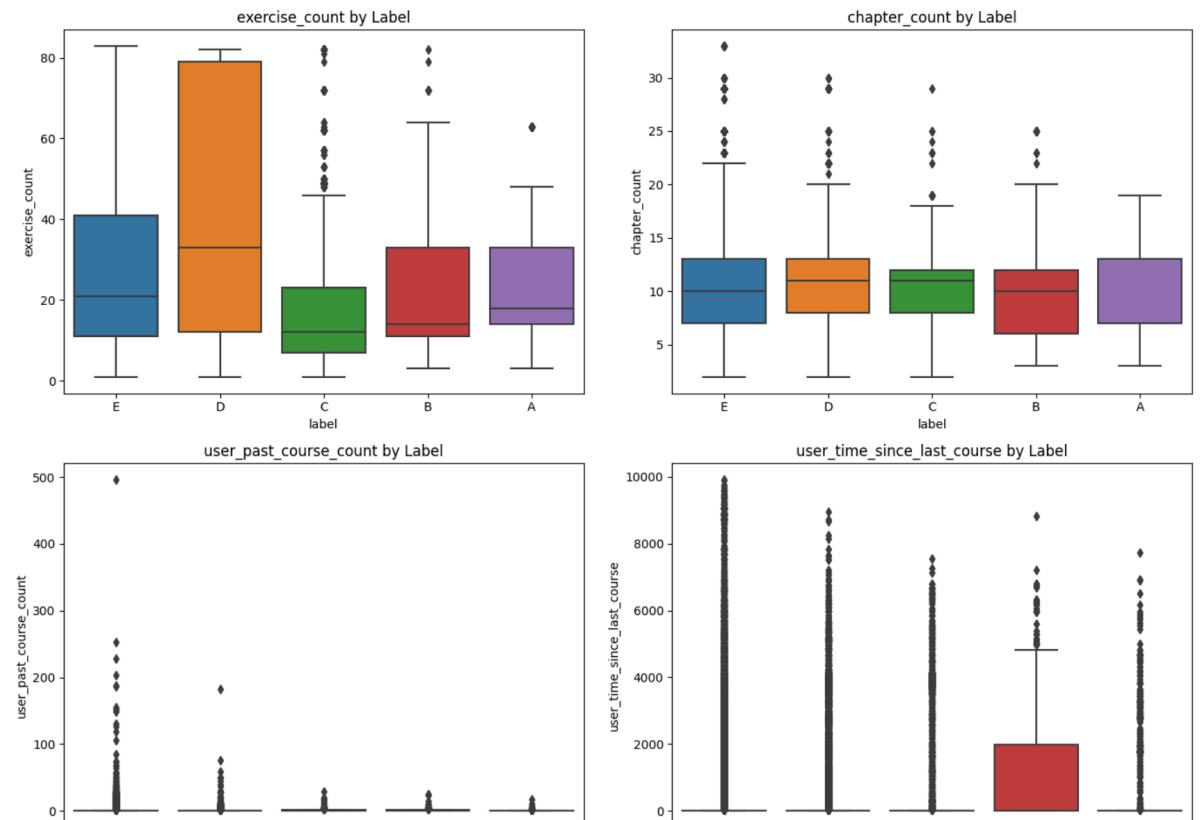
# Create subplots
fig, axes = plt.subplots(n_rows, n_cols, figsize=(14, 5 * n_rows))
axes = axes.flatten()

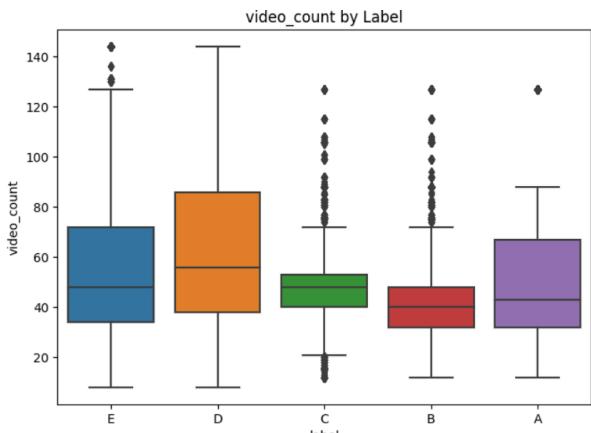
# Plot each feature
for i, feature in enumerate(numeric_cols):
    sns.boxplot(
        data=user_train_phase_1,
        x='label',
        y=feature,
        order=['E', 'D', 'C', 'B', 'A'],
        ax=axes[i]
    )
    axes[i].set_title(f'{feature} by Label')

# Hide unused axes
for j in range(i + 1, len(axes)):
    fig.delaxes(axes[j])

plt.tight_layout()
plt.show()

```





Cập nhật đầu đủ trong file code.

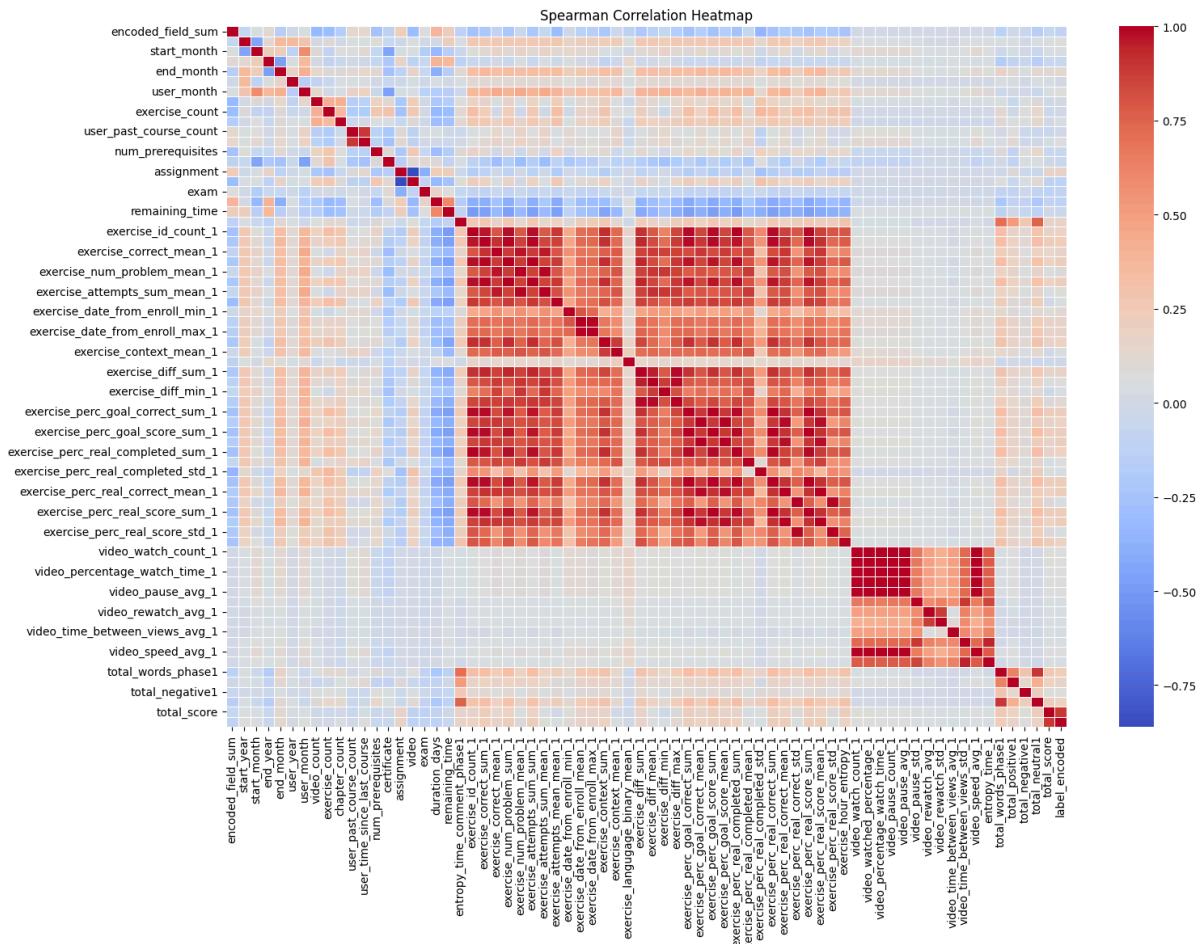
Nhận xét:

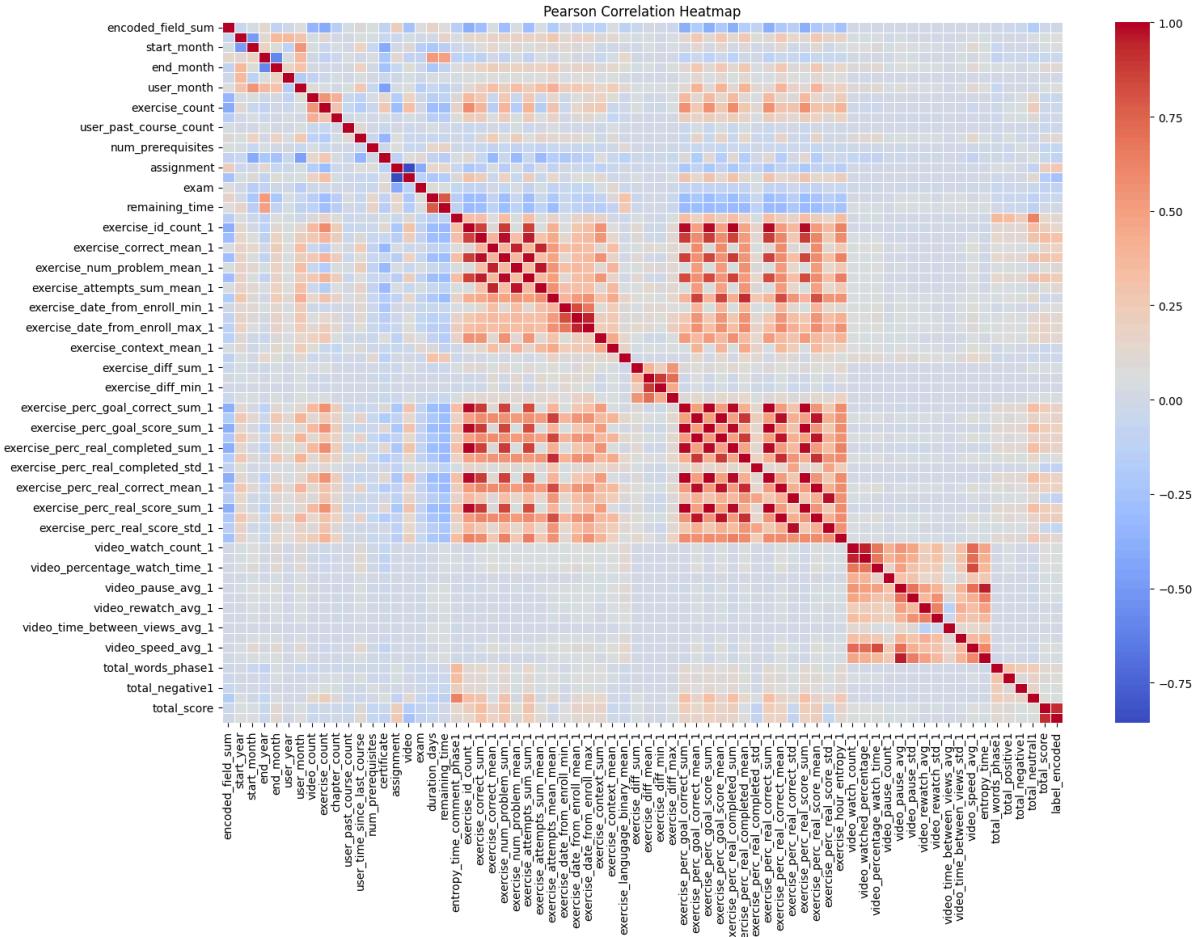
- Hành vi làm bài tập
 - Các nhóm C, B, A xuất hiện những học viên có số lượng bài tập vượt xa mức phổ biến, thể hiện sự đa dạng hành vi học tập trong nhóm.
 - Ngược lại, các nhóm D và E có hành vi làm bài tập trung hơn, không có ngoại lệ đáng chú ý.
 - Tổng số bài tập có xu hướng tăng dần từ E → A, tuy nhiên: Nhóm D có phân bố rộng (khoảng từ 10 đến 80 bài), cho thấy sự phân tán cao trong hành vi làm bài dù điểm số không quá cao.
- Hành vi xem video
 - Nhóm A có hành vi xem video đồng đều, không có sự chênh lệch lớn giữa các học viên.
 - Nhóm C thể hiện sự phân hóa rõ rệt: một số học viên xem video rất nhiều, trong khi một số khác lại rất ít.
 - Nhóm B cũng có sự chênh lệch nhất định.
 - Tổng số lượt xem video tăng dần từ nhóm E đến A, phản ánh mối liên hệ giữa mức độ theo dõi nội dung và kết quả học tập.
 - Số lượng khóa học đã đăng ký
- Nhóm E có xu hướng đăng ký nhiều khóa học, tuy nhiên kết quả học tập lại thấp. Điều này có thể do thiếu cam kết, quá tải, hoặc thiếu chiến lược học tập hiệu quả.
- Kết luận: Nhóm học viên có điểm số cao (A, B, C) thể hiện sự chủ động và đa dạng trong hành vi học tập, đặc biệt là làm bài và xem video. Trong khi đó, nhóm điểm thấp (D, E) có hành vi học đồng đều hơn nhưng kém tích cực, nhất là với nhóm E – số lượng khóa học lớn nhưng hiệu quả học tập không cao.

5.2.5.4. Tính toán hệ số tương quan

Hệ số tương quan đo lường mức độ và hướng của mối quan hệ tuyến tính giữa hai biến số. Các bước áp dụng

- **Chọn các cột số:** Chọn các cột có kiểu dữ liệu số từ DataFrame đã làm sạch (user_train_phase_1_cleaned).
- **Tính toán ma trận tương quan:**
 - Sử dụng phương thức .corr() trên DataFrame chứa các cột số để tính toán ma trận tương quan.
 - Tính toán cả tương quan Spearman và Pearson.
 - **Tương quan Pearson:** Đo lường mối quan hệ tuyến tính. Nó giả định dữ liệu có phân phối chuẩn và mối quan hệ là tuyến tính.
 - **Tương quan Spearman:** Đo lường mối quan hệ đơn điệu (monotonic). Nó không giả định phân phối chuẩn hoặc mối quan hệ tuyến tính, mà dựa trên thứ hạng của dữ liệu.
- **Trực quan hóa ma trận tương quan:**
 - Sử dụng heatmap (sns.heatmap) để trực quan hóa ma trận tương quan. Màu sắc trên heatmap biểu thị giá trị của hệ số tương quan, giúp bạn dễ dàng xác định các cặp feature có tương quan cao (cả dương và âm).





- Nhìn chung, không có đặc trưng nào có độ tương quan tuyệt đối cao với `total_score`. Điều này cho thấy kết quả học tập bị ảnh hưởng bởi nhiều yếu tố kết hợp, không phụ thuộc duy nhất vào một đặc trưng nào.
- Tuy nhiên, một số nhóm đặc trưng có tương quan tương đối cao, bao gồm:
 - Các đặc trưng liên quan đến bài tập: như `exercise_correct_mean_1`, `exercise_diff_mean_1`, `exercise_perc_real_score_mean_1`...
 - Các thành phần điểm trong khóa học: như `assignment`, `exam`, `video`, `certificate`, v.v.
 - Điều này cho thấy hiệu suất làm bài và mức độ hoàn thành các yêu cầu trong khóa học là những yếu tố đóng vai trò then chốt trong việc xác định điểm tổng kết `total_score`.

5.2.5.5. Loại bỏ những feature có tương quan cao

- **Xác định các cặp feature có tương quan cao:** Hàm `get_high_corr_pairs` để tìm các cặp feature có trị tuyệt đối của hệ số tương quan lớn hơn một ngưỡng nhất định (mặc định là 0.8).

```
# Improved function to get feature pairs with correlation > threshold
def get_high_corr_pairs(corr_matrix, threshold=0.8):
    corr_pairs = []
    cols = corr_matrix.columns
    for i in range(len(cols)):
        for j in range(i + 1, len(cols)):
            feature_1 = cols[i]
            feature_2 = cols[j]
            corr_value = corr_matrix.loc[feature_1, feature_2]
            if abs(corr_value) > threshold:
                corr_pairs.append((feature_1, feature_2, corr_value))
    return sorted(corr_pairs, key=lambda x: -abs(x[2]))

# Get high correlation pairs
high_spearman = get_high_corr_pairs(spearman_corr, threshold=0.8)
high_pearson = get_high_corr_pairs(pearson_corr, threshold=0.8)

# Print results
print("Highly correlated pairs (Spearman > 0.8):")
for pair in high_spearman:
    print(f"{pair[0]} & {pair[1]}: {pair[2]:.3f}")

print("\nHighly correlated pairs (Pearson > 0.8):")
for pair in high_pearson:
    print(f"{pair[0]} & {pair[1]}: {pair[2]:.3f}")
```

```
Highly correlated pairs (Spearman > 0.8):
video_watch_count_1 & video_watched_percentage_1: 1.000
video_watch_count_1 & video_pause_count_1: 1.000
video_pause_count_1 & video_pause_avg_1: 1.000
video_watched_percentage_1 & video_pause_count_1: 1.000
video_watched_percentage_1 & video_speed_avg_1: 1.000
video_watch_count_1 & video_speed_avg_1: 1.000
video_pause_count_1 & video_speed_avg_1: 1.000
video_pause_avg_1 & video_speed_avg_1: 1.000
video_watch_count_1 & video_pause_avg_1: 1.000
video_watched_percentage_1 & video_pause_avg_1: 1.000
video_watched_percentage_1 & video_percentage_watch_time_1: 1.000
video_watch_count_1 & video_percentage_watch_time_1: 1.000
video_percentage_watch_time_1 & video_speed_avg_1: 1.000
video_percentage_watch_time_1 & video_pause_count_1: 1.000
video_percentage_watch_time_1 & video_pause_avg_1: 1.000
exercise_num_problem_sum_1 & exercise_attempts_sum_sum_1: 0.999
exercise_perc_goal_correct_sum_1 & exercise_perc_goal_score_sum_1: 0.999
exercise_perc_real_correct_sum_1 & exercise_perc_real_score_sum_1: 0.999
exercise_perc_goal_correct_sum_1 & exercise_perc_real_correct_sum_1: 0.998
exercise_id_count_1 & exercise_perc_real_completed_sum_1: 0.998
exercise_perc_goal_score_sum_1 & exercise_perc_real_score_sum_1: 0.998
exercise_perc_goal_score_sum_1 & exercise_perc_real_correct_sum_1: 0.997
exercise_perc_goal_correct_sum_1 & exercise_perc_real_score_sum_1: 0.997
exercise_perc_real_correct_mean_1 & exercise_perc_real_score_mean_1: 0.996
exercise_perc_goal_correct_mean_1 & exercise_perc_goal_score_mean_1: 0.995
exercise_num_problem_mean_1 & exercise_attempts_sum_mean_1: 0.992
exercise_perc_real_correct_std_1 & exercise_perc_real_score_std_1: 0.990
exercise_date_from_enroll_mean_1 & exercise_date_from_enroll_max_1: 0.990
exercise_correct_sum_1 & exercise_num_problem_sum_1: 0.989
exercise_correct_sum_1 & exercise_attempts_sum_sum_1: 0.988
exercise_perc_real_completed_sum_1 & exercise_perc_real_correct_sum_1: 0.986
exercise_perc_real_completed_sum_1 & exercise_perc_real_score_sum_1: 0.985
```

Nhận xét:

- Rất nhiều biến về hành vi xem video có tương quan gần như tuyệt đối với nhau (Spearman = 1.000), ví dụ như: video_watch_count_1, video_pause_count_1, video_watched_percentage_1, video_speed_avg_1,... Điều này cho thấy các biến này gần như đang đo lường cùng một hành vi học tập – có thể gây dư thừa thông tin.
- Các biến liên quan đến bài tập (exercise) cũng thể hiện mức độ tương quan rất cao, đặc biệt giữa các biến tổng và trung bình như:
exercise_num_problem_sum_1 & exercise_attempts_sum_sum_1,
exercise_perc_goal_correct_sum_1 &
exercise_perc_real_correct_sum_1,... Điều này phản ánh một mối quan hệ chặt chẽ giữa khối lượng bài làm và hiệu quả làm bài.
- Nhiều biến về hiệu suất làm bài và độ khó bài tập có liên hệ chặt chẽ, ví dụ: exercise_diff_sum_1 & exercise_perc_real_completed_sum_1, exercise_correct_mean_1 & exercise_diff_sum_1,... Điều này cho thấy người học làm bài nhiều hơn thường có xu hướng đạt kết quả tốt hơn trong các bài tập có độ khó cao.

- Tính dư thừa thông tin là đáng kể, đặc biệt trong nhóm video và nhóm bài tập – việc giữ tất cả các biến có thể gây đa cộng tuyến nếu dùng trong các mô hình tuyến tính hoặc làm tăng độ phức tạp mô hình không cần thiết.
- Có một vài mối tương quan âm đáng chú ý, chẳng hạn assignment & video (Spearman = -0.861), cho thấy có thể tồn tại mối quan hệ bù trừ giữa các hoạt động học tập như xem video và làm bài tập. Việc khai thác mối quan hệ này có thể giúp cải thiện mô hình dự báo hành vi học tập.
- **Loại bỏ các feature có tương quan cao:** Hàm remove_high_corr_features để xác định và loại bỏ một trong hai feature trong các cặp có tương quan rất cao (ngưỡng 0.98 trong mã của bạn). Việc loại bỏ này nhằm giảm thiểu đa cộng tuyến, có thể gây vấn đề cho một số mô hình máy học. Khi loại bỏ ưu tiên giữ lại feature có tương quan cao hơn với biến mục tiêu (label_encoded).

```
# Function to remove highly correlated features
def remove_high_corr_features(corr_matrix, threshold=0.90):
    # List of features to drop
    to_drop = set()

    # Iterate over pairs of features
    for i in range(len(corr_matrix.columns)):
        for j in range(i+1, len(corr_matrix.columns)):
            # Check if correlation is greater than the threshold
            if abs(corr_matrix.iloc[i, j]) > threshold:
                feature_1 = corr_matrix.columns[i]
                feature_2 = corr_matrix.columns[j]

                # Choose to drop the feature that is least important or least correlated with the label
                if feature_1 in to_drop or feature_2 in to_drop:
                    continue
                # Keep the feature that is more correlated with the label (using 'label_encoded' or target)
                if pearson_corr.loc[feature_1, 'label_encoded'] > pearson_corr.loc[feature_2, 'label_encoded']:
                    to_drop.add(feature_2) # Remove feature_2
                else:
                    to_drop.add(feature_1) # Remove feature_1

    return to_drop

# Get high correlation features to remove
features_to_remove = remove_high_corr_features(pearson_corr, threshold=0.98)
```

Quy trình hợp nhất dữ liệu, xử lý và EDA dữ liệu lặp lại cho 4 phase.

5.2.6. Ghi nhãn dữ liệu

5.2.6.1. Cơ sở tính điểm

Nhóm vẫn duy trì phương pháp tính điểm như trên nền tảng **XuetangX**, vốn sử dụng mô hình đánh giá dựa trên nhiều thành phần để phản ánh toàn diện quá trình học tập của học viên. Tuy nhiên, để phục vụ mục đích gán nhãn và phân tích dữ liệu học tập, nhóm xác định rõ **3 thành phần điểm chính** như sau:

Thành phần điểm:

Thành phần	Bao gồm	Ghi chú thêm
Assignment	Bài tập trắc nghiệm, tự luận, câu hỏi chương	Gộp cả điểm Discussion (nếu có)
Exam	Chỉ tính điểm Final Exam	Bỏ qua điểm giữa kỳ nếu có
Video	Tỉ lệ xem video, mức độ hoàn thành tài liệu học	Gộp thêm phần Reading (nếu có)

Công thức tính điểm tổng kết:

$$\text{Score} = (w_1 \times \text{Assignment}) + (w_2 \times \text{Exam}) + (w_3 \times \text{Video})$$

- Trong đó: $w_1 + w_2 + w_3 = 1$
- Trọng số w_1, w_2, w_3 được lấy theo từng khóa học cụ thể trên XuetangX.
- Các khóa học có thể thay đổi trọng số, nhưng nhìn chung:
 - Assignment: 20% – 40%
 - Exam (final): 30% – 50%
 - Video/Reading: 20% – 30%

5.2.6.2. Chiến lược xếp loại kết quả của học viên.

Nền tảng XuetangX mặc định chia thành 2 mức:

- A: Đạt $\geq 60\%$
- F: Không đạt $< 60\%$

Tuy nhiên, nhóm mở rộng thành **5 mức đỗ**:

- Phân tích tốt hơn chất lượng học tập ở các cấp độ khác nhau
- Phục vụ gán nhãn mô hình học máy
- Vẫn đảm bảo **tính tương thích với hệ thống gốc** bằng cách **giữ nguyên 60%** là điều kiện đạt

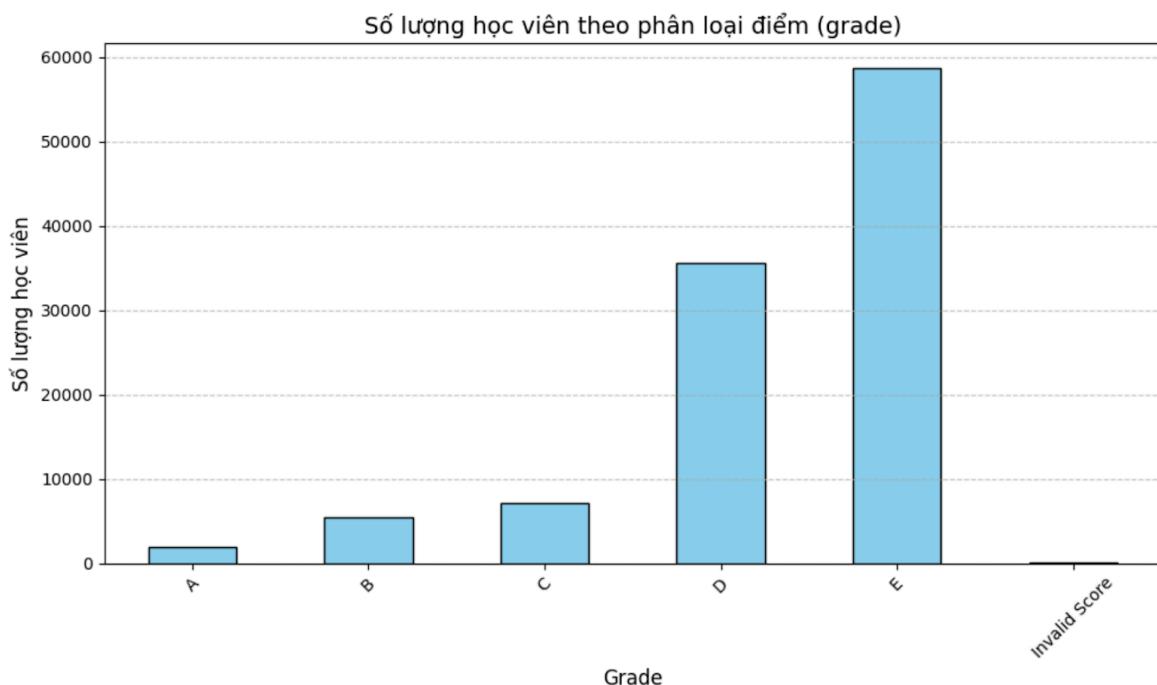
Nguyên tắc gán nhãn:

- **Mốc 60%** là ranh giới rõ ràng để phân biệt học viên **đạt** và **không đạt**.
- Các học viên có điểm $< 60\%$ đều bị gán nhãn là **Fail**, không đủ điều kiện nhận chứng chỉ.
- Các học viên có điểm $\geq 60\%$ sẽ được chia tiếp thành **4 mức độ thành công**, phản ánh mức độ hoàn thành từ cơ bản đến xuất sắc.

Bảng phân loại chi tiết:

Nhãn	Điểm số (Score)	Mức độ hoàn thành
A (Excellent)	$85 \leq \text{Score} \leq 100$	Hoàn thành xuất sắc; thành thạo kiến thức và kỹ năng.
B (Good)	$70 \leq \text{Score} < 85$	Nắm vững nội dung; vượt yêu cầu cơ bản.
C (Pass)	$60 \leq \text{Score} < 70$	Đủ điều kiện để nhận tín chỉ của khóa học. Đáp ứng phần lớn mục tiêu học tập.
D (Fail)	$30 \leq \text{Score} < 60$	Không đạt để nhận tín chỉ.
E (Inactive)	$0 \leq \text{Score} < 30$	Học viên hầu như không hoạt động.

Dữ liệu sau khi gán nhãn



5.2.7. Chia tập dữ liệu

5.2.7.1. Xác định thời gian cho tập dữ liệu test

5.2.7.1.1. Thông tin về thời gian bắt đầu, kết thúc khóa học

Thời gian bắt đầu khóa học

Dữ liệu thu thập cho thấy các khóa học được giới hạn bắt đầu trong hai năm chính là 2019 và 2020. Trong đó, năm 2020 là năm có số lượng khóa học được khai giảng nhiều nhất. Đặc biệt, thời gian bắt đầu các khóa học trong năm 2020 có sự phân bố không đồng đều qua các tháng.

- Hai thời điểm cao điểm mở khóa học được học viên đăng ký nhiều là tháng 2 và tháng 9.
- Trong số đó, tháng 2 nổi bật là tháng có số lượng khóa học khai giảng cao nhất, với 26.129 khóa học được mở, vượt xa các tháng còn lại.

```
user_train_phase_1_cleaned[['start_year', 'start_month']].value_counts()
```

```
start_year  start_month
2020.0      2.0          26129
              6.0          19543
              9.0          14518
              4.0          8874
              7.0          8623
2019.0      12.0         6262
2020.0      10.0         6216
              3.0          6083
              8.0          5270
              5.0          2304
              11.0         1421
2019.0      9.0          1171
              8.0          913
2020.0      1.0          385
2019.0      11.0         323
              10.0         73
2020.0      12.0         14
Name: count, dtype: int64
```

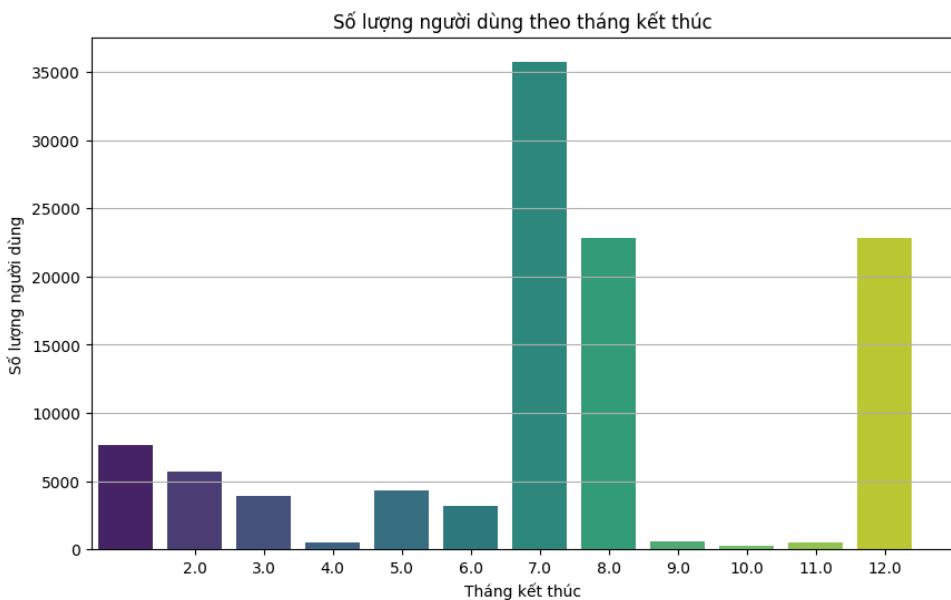
Thời gian kết thúc khóa học

Về mặt thời gian kết thúc, các khóa học có thời điểm kết thúc trải dài từ năm 2019 đến năm 2021. Trong đó:

- Năm 2020 là năm có số lượng khóa học kết thúc nhiều nhất, trùng với thời điểm nhiều khóa học cũng được khai giảng.
- Các tháng có nhiều khóa học kết thúc nhất là tháng 7, tháng 8 và tháng 12, cho thấy chu kỳ kết thúc khóa học thường rơi vào giữa năm và cuối năm.

```
: user_info['end_year'].value_counts()
```

```
: end_year
2020.0    94451
2021.0    13633
2019.0     38
Name: count, dtype: int64
```



Điểm đáng chú ý là một số khóa học bắt đầu vào tháng 7 năm 2020 nhưng lại kéo dài và kết thúc vào năm 2021. Điều này cho thấy có sự đa dạng rõ rệt về độ dài của các khóa học – từ các khóa ngắn hạn (vài tuần hoặc vài tháng) đến các chương trình đào tạo dài hạn (trên 6 tháng, thậm chí 1 năm).

Chính vì sự khác biệt này, việc phân loại và phân tích dữ liệu dựa theo thời gian bắt đầu hoặc kết thúc của khóa học có thể gặp khó khăn. Cần lưu ý đến khoảng thời gian diễn ra toàn bộ khóa học, thay vì chỉ xem xét ngày bắt đầu hoặc ngày kết thúc một cách riêng lẻ.

```
user_info[(user_info['start_year'] >= 2020) & (user_info['start_month'] >= 7)][['end_year', 'end_month']].value_counts()
```

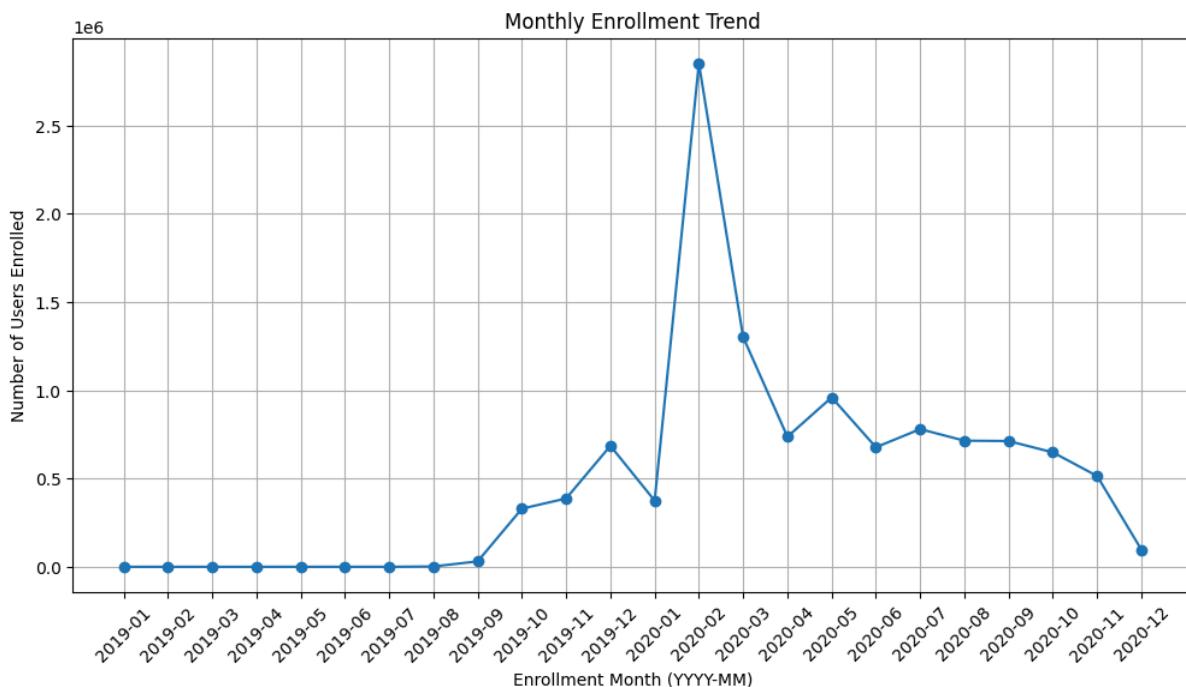
end_year	end_month	count
2020.0	12.0	22067
2021.0	1.0	6639
	2.0	4339
2020.0	8.0	1147
2021.0	8.0	530
2020.0	9.0	478
	11.0	445
2021.0	5.0	169
2020.0	10.0	151
2021.0	3.0	77
	9.0	12
	4.0	8

Name: count, dtype: int64

Thời gian tham gia khóa học của user

Thời gian tham gia khóa học của người học được khảo sát trong khoảng từ tháng 1 năm 2019 đến tháng 12 năm 2020. Dữ liệu này giúp hiểu rõ hơn về mức độ quan tâm, thời điểm đăng ký và xu hướng học tập của người dùng theo từng giai đoạn.

- Tháng 1 và tháng 2 năm 2020 là hai tháng có số lượng người đăng ký khóa học cao nhất trong toàn bộ khoảng thời gian khảo sát.
- Từ tháng 7 năm 2020 trở đi, dữ liệu cho thấy số lượng người dùng đăng ký bắt đầu giảm dần.



5.2.7.1.2. Chia tập train và test theo thời gian

Để tránh việc dữ liệu bị rõ rẽ, nhóm đã chọn giới hạn khoảng thời gian cho tập train và test

a. Tập test

- Những học viên đăng ký các khóa học bắt đầu từ tháng 7 năm 2020.
- Những khóa học này có thể có thời gian khai giảng trước tháng 7 năm 2020.

```
user_train_phase_1_filtered = user_train_phase_1_filtered[
    (
        (user_train_phase_1_filtered['user_year'] >= 2020) & (user_train_phase_1_filtered['user_month'] >= 6)
    ) |
    (user_train_phase_1_filtered['user_year'] > 2020)
]
```

b. Tập train

- Những học viên đăng ký khóa học không có trong tập test.
- Những khóa học từ năm 2019 và kết thúc trước tháng 7 năm 2020.

```
filtered = user_train_phase_1_cleaned[
    ~user_train_phase_1_cleaned.apply(lambda row: (row['user_id'], row['course_id']) in latest_use
r_course_pairs, axis=1)
]

filtered = filtered[
    ((filtered['end_year'] == 2020) & (filtered['end_month'] < 6)) | (filtered['end_year'] == 201
9)
]
```

5.2.7.2. Tổng quan tập train và test

5.2.7.2.1. Giai đoạn 1 (phase 1)

Một object (kết hợp giữa user và course) được đưa vào giai đoạn 1 nếu thỏa mãn tất cả các điều kiện sau:

1. Thời gian kể từ ngày học viên đăng ký khóa học đã kéo dài ít nhất 14 ngày.
2. Thời gian còn lại của khóa học (tính từ thời điểm học viên đăng ký đến ngày kết thúc khóa học) vẫn còn nhiều hơn 14 ngày.
3. Học viên đã có thực hiện ít nhất một bài tập trong khoảng thời gian đó.

Tại thời điểm đủ điều kiện (sau 14 ngày kể từ ngày đăng ký), các đặc trưng (features) liên quan đến object sẽ được ghi nhận và cập nhật.

Dữ liệu tập train: 10570 hàng × 70 cột

Dữ liệu tập test: 756 hàng × 70 cột

	user_id	school	course_id	field_encoded_1	field_encoded_2	start_year	start_month	end_year	end_month	user_year	user_month
1	U_1000979	云南大学	C_947149	38	36	2019.0	12.0	2020.0	4.0	1	1
3	U_1001176	云南大学	C_947149	38	36	2019.0	12.0	2020.0	4.0	1	1
15	U_10035349	南开大学	C_707456	38	36	2019.0	8.0	2020.0	2.0	1	1
22	U_1004792	昆明理工大学	C_735164	13	0	2019.0	9.0	2019.0	12.0	1	1
41	U_10060293	南开大学	C_707456	38	36	2019.0	8.0	2020.0	2.0	1	1
...
107926	U_9894275	中国人民解放军空军军西安飞行学院	C_947244	38	36	2019.0	12.0	2020.0	3.0	1	1
107990	U_9936073	北京林业大学	C_737592	38	36	2019.0	9.0	2020.0	1.0	1	1
108076	U_9949439	中南财经政法大学	C_682549	38	36	2019.0	12.0	2020.0	3.0	1	1
108081	U_9953958	中南财经政法大学	C_682549	38	36	2019.0	12.0	2020.0	3.0	1	1
108084	U_995473	昆明理工	C_735164	13	0	2020.0	2.0	2020.0	5.0	1	1

Gồm các đặc trưng sau: course_id, user_id, school, field_encoded_1, field_encoded_2, start_year, start_month, end_year, end_month, user_year , user_month, video_count, exercise_count, chapter_count, user_past_course_count, user_time_since_last_course, num_prerequisites, certificate, assignment, video, exam, type, duration_days, remaining_time, entropy_time_comment_phase1, exercise_id_count_1, exercise_correct_sum_1, exercise_correct_mean_1, exercise_num_problem_sum_1, exercise_num_problem_mean_1, exercise_attempts_sum_mean_1, exercise_attempts_mean_mean_1, exercise_date_from_enroll_min_1, exercise_date_from_enroll_mean_1, exercise_date_from_enroll_max_1, exercise_context_sum_1, exercise_context_mean_1, exercise_language_binary_mean_1, exercise_diff_sum_1, exercise_diff_mean_1, exercise_diff_min_1, exercise_diff_max_1, exercise_perc_goal_score_mean_1, exercise_perc_real_completed_mean_1, exercise_perc_real_completed_std_1, exercise_perc_real_correct_mean_1, exercise_perc_real_correct_std_1, exercise_perc_real_score_sum_1, exercise_perc_real_score_mean_1, exercise_perc_real_score_std_1, exercise_hour_entropy_1, video_watch_count_1, video_WATCHED_PERCENTAGE_1, video_PERCENTAGE_WATCH_TIME_1, video_PAUSE_COUNT_1, video_PAUSE_AVG_1, video_PAUSE_STD_1, video_REWATCH_AVG_1, video_REWATCH_STD_1, video_TIME_BETWEEN_VIEWS_AVG_1, video_TIME_BETWEEN_VIEWS_STD_1, video_SPEED_AVG_1, entropy_time_1, total_words_phase1, total_positive1, total_negative1, total_neutral1.

Nhãn: total_score, label, label_encoded

5.2.7.2.2. Giai đoạn 2 (phase 2)

Một object (tổ hợp giữa user và course) sẽ được xét vào giai đoạn 2 nếu đáp ứng tất cả các điều kiện sau:

1. Thời gian kể từ ngày học viên đăng ký khóa học đã kéo dài ít nhất 28 ngày.
2. Thời gian còn lại của khóa học (tính từ thời điểm đăng ký đến ngày kết thúc) vẫn còn lớn hơn 28 ngày.
3. Object đã từng xuất hiện trong giai đoạn 1, tức là đã thỏa mãn các điều kiện ban đầu sau 14 ngày và được ghi nhận từ trước.

Tại thời điểm đủ điều kiện (sau 28 ngày kể từ ngày đăng ký), các đặc trưng (features) của object sẽ tiếp tục được cập nhật để phản ánh hành vi học tập mới nhất của học viên trong khoảng thời gian dài hơn.

Tập train: 8548 hàng × 113 cột

Tập test: 746 hàng × 113 cột

	course_id	user_id	school	field_encoded_1	field_encoded_2	start_year	start_month	end_year	end_month	use
0	C_1123979	U_30144337	河北地质大学	22	38	2020.0	5.0	2020.0	8.0	2020.0
1	C_1159827	U_12083380	北京体育大学	38	36	2020.0	2.0	2020.0	8.0	2020.0
2	C_1410076	U_16214293	潍坊学院	38	36	2020.0	2.0	2020.0	7.0	2020.0
3	C_1410117	U_20814393	None	38	36	2020.0	2.0	2020.0	7.0	2020.0
4	C_1410126	U_102435	南方科技大学	38	36	2020.0	2.0	2020.0	7.0	2020.0
...
741	C_948431	U_15004330	None	38	36	2020.0	3.0	2020.0	7.0	2020.0
742	C_948435	U_10194888	青海大学	38	36	2020.0	4.0	2020.0	7.0	2020.0
743	C_948436	U_10158440	贵州师范大学	38	36	2020.0	2.0	2020.0	7.0	2020.0
744	C_949541	U_13735255	西京学院	38	36	2020.0	5.0	2020.0	7.0	2020.0
745	C_956130	U_14147218	计算机学院	38	36	2020.0	9.0	2020.0	12.0	2020.0

Gồm các đặc trưng: user_id, school, course_id, field_encoded_1, field_encoded_2, start_year, start_month, end_year, end_month, user_year, user_month, video_count, exercise_count, chapter_count, user_past_course_count, user_time_since_last_course, num_prerequisites, certificate, assignment, video, exam, type, duration_days, remaining_time, entropy_time_comment_phase1, exercise_id_count_1, exercise_correct_sum_1, exercise_correct_mean_1, exercise_num_problem_sum_1, exercise_num_problem_mean_1, exercise_attempts_sum_mean_1,

exercise_attempts_mean_mean_1, exercise_date_from_enroll_min_1,
exercise_date_from_enroll_mean_1, exercise_date_from_enroll_max_1,
exercise_context_sum_1, exercise_context_mean_1,
exercise_language_binary_mean_1, exercise_diff_sum_1, exercise_diff_mean_1,
exercise_diff_min_1, exercise_diff_max_1, exercise_perc_goal_score_mean_1,
exercise_perc_real_completed_mean_1, exercise_perc_real_completed_std_1,
exercise_perc_real_correct_mean_1, exercise_perc_real_correct_std_1,
exercise_perc_real_score_sum_1, exercise_perc_real_score_mean_1,
exercise_perc_real_score_std_1, exercise_hour_entropy_1, video_watch_count_1,
video_watched_percentage_1, video_percentage_watch_time_1,
video_pause_count_1, video_pause_avg_1, video_pause_std_1,
video_rewatch_avg_1, video_rewatch_std_1, video_time_between_views_avg_1,
video_time_between_views_std_1, video_speed_avg_1, entropy_time_1,
total_words_phase1, total_positive1, total_negative1, total_neutral1,
entropy_time_comment_phase2, exercise_id_count_2, exercise_correct_sum_2,
exercise_correct_mean_2, exercise_num_problem_sum_2,
exercise_num_problem_mean_2, exercise_attempts_sum_mean_2,
exercise_attempts_mean_mean_2, exercise_date_from_enroll_min_2,
exercise_date_from_enroll_mean_2, exercise_date_from_enroll_max_2,
exercise_context_sum_2, exercise_context_mean_2,
exercise_language_binary_mean_2, exercise_diff_sum_2, exercise_diff_mean_2,
exercise_diff_min_2, exercise_diff_max_2, exercise_perc_goal_score_mean_2,
exercise_perc_real_completed_mean_2, exercise_perc_real_completed_std_2,
exercise_perc_real_correct_mean_2, exercise_perc_real_correct_std_2,
exercise_perc_real_score_sum_2, exercise_perc_real_score_mean_2,
exercise_perc_real_score_std_2, exercise_hour_entropy_2, video_watch_count_2,
video_watched_percentage_2, video_percentage_watch_time_2,
video_pause_count_2, video_pause_avg_2, video_pause_std_2,
video_rewatch_avg_2, video_rewatch_std_2, video_time_between_views_avg_2,
video_time_between_views_std_2, video_speed_avg_2, entropy_time_2,
total_words_phase2, total_positive2, total_negative2, total_neutral2

(có thêm các đặc trưng trong giai đoạn 2)

Nhãn: total_score, label, label_encoded

5.2.7.2.3. Giai đoạn 3 (phase 3)

Một object (kết hợp giữa user và course) sẽ được xét vào giai đoạn 3 nếu thỏa mãn tất cả các điều kiện sau:

1. Thời gian kể từ ngày học viên đăng ký khóa học đã kéo dài ít nhất 42 ngày.

2. Thời gian còn lại đến khi khóa học kết thúc (tính từ thời điểm đăng ký) vẫn còn lớn hơn 42 ngày.
3. Object đã từng xuất hiện trong giai đoạn 2, tức là đã được ghi nhận và theo dõi từ trước ở các giai đoạn trước đó.

Tại thời điểm đủ điều kiện (sau 42 ngày kể từ ngày đăng ký), các đặc trưng (features) của object tiếp tục được cập nhật nhằm phản ánh hành vi học tập mới nhất và giúp theo dõi tiến trình học tập dài hạn của học viên.

Tập train: 7803 hàng × 156 cột

Tập test: 721 hàng × 156 cột

5.2.7.2.4. Giai đoạn 4 (phase 4)

Một object (tổ hợp giữa user và course) sẽ được xét vào giai đoạn 4 nếu thỏa mãn tất cả các điều kiện sau:

1. Thời gian kể từ ngày học viên đăng ký khóa học đã kéo dài ít nhất 56 ngày.
2. Thời gian còn lại của khóa học (tính từ thời điểm đăng ký) vẫn còn lớn hơn 56 ngày.
3. Object đã từng xuất hiện trong giai đoạn 3, nghĩa là đã được theo dõi liên tục từ các giai đoạn trước.

Khi đủ điều kiện tại mốc 56 ngày kể từ ngày đăng ký, các đặc trưng (features) của object sẽ tiếp tục được cập nhật, nhằm phản ánh hành vi và tiến độ học tập mới nhất của học viên trong giai đoạn dài hạn.

Tập train: 5363 hàng x 199 cột

Tập test: 687 hàng x 199 cột

5.2.7.2.5. Tổng hợp các giai đoạn

Giai đoạn	Tập dữ liệu	Hàng x Cột
Giai đoạn 1	Train	10570 hàng × 70 cột
	Test	756 hàng × 70 cột
Giai đoạn 2	Train	8548 hàng × 113 cột
	Test	746 hàng × 113 cột
Giai đoạn 3	Train	7803 hàng × 156 cột
	Test	721 hàng × 156 cột

Giai đoạn 4	Train	5363 hàng x 199 cột
	Test	687 hàng x 199 cột

5.2.7.3. Tập dữ liệu validation

Nhóm thống nhất sử dụng Stratified K-Fold, cho việc chia dữ liệu validation trong dữ liệu train, giúp tìm các siêu tham số cho mô hình.

Stratified K-Fold là cải tiến của K-Fold và rất quan trọng khi làm việc với tập dữ liệu bị mất cân bằng.

- K-Fold Cross-Validation (CV):
 - Chia ngẫu nhiên dữ liệu thành K phần (folds) bằng nhau.
 - Mỗi fold có thể không đảm bảo giữ tỷ lệ phân bố của các lớp (labels) giống nhau.
 - Khi dữ liệu mất cân bằng (ví dụ: 90% lớp A, 10% lớp B), một số fold có thể gần như không có mẫu thuộc lớp thiểu số → gây ra đánh giá sai lệch.
- Stratified K-Fold Cross-Validation:
 - Cũng chia dữ liệu thành K folds, nhưng đảm bảo mỗi fold giữ nguyên tỷ lệ phân bố các lớp như trong toàn bộ tập dữ liệu.
 - Tức là nếu toàn bộ tập có 10% lớp B, thì mỗi fold cũng sẽ có ~10% mẫu lớp B.

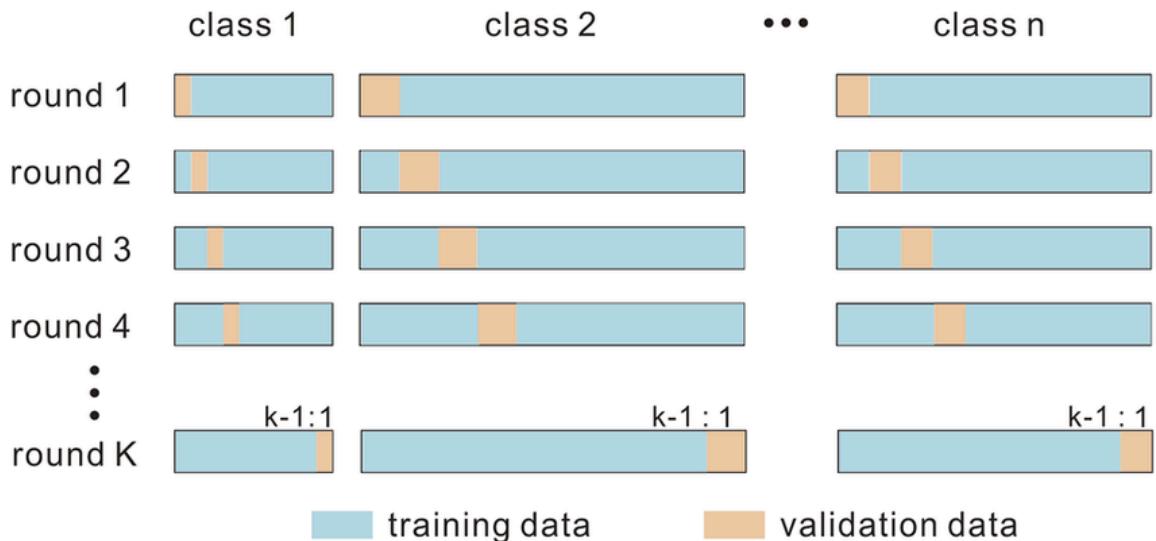
Lợi ích khi dùng Stratified K-Fold:

- Đảm bảo đại diện công bằng của tất cả các lớp trong mỗi fold.
- Giúp mô hình được huấn luyện và đánh giá trên tập con dữ liệu đầy đủ thông tin, không bị lệch lớp.
- Tránh overfitting hoặc underfitting cục bộ, đặc biệt với lớp thiểu số.
- Đánh giá chính xác hơn hiệu suất thực tế của mô hình, đặc biệt với các chỉ số như precision, recall, F1-score cho lớp thiểu số.

Siêu tham số k-fold được chọn là 5.

- $K = 5$ là một lựa chọn phổ biến khi sử dụng K-Fold hoặc Stratified K-Fold. Việc chọn $K = 5$ giúp có đủ số lần huấn luyện và kiểm tra mà không làm mất quá nhiều thời gian tính toán.
- Với $K = 5$, mô hình sẽ được huấn luyện 5 lần, giúp tăng độ tin cậy trong việc đánh giá và chọn lựa siêu tham số, đồng thời vẫn giữ được hiệu quả tính toán.

- Với các mô hình mạng học sâu, sẽ chỉ lấy một lần fold để làm tập dữ liệu train (huấn luyện mô hình) và valid (để theo dõi kết quả của mô hình qua các epoch).



```

X = df.drop(columns=[label_column])
y = df[label_column]

skf = StratifiedKFold(n_splits=k, shuffle=True, random_state=42)

for fold, (train_idx, val_idx) in enumerate(skf.split(X, y)):
    print(f"\n➡ Phase {phase_num} - Fold {fold + 1}")

    X_train, X_val = X.iloc[train_idx], X.iloc[val_idx]
    y_train, y_val = y.iloc[train_idx], y.iloc[val_idx]

    # Thư mục lưu kết quả fold
    fold_dir = f'outputs/phase{phase_num}/fold{fold+1}'
    os.makedirs(fold_dir, exist_ok=True)

    # Lưu train/val trước SMOTE
    X_train.to_csv(f'{fold_dir}/X_train.csv', index=False)
    y_train.to_csv(f'{fold_dir}/y_train.csv', index=False)
    X_val.to_csv(f'{fold_dir}/X_val.csv', index=False)
    y_val.to_csv(f'{fold_dir}/y_val.csv', index=False)

```

5.3. Chất lượng dữ liệu và thông tin về dữ liệu

5.3.1. Input của bài toán

Nhóm	Tên cột	Ý nghĩa
------	---------	---------

Thông tin khóa học (course_*)	course_id	Mã khóa học
	num_prerequisites	Số lượng môn học tiên quyết
	encoded_field	Lĩnh vực đã được mã hóa
	start_date	Ngày bắt đầu khóa học
	end_date	Ngày kết thúc khóa học
	duration_days	Độ dài của khóa học (ngày)
Tài liệu khóa học (resource_*)	video_count	Tổng số lượng video
	exercise_count	Tổng số lượng bài tập
	chapter_count	Tổng số lượng chương
Thành phần điểm (score_*)	assignment	Điểm bài tập
	exam	Điểm bài thi cuối kỳ
	video	Điểm từ xem video
	certificate	Có nhận chứng chỉ hay không (0, 1)
Thông tin người dùng (user_*)	user_id	ID người dùng
	school	Trường học
	user_enroll_time	Thời gian người dùng đăng ký khóa học
	user_past_course_count	Số lượng khóa học đã đăng ký trước đó
	user_duration_since_last_course	Khoảng cách (ngày) từ lần đăng ký hiện tại đến lần gần nhất
Hành vi học tập - Bình luận (comment_*)	comment_count_phase{i}	Số lượng bình luận trong giai đoạn i

- Theo từng đợt		
	total_words_phase{i}_x	Tổng số từ trong các bình luận trong giai đoạn i
	entropy_time_comment_phase{i}	Mức độ phân tán thời gian của bình luận (entropy) trong giai đoạn i
Hành vi học tập - Phản hồi (reply_*) - Theo từng đợt	reply_count_phase{i}	Số lượng phản hồi trong giai đoạn i
	total_words_phase{i}_y	Tổng số từ trong các phản hồi trong giai đoạn i
	entropy_time_reply_phase{i}	Mức độ phân tán thời gian của phản hồi (entropy) trong giai đoạn i
Hành vi học tập - Bài tập (exercise_*) - Theo từng đợt	exercise_id_count_{i}	Số lượng bài tập (exercise) được làm trong giai đoạn i
	exercise_correct_sum_{i}	Tổng số câu trả lời đúng trong giai đoạn i
	exercise_correct_mean_{i}	Tỷ lệ trả lời đúng trung bình của mỗi bài tập trong giai đoạn i
	exercise_num_problem_sum_{i}	Tổng số câu hỏi trong tất cả các bài tập trong giai đoạn i
	exercise_num_problem_mean_{i}	Số câu hỏi trung bình mỗi bài trong giai đoạn i
	exercise_attempts_sum_sum_{i}	Tổng số lần thử làm các bài tập trong giai đoạn i
	exercise_attempts_sum_mean_{i}	Số lần thử trung bình mỗi bài trong giai đoạn i
	exercise_attempts_mean_mean_{i}	Trung bình số lần thử mỗi câu hỏi (problem) trong giai đoạn i

	exercise_date_from_enrollment_min_{i}	Ngày đầu tiên làm bài tập (so với ngày đăng ký) trong giai đoạn i
	exercise_date_from_enrollment_mean_{i}	Ngày trung bình làm bài (so với ngày đăng ký) trong giai đoạn i
	exercise_date_from_enrollment_max_{i}	Ngày cuối cùng làm bài (so với ngày đăng ký)
	exercise_context_sum_{i}	Tổng số ngữ cảnh làm bài tập trong giai đoạn i
	exercise_context_mean_{i}	Trung bình số ngữ cảnh làm bài tập trong giai đoạn i
	exercise_language_binary_mean_{i}	Trung bình giá trị ngôn ngữ bài tập (1 = Tiếng Anh, 0 = Tiếng Trung) trong giai đoạn i
	exercise_diff_sum_{i}	Tổng thời gian làm bài tập trong giai đoạn i
	exercise_diff_mean_{i}	Trung bình thời gian làm mỗi bài tập trong giai đoạn i
	exercise_diff_min_{i}	Thời gian làm bài ít nhất trong giai đoạn i
	exercise_diff_max_{i}	Thời gian làm bài nhiều nhất trong giai đoạn i
	exercise_perc_goal_correct_sum_{i}	Tổng của tỷ lệ trả lời đúng giai đoạn i
	exercise_perc_goal_correct_mean_{i}	Trung bình của tỷ lệ trả lời đúng trong giai đoạn i
	exercise_perc_goal_score_sum_{i}	Tổng điểm đạt được trong giai đoạn i
	exercise_perc_goal_score_mean_{i}	Trung bình điểm đạt được của mỗi bài tập trong giai đoạn i

	exercise_perc_real_completed_sum_{i}	Tổng tỷ lệ hoàn thành bài tập thực tế trong giai đoạn i
	exercise_perc_real_completed_mean_{i}	Trung bình tỷ lệ hoàn thành thực tế
	exercise_perc_real_completed_std_{i}	Độ lệch chuẩn của tỷ lệ hoàn thành thực tế
	exercise_perc_real_correct_sum_{i}	Tổng tỷ lệ đúng thực tế
	exercise_perc_real_correct_mean_{i}	Tỷ lệ đúng thực tế trung bình
	exercise_perc_real_correct_std_{i}	Độ lệch chuẩn của tỷ lệ đúng thực tế
	exercise_perc_real_score_sum_{i}	Tổng điểm thực tế đạt được
	exercise_perc_real_score_mean_{i}	Trung bình điểm thực tế
	exercise_perc_real_score_std_{i}	Độ lệch chuẩn điểm thực tế
	exercise_hour_entropy_{i}	Entropy của thời gian làm bài tập theo giờ (phân tán trong ngày)
Hành vi học tập - Video (video_*) - Theo từng đợt	video_watched_count_{i}	Số lượng video được xem trong đợt i (user coi ở giây cuối cùng được coi là hoàn thành)
	video_watched_percentange_{i}	Tỷ lệ video đã xem trên tổng số video của khóa học trong từng đợt
	video_watch_time_{i}	Tổng thời gian user xem video trong đợt i
	video_time_{i}	Tổng thời gian video được chọn để coi đợt i

	video_percentage_watch_{i}	Phần trăm thời lượng user coi trên tổng thời lượng của video
	video_pause_count_{i}	Số lượng ngắt quãng trong tổng số video đợt i
	video_pause_avg_{i}	Trung bình số lần ngắt quãng đợt i
	video_pause_std_{i}	Độ lệch chuẩn số lần ngắt quãng đợt i
	video_rewatch_count_{i}	Số lần xem lại đơn vị video (segment) đợt i
	video_rewatch_avg_{i}	Trung bình số lần xem lại trên mỗi video đợt i
	video_rewatch_std_{i}	Độ lệch chuẩn số lần xem lại trên mỗi video đợt i
	video_time_between_videos_avg_{i}	Trung bình thời gian giữa các lần xem video đợt i
	video_speed_avg_{i}	Trung bình tốc độ xem video đợt i
	video_entropy_time_{i}	Entropy thời điểm user coi video đợt i
	video_final_score_percentage_{i}	Phần trăm điểm đặt được vào điểm cuối khóa
Nhãn	label	A, B, C, D, E thể hiện kết quả học tập cuối cùng của học viên

5.3.2. Metadata của dữ liệu

5.3.2.1. Đặc điểm dữ liệu

Dữ liệu được thiết kế dưới dạng là Dataframe.

Thành Phần	Ý Nghĩa	DataFrame

Objects	Hàng đại diện cho thực thể	Mỗi dòng là một học viên đăng ký một khóa học
Attributes	Cột đại diện cho đặc điểm của object	"user_id", "course_id", "course_name", "user_school", "exam", "exercise_count"...
Dữ liệu cụ thể	Giá trị trong ô	"U_123456", "C_584313", "微积分——极限理论与一元函数", "Tsinghua University", 85.0, 12

Ví dụ:

user_id	course_id	course_name	user_school	score_final_exam	exercise_count_1	video_watched_count_1
U_123456	C_584313	微积分——极限理论与一元函数	Tsinghua University	85.0	12	10

5.3.2.2 Thông tin chi tiết của các cột

Bảng course (Thông tin khóa học)

Tên cột	Mô tả	Kiểu dữ liệu	Khoảng giá trị/Giới hạn
course_id	Mã khóa học	string	Dạng C_xxxxxx, duy nhất
num_prerequisites	Số lượng môn tiên quyết	int	0 - 10 10 môn bắt buộc
field_x	Lĩnh vực khóa học	list[string]	0 - 10 lĩnh vực
num_field_x	Số lượng lĩnh vực liên quan	int	0-10
start_date	Số lượng lĩnh vực liên quan	int	Trong khoảng 2019 - 2021

end_date	Ngày kết thúc khóa học	string	Trong khoảng 2019 - 2021
duration_days	Độ dài khóa học (ngày)	int	7 - 365

Bảng resource (Tài liệu khóa học)

Tên cột	Mô tả	Kiểu dữ liệu	Khoảng giá trị/Giới hạn
course_id	Mã khóa học	string	Dạng C_xxxxxx, duy nhất
video_count	Tổng số lượng video	int	1 - 200
exercise_count	Tổng số lượng bài tập	int	1 - 300
chapter_count	Tổng số lượng chương	int	1 - 60

Bảng score (Thành phần điểm)

Tên cột	Mô tả	Kiểu dữ liệu	Khoảng giá trị/Giới hạn
course_id	Mã khóa học	string	Dạng C_xxxxxx
assignment	Điểm bài tập	float	0.0 - 100.0
exam	Điểm thi	float	0.0 - 100.0
video	Điểm từ xem video	float	0.0 - 100.0

Bảng user (Thông tin người dùng)

Tên cột	Mô tả	Kiểu dữ liệu	Khoảng giá trị/Giới hạn
user_id	ID người dùng	string	Dạng U_xxxxxx, duy nhất
school	Trường học	string	3 - 100 ký tự

user_enroll_time	Thời gian đăng ký	datetime	2019 - 2021
user_past_course_count	Số lượng khóa học đã đăng ký trước đó	int	0 - 100
user_time_since_last_course	Khoảng cách (ngày) từ lần đăng ký hiện tại đến lần gần nhất	float	0 - 1095 (~3 năm)

Bảng comment (Hoạt động bình luận của user theo đợt 2 tuần)

Tên cột	Mô tả	Kiểu dữ liệu	Khoảng giá trị/Giới hạn
user_id	ID người dùng	string	Dạng U_XXXXXX
course_id	Mã khóa học	string	Dạng C_XXXXXX
comment_count_phase{i}	Số lượng bình luận trong giai đoạn i	int	0 - 100
total_words_phase{i}_x	Tổng số từ trong các bình luận trong giai đoạn i	int	0 - 1000
entropy_time_comment_phase{i}	Mức độ phân tán thời gian của bình luận (entropy) trong giai đoạn i	float	0.0 - 1.0

Bảng reply (Hoạt động bình luận của user theo đợt 2 tuần)

Tên cột	Mô tả	Kiểu dữ liệu	Khoảng giá trị/Giới hạn
user_id	ID người dùng	string	Dạng U_XXXXXX
course_id	Mã khóa học	string	Dạng C_XXXXXX
reply_count_phase{i}	Số lượng phản hồi trong giai đoạn i	int	0 - 100
total_words_phase{i}_y	Tổng số từ trong các phản hồi trong giai đoạn i	int	0 - 1000

entropy_time_reply_phase_{i}	Mức độ phân tán thời gian của phản hồi (entropy) trong giai đoạn i	float	0.0 - 1.0
------------------------------	--	-------	-----------

Bảng exercise (Hành vi làm bài tập của user theo đợt 2 tuần)

Tên cột	Mô tả	Kiểu dữ liệu	Khoảng giá trị/Giới hạn
exercise_id_count_{i}	Số lượng bài tập (exercise) được làm trong giai đoạn i	int	0 - 50
exercise_correct_sum_{i}	Tổng số câu trả lời đúng trong giai đoạn i	int	0 - 50
exercise_correct_mean_{i}	Tỷ lệ trả lời đúng trung bình của mỗi bài tập trong giai đoạn i	float	0.0 - 1.0
exercise_num_problem_sum_{i}	Tổng số câu hỏi trong tất cả các bài tập trong giai đoạn i	int	≥ 0
exercise_num_problem_mean_{i}	Số câu hỏi trung bình mỗi bài trong giai đoạn i	float	≥ 0
exercise_attempts_sum_sum_{i}	Tổng số lần thử làm các bài tập trong giai đoạn i	int	≥ 0
exercise_attempts_sum_mean_{i}	Số lần thử trung bình mỗi bài trong giai đoạn i	float	≥ 0
exercise_attempts_mean_mean_{i}	Trung bình số lần thử mỗi câu hỏi (problem) trong giai đoạn i	float	≥ 0
exercise_date_from_enrol1_min_{i}	Ngày đầu tiên làm bài tập (so với ngày đăng ký) trong giai đoạn i	int	≥ 0
exercise_date_from_enrol1_mean_{i}	Ngày trung bình làm bài (so với ngày đăng ký) trong giai đoạn i	float	≥ 0

exercise_date_from_enrol_l_max_{i}	Ngày cuối cùng làm bài (so với ngày đăng ký)	int	≥ 0
exercise_context_sum_{i }	Tổng số ngữ cảnh làm bài tập trong giai đoạn i	int	≥ 0
exercise_context_mean_{ i}	Trung bình số ngữ cảnh làm bài tập trong giai đoạn i	float	≥ 0
exercise_language_binary_mean_{i}	Trung bình giá trị ngôn ngữ bài tập (1 = Tiếng Anh, 0 = Tiếng Trung) trong giai đoạn i	int	0 / 1
exercise_diff_sum_{i}	Tổng thời gian làm bài tập trong giai đoạn i	float	≥ 0
exercise_diff_mean_{i}	Trung bình thời gian làm mỗi bài tập trong giai đoạn i	float	≥ 0
exercise_diff_min_{i}	Thời gian làm bài ít nhất trong giai đoạn i	float	≥ 0
exercise_diff_max_{i}	Thời gian làm bài nhiều nhất trong giai đoạn i	float	≥ 0
exercise_perc_goal_correct_sum_{i}	Tổng của tỷ lệ trả lời đúng giai đoạn i	float	0.0 - 100.0
exercise_perc_goal_correct_mean_{i}	Trung bình của tỷ lệ trả lời đúng trong giai đoạn i	float	0.0 - 100.0
exercise_perc_goal_score_sum_{i}	Tổng điểm đạt được trong giai đoạn i	float	0.0 - 100.0
exercise_perc_goal_score_mean_{i}	Trung bình điểm đạt được của mỗi bài tập trong giai đoạn i	float	0.0 - 100.0
exercise_perc_real_completed_sum_{i}	Tổng tỷ lệ hoàn thành bài tập thực tế trong giai đoạn i	float	0.0 - 100.0
exercise_perc_real_completed_mean_{i}	Trung bình tỷ lệ hoàn thành thực tế	float	0.0 - 100.0

exercise_perc_real_compl eted_std_{i}	Độ lệch chuẩn của tỷ lệ hoàn thành thực tế	float	≥ 0
exercise_perc_real_correc t_sum_{i}	Tổng tỷ lệ đúng thực tế	float	0.0 - 100.0
exercise_perc_real_correc t_mean_{i}	Tỷ lệ đúng thực tế trung bình	float	0.0 - 100.0
exercise_perc_real_correc t_std_{i}	Độ lệch chuẩn của tỷ lệ đúng thực tế	float	≥ 0
exercise_perc_real_score_ sum_{i}	Tổng điểm thực tế đạt được	float	0.0 - 100.0
exercise_perc_real_score_ mean_{i}	Trung bình điểm thực tế	float	0.0 - 100.0
exercise_perc_real_score_ std_{i}	Độ lệch chuẩn điểm thực tế	float	≥ 0
exercise_hour_entropy_{i }	Entropy của thời gian làm bài tập theo giờ (phân tán trong ngày)	float	0.0 - 1.0

Thống kê dataset theo phase

```
# Đọc các file
df1 = pl.read_csv("/kaggle/input/final-data/user_train_phase_1.csv")
df2 = pl.read_csv("/kaggle/input/final-data/user_train_phase_2.csv")
df3 = pl.read_csv("/kaggle/input/final-data/user_train_phase_3.csv")
df4 = pl.read_csv("/kaggle/input/final-data/user_train_phase_4.csv")

print(f"Phase 1 có : {df1.shape[0]} user và {df1.shape[1]} feature" )
print(f"Phase 2 có : {df2.shape[0]} user và {df2.shape[1]} feature" )
print(f"Phase 3 có : {df3.shape[0]} user và {df3.shape[1]} feature" )
print(f"Phase 4 có : {df4.shape[0]} user và {df4.shape[1]} feature" )

Phase 1 có : 108810 user và 60 feature
Phase 2 có : 102252 user và 98 feature
Phase 3 có : 95400 user và 136 feature
Phase 4 có : 84241 user và 174 feature
```

Kết quả thống kê dataset theo phase

Xác định thông tin về bảng, số cột, kiểu dữ liệu: df.schema() và df.describe()

```
df.schema
:
Schema([('user_id', String),
         ('school', String),
         ('course_id', String),
         ('user_enroll_time', String),
         ('user_past_course_count', Int64),
         ('user_time_since_last_course', Float64),
         ('video_count', Int64),
         ('exercise_count', Int64),
         ('chapter_count', Int64),
         ('field_x', String),
         ('num_field_x', Int64),
         ('num_prerequisites', Int64),
         ('certificate', Float64),
         ('assignment', Float64),
         ('video', Float64),
         ('exam', Float64),
         ('type', Float64),
```

Xác định thông tin về bảng

Kiểm tra số lượng bản ghi (rows) và số cột (columns): df.shape

```
df.shape
:
(84241, 174)
```

Kết quả kiểm tra số lượng bản ghi

Dữ liệu gồm 84241 hàng (object) và có 174 cột (đặc trưng).

5.3.3. Phân tích thống kê dữ liệu

Kiểm tra thống kê tổng quát với .describe() (chỉ áp dụng cho cột số): df.describe()

In [5]: df.describe()

Out[5]: shape: (9, 175)

statistic	user_id	school	course_id	user_enroll_time	user_past_course_count	user_time_since_last_course	video_count	exercise
str	str	str	str	f64	f64	f64	f64	f64
"count"	"84241"	"39065"	"84241"	"84241"	84241.0	84241.0	84241.0	84241.0
"null_count"	"0"	"45176"	"0"	"0"	0.0	0.0	0.0	0.0
"mean"	null	null	null	null	0.44	431.968661	51.670374	26.9635
"std"	null	null	null	null	3.039712	1224.537762	23.283957	22.4033
		".重庆三峡医 药高等 专科学 校"	"C_1017355"	"2019-09-11"	0.0	0.0	8.0	1.0
"min"	"U_10000"							
"25%"	null	null	null	null	0.0	0.0	34.0	10.0
"50%"	null	null	null	null	0.0	0.0	47.0	17.0
"75%"	null	null	null	null	0.0	0.0	63.0	38.0
"max"	"U_99772"	"龙岩侨 中"	"C_956130"	"2020-12-08"	496.0	9912.0	144.0	83.0

Kết quả kiểm tra thống kê tổng quát

5.3.3.1. Kiểm tra số giá trị NULL trong từng cột:

```
# 1. Các kiểu dữ liệu số
numeric_types = (pl.Int8, pl.Int16, pl.Int32, pl.Int64,
                  pl.UInt8, pl.UInt16, pl.UInt32, pl.UInt64,
                  pl.Float32, pl.Float64)

# 2. Lọc các cột số
numeric_cols = [name for name, dtype in df.schema.items() if isinstance(dtype, numeric_types)]

# 3. Đếm null cho mỗi cột số
null_counts = df.select([
    (pl.col(col).is_null().sum().alias(col)) for col in numeric_cols
])

# 4. Chuyển từ wide -> long format để vẽ biểu đồ
null_melted = null_counts.unpivot(index=[], variable_name="column", value_name="null_count")

# 5. Sắp xếp giảm dần
null_melted = null_melted.sort("null_count", descending=True)

null_melted
```

Kiểm tra giá trị null

- df.schema.items(): Trả về danh sách tên cột và kiểu dữ liệu của từng cột trong DataFrame. Dùng để kiểm tra kiểu dữ liệu của từng cột.
- numeric_cols: Danh sách tên các cột có kiểu dữ liệu số, được lọc từ schema ở trên. Mục đích là chỉ xử lý các cột số.
- df.select([...]): Lấy ra một bảng chỉ chứa các cột được chọn và các thao tác áp dụng lên cột đó, dùng để tính tổng số giá trị null cho từng cột số.

- pl.col(col).is_null().sum(): Kiểm tra null trên một cột và tính tổng số giá trị null trong cột đó. Mục đích là đếm số lượng giá trị thiếu
- .alias(col): Đặt tên lại cho cột kết quả (giúp giữ nguyên tên cột gốc trong bảng kết quả).
- unpivot(...): Chuyển bảng từ dạng rộng (wide format) sang dạng dài (long format), mỗi dòng tương ứng với một cột trong bảng gốc để dễ trực quan hóa dữ liệu.
- sort("null_count", descending=True): Sắp xếp bảng theo số lượng giá trị null giảm dần. Mục đích để biết cột nào bị thiếu dữ liệu nhiều nhất.

5.3.3.2 Kiểm tra số lượng giá trị duy nhất (độ phân tán của dữ liệu): df.nunique()

```
# Kiểm tra số lượng giá trị duy nhất (nunique) cho tất cả các cột:
nunique_df = df.select([
    pl.col(col).n_unique().alias(f"{col}_nunique") for col in df.columns
])
nunique_df
```

shape: (1, 174)

user_id_nunique	school_nunique	course_id_nunique	user_enroll_time_nunique	user_past_course_count_nunique	user_time_since_last_course_nu
u32	u32	u32	u32	u32	u32
81941	2065	799	426	69	375

Kết quả kiểm tra số lượng giá trị duy nhất

- df.columns: Lấy danh sách tên tất cả các cột trong DataFrame.
- pl.col(col): Truy cập vào từng cột trong DataFrame theo tên.
- .n_unique(): Hàm đếm số lượng giá trị duy nhất (khác nhau) trong cột.
- df.select([...]): Trả về một DataFrame mới chỉ chứa các cột với số lượng giá trị duy nhất tương ứng.

5.3.3.3. Kiểm tra độ dài trung bình của chuỗi đối với các cột kiểu object (chuỗi văn bản): df.select_dtypes(include=['object']).apply(lambda x: x.str.len().mean())

```
#Tính độ dài trung bình của chuỗi cho các cột kiểu chuỗi (string):
# Lọc các cột kiểu string
string_cols = [name for name, dtype in df.schema.items() if dtype == pl.Utf8]

# Tính độ dài trung bình
avg_len_df = df.select([
    pl.col(col).str.len_chars().mean().alias(f"{col}_avg_len") for col in string_cols
])

avg_len_df
```

shape: (1, 7)

user_id_avg_len	school_avg_len	course_id_avg_len	user_enroll_time_avg_len	field_x_avg_len	start_date_avg_len	end_date_avg_len
f64	f64	f64	f64	f64	f64	f64
9.815672	6.912326	8.530858	10.0	2.874717	10.0	10.0

Kết quả kiểm tra độ dài trung bình chuỗi

- df.schema.items(): Trả về cặp (tên cột, kiểu dữ liệu) cho toàn bộ DataFrame.
- if dtype == pl.Utf8: Lọc các cột có kiểu dữ liệu là chuỗi (string trong Polars là Utf8).
- string_cols: Danh sách tên các cột kiểu chuỗi trong DataFrame.
- pl.col(col).str.len_chars(): Tính độ dài (số ký tự) của mỗi chuỗi trong cột.
- mean(): Tính trung bình độ dài chuỗi trong cột đó.
- alias(f'{col}_avg_len'): Đặt tên lại cho cột kết quả theo dạng <tên_cột>_avg_len để dễ hiểu.

5.3.3.4 Kiểm tra điều kiện dữ liệu có hợp lệ không theo quy tắc nghiệp vụ và duy tắc kinh doanh:

Các cột số không âm, cột user_id bắt đầu bằng “U_”, cột course_id bắt đầu bằng “C_”

```

#Lấy danh sách các cột số
numeric_cols = [name for name, dtype in df.schema.items() if isinstance(dtype, numeric_types)]

#Kiểm tra các giá trị số > 0 và định dạng user/course_id
# Điều kiện: tất cả cột số đều > 0
numeric_conditions = [pl.col(col) >= 0.0 for col in numeric_cols]

# Điều kiện: user_id bắt đầu bằng "U_" và course_id bắt đầu bằng "C_"
id_conditions = [
    pl.col("user_id").str.starts_with("U_"),
    pl.col("course_id").str.starts_with("C_")
]

# Gộp tất cả điều kiện lại bằng AND
all_conditions = reduce(operator.and_, numeric_conditions + id_conditions)

# Lọc dữ liệu hợp lệ
valid_df = df.filter(all_conditions)

# Lọc dữ liệu vi phạm
invalid_df = df.filter(~all_conditions)

print(f"Số dòng hợp lệ: {valid_df.shape[0]}")
print(f"Số dòng vi phạm: {invalid_df.shape[0]}")

```

Kiểm tra điều kiện dữ liệu hợp lệ

- pl.col("user_id").str.starts_with("U_"): Kiểm tra xem cột user_id có bắt đầu bằng "U_" không.
- pl.col("course_id").str.starts_with("C_"): Kiểm tra xem cột course_id có bắt đầu bằng "C_" không.
- id_conditions: Một danh sách gồm các điều kiện chuỗi (ID bắt đầu đúng định dạng).
- numeric_conditions: (được tạo ở phần trước, thường là kiểm tra các giá trị số không bị âm, nằm trong khoảng hợp lý...).
- reduce(operator.and_, ...): Gộp toàn bộ các điều kiện lại bằng toán tử AND, nghĩa là **bản ghi hợp lệ là bản ghi thỏa tất cả các điều kiện**.
- df.filter(all_conditions): Trả ra valid_df, là những dòng dữ liệu **hợp lệ**.
- df.filter(~all_conditions): Trả ra invalid_df, là những dòng dữ liệu **vi phạm** một trong các điều kiện.

Số dòng hợp lệ: 84138

Số dòng vi phạm: 103

Kết quả kiểm tra điều kiện dữ liệu hợp lệ

5.3.4. Phát hiện những giá trị bất thường và dữ liệu trùng lặp

5.3.4.1. Xác định outliers

Xác định outliers bằng IQR (Interquartile Range) đối với dữ liệu số:

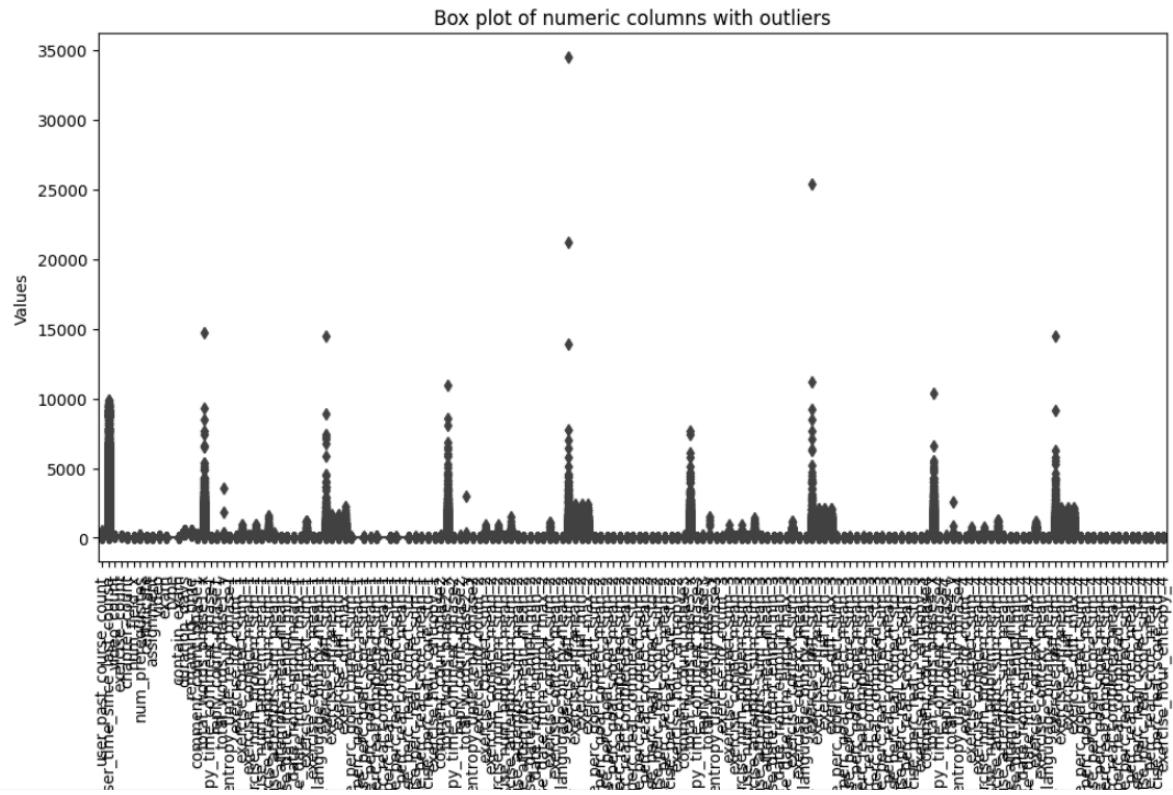
$$Q1 = \text{df.quantile}(0.25), Q3 = \text{df.quantile}(0.75), IQR = Q3 - Q1$$

```
# Lọc các cột có kiểu số
numeric_types = (pl.Int32, pl.Int64, pl.Float32, pl.Float64)
numeric_cols = [name for name, dtype in df.schema.items() if isinstance(dtype, numeric_types)]\n\n# Phát hiện outliers cho từng cột số
outliers = {}\\n\\nfor col in numeric_cols:
    q1 = df.select(pl.col(col)).quantile(0.25).to_numpy()[0]
    q3 = df.select(pl.col(col)).quantile(0.75).to_numpy()[0]
    iqr = q3 - q1\\n\\n    lower_bound = q1 - 1.5 * iqr
    upper_bound = q3 + 1.5 * iqr\\n\\n    # Lọc các outliers
    outliers_for_col = df.filter((pl.col(col) < lower_bound) | (pl.col(col) > upper_bound))
    outliers[col] = outliers_for_col\\n\\n# In kết quả
outliers
```

Xác định outlier

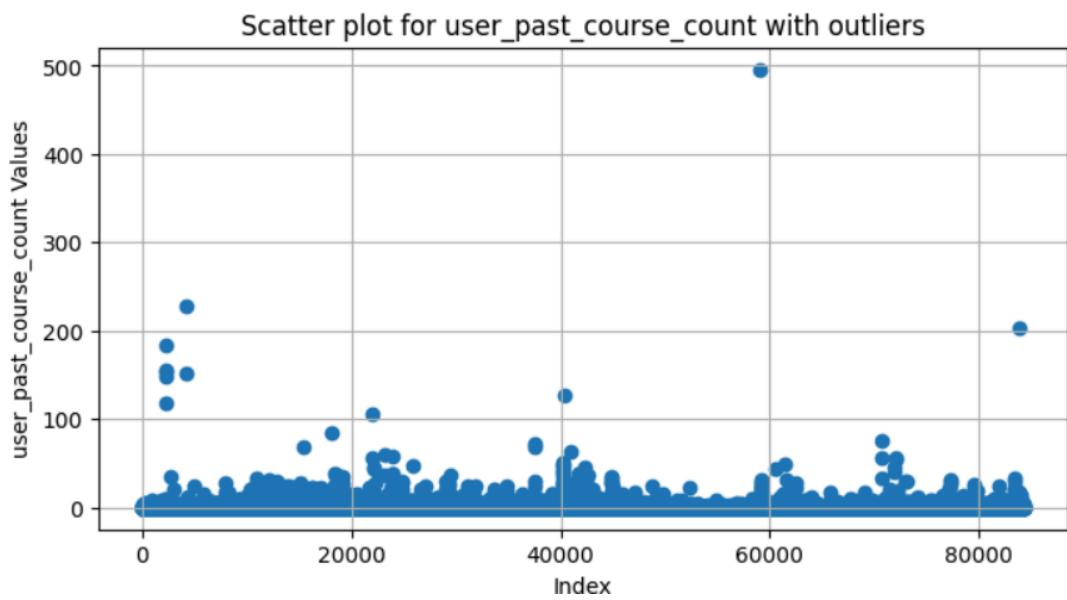
- $q1$ và $q3$: là quartile 1 (25%) và quartile 3 (75%) của cột hiện tại.
- $iqr = q3 - q1$: tính khoảng tứ phân vị (Interquartile Range).
- $lower_bound$, $upper_bound$: tính ngưỡng dưới và ngưỡng trên để xác định outliers. Theo quy tắc IQR, các giá trị nhỏ hơn $q1 - 1.5*iqr$ hoặc lớn hơn $q3 + 1.5*iqr$ là outliers.
- $\text{df.filter}(...)$: lọc ra các dòng có giá trị nằm ngoài khoảng $[lower_bound, upper_bound]$.
- $\text{outliers}[col] = \text{outliers}_\text{for_col}$: lưu các dòng chứa outliers của từng cột vào dictionary outliers, với key là tên cột.

Vẽ biểu đồ thể hiện giá trị outlier cho các cột:

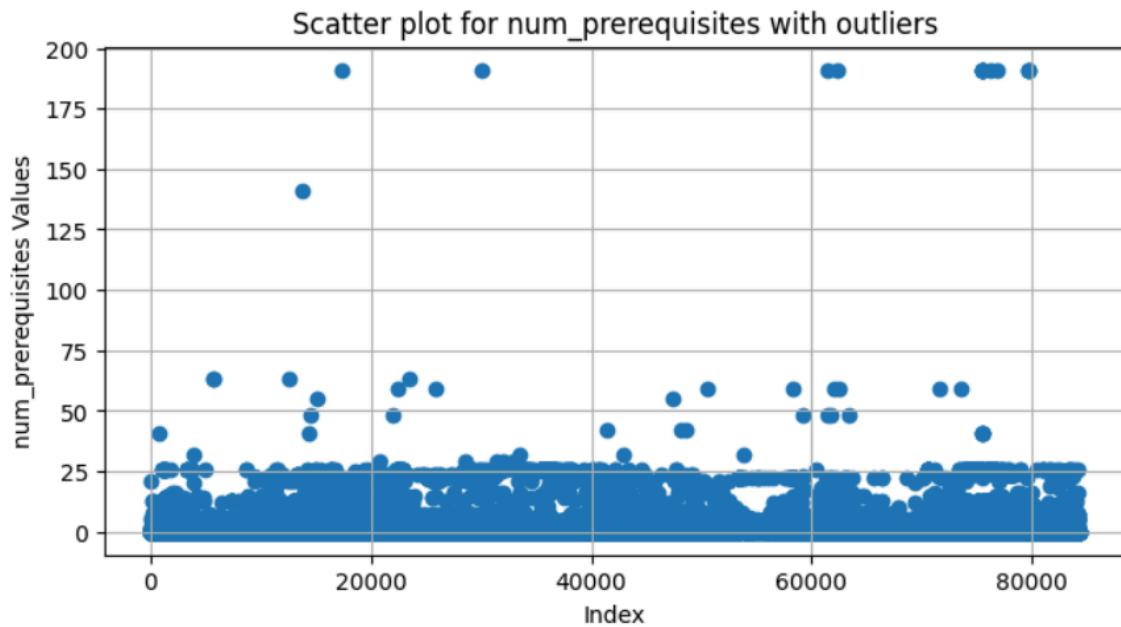


Kết quả kiểm tra outlier tất cả các cột

Vẽ biểu đồ thể hiện giá trị outlier cho từng cột:



Kết quả kiểm tra outlier cột user_past_course_count



Kết quả kiểm tra outlier cột num_prerequisites

5.3.4.2. Xác định dữ liệu bị trùng lặp

Kiểm tra dữ liệu trùng lặp: df.duplicated().sum()

```
# 1. Kiểm tra dữ liệu trùng lặp
# Kiểm tra số lượng dòng ban đầu
initial_count = df.height

# Loại bỏ các dòng trùng lặp
df_unique = df.unique()

# Kiểm tra số lượng dòng sau khi loại bỏ trùng lặp
unique_count = df_unique.height

# Số dòng bị trùng lặp
duplicate_count = initial_count - unique_count
print(f"Duplicated rows count: {duplicate_count}")
```

Duplicated rows count: 0

Kết quả kiểm tra dữ liệu trùng lặp

df.height: trả về số dòng của DataFrame.

df.unique(): loại bỏ các dòng trùng lặp hoàn toàn và trả về DataFrame mới chỉ chứa các dòng duy nhất.

df_unique.height: trả về số dòng sau khi đã loại bỏ trùng lặp.

`duplicate_count = initial_count - unique_count`: tính số dòng bị trùng lặp bằng cách lấy hiệu giữa số dòng ban đầu và số dòng duy nhất còn lại.

5.3.5. Kiểm tra tính nhất quán giữa nhiều nguồn dữ liệu

Kiểm tra khóa học có dữ liệu nhưng không có user đăng ký:

```
# Kiểm tra khóa học có dữ liệu nhưng không có user đăng ký
missing_users = df.filter(pl.col("user_id").is_null())
print(f"Missing users: {missing_users.shape[0]}")
```

Missing users: 0

Kết quả kiểm tra user bị null khi có thông tin về khóa học

5.3.6. Accuracy

- Độ chính xác: Dữ liệu phải phản ánh đúng thực tế, không có lỗi, sai sót hoặc thông tin sai lệch.
- Ý tưởng đo lường: Đo lường mức độ chính xác của dữ liệu so với giá trị thực tế hoặc nguồn tham chiếu.

Accuracy theo Object

- Đo lường độ chính xác của từng Object trong tập dữ liệu bằng cách so sánh với dữ liệu tham chiếu. Nó phản ánh mức độ dữ liệu của từng cá nhân hoặc thực thể đúng so với thực tế. Cho một object O có n thuộc tính, nếu x là giá trị của object trong dataset và x_i (ref) là giá trị tham chiếu. Khi một số thuộc tính quan trọng hơn những thuộc tính khác, ta có thể sử dụng trọng số w_i cho từng thuộc tính để điều chỉnh mức độ ảnh hưởng của chúng.

$$Accuracy(O) = \frac{\sum_{i=1}^n w_i \times 1(x_i = x_i^{(ref)})}{\sum_{i=1}^n w_i}$$

Trong đó:

- w_i là trọng số của thuộc tính i .
- $1(x_i = x_i^{(ref)})$ nhận giá trị 1 nếu đúng, 0 nếu sai.
- n là số lượng thuộc tính cần so sánh.
- Mẫu số $\sum_{i=1}^n w_i$ đảm bảo tổng trọng số chuẩn hóa về 1.

Accuracy toàn bộ dataset

Accuracy trung bình trên tất cả các objects với m là số lượng object.

$$Accuracy = \frac{\sum_{i=1}^m Accuracy(O_i)}{m}$$

Nghiên cứu về phương pháp đo lường Accuracy khi không có ground truth:
Trong lĩnh vực DQ, ta có:

- Accuracy (Độ chính xác): Dữ liệu phải phản ánh đúng thực tế, không có lỗi, sai sót hoặc thông tin sai lệch.
- Reliability (Độ Tin Cậy): Đo lường mức độ đúng đắn và ổn định của dữ liệu, đảm bảo rằng dữ liệu phản ánh đúng thực tế.
- Relevance (Độ Liên Quan): Đánh giá mức độ dữ liệu phù hợp với mục tiêu phân tích hoặc dự đoán.

Trong DQ, độ chính xác (Accuracy) thường được đo lường bằng cách so sánh dữ liệu với một nguồn tham chiếu đáng tin cậy (ground truth). Tuy nhiên, nếu không có dữ liệu tham chiếu, việc đo Accuracy trở nên không khả thi.

⇒ **Dùng Reliability và Relevance Để Đánh Giá Accuracy Gián Tiếp:**

- Nếu dữ liệu có độ tin cậy cao (Reliability), có khả năng nó cũng chính xác.
 - Dữ liệu ổn định giữa các lần đo lường → Ít có khả năng chứa lỗi ngẫu nhiên.
 - Nếu dữ liệu có sai số nhỏ giữa các lần huấn luyện mô hình, có thể đáng tin cậy.
- Dữ liệu phù hợp với mục tiêu dự đoán (Relevance) có thể có độ chính xác cao hơn.
 - Nếu các đặc trưng dữ liệu giải thích tốt kết quả dự đoán, có khả năng dữ liệu liên quan và có độ chính xác cao.
 - Nếu nhiều đặc trưng không quan trọng, dữ liệu có thể chứa thông tin nhiễu.
- Kết hợp 2 yếu tố:
 - Reliability cao + Relevance cao → Dữ liệu có khả năng chính xác.
 - Reliability thấp nhưng Relevance cao → Dữ liệu có thể có nhiễu.
 - Reliability cao nhưng Relevance thấp → Dữ liệu có thể không đủ thông tin quan trọng.

Vì bộ dữ liệu này không có groundtruth, cho nên nhóm đã quyết định đánh giá Accuracy của dữ liệu bằng cách gián tiếp qua Reliability và Relevance.

Sau đây là kết quả của 2 chỉ số trên:

5.3.6.1. Reliability:

Sử Dụng Accuracy và F1-score Để Đánh Giá Reliability

- Accuracy cao và F1-score cao → Dữ liệu có độ tin cậy tốt, mô hình hoạt động ổn định.
- Accuracy cao nhưng F1-score thấp → Dữ liệu có thể bị mất cân bằng, thiếu độ tin cậy trong một số lớp.

Kiểm Tra Độ Ôn Định (Consistency) của Reliability Bằng Cross-Validation: Dữ liệu có độ tin cậy cao nếu mô hình cho kết quả ổn định trên các tập kiểm tra khác nhau.

- Nếu độ lệch (variance) giữa các folds thấp, dữ liệu có độ tin cậy cao.
- Nếu kết quả biến động nhiều, dữ liệu có thể chứa lỗi hoặc không đầy đủ.

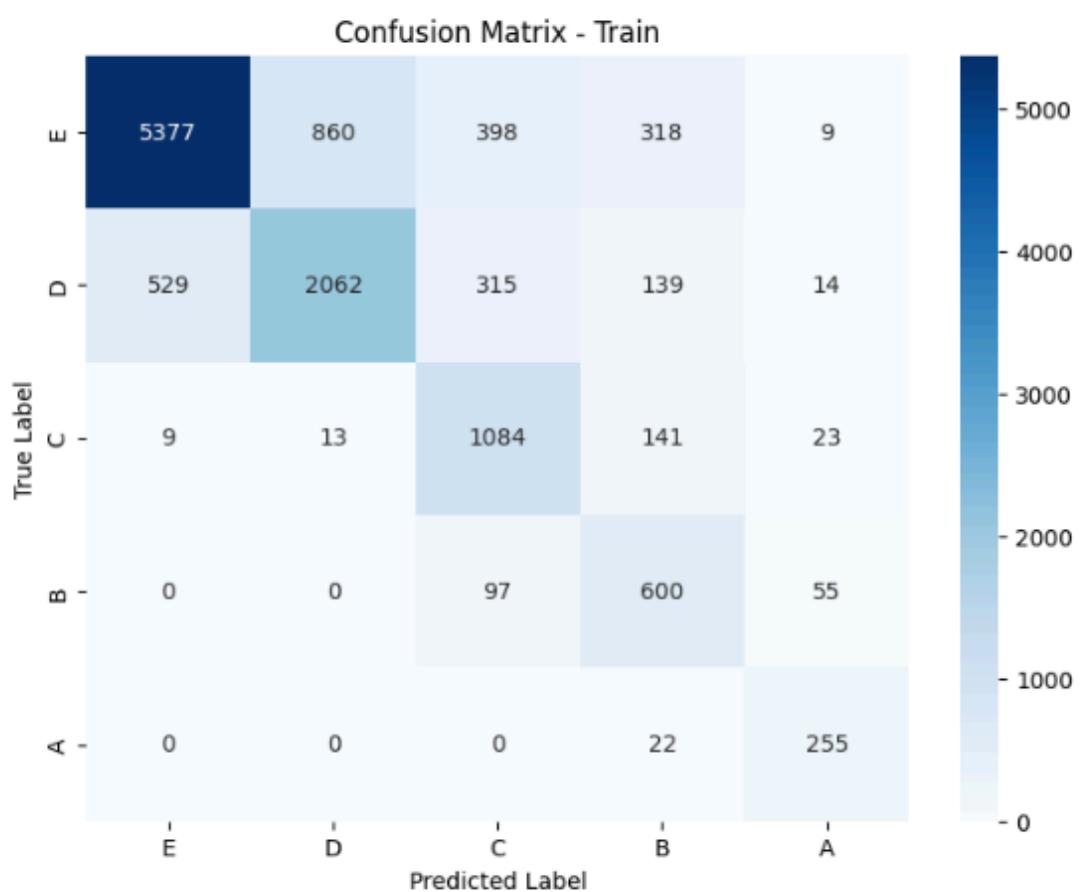
Sau đây là kết quả áp dụng các thuật toán để tính chỉ số Accuracy và F1-score.

Accuracy cho thuật toán Randomforest

Phase 1:

```
--- Train Metrics ---  
F1 Score (per class): [0.83513241 0.68802135 0.6852086 0.60851927 0.8056872 ]  
Precision (per class): [0.9090448 0.70255537 0.57233369 0.49180328 0.71629213]  
Recall (per class): [0.77233554 0.6740765 0.85354331 0.79787234 0.92057762]  
Accuracy: 0.7612  
AUC (One-vs-Rest): 0.9494
```

	precision	recall	f1-score	support
E	0.91	0.77	0.84	6962
D	0.70	0.67	0.69	3059
C	0.57	0.85	0.69	1270
B	0.49	0.80	0.61	752
A	0.72	0.92	0.81	277
accuracy			0.76	12320
macro avg	0.68	0.80	0.72	12320
weighted avg	0.79	0.76	0.77	12320

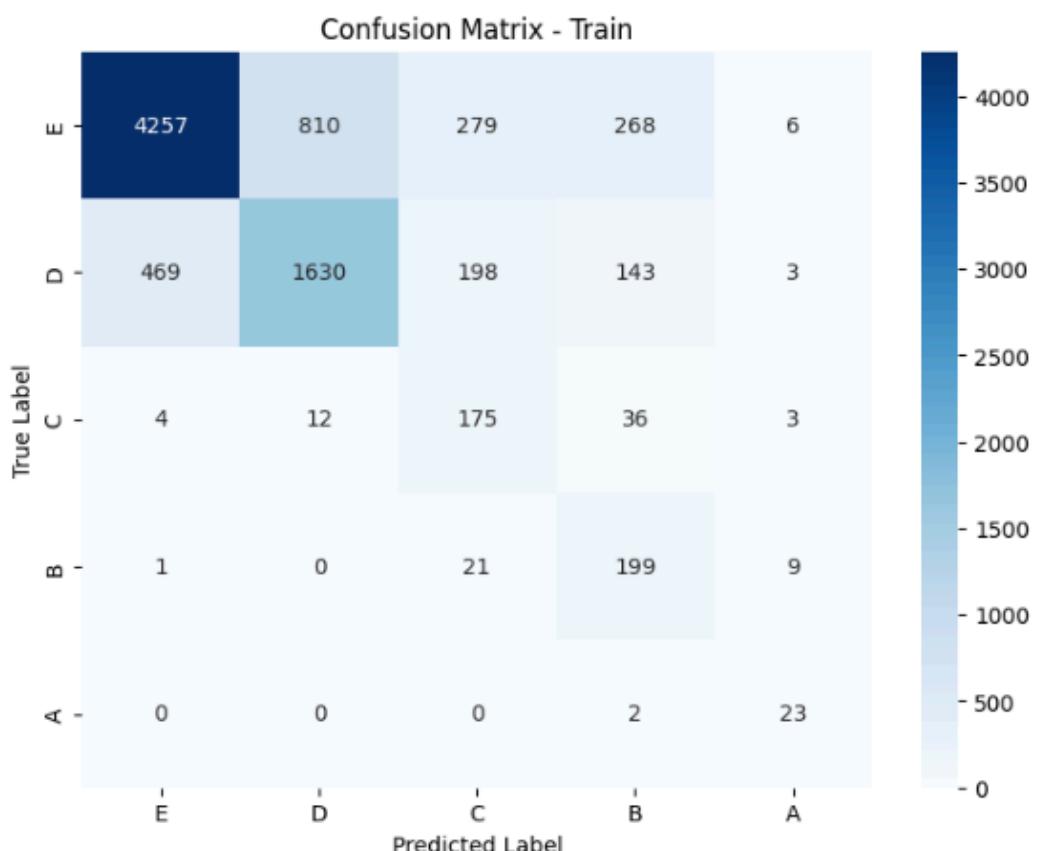


```
--- Test Metrics ---
F1 Score (per class): [0.92018072 0.36231884 0.          0.          0.          ]
Precision (per class): [0.85814607 0.56818182 0.          0.          0.          ]
Recall (per class): [0.99188312 0.26595745 0.          0.          0.          ]
Accuracy: 0.8413
AUC (One-vs-Rest): 0.8469
```

Phase 2:

```
--- Train Metrics ---
F1 Score (per class): [0.82252922 0.6659857 0.3875969 0.45330296 0.66666667]
Precision (per class): [0.89980977 0.66476346 0.26002972 0.30709877 0.52272727]
Recall (per class): [0.75747331 0.66721244 0.76086957 0.86521739 0.92      ]
Accuracy: 0.7351
AUC (One-vs-Rest): 0.9270
```

	precision	recall	f1-score	support
E	0.90	0.76	0.82	5620
D	0.66	0.67	0.67	2443
C	0.26	0.76	0.39	230
B	0.31	0.87	0.45	230
A	0.52	0.92	0.67	25
accuracy			0.74	8548
macro avg	0.53	0.79	0.60	8548
weighted avg	0.80	0.74	0.76	8548

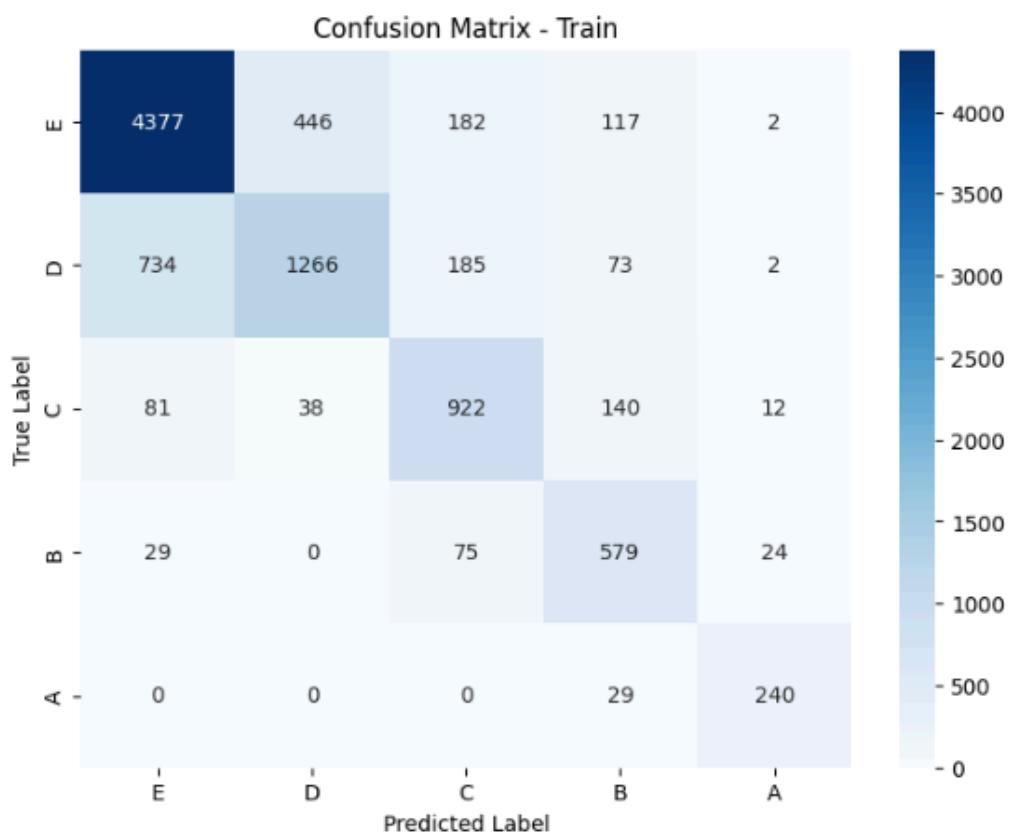


```
--- Test Metrics ---
F1 Score (per class): [0.90289103 0.06122449 0.          0.          0.          ]
Precision (per class): [0.82297297 0.5         0.          0.          0.          ]
Recall (per class): [1.          0.0326087 0.          0.          0.          ]
Accuracy: 0.8204
AUC (One-vs-Rest): 0.7475
```

Phase 3:

```
--- Train Metrics ---  
F1 Score (per class): [0.8462059 0.63142145 0.72115761 0.70395137 0.87431694]  
Precision (per class): [0.83834514 0.72342857 0.67595308 0.61727079 0.85714286]  
Recall (per class): [0.85421546 0.56017699 0.77284158 0.81895332 0.89219331]  
Accuracy: 0.7730  
AUC (One-vs-Rest): 0.9476
```

	precision	recall	f1-score	support
E	0.84	0.85	0.85	5124
D	0.72	0.56	0.63	2260
C	0.68	0.77	0.72	1193
B	0.62	0.82	0.70	707
A	0.86	0.89	0.87	269
accuracy			0.77	9553
macro avg	0.74	0.78	0.76	9553
weighted avg	0.78	0.77	0.77	9553



```

--- Test Metrics ---
F1 Score (per class): [0.89824024 0.          0.          0.          0.          ]
Precision (per class): [0.81641168 0.          0.          0.          0.          ]
Recall (per class): [0.99829932 0.          0.          0.          0.          ]
Accuracy: 0.8141
AUC (One-vs-Rest): 0.6318

```

Phase 4:

--- Train Metrics ---

F1 Score (per class): [0.82454784 0.69335055 0.76037588 0.75526316 0.9025641]

Precision (per class): [0.90040513 0.70844327 0.66827254 0.66666667 0.82242991]

Recall (per class): [0.76047904 0.67888748 0.88192552 0.87101669 1.]

Accuracy: 0.7803

AUC (One-vs-Rest): 0.9525

	precision	recall	f1-score	support
E	0.90	0.76	0.82	3507
D	0.71	0.68	0.69	1582
C	0.67	0.88	0.76	1101
B	0.67	0.87	0.76	659
A	0.82	1.00	0.90	264
accuracy			0.78	7113
macro avg	0.75	0.84	0.79	7113
weighted avg	0.80	0.78	0.78	7113

Confusion Matrix - Train



```

--- Test Metrics ---
F1 Score (per class): [0.88888889 0.0212766 0.          0.          0.          0.          ]
Precision (per class): [0.80825959 0.11111111 0.          0.          0.          0.          ]
Recall (per class): [0.98738739 0.01176471 0.          0.          0.          0.          ]
Accuracy: 0.7991
AUC (One-vs-Rest): 0.5553

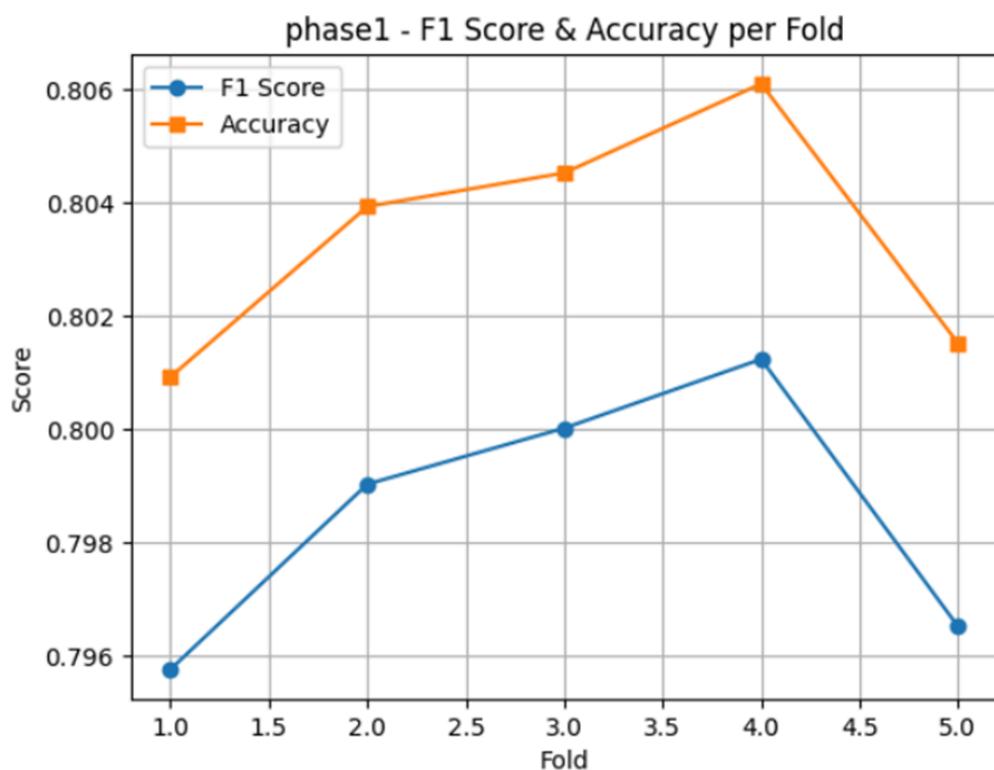
```

Nhận xét:

- Cả 4 phase đều cho kết quả khá cao, trong khoảng 0.79 - 0.84 cho thấy hiệu suất của mô hình khá tốt.

CatBoost - StandardScaler()

Phase1:



Nhận xét:

- Độ ổn định chỉ số qua các fold
 - F1 Score dao động từ khoảng 0.796 đến 0.801, và Accuracy dao động từ 0.801 đến 0.806.
 - Các đường biểu diễn khá ổn định, không có fold nào lệch quá mạnh so với trung bình.

- Fold 4 có điểm cao nhất, fold 1 và fold 5 có điểm thấp hơn, nhưng variance giữa các fold rất thấp.

2. Khoảng cách giữa F1 Score và Accuracy

- Chênh lệch giữa hai chỉ số này rất nhỏ, cho thấy dữ liệu không có mâu thuẫn trong lớp nghiêm trọng.
- Mô hình giữ được độ chính xác và độ cân bằng tốt trong các dự đoán.

3. Giá trị tuyệt đối của chỉ số

- F1 Score và Accuracy đều không quá cao so với các phase khác, nhưng vẫn ở mức khá (xấp xỉ 0.80).
- Điều này cho thấy dữ liệu có thể có mức độ khó cao hơn, hoặc mô hình chưa tối ưu hoàn toàn với phase này.

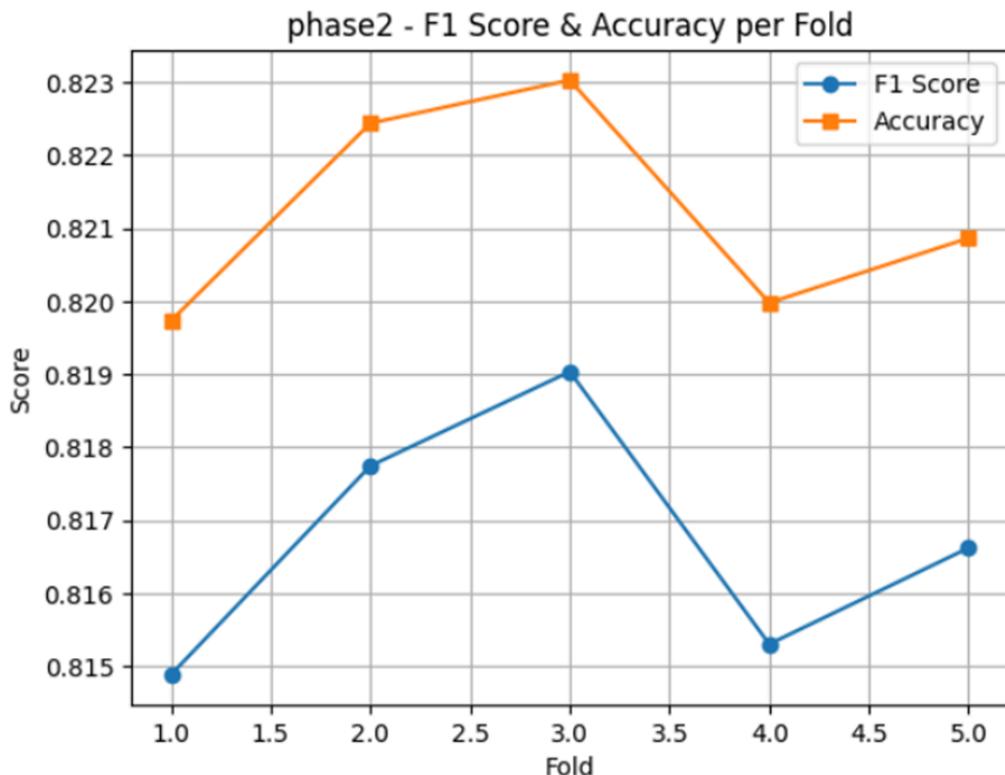
4. Tính nhất quán (Consistency/Reliability)

- Sự dao động giữa các fold rất nhỏ, mô hình cho ra kết quả ổn định trên các tập kiểm tra khác nhau.
- Không có dấu hiệu bất thường, dữ liệu đủ reliability cho các ứng dụng mô hình hóa thực tế.

Kết luận về reliability phase1:

Dữ liệu phase1 có reliability tốt, vì các chỉ số ổn định và variance thấp giữa các fold. Tuy nhiên, giá trị tuyệt đối của F1 Score và Accuracy thấp hơn một chút so với các phase còn lại (các phase 2, 3, 4 cao hơn).

Phase 2:



Nhận xét:

1. Độ ổn định của chỉ số qua các fold

- Cả F1 Score và Accuracy đều dao động nhẹ quanh mức $\sim 0.815\text{--}0.823$.
- Đường biểu diễn các chỉ số qua các fold rất “phẳng”, không có fold nào tụt mạnh hoặc nhảy vọt.
- Điều này cho thấy mô hình duy trì hiệu suất ổn định trên các tập kiểm tra khác nhau (variance thấp).

2. Trị số tuyệt đối của F1 và Accuracy

- Accuracy ổn định quanh mức $0.820\text{--}0.823$, F1 Score quanh mức $0.815\text{--}0.819$.
- Đây đều là các giá trị khá cao và sát nhau giữa các fold, không có sự chênh lệch lớn.
- Độ chênh giữa accuracy và F1-score không nhiều, chứng tỏ dữ liệu không bị mất cân bằng nghiêm trọng.

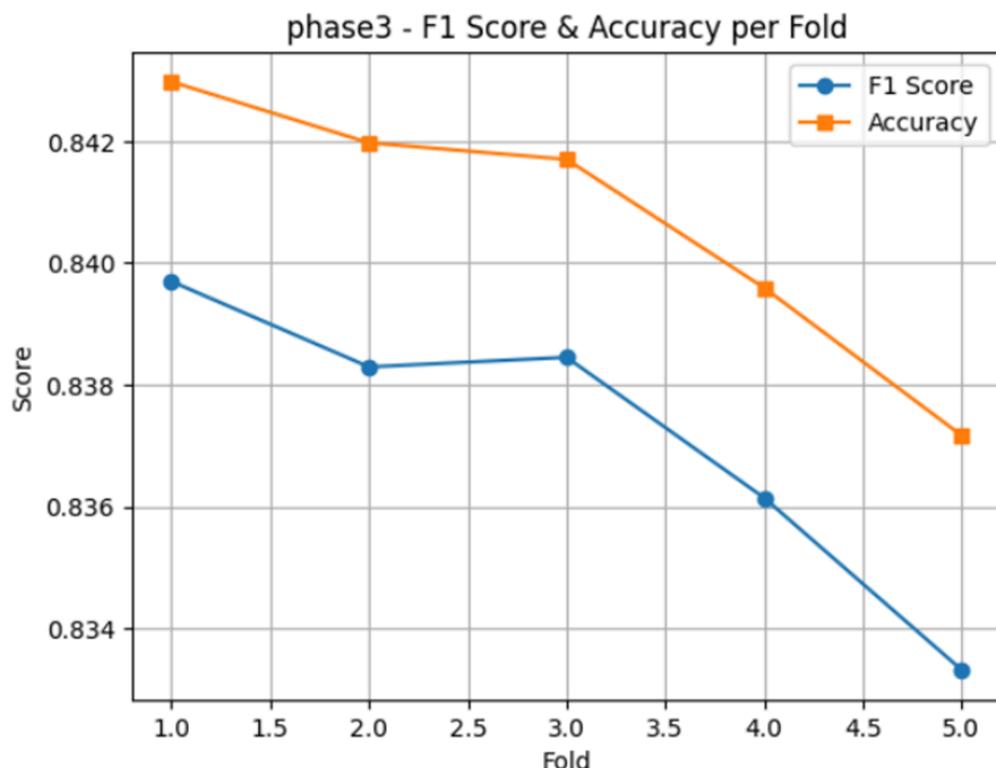
3. Reliability (Độ tin cậy)

- Độ lệch nhỏ giữa các fold, không xuất hiện dấu hiệu dữ liệu có vấn đề (ví dụ: class imbalance nghiêm trọng, lỗi trong dữ liệu...).
- Độ tin cậy của dữ liệu phase2 là tốt: mô hình học được các đặc trưng tổng quát, dữ liệu nhất quán và không bị lệ thuộc vào một fold cụ thể.

Kết luận về reliability phase2: Dữ liệu phase2 đạt reliability tốt.

Chỉ số F1-score và accuracy ổn định giữa các fold, variance thấp, không có hiện tượng outlier hay nhiễu loạn bất thường.

Phase 3:



Nhận xét:

Độ ổn định qua các fold (Consistency)

- **F1 Score** và **Accuracy** ở mức khá cao (0.833–0.843), nhưng có xu hướng giảm dần nhẹ từ fold 1 đến fold 5.
- Độ chênh lệch giữa các fold vẫn **không lớn** (khoảng 0.01), không có hiện tượng outlier mạnh hay sự sụt giảm đột ngột.

Đánh giá tổng thể

- **F1 Score** và **Accuracy** khá gần nhau, không có chênh lệch nhiều, chứng tỏ dữ liệu vẫn khá cân bằng.
- Biểu đồ cho thấy mô hình vẫn duy trì hiệu suất khá ổn định trên nhiều fold, tuy có xu hướng giảm dần.

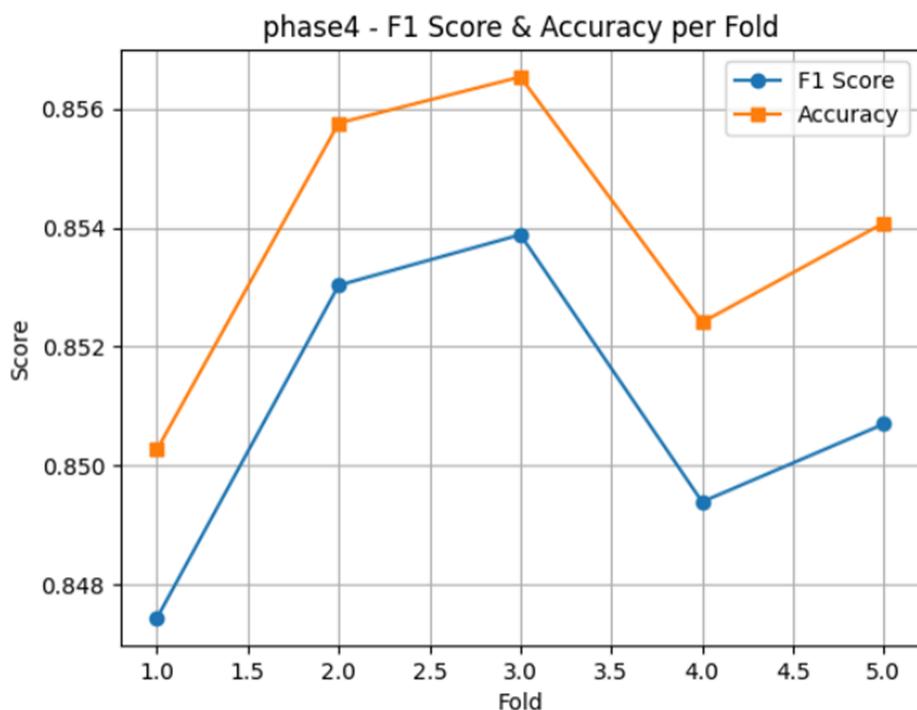
Reliability

- **Variance** của các chỉ số không quá cao, vẫn đảm bảo reliability tốt.
- Tuy nhiên, việc cả hai chỉ số đều giảm dần qua các fold có thể là dấu hiệu nhẹ về sự khác biệt nhất định giữa các phần nhỏ của dữ liệu (ví dụ: fold cuối chứa nhiều sample khó hơn, hoặc có thể là ngẫu nhiên).
- Không phát hiện dấu hiệu mất cân bằng nghiêm trọng hoặc lỗi lớn trong dữ liệu.

Kết luận

- **Dữ liệu phase3 có reliability tốt**, chỉ số ổn định và ở mức cao, phù hợp để train/test mô hình.
- Xu hướng giảm nhẹ qua các fold có thể kiểm tra thêm, nhưng hiện tại chưa ảnh hưởng đáng kể đến đánh giá độ tin cậy tổng thể.

Phase 4:



Nhận xét:

1. Độ ổn định các chỉ số qua từng fold
 - F1 Score và Accuracy đều duy trì ở mức cao:
 - F1 Score: dao động quanh 0.848 – 0.854.
 - Accuracy: dao động quanh 0.850 – 0.857.
 - Đường biểu diễn các chỉ số khá “mềm”, biến động nhẹ, không có fold nào bị outlier (tụt hoặc tăng bất thường).
2. So sánh giữa F1 và Accuracy
 - Cả hai chỉ số đều ổn định và gần nhau trên tất cả các fold, điều này cho thấy dữ liệu không có class bị mất cân bằng nghiêm trọng.
 - Không có sự chênh lệch lớn giữa F1 Score và Accuracy → Dữ liệu tốt, mô hình phân biệt các lớp đều.
3. Variance giữa các fold (tính nhất quán)
 - Variance thấp, các fold chỉ dao động nhẹ quanh trung bình, không fold nào bị lệch nhiều.
 - Sự lén xuống nhỏ này hoàn toàn nằm trong phạm vi bình thường của phân phối dữ liệu khi cross-validation.

Kết luận về reliability phase 4:

Dữ liệu phase4 có độ reliability cao.

Các chỉ số đánh giá đều cao và ổn định qua các fold → mô hình học tốt và tổng quát hóa tốt trên dữ liệu này.

5.3.6.2. Relevance

- Sử dụng Importance Feature

Phase 1

Feature Importances:	
remaining_time	0.185053
school	0.077967
total_words_phase1	0.076819
assignment	0.061977
video	0.051491
duration_days	0.046797
exercise_correct_sum_1	0.043589
exercise_count	0.041898
encoded_field_sum	0.041856
user_month	0.041328
video_count	0.038888
total_neutral1	0.037761
chapter_count	0.027658
user_time_since_last_course	0.027393
entropy_time_comment_phase1	0.027178
exercise_perc_real_score_mean_1	0.026989
total_positive1	0.021328
exercise_correct_mean_1	0.021093
user_past_course_count	0.016221
exercise_context_sum_1	0.014792
num_prerequisites	0.013899
exercise_num_problem_sum_1	0.012218
exercise_diff_mean_1	0.011847
exercise_id_count_1	0.009117
exercise_attempts_sum_mean_1	0.006892
exercise_hour_entropy_1	0.006508
exam	0.005746
certificate	0.005839
exercise_perc_real_completed_mean_1	0.004799
total_negative1	0.004272
exercise_language_binary_mean_1	0.001814
end_year	0.000367
video_percentage_watch_time_1	0.000185
video_time_between_views_std_1	0.000106
video_pause_count_1	0.000071
video_watched_percentage_1	0.000056
video_speed_avg_1	0.000057

Nhận xét:

Nhóm đặc trưng quan trọng nhất ($FI > 0.04$):

Feature Importance

remaining_time 0.1856

school 0.0779

total_words_phase1 0.0768

assignment 0.0519

Đây là các đặc trưng cốt lõi – mô hình dựa chủ yếu vào chúng để phân loại → Cần giữ lại, và thậm chí có thể khai thác thêm đặc trưng mở rộng từ các yếu tố này.

Phase 2:

Feature Importances:	
remaining_time	0.215813
school	0.083155
total_words_phase1	0.076926
exercise_count	0.042856
encoded_field_sum	0.042558
user_month	0.037883
video_count	0.037556
duration_days	0.037465
total_words_phase2	0.036843
total_neutral1	0.035842
video	0.035361
assignment	0.035015
user_time_since_last_course	0.030773
chapter_count	0.024514
entropy_time_comment_phase1	0.021779
total_positive1	0.019285
total_neutral2	0.019167
user_past_course_count	0.015081
num_prerequisites	0.013701
exercise_correct_sum_1	0.011247
entropy_time_comment_phase2	0.011091
exercise_perc_real_score_mean_1	0.008049
total_positive2	0.007631
exam	0.007316
exercise_correct_sum_2	0.007148
exercise_context_sum_1	0.006998
exercise_correct_mean_1	0.006743
certificate	0.006565
exercise_correct_mean_2	0.006061
exercise_perc_real_score_mean_2	0.005926
exercise_num_problem_sum_1	0.005878
exercise_id_count_1	0.005005
exercise_hour_entropy_1	0.004595
exercise_diff_mean_1	0.004573
exercise_diff_mean_2	0.004533
total_negative1	0.004484
exercise_num_problem_sum_2	0.003553
exercise_context_sum_2	0.003293
exercise_attempts_sum_mean_1	0.003023
exercise_id_count_2	0.002763
exercise_hour_entropy_2	0.002203
exercise_attempts_sum_mean_2	0.002167
total_negative2	0.002096
exercise_perc_real_completed_mean_2	0.001988
exercise_perc_real_completed_mean_1	0.001936
exercise_language_binary_mean_1	0.000499
exercise_language_binary_mean_2	0.000437
end_year	0.000187
video_percentage_watch_time_1	0.000127
video_watched_percentage_1	0.000119
video_pause_count_1	0.000107
video_time_between_views_std_1	0.000104

Nhóm đặc trưng quan trọng nhất (FI > 0.04):

Feature	Importance
remaining_time	0.2158

school 0.083155
total_words_phase1 0.0769
exercise_count 0.0428

Đây là các đặc trưng cốt lõi – mô hình dựa chủ yếu vào chúng để phân loại → Cân giữ lại, và thậm chí có thể khai thác thêm đặc trưng mở rộng từ các yếu tố này.

Phase 3:

Feature Importances:	
remaining_time	0.158166
school	0.070485
video	0.052388
encoded_field_sum	0.051347
video_count	0.049482
duration_days	0.049357
exercise_count	0.048428
assignment	0.045362
total_words_phase1	0.043889
user_month	0.037320
chapter_count	0.034064
total_neutral1	0.026291
total_words_phase3	0.026122
total_words_phase2	0.024283
user_time_since_last_course	0.021613
total_neutrals3	0.018311
num_prerequisites	0.016878
entropy_time_comment_phase1	0.015765
total_positive1	0.015646
total_neutral2	0.015616
user_past_course_count	0.013932
entropy_time_comment_phase3	0.009678
entropy_time_comment_phase2	0.009216
exercise_correct_sum_1	0.008596
certificate	0.008230
exam	0.007473
total_positive2	0.006839
exercise_correct_sum_3	0.006789
total_positive3	0.006024
exercise_context_sum_1	0.005945
exercise_correct_sum_2	0.005766
exercise_perc_real_score_mean_1	0.005198
exercise_num_problem_sum_1	0.005118
exercise_num_problem_sum_3	0.004967
exercise_id_count_1	0.004670
exercise_context_sum_3	0.004630
exercise_correct_mean_1	0.004499
exercise_correct_mean_3	0.004185
exercise_perc_real_score_mean_2	0.004185
exercise_perc_real_score_mean_3	0.003841
exercise_id_count_3	0.003811
exercise_correct_mean_2	0.003711
exercise_hour_entropy_1	0.003380
exercise_num_problem_sum_2	0.003160
exercise_diff_mean_3	0.003029
exercise_context_sum_2	0.002915
exercise_id_count_2	0.002853
exercise_attempts_sum_mean_3	0.002736
total_negative1	0.002788
total_negative2	0.002408
exercise_attempts_sum_mean_1	0.002244
exercise_hour_entropy_3	0.002107
exercise_diff_mean_2	0.002061

Nhận xét:

Nhóm đặc trưng quan trọng nhất ($FI > 0.04$):

Feature	Importance
remaining_time	0.158166
school	0.070405
video	0.0523
encoded_field_sum	0.0513

Đây là các đặc trưng cốt lõi – mô hình dựa chủ yếu vào chúng để phân loại → Cần giữ lại, và thậm chí có thể khai thác thêm đặc trưng mở rộng từ các yếu tố này.

Phase 4:

```

Feature importances:
remaining_time          0.083049
school                  0.058783
assignment               0.054043
duration_days            0.052171
video_count                0.051823
encoded_field_sum        0.049183
video                   0.047412
exercise_count             0.046882
chapter_count              0.038888
user_month                 0.032893
total_words_phase1       0.038324
exercise_correct_mean_4    0.026495
total_words_phase4       0.019598
user_time_since_last_course 0.017611
total_neutral1              0.016568
exercise_context_mean_4    0.016025
num_prerequisites           0.015772
total_words_phase2       0.015418
total_words_phase3       0.015254
exercise_num_problem_mean_4 0.014765
exercise_correct_mean_4    0.014369
exercise_perc_real_score_mean_4 0.014212
total_neutral4              0.013824
entropy_time_comment_phase1 0.012476
total_positive1             0.011513
total_neutral3              0.010755
exercise_correct_mean_1    0.009868
user_past_course_count      0.009714
exercise_id_count_4         0.009711
entropy_time_comment_phase4 0.009091
exercise_diff_mean_4         0.008477
exercise_attempts_mean_mean_4 0.008480
certificate                 0.008183
exercise_context_mean_1    0.008145
total_neutral2              0.007428
exercise_perc_real_completed_mean_4 0.006592
exercise_hour_entropy_4     0.006439
exercise_num_problem_mean_1 0.006012
exercise_perc_real_score_mean_1 0.005626
exercise_correct_mean_3     0.005596
total_positive4             0.005542
exercise_correct_mean_2     0.005418
exercise_correct_mean_3     0.005287
exercise_context_mean_3     0.005265
exercise_perc_real_score_mean_2 0.005182
exercise_correct_mean_2     0.005179
exam                      0.004919
total_positive2             0.004818
entropy_time_comment_phase3 0.004719
exercise_hour_entropy_3     0.004478
exercise_id_count_1         0.004182
exercise_num_problem_mean_3 0.004054
entropy_time_comment_phase2 0.004032
exercise_perc_real_score_mean_3 0.003753
total_positive3             0.003729
exercise_id_count_3         0.003397
exercise_correct_mean_3     0.003373
exercise_diff_mean_1         0.003178
exercise_language_binary_mean_4 0.002773
exercise_contact_mean_2     0.002559
exercise_num_problem_mean_2 0.002495
exercise_diff_mean_2         0.002279
exercise_diff_mean_3         0.002252
exercise_attempts_mean_mean_1 0.002142
exercise_attempts_mean_mean_3 0.002042
exercise_id_count_2         0.001918
exercise_perc_real_completed_mean_1 0.001887
exercise_hour_entropy_3     0.001737
exercise_perc_real_completed_mean_3 0.001556
exercise_perc_real_completed_mean_2 0.001247
exercise_attempts_mean_mean_2 0.001221
total_negative2             0.001128
exercise_hour_entropy_2     0.001068
total_negative4             0.000958
total_negative1             0.000824
total_negative3             0.0008595
exercise_language_binary_mean_3 0.0008184
exercise_language_binary_mean_1 0.0008183
end_year                    0.000753
video_speed_avg_3           0.000000

```

Nhận xét:

Nhóm đặc trưng quan trọng nhất ($FI > 0.04$):

Feature Importance

remaining_time 0.0830

school 0.0587

assignment 0.0548
duration_days 0.0521

Đây là các đặc trưng cốt lõi – mô hình dựa chủ yếu vào chúng để phân loại → Cần giữ lại, và thậm chí có thể khai thác thêm đặc trưng mở rộng từ các yếu tố này

- Sử dụng AUC

Phase 1:

Test AUC (macro-average, OVR): 0.8476

```
📊 AUC-ROC per feature (Relevance):  
total_score 1.000000  
exercise_perc_real_score_sum_1 0.608345  
exercise_correct_sum_1 0.608138  
exercise_perc_real_score_mean_1 0.602467  
total_words_phase1 0.599964  
...  
start_year 0.478836  
exercise_perc_real_score_std_1 0.470034  
exercise_perc_real_correct_std_1 0.469191  
exercise_diff_min_1 0.464494  
exercise_context_mean_1 0.456278
```

Phase 2:

Test AUC (macro-average, OVR): 0.8068

```
📊 AUC-ROC per feature (Relevance):  
total_score 1.000000  
exercise_perc_real_score_sum_1 0.608345  
exercise_correct_sum_1 0.608138  
exercise_perc_real_score_mean_1 0.602467  
total_words_phase1 0.599964  
...  
start_year 0.478836  
exercise_perc_real_score_std_1 0.470034  
exercise_perc_real_correct_std_1 0.469191  
exercise_diff_min_1 0.464494  
exercise_context_mean_1 0.456278
```

Phase 3:

Test AUC (macro-average, OVR): 0.8154

AUC-ROC per feature (Relevance):	
total_score	1.000000
exercise_perc_real_score_sum_1	0.604195
exercise_correct_sum_1	0.603320
assignment	0.602370
video	0.599462
	...
exercise_diff_min_1	0.483087
start_year	0.481111
exercise_perc_real_correct_std_1	0.467596
exercise_context_mean_1	0.459294
exercise_perc_real_score_std_1	0.458247

Phase 4:

Test AUC (macro-average, OVR): 0.7799

AUC-ROC per feature (Relevance):	
total_score	1.000000
assignment	0.607792
video	0.606986
exercise_perc_real_score_sum_1	0.597216
total_words_phase1	0.596465
	...
exercise_diff_min_1	0.485846
start_year	0.482234
exercise_perc_real_score_std_1	0.463257
exercise_perc_real_correct_std_1	0.461394
exercise_context_mean_1	0.461238

Độ tin cậy tổng quan

- AUC là thước đo tổng quát, phản ánh khả năng phân biệt các lớp của mô hình trên dữ liệu test.
- AUC > 0.8 thường được xem là tốt, > 0.85 là rất tốt, còn dưới 0.8 thì là khá, nhưng vẫn chấp nhận được nếu không thấp quá nhiều.

- Các phase đều có AUC trên 0.77 – không có phase nào bị cực thấp hoặc chênh lệch nghiêm trọng so với phần còn lại.

Đánh giá cụ thể từng phase

- Phase 1 có reliability tốt nhất, AUC cao nhất (**0.8476**), dữ liệu này giúp mô hình dự đoán ổn định và có khả năng tổng quát hóa tốt nhất trong các phase.
- Phase 2 và Phase 3 có AUC khá tốt (**0.8068** và **0.8154**), reliability của dữ liệu ở mức tốt, chỉ số gần nhau, không có dấu hiệu bị lỗi nghiêm trọng hay mất cân bằng lớn.
- Phase 4 có reliability thấp nhất (**0.7799**), tuy nhiên vẫn trên mức trung bình, mô hình vẫn có khả năng phân biệt các lớp, chỉ là thấp hơn các phase còn lại. Cần kiểm tra lại phase này nếu muốn tăng reliability (có thể do dữ liệu ít, nhiễu, mất cân bằng hoặc class khó phân biệt hơn).

Sự chênh lệch giữa các phase

- Sự chênh lệch AUC giữa phase cao nhất (1) và thấp nhất (4) là khoảng **0.0677**, không phải là cực lớn nhưng cũng cần chú ý nếu bạn muốn tính nhât quán cao giữa các phase.
- Tuy vậy, **tất cả các phase đều có AUC cao hơn nhiều so với mức random (0.5)**, nên reliability tổng thể của bộ dữ liệu là **ổn định và đủ tin cậy**.

Kết luận tổng quan

- Phase 1 có độ reliability cao nhất, phase 2 và 3 khá ổn định, phase 4 thấp nhất nhưng vẫn ổn.
- Tất cả các phase đều đủ tin cậy để sử dụng cho train/test mô hình thực tế, tuy nhiên nên kiểm tra kỹ hơn về đặc trưng dữ liệu ở phase 4 nếu muốn tối ưu hiệu suất.

Kết luận: Reliability cao + Relevance cao → Dữ liệu có khả năng chính xác.

5.3.7 Completeness

- **Khái niệm:** Tính đầy đủ là mức độ mà tất cả các dữ liệu cần thiết và có liên quan đã được thu thập và lưu trữ đầy đủ trong một tập dữ liệu. Tính đầy đủ không yêu cầu rằng 100% tất cả các trường dữ liệu phải đầy đủ, mà chủ yếu tập trung vào việc đảm bảo rằng những trường dữ liệu quan trọng và có ý nghĩa cho mục đích sử dụng của dữ liệu là đầy đủ và chính xác.

5.3.7.1. Completeness theo Object

- **Completeness theo Object** là một khái niệm trong quản lý dữ liệu và chất lượng dữ liệu, đề cập đến mức độ mà một đối tượng dữ liệu (object) có đầy đủ tất cả các thuộc tính cần thiết để phục vụ một mục đích cụ thể
- Tính toán tỷ lệ giữa số lượng phần tử hoàn chỉnh và tổng số phần tử trong một tập dữ liệu. Công thức tính completeness được biểu diễn như sau:

$$\text{Completeness}(O) = \frac{\text{Số lượng thuộc tính không bị thiếu}}{\text{Tổng số thuộc tính}}$$

```
# Tính completeness tổng thể
completeness_value = df.notnull().sum()
print(completeness_value)

user_id                      108810
school                       46493
course_id                     108810
user_enroll_time              108810
user_past_course_count        108810
...
exercise_perc_real_score_sum_1 108810
exercise_perc_real_score_mean_1 108810
exercise_perc_real_score_std_1 108810
exercise_hour_entropy_1       108810
row_completeness               108810
Length: 61, dtype: int64
```

Hình 2.1: Kiểm tra giá trị không null bằng hàm notnull()

```
# Tính completeness cho mỗi hàng
row_completeness = df.notnull().mean(axis=1)

# Gắn kết quả vào DataFrame nếu muốn
df['row_completeness'] = (row_completeness * 100).round(2)

# In 10 hàng đầu để xem kết quả
print(df[['row_completeness']].head(100))

row_completeness
0           98.36
1          100.00
2          100.00
3          100.00
4          100.00
...
95         98.36
96         100.00
97         98.36
98         100.00
99         100.00
```

Hình 2.2.: Giá trị completeness cho từng dòng dữ liệu

5.3.7.2. Completeness toàn bộ Dataset

- Công thức tính:

$$\text{Completeness} = \frac{\sum_{i=1}^m \text{Completeness}(O_i)}{m}$$

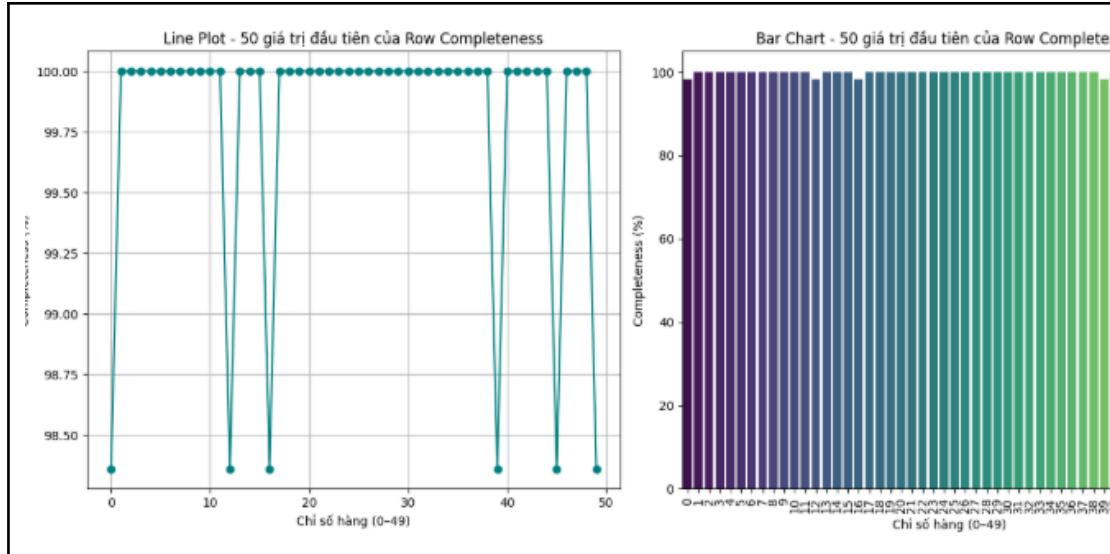
- Trong đó:
 - + m: tổng thuộc tính trong bộ dataset

```
completeness_total = df['row_completeness'].mean()
print(f"Giá trị completeness trung bình theo hàng là: {average_row_compl}

Giá trị completeness trung bình theo hàng là: 99.06%
```

Hình 2.3. Giá trị completeness cho toàn bộ dữ liệu

- Đánh giá sự phân bố của dữ liệu để xác định các giá trị bị thiếu một cách có hệ thống. Sử dụng biểu đồ histogram để thống kê độ completeness của từng hàng dữ liệu.



Hình 2.4: Biểu đồ histogram biểu thị giá trị completeness cho từng hàng

- Nhận xét:

- + Mức completeness nhìn chung khá cao, đôi khi có những hàng do thiếu giá trị ở trường school nên giá trị completeness giảm xuống còn 98.36%

5.3.8. Consistency

5.3.8.1 **Tính nhất quán:** Dữ liệu phải đồng nhất giữa các nguồn và hệ thống khác nhau, không có xung đột hoặc trùng lặp.

5.3.8.2 **Ý tưởng đo lường:** Được tính bằng tỷ lệ giữa số lượng điều kiện hợp lệ và tổng số điều kiện cần kiểm tra.

5.3.8.3 Sử dụng bộ dữ liệu để giao diện 2 từ người dùng đã học 1 tháng của khóa học để đánh giá độ Consistency. Sẽ bổ sung thêm đầy đủ vào báo cáo đồ án.

5.3.8.4 Consistency theo Object

a. Miền giá trị (Domain Range)

Dựa vào metadata ta có các ràng buộc của các đặc trưng.

- Các đặc trưng course_id và user_id có định dạng string và bắt đầu bằng “C_xxx” và “U_xxx” (tương tự).

```
# 1. Kiểm tra định dạng user_id, course_id
if not isinstance(row['user_id'], str) or not row['user_id'].startswith("U_"):
    return 0
if not isinstance(row['course_id'], str) or not row['course_id'].startswith("C_"):
    return 0
```

- Tên trường học (school) có dạng string và có độ dài từ 3-100 ký tự

```
# 2. Trường học: string, 3-100 ký tự
if not isinstance(row['school'], str) or not (3 <= len(row['school']) <= 100):
    return 0, 'school không hợp lệ'
```

- Các năm liên quan đến user và khóa học trong thời gian khảo sát (từ năm 2019 đến năm 2021).

```
# 3. các năm: phải nằm trong khoảng năm 2019-2021
if not (2019 <= row['end_year'] <= 2021) or not (2019 <= row['start_year'] <= 2021) or not (2019 <= row['year'] <= 2021):
    return 0, 'năm nằm ngoài khoảng 2019-2021'
```

- Số lượng khóa học mà học viên đã đăng kí trước đây (user_past_course_count) phải nhỏ hơn 1000 khóa học.

```
# 4. user_past_course_count: 0-1000
if not (0 <= row['user_past_course_count'] <= 1000):
    return 0
```

- Thời gian đăng kí khóa học đó kể từ khóa học gần nhất được đăng kí (user_time_since_last_course) không được lớn hơn 3 năm (trong khoảng thời gian khảo sát từ 2019 đến 2021).

```
# 5. user_time_since_last_course: 0-1095
if not (0 <= row['user_time_since_last_course'] <= 1095):
    return 0
```

- Số lượng video (video_count) không được nhỏ hơn 1.

```
# 6. video_count: >=1
if not (1 <= row['video_count']):
    return 0, 'video_count ít hơn 1'
```

- Số lượng bài tập của khóa học không được nhỏ hơn 1.

```
# 7. exercise_count: >= 1
if not (1 <= row['exercise_count']):
    return 0, 'exercise_count ít hơn 1'
```

- Số lượng chương không được nhỏ hơn 1 và lớn hơn 60.

```
# 8. chapter_count: >= 1
if not (1 <= row['chapter_count']):
    return 0, 'chapter_count ít hơn 1'
```

- Kiểm tra encoded_field_sum không được âm.

```
# 9. encoded_field_sum không âm
if row['encoded_field_sum'] < 0:
    return 0, 'encoded_field_sum có giá trị âm'
```

- Số lượng môn học tiên quyết không được là số âm và lớn hơn 50.

```
# 11. num_prerequisites: 0-50
if not (0 <= row['num_prerequisites'] <= 50):
    return 0, 'num_prerequisites ngoài khoảng 0-50'
```

- Các thành phần điểm assignment, video, exam trong khoảng 0 đến 100.

```
# 12. assignment, video, exam: float 0-100
for score_col in ['assignment', 'video', 'exam']:
    if not (0.0 <= row[score_col] <= 100.0):
        return 0
```

- Các trường liên quan đến tháng phải từ 1 đến 12.

```
# 13. start_month, end_month, user_month: tháng hợp lệ 1-12
for date_col in ['start_month', 'end_month', 'user_month']:
    if not (1 <= row[date_col] <= 12):
        return 0, f'{date_col} có tháng nằm ngoài 1-12'
```

- Thời gian của khóa học không được ngắn hơn 1 tuần và dài hơn 1 năm.

```
# 14. duration_days: 7 - 365
if not (7 <= row['duration_days'] <= 365):
    return 0
```

- Kiểm tra các chỉ số liên quan đến bình luận (Comment) và trả lời câu hỏi (Reply) phải không âm:

```

# Sentiment features
for col in [f'total_words_{phase}', f'total_positive_{phase}', f'total_negative_{phase}', f'total_neutral_{phase}']:
    if row[col] < 0:
        return 0, f'{col} có giá trị âm'

```

- Thống kê cơ bản về bài tập:

exercise_id_count_{phase}: số ID bài tập khác nhau được làm, từ 0–50

exercise_correct_sum_{phase}: tổng số bài làm đúng, từ 0–50,

exercise_correct_mean_{phase}: tỉ lệ làm đúng trung bình, từ 0.0–1.0

Thống kê số lượng câu hỏi:

exercise_num_problem_sum_{phase} và

exercise_num_problem_mean_{phase}: tổng và trung bình số câu hỏi
trong các bài tập, yêu cầu ≥ 0

Thống kê số lần làm bài:

exercise_attempts_sum_sum_{phase},

exercise_attempts_sum_mean_{phase},

exercise_attempts_mean_mean_{phase}: thống kê số lần làm bài dưới
nhiều hình thức, yêu cầu ≥ 0

Thời gian làm bài so với thời điểm ghi

danh: exercise_date_from_enroll_min_{phase}, mean, max: khoảng cách
thời gian (tính bằng ngày) từ lúc ghi danh đến khi làm bài, yêu cầu ≥ 0

Ngữ cảnh : exercise_context_sum_{phase} và mean: thống kê về mức độ
ngữ cảnh học tập, yêu cầu ≥ 0

Thời gian làm exercise: exercise_diff_sum_{phase}, mean, min, max: độ
khó bài tập, cũng phải ≥ 0 , exercise_language_binary_mean_{phase}: giá
trị nhị phân (0 hoặc 1), thể hiện bài tập có liên quan đến ngôn ngữ lập trình
hay không

Mức độ hoàn thành mục tiêu: Lặp qua các loại thống kê gồm: correct,
score, completed: kiểm tra tỉ lệ hoàn thành mục tiêu (perc_goal_* và
perc_real_*), với thống kê sum và mean: phải nằm trong 0.0–100.0, kiểm
tra độ lệch chuẩn (std) của perc_real_*: phải ≥ 0

Mức độ phân bố thời gian làm bài: exercise_hour_entropy_{phase}:
entropy thời gian làm bài theo giờ, phải nằm trong 0.0–1.0

```

# Exercise
if not (0 <= row[f'exercise_id_count_{phase}'] <= 50): return 0
if not (0 <= row[f'exercise_correct_sum_{phase}'] <= 50): return 0
if not (0.0 <= row[f'exercise_correct_mean_{phase}'] <= 1.0): return 0
if row[f'exercise_num_problem_sum_{phase}'] < 0: return 0
if row[f'exercise_num_problem_mean_{phase}'] < 0: return 0
if row[f'exercise_attempts_sum_sum_{phase}'] < 0: return 0
if row[f'exercise_attempts_sum_mean_{phase}'] < 0: return 0
if row[f'exercise_attempts_mean_mean_{phase}'] < 0: return 0
if row[f'exercise_date_from_enroll_min_{phase}'] < 0: return 0
if row[f'exercise_date_from_enroll_mean_{phase}'] < 0: return 0
if row[f'exercise_date_from_enroll_max_{phase}'] < 0: return 0
if row[f'exercise_context_sum_{phase}'] < 0: return 0
if row[f'exercise_context_mean_{phase}'] < 0: return 0
if not (row[f'exercise_language_binary_mean_{phase}'] in [0, 1]): return 0
if row[f'exercise_diff_sum_{phase}'] < 0: return 0
if row[f'exercise_diff_mean_{phase}'] < 0: return 0
if row[f'exercise_diff_min_{phase}'] < 0: return 0
if row[f'exercise_diff_max_{phase}'] < 0: return 0
for col_suffix in ['correct', 'score', 'completed']:
    for stat in ['sum', 'mean']:
        if not (0.0 <= row[f'exercise_perc_goal_{col_suffix}_{stat}_{phase}'] <= 100.0): return 0
        if not (0.0 <= row[f'exercise_perc_real_{col_suffix}_{stat}_{phase}'] <= 100.0): return 0
        std_col = f'exercise_perc_real_{col_suffix}_std_{phase}'
        if row[std_col] < 0: return 0
if not (0.0 <= row[f'exercise_hour_entropy_{phase}'] <= 1.0): return 0

```

- Thống kê cơ bản về hành vi xem video: Các giá trị phải không được âm và các trường liên quan đến phần trăm như (phần trăm thời gian coi video) không lớn 100.

```

# Video features
video_features = [
    f'video_watch_count_{phase}',
    f'video_WATCHED_PERCENTAGE_{phase}',
    f'video_PERCENTAGE_WATCH_TIME_{phase}',
    f'video_PAUSE_COUNT_{phase}',
    f'video_PAUSE_AVG_{phase}',
    f'video_PAUSE_STD_{phase}',
    f'video_REWATCH_AVG_{phase}',
    f'video_REWATCH_STD_{phase}',
    f'video_TIME_BETWEEN_VIEWS_AVG_{phase}',
    f'video_TIME_BETWEEN_VIEWS_STD_{phase}',
    f'video_SPEED_AVG_{phase}',
    f'entropy_time_{phase}',
]
for col in video_features:
    if row[col] < 0:
        return 0, f'{col} có giá trị âm'
    if "percentage" in col or "watched" in col or "entropy" in col:
        if row[col] > 100:
            return 0, f'{col} vượt quá 100%'

```

b. Dữ liệu không rỗng (Not-Null)

Hàm kiểm tra một object (hàng trong dataframe) có giá trị rỗng hay không

```
def is_not_null_pandas(row):
    return not row.isnull().any()
```

c. Loại dữ liệu (Data Type)

- Kiểm tra user_id, course_id và school có phải dạng chuỗi không.

```
# 1. user_id, course_id: phải là str
if not isinstance(row['user_id'], str): return 0
if not isinstance(row['course_id'], str): return 0

# 2. school: str
if not isinstance(row['school'], str): return 0
```

- Kiểm tra user_enroll_time có phải dạng số hoặc chuỗi không.

```
# 3. user_enroll_time: dạng số hoặc chuỗi có thể trích xuất năm
if not isinstance(row['user_enroll_time'], (str, int, float)): return 0
```

- Kiểm tra có trường có kiểu dữ liệu là số nguyên là:
'user_past_course_count', 'user_time_since_last_course', 'video_count',
'exercise_count', 'chapter_count', 'num_field_x', 'num_prerequisites',
'duration_days'.

```
# 4. Các trường kiểu int
int_fields = [
    'user_past_course_count', 'user_time_since_last_course',
    'video_count', 'exercise_count', 'chapter_count',
    'num_field_x', 'num_prerequisites', 'duration_days'
]
for field in int_fields:
    if not isinstance(row[field], int): return 0
```

- Kiểm tra field_x có phải dạng list không.

```
# 5. field_x: list[str]
if not isinstance(row['field_x'], list): return 0
if not all(isinstance(x, str) for x in row['field_x']): return 0
```

- Kiểm tra assignment, video và exam có kiểu dữ liệu là float hoặc int.

```
# 6. assignment, video, exam: float/int
for field in ['assignment', 'video', 'exam']:
    if not isinstance(row[field], (float, int)): return 0
```

- Kiểm tra date có kiểu dữ liệu là str, int, float.

```
# 7. start_date, end_date: str hoặc số
for field in ['start_date', 'end_date']:
    if not isinstance(row[field], (str, int, float)): return 0
```

- Các trường phải thuộc kiểu **số nguyên (int)** gồm:
comment_count_phase{phase}, total_words_phase{phase}_x,
reply_count_phase{phase}, total_words_phase{phase}_y,
exercise_id_count_{phase}, và exercise_correct_sum_{phase}. Các trường
này phản ánh số lượng hành vi (bình luận, phản hồi, làm bài đúng) nên yêu
cầu kiểu dữ liệu nguyên. Trong khi đó, các trường như
entropy_time_comment_phase{phase}, entropy_time_reply_phase{phase},
exercise_num_problem_sum_{phase},
exercise_num_problem_mean_{phase},
exercise_attempts_sum_sum_{phase},
exercise_attempts_sum_mean_{phase},
exercise_attempts_mean_mean_{phase},
exercise_date_from_enroll_min_{phase},
exercise_date_from_enroll_mean_{phase},
exercise_date_from_enroll_max_{phase}, exercise_context_sum_{phase},
exercise_context_mean_{phase}, exercise_diff_sum_{phase},
exercise_diff_mean_{phase}, exercise_diff_min_{phase},
exercise_diff_max_{phase}, và exercise_hour_entropy_{phase} đều được
chấp nhận là kiểu **số thực (float) hoặc số nguyên (int)**,

```

# int
int_phase_fields = [
    f'comment_count_phase{phase}',
    f'total_words_phase{phase}_x',
    f'reply_count_phase{phase}',
    f'total_words_phase{phase}_y',
    f'exercise_id_count_{phase}',
    f'exercise_correct_sum_{phase}'
]
for field in int_phase_fields:
    if not isinstance(row[field], int): return 0

# float
float_phase_fields = [
    f'entropy_time_comment_phase{phase}',
    f'entropy_time_reply_phase{phase}',
    f'exercise_num_problem_sum_{phase}',
    f'exercise_num_problem_mean_{phase}',
    f'exercise_attempts_sum_sum_{phase}',
    f'exercise_attempts_sum_mean_{phase}',
    f'exercise_attempts_mean_mean_{phase}',
    f'exercise_date_from_enroll_min_{phase}',
    f'exercise_date_from_enroll_mean_{phase}',
    f'exercise_date_from_enroll_max_{phase}',
    f'exercise_context_sum_{phase}',
    f'exercise_context_mean_{phase}',
    f'exercise_diff_sum_{phase}',
    f'exercise_diff_mean_{phase}',
    f'exercise_diff_min_{phase}',
    f'exercise_diff_max_{phase}',
    f'exercise_hour_entropy_{phase}'
]

```

- Trường `exercise_language_binary_mean_{phase}` phải là số thực (float), vì giá trị trung bình của một biến nhị phân (0 hoặc 1) sẽ nằm trong khoảng thập phân. Tiếp theo, các trường bắt đầu bằng `exercise_perc_real_` và `exercise_perc_goal_` đại diện cho tỷ lệ đúng, điểm số và tỷ lệ hoàn thành - đều là giá trị phần trăm, vì vậy yêu cầu kiểu số thực (float) hoặc số nguyên (int). Với mỗi suffix là `correct`, `score`, hoặc `completed`, chương trình duyệt qua ba loại thống kê là sum, mean, và std, đảm bảo tất cả đều đúng kiểu. Riêng với `exercise_perc_goal_`, chỉ kiểm tra kiểu của sum và mean, vì không có trường std cho nhóm này.

```

for field in float_phase_fields:
    if not isinstance(row[field], (float, int)): return 0

# binary 0/1: cũng là int
if not isinstance(row[f'exercise_language_binary_mean_{phase}'], int): return 0

# các phần trăm: float
for suffix in ['correct', 'score', 'completed']:
    for stat in ['sum', 'mean', 'std']:
        col = f'exercise_perc_real_{suffix}_{stat}_{phase}'
        if not isinstance(row[col], (float, int)): return 0
        if stat != 'std':
            col_goal = f'exercise_perc_goal_{suffix}_{stat}_{phase}'
            if not isinstance(row[col_goal], (float, int)): return 0

```

- Hành vi xem video phải có kiểu giá trị là float:

```

# 7. Các chỉ số video theo phase 1
video_fields = [
    'video_watch_count_1',
    'video_WATCHED_PERCENTAGE_1',
    'video_PERCENTAGE_WATCH_TIME_1',
    'video_PAUSE_COUNT_1',
    'video_PAUSE_AVG_1',
    'video_PAUSE_STD_1',
    'video_REWATCH_AVG_1',
    'video_REWATCH_STD_1',
    'video_TIME_BETWEEN_VIEWS_AVG_1',
    'video_TIME_BETWEEN_VIEWS_STD_1',
    'video_SPEED_AVG_1'
]
for field in video_fields:
    if not isinstance(row[field], (float, int, np.floating, np.integer)):
        return 0, f'{field} không phải float/int'

```

d. Ràng buộc logic (Logical Constraints)

- Tổng điểm thành phần: Với mỗi hàng, tổng của assignment + video + exam phải đúng bằng 100. Đây là điều kiện bắt buộc để đảm bảo phân bổ điểm hợp lý.

```

# 1. Tổng điểm thành phần phải bằng 100
total_score = round(row['assignment'] + row['video'] + row['exam'], 2)
if total_score != 100.0:
    return False, f'Tổng điểm không bằng 100: {total_score}'

```

- Thời gian hợp lệ: Thời gian còn lại của người dùng cho khóa học (remaining_time) phải nhỏ hơn hoặc bằng thời lượng khóa học (duration_days).

```

# 2. Thời gian còn lại phải nhỏ hơn hoặc bằng thời lượng khóa học
if row['remaining_time'] > row['duration_days']:
    return False, f'Thời gian còn lại ({row['remaining_time']}) > thời lượng khóa học ({row['duration_days']})'

```

- Comment / Reply logic:

- Nếu $\text{comment_count_phase}\{i\} > 0$ thì $\text{total_words_phase}\{i\}_x$ cũng phải > 0 .
- Ngược lại, nếu không có comment thì tổng số từ phải bằng 0.
- Tương tự với $\text{reply_count_phase}\{i\}$ và $\text{total_words_phase}\{i\}_y$.

```
# 3.5 Emotion / comment logic
total_words = row.get(f'total_words_phase{phase}', 0)
if total_words > 0:
    pos = row.get(f'total_positive{phase}', 0)
    neg = row.get(f'total_negative{phase}', 0)
    neu = row.get(f'total_neutral{phase}', 0)
    if (pos + neg + neu) == 0:
        return False, f"Phase {phase}: Có bình luận nhưng không có cảm xúc nào được ghi nhận"
```

- Exercise logic:
 - Nếu $\text{exercise_id_count}\{i\} > 0$ thì tổng số câu trả lời đúng $\text{exercise_correct_sum}\{i\}$ phải > 0 các số liên quan đến exercise > 0 .
 - Ngược lại, nếu không làm bài nào thì tổng số câu trả lời đúng cũng phải bằng 0.

```
# 5. Exercise Phase 1
if row['exercise_id_count_1'] > 0:
    if row['exercise_correct_sum_1'] <= 0:
        return False, "Có làm bài tập phase 1 nhưng số câu đúng = 0"
    # Có thể kiểm tra thêm nhiều cột liên quan nếu cần
else:
    if row['exercise_correct_sum_1'] != 0:
        return False, "Không làm bài tập phase 1 nhưng có số câu đúng"
```

- Video logic:
 - Có video xem nhưng phần trăm xem = 0
 - Phần trăm video xem $> 100\%$
 - Có pause nhưng avg pause = 0

```
# 3.2 Video logic
watch_count = row.get(f'video_watch_count_{phase}', 0)
if watch_count > 0:
    if row.get(f'video_watched_percentage_{phase}', 0) <= 0:
        return False, f"Phase {phase}: Có video xem nhưng phần trăm xem = 0"
    if row.get(f'video_watched_percentage_{phase}', 0) > 100:
        return False, f"Phase {phase}: Phần trăm video xem > 100%"

# 3.3 Video pause logic
pause_count = row.get(f'video_pause_count_{phase}', 0)
if pause_count > 0:
    if row.get(f'video_pause_avg_{phase}', 0) <= 0:
        return False, f"Phase {phase}: Có pause nhưng avg pause = 0"
```

Tương tự với giai đoạn 2, 3, 4:

```
# 6. Comment/Reply/Exercise Phase 2 tương tự
if row['comment_count_phase2'] > 0 and row['total_words_phase2_x'] <= 0:
    return False, "Có comment nhưng tổng số từ bình luận phase 2 = 0"
if row['comment_count_phase2'] == 0 and row['total_words_phase2_x'] != 0:
    return False, "Không có comment nhưng vẫn có từ bình luận phase 2"

if row['reply_count_phase2'] > 0 and row['total_words_phase2_y'] <= 0:
    return False, "Có reply nhưng tổng số từ reply phase 2 = 0"
if row['reply_count_phase2'] == 0 and row['total_words_phase2_y'] != 0:
    return False, "Không có reply nhưng vẫn có từ reply phase 2"

if row['exercise_id_count_2'] > 0:
    if row['exercise_correct_sum_2'] <= 0:
        return False, "Có làm bài tập phase 2 nhưng số câu đúng = 0"
else:
    if row['exercise_correct_sum_2'] != 0:
        return False, "Không làm bài tập phase 2 nhưng có số câu đúng"
```

e. Tính duy nhất (Uniqueness)

Kiểm tra một object có bị trùng lặp với các hàng khác không.

```
def check_duplicates(df):
    # Đếm số dòng trùng lặp hoàn toàn
    total_duplicates = df.duplicated().sum()
    print(f"Số dòng bị trùng lặp hoàn toàn: {total_duplicates}")

    # Kiểm tra trùng khi bỏ user_id và course_id
    subset_columns = [col for col in df.columns if col not in ['user_id', 'course_id']]
    duplicates_without_ids = df.duplicated(subset=subset_columns).sum()
    print(f"Số dòng trùng lặp khi bỏ 'user_id' và 'course_id': {duplicates_without_ids}")
```

Số dòng bị trùng lặp hoàn toàn: 0

Số dòng trùng lặp khi bỏ 'user_id' và 'course_id': 19483

- Dữ liệu cho thấy không có dòng bị trùng lặp hoàn toàn. Tuy nhiên khi bỏ đi khóa chính là {user_id; course_id} thì số dòng có kết quả trùng nhau là 19483 dòng.

f. Tính khóa ngoại (Foreign Key Integrity)

Trong dữ liệu có 2 khóa ngoại: user_id và course_id

- Kiểm tra các user_id có trong file user.json (file thông tin người dùng)

```
user_info = pd.read_json("/kaggle/input/lightmooccubex/entities/user.json", lines = True)
user_info
```

```
/usr/local/lib/python3.10/dist-packages/pandas/io/format.py:1458: RuntimeWarning: invalid value encountered in greater
    has_large_values = (abs_vals > 1e6).any()
/usr/local/lib/python3.10/dist-packages/pandas/io/format.py:1459: RuntimeWarning: invalid value encountered in less
    has_small_values = ((abs_vals < 10 ** (-self.digits)) & (abs_vals > 0)).any()
/usr/local/lib/python3.10/dist-packages/pandas/io/format.py:1459: RuntimeWarning: invalid value encountered in greater
    has_small_values = ((abs_vals < 10 ** (-self.digits)) & (abs_vals > 0)).any()
```

	id	name	gender	school	year_of_birth	course_order	enroll_time
0	U_22	我	0.0		2015.0	[682129, 2294668]	[2019-10-12 10:28:02, 2020-11-21 14:03:28]
1	U_24	王帅国	1.0	清华大学	6558.0	[597214, 605512, 597211, 597314, 597208, 62950...]	[2019-05-20 16:06:48, 2019-05-24 19:34:43, 201...
2	U_25	王帅国	0.0	清华大学	NaN	[1903985]	[2020-08-07 18:59:13]
3	U_53	于欽杰	1.0	清华大学	1973.0	[696679, 1704639, 943255, 1729417, 682164, 177...	[2020-03-01 21:24:30, 2020-03-12 16:17:02, 202...
4	U_54	马昱春	2.0	清华大学	NaN	[682442, 682164, 1748240, 1778890, 1829031, 17...	[2019-10-09 02:17:49, 2019-11-08 00:49:03, 202...
...
3330289	U_34712108		0.0		NaN	[697791, 782490, 799796]	[2020-10-12 03:39:14, 2020-10-12 03:41:00, 202...

- Kiểm tra các course_id mà user đăng kí có trong file course.json (file thông tin khóa học)

```
course_info = pd.read_json("/kaggle/input/lightmooccubex/entities/course.json", lines = True)
course_info
```

	id	name	field	prerequisites	about	resource
0	C_584313	《资治通鉴》导读	[历史学, 中国语言文学]		通过老师导读，同学们可深入这一经典文本内部，得以纵览千年历史，提升国学素养，体味人生智慧。	[{"titles": ["第一课 导论与三家分晋", "导论", "导论"], "resou...
1	C_584329	微积分——极限理论与一元函数	[应用经济学, 数学, 物理学, 理论经济学]		本课程是理工科的一门数学基础课，系统、全面地介绍了一元函数微积分学。课程既保持了数学的严谨和...	[{"titles": ["序言", "序言", "序言"], "resource_id": "..."]]
2	C_584381	新闻摄影	[艺术学, 新闻传播学]		掌握基本的摄影技能，了解图片新闻的工作方式，训练对生活的观察和热爱，发展对图像的审美和批评能...	[{"titles": ["第一章 绪论", "第一章 绪论", "第一章 绪论"], "resou...
3	C_597208	数据挖掘：理论与算法	[计算机科学与技术]		最有趣的理论+最有用的算法=不得不学的数据科学。	[{"titles": ["走进数据科学：博大精深，美不胜收", "整装待发", "Vide..."]]

```
def valid_foreign_key(row, valid_user_ids, valid_course_ids):
    user_id = str(row['user_id']).strip()
    course_id = str(row['course_id']).strip()

    print(user_id in valid_user_ids)
    print(course_id in valid_course_ids)

    return user_id in valid_user_ids and course_id in valid_course_ids

# Chuyển thành list để dùng `in`
valid_user_ids = user_info["id"].astype(str).str.strip().tolist()
valid_course_ids = course_info["id"].astype(str).str.strip().tolist()
```

g. Tính Consistency cho một Object

Là tỷ lệ giữa số lượng điều kiện hợp lệ trong object và tổng số điều kiện cần kiểm tra.

$$Consistency(O) = \frac{\text{Số lượng điều kiện hợp lệ}}{\text{Tổng số điều kiện cần kiểm tra}} \times 100\%$$

Tính Consistency (tính nhất quán) đánh giá mức độ dữ liệu không mâu thuẫn và tuân theo các quy tắc logic.

Áp dụng cho dữ liệu khảo sát.

```
: row0 = data.iloc[1]
row0
```

```
: user_id                  U_1000979
school                   云南大学
course_id                C_947149
user_enroll_time        2020-03-03
user_past_course_count   0
...
exercise_perc_real_correct_std_2    0.0
exercise_perc_real_score_sum_2     0.0
exercise_perc_real_score_mean_2    0.0
exercise_perc_real_score_std_2     0.0
exercise_hour_entropy_2           0.0
Name: 1, Length: 98, dtype: object
```

```
print("Kiểm tra 1 dòng (dòng 0):")
print("Không null:", is_not_null_pandas(row0))
print("Kiểu dữ liệu hợp lệ:", validate_data_type(row0))
print("Range hợp lệ:", validate_range(row0))
print("Logic hợp lệ:", check_valid_logic(row0))
print("Khóa ngoại hợp lệ:", valid_foreign_key(row0, valid_user_ids, valid_course_ids))
print("Bị lặp không:", is_row_duplicated(row0, data))
```

```
Kiểm tra 1 dòng (dòng 0):
Không null: True
Kiểu dữ liệu hợp lệ: (1, 'Hợp lệ')
Range hợp lệ: (1, 'Hợp lệ')
Logic hợp lệ: (True, 'Hợp lệ')
Khóa ngoại hợp lệ: True
Bị lặp không: False
```

Áp dụng công thức tính phần trăm đảm bảo tính consistency cho mỗi object.

```

def check_all_criteria(row):
    total = 6
    passed = 0
    details = {}

    criteria = {
        "Không null": is_not_null_pandas(row),
        "Kiểu dữ liệu hợp lệ": validate_data_type(row)[0] == 1,
        "Range hợp lệ": validate_range(row)[0] == 1,
        "Logic hợp lệ": check_valid_logic(row)[0] == 1,
        "Khóa ngoại hợp lệ": valid_foreign_key(row, valid_user_ids, valid_course_ids),
        "Bị lặp không": not is_row_duplicated(row, data)
    }

    for name, passed_check in criteria.items():
        if passed_check:
            passed += 1
        details[name] = passed_check

    percentage = round((passed / total) * 100, 2)
    return percentage, details

```

Bao gồm 6 mục tiêu chí và trong mỗi mục sẽ có các tiêu chí được trình bày chi tiết ở trên.

Nếu như object không hoàn thành một mục trong tiêu chí thì tính không thỏa.

Ví dụ: object thỏa các mục tiêu chí trừ range hợp lệ thì phần trăm consistency của object là % = 83%

```

> percent, detail = check_all_criteria(row0)
print(f"Phần trăm tiêu chí đạt: {percent}%")
print("Chi tiết:")
for k, v in detail.items():
    print(f"{k}: {'Hợp lệ' if v else 'Không hợp lệ'}")

```

Phần trăm tiêu chí đạt: 100.0%
Chi tiết:
Không null: Hợp lệ
Kiểu dữ liệu hợp lệ: Hợp lệ
Range hợp lệ: Không hợp lệ
Logic hợp lệ: Hợp lệ
Khóa ngoại hợp lệ: Hợp lệ
Bị lặp không: Hợp lệ

5.3.8.5 Consistency toàn bộ Dataset

Consistency trung bình trên tất cả các objects với m là số lượng object.

$$Consistency = \frac{\sum_{i=1}^m Consistency(O_i)}{m}$$

```

# Áp dụng hàm cho mỗi dòng
percent_list = []
details_list = []

for _, row in data.iterrows():
    percent, details = check_all_criteria(row)
    percent_list.append(percent)
    details_list.append(details)

# Tạo DataFrame kết quả
criteria_df = pd.DataFrame(details_list)
criteria_df['percent_passed'] = percent_list

# Trung bình phần trăm tiêu chí đạt mỗi dòng
average_percent = round(criteria_df['percent_passed'].mean(), 2)

# In kết quả
print("Trung bình % tiêu chí thỏa mãn trên toàn bộ dataset:", average_percent)

# (Tuỳ chọn) In tỷ lệ từng tiêu chí thỏa mãn (%)
print("\nTỷ lệ từng tiêu chí thỏa mãn (%):")
print(criteria_df.drop(columns='percent_passed').mean() * 100).round(2)

```

Do dữ liệu lớn nên chỉ đánh giá khoảng 80.000 dòng cho phase 1, 70.000 dòng cho phase 2, 60.000 dòng cho phase 3, 40.000 dòng cho phase 4.

Trung bình % tiêu chí thỏa mãn: 93.47%

Tỷ lệ từng tiêu chí thỏa mãn (%):

Không null	92.8
Kiểu dữ liệu hợp lệ	100.0
Range hợp lệ	69.5
Logic hợp lệ	98.5
Khóa ngoại hợp lệ	100.0
Bị lặp không	100.0

Kết quả trong phase 1, các phase khác trình bày trong excel.

Nhận xét:

- Chất lượng dữ liệu khá tốt:
 - Mức độ 93,47% cho thấy phần lớn dữ liệu đạt yêu cầu về định dạng, kiểu dữ liệu, logic và ràng buộc.
 - Đây là một tỷ lệ cao, đặc biệt nếu dataset có nhiều dòng và tiêu chí kiểm tra nghiêm ngặt.
- Vẫn còn một tỷ lệ nhỏ không hợp lệ (~6,53%) xảy ra vì lý do có giá trị null và rāne , logic không hợp lệ:
 - Cần xem xét chi tiết những dòng không đạt: là lỗi hệ thống, nhập liệu hay dữ liệu thiếu?
 - Có thể xác định những tiêu chí nào bị vi phạm nhiều nhất (ví dụ: kiểu dữ liệu, null, logic...).
- Dataset đã sẵn sàng cho phân tích hoặc mô hình hóa, nhưng:

- Cần làm sạch tiếp phần còn lỗi nếu mục tiêu là xây dựng mô hình machine learning hoặc phân tích thống kê chính xác.
- Có thể cân nhắc loại bỏ các dòng lỗi hoặc thay thế/điền giá trị hợp lý (imputation).

5.3.9. Timeliness

Tính kịp thời: Dữ liệu phải được cập nhật và phản ánh tình hình hiện tại, không bị lỗi thời.

Ý tưởng đo lường: Đo lường mức độ dữ liệu có sẵn đúng thời điểm cần thiết để sử dụng.

5.3.9.1 Khảo sát dữ liệu và xác định thời gian chủ yếu được cập nhật

Timeliness (tính kịp thời) trong phân tích dữ liệu đề cập đến việc đo lường khoảng thời gian giữa các lần cập nhật dữ liệu hoặc thời gian gần đây nhất dữ liệu được cập nhật so với thời điểm hiện tại. Nó giúp xác định mức độ mới mẻ và hợp thời của dữ liệu.

Trong bộ dữ liệu này, các thuộc tính về thời gian (các trường user_enroll_time, user_time_since_last_course, start_date, end_date, duration_days, remaining_time) đều không mang ý nghĩa về mặt cập nhật dữ liệu thường nên nhóm không thể demo được cho tính chất kịp thời của chất lượng dữ liệu.

Do đó, nhóm chúng em đã đề xuất một kịch bản mới để bổ sung và cải thiện phân tích, tập trung vào việc theo dõi, kiểm tra những dữ liệu ngoài vùng khảo sát, hoặc trẽ qua boxplot.

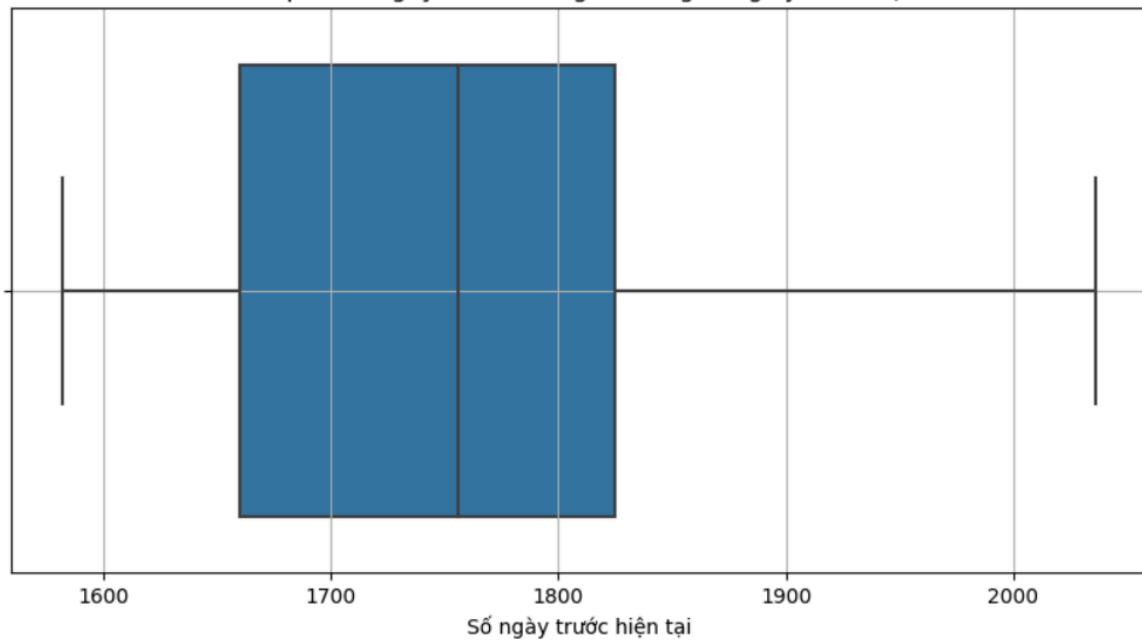
Đó là những điểm dữ liệu nằm ngoài vùng giá trị hợp lý (Domain Range), quá xa trung bình so với dữ liệu chung (dùng IQR hoặc Z-score).

Những mục cần kiểm tra:

- User_past_course (số lượng khóa học) quá xa thời gian hiện tại
- Duration_days (độ dài khóa học) quá dài hoặc quá ngắn bất thường.
- user_time_since_last_course (thời gian giữa lần đăng ký khóa học gần nhất đến khóa học hiện tại) có giá trị quá lớn.
- User_enroll_time (thời điểm đăng ký khóa học) quá xa so với thời gian hiện tại.

Enroll_time:

Boxplot số ngày kể từ khi người dùng đăng ký khóa học



Khoảng giá trị chính (IQR):

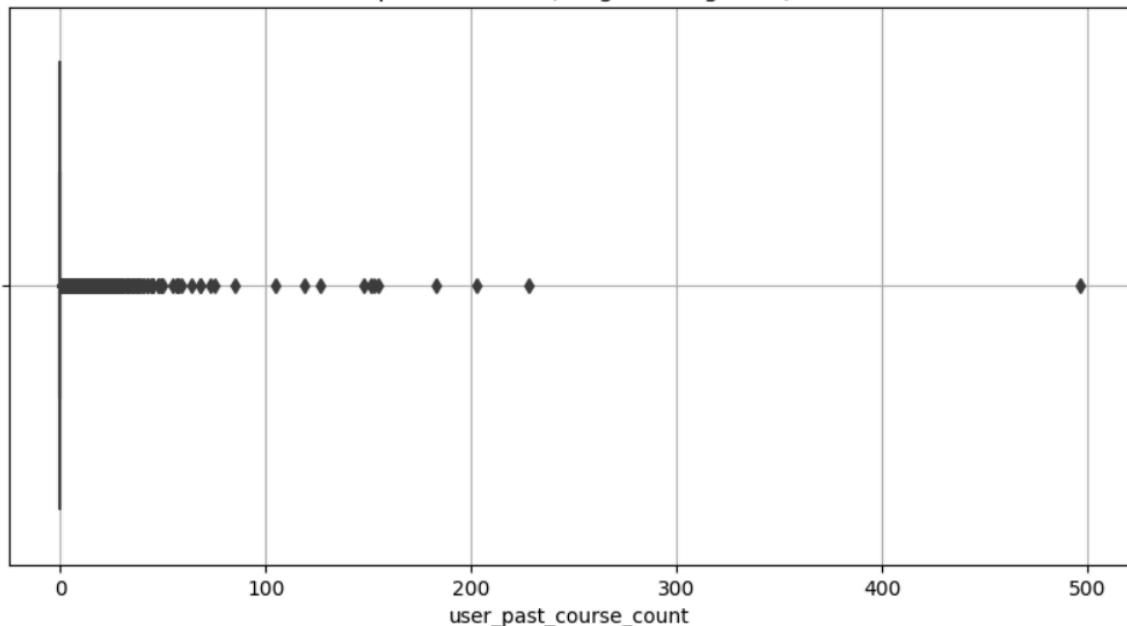
- Phần lớn người dùng đăng ký khóa học trong khoảng **1600 đến 2000 ngày** trước hiện tại (tức là khoảng **4.4 đến 5.5 năm trước**). Điều này cho thấy bộ dataset được thu thập vào khoảng cuối năm 2021
- Phân tích so sánh với móc thời gian hiện tại thi cho thấy dataset khá là cũ.
- Median (trung vị) nằm đâu đó quanh **1750 ngày** (~ 4.8 năm).

Outliers:

- Những người này đăng ký quá **sớm** hoặc quá **gần đây** so với phần lớn người dùng khác → đây là **outliers**.
- Nhưng qua boxplot ta nhận thấy không có outlier nào.

Số lượng khóa học đã đăng kí:

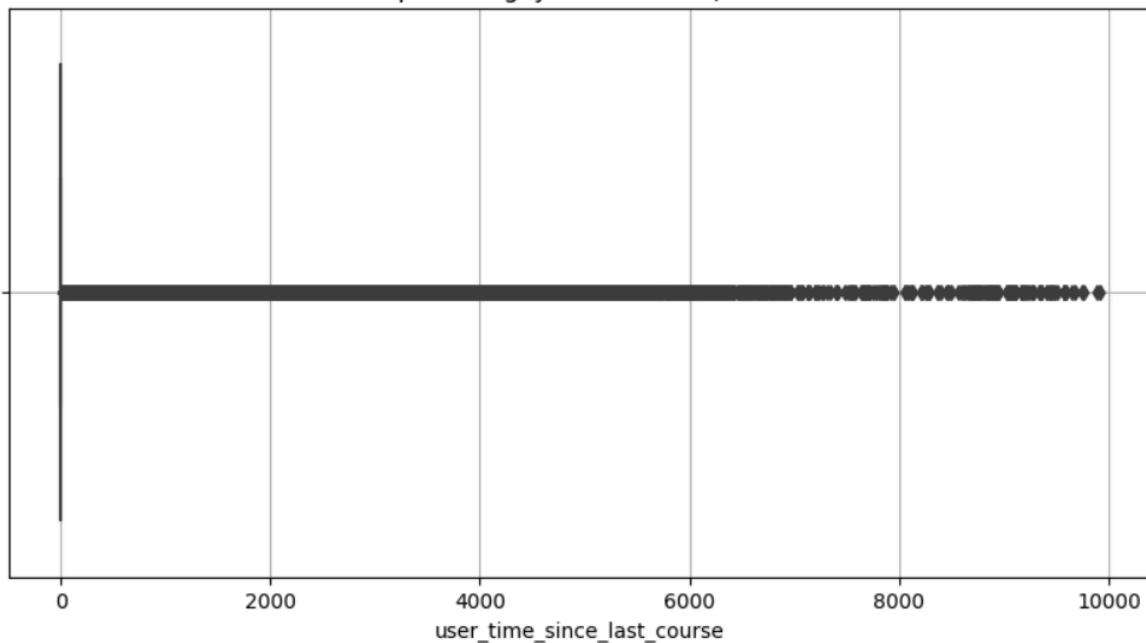
Boxplot số khóa học người dùng đã học



- **Phần lớn người dùng học rất ít khóa học trước đó:**
Median (giá trị trung vị) nằm rất gần 0, cho thấy đa số người dùng có số lượng khóa học cũ khá thấp.
- **Phân phối lệch phải rất mạnh (right-skewed):**
Một số người học rất nhiều khóa học trước đó (từ 100 đến gần 500), trong khi đa số còn lại học rất ít.
Điều này thể hiện qua phần "đuôi dài" bên phải của boxplot.
- **Nhiều outliers:**
Rất nhiều điểm nằm ngoài vùng whiskers (râu), tức là vượt xa giới hạn IQR. Đặc biệt có một vài điểm cực kỳ cao (~500), là **outliers mạnh (extreme outliers)**.

Thời gian từ lần đăng kí khóa học trước đến khóa học hiện tại

Boxplot số ngày kể từ khóa học trước



- **Phân phối rất lệch phải (right-skewed):**

Phần lớn giá trị tập trung ở phía gần 0, tức là đa số người dùng mới học khóa trước đây không lâu hoặc do đăng ký nhiều khóa học cùng lúc.

- **Outlier rất rõ ràng:**

Có rất nhiều điểm nằm **rất xa phía bên phải** của boxplot (từ 4000 ngày trở lên đến gần 10,000 ngày).

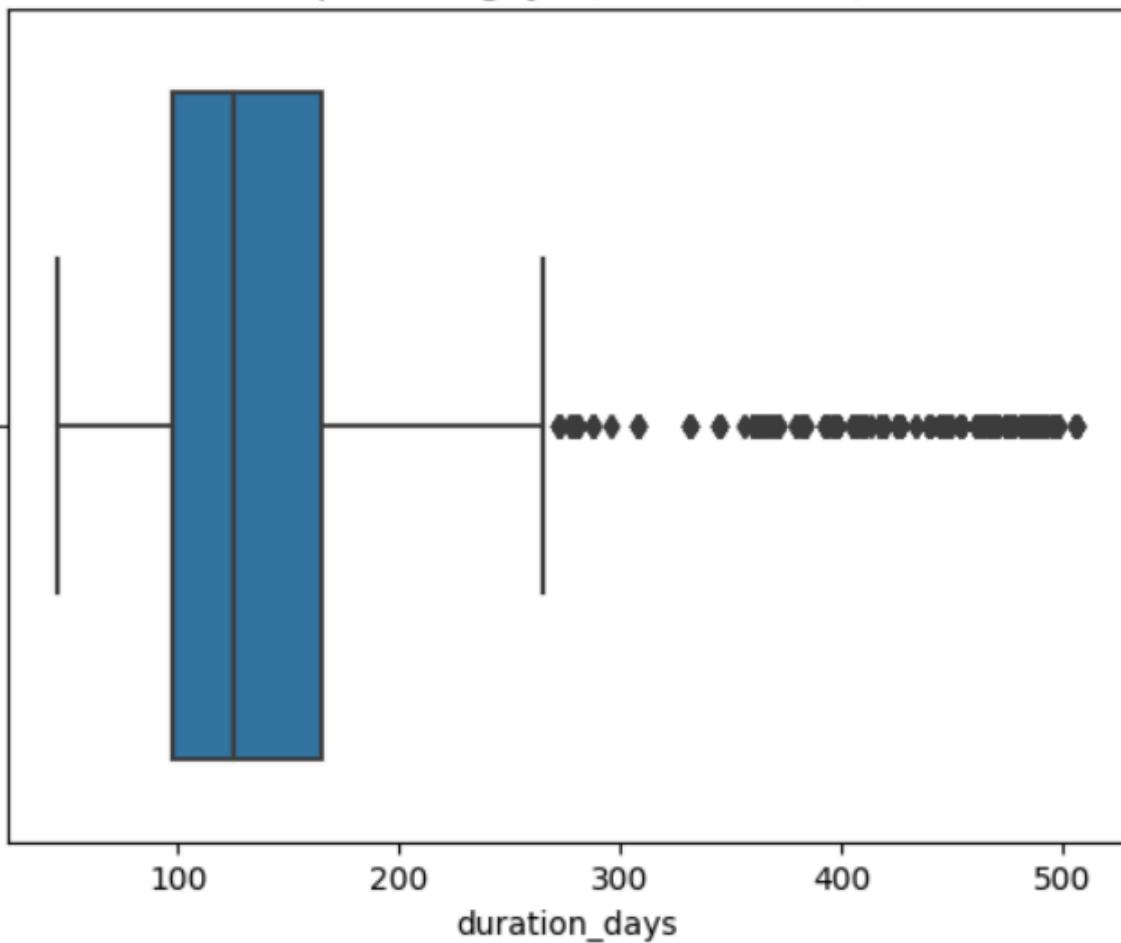
Những điểm này là **outliers** vì nằm ngoài khoảng xác định bởi quy tắc IQR ($Q3 + 1.5 \times IQR$).

- **Có thể có dữ liệu lỗi hoặc đặc biệt:**

Những giá trị như 9000–10000 ngày (~25 năm) có thể là lỗi dữ liệu, hoặc hệ thống ghi nhận không đúng thời điểm.

Số ngày của khóa học:

Boxplot số ngày học của khóa học



Phần lớn các giá trị tập trung trong khoảng 90 đến 220 ngày.

- Đây là phần thân hộp (box) — tức từ Q1 đến Q3 (25% đến 75% dữ liệu).

Có khá nhiều giá trị ngoại lai (outliers) nằm ở phía bên phải — tức các khóa học có thời lượng dài bất thường (trên ~250 ngày).

- Các dấu chấm nhỏ bên ngoài "râu" phải biểu thị những outliers.
- Một số khóa học kéo dài tới gần 500 ngày, điều này **không phổ biến** và có thể là ngoại lệ, nhập sai hoặc các trường hợp đặc biệt.

5.3.9.2 Áp dụng công thức tính timeliness giả sử thời gian cập nhật

Giả sử sau 2 tuần kể từ ngày đăng ký khóa học thì dữ liệu của user tự động cập nhật để hỗ trợ việc dự đoán.

Timeliness phản ánh mức độ kịp thời của việc cập nhật dữ liệu người học. Dữ liệu được coi là cập nhật đúng hạn nếu trong mỗi lần cập nhật định kỳ, các cột hành vi học tập như: Xem video, Làm bài tập, Thảo luận nhóm -> không chứa giá trị NaN tại tuần tương ứng.

Sau khi người dùng đăng ký khóa học, dữ liệu học tập sẽ tự động được cập nhật định kỳ 2 tuần một lần. Quá trình cập nhật này diễn ra tổng cộng 4 lần trong vòng 8 tuần, như sau:

Giai đoạn (Phase)	Số tuần sau đăng ký	Số lần cập nhật tích lũy
Phase 1	2 tuần	1 lần
Phase 2	4 tuần	2 lần
Phase 3	6 tuần	3 lần
Phase 4	8 tuần	4 lần

Timeliness theo Object

Với mỗi người học (object), Timeliness được tính là:

$$Timeliness(O) = \frac{\text{Số lần cập nhật đúng hạn}}{\text{Tổng số lần cập nhật}} \times 100\%$$

Trong đó, mỗi lần cập nhật đúng hạn yêu cầu không có NaN trong dữ liệu hành vi tại các cột tương ứng của tuần cập nhật đó.

Các bước thực hiện:

- Trong Phase 1 (2 tuần đầu), chỉ có 1 lần cập nhật.
- Ta cần kiểm tra xem dữ liệu ở lần cập nhật này có bị thiếu (NaN) ở các cột hành vi học tập hay không.
- Nếu ít nhất một cột hành vi ở phase 1 bị NaN → cập nhật không kịp thời → điểm timeliness = 0.
- Nếu không có NaN trong các cột hành vi ở phase 1 → điểm timeliness = 1.

```
def calculate_timeliness_from_files(base_dir, num_phases=4):
    """
    Tính timeliness của từng user từ các file phase riêng biệt.

    base_dir: thư mục gốc chứa các thư mục phaseX
    num_phases: số lượng phase (mặc định 4)

    Trả về:
        df_final: DataFrame chứa user_id, timeliness theo từng phase, và trung bình timeliness
    """

    timeliness_dfs = []

    for phase in range(1, num_phases + 1):
        file_path = os.path.join(base_dir, f'phase{phase}', f'user_train_phase_{phase}.csv')

        if not os.path.exists(file_path):
            print(f"[!] Không tìm thấy file {file_path}. Bỏ qua Phase {phase} .")
            continue

        df = pd.read_csv(file_path)

        if 'user_id' not in df.columns:
            raise ValueError(f"[!] File {file_path} không chứa cột 'user_id'.")

        # Lấy các cột hành vi cần kiểm tra trong phase này (có hậu tố _<phase> hoặc chứa 'phaseX')
        behavior_cols = [col for col in df.columns if col.endswith(f"_phase{phase}") or f"phase{phase}" in col]

        if not behavior_cols:
            print(f"[!] Không tìm thấy cột hành vi trong {file_path}.")
            df[f'timeliness_phase{phase}'] = pd.NA
        else:
            timely_flags = ~df[behavior_cols].isnull().any(axis=1)
            df[f'timeliness_phase{phase}'] = timely_flags.astype(int)

        # Giữ lại user_id và timeliness
        timeliness_df = df[['user_id', 'course_id', f'timeliness_phase{phase}']]
        timeliness_dfs.append(timeliness_df)

    # Gộp tất cả các phase theo user_id, course_id
    df_final = timeliness_dfs[0]
    for df in timeliness_dfs[1:]:
        df_final = df_final.merge(df, on=['user_id', 'course_id'], how='outer')

    # Tính trung bình timeliness
    phase_cols = [f'timeliness_phase{p}' for p in range(1, num_phases + 1)]
    df_final['timeliness_avg'] = df_final[phase_cols].mean(axis=1, skipna=True)

    return df_final
```

	user_id	course_id	timeliness_phase1	timeliness_phase2	timeliness_phase3	timeliness_phase4	timeliness_avg
0	U_10000	C_2033958	1	1.0	1.0	1.0	1.0
1	U_1000979	C_947149	1	1.0	1.0	1.0	1.0
2	U_1000982	C_947149	1	1.0	1.0	NaN	1.0
3	U_1001176	C_947149	1	1.0	1.0	1.0	1.0
4	U_1001413	C_735164	1	1.0	NaN	NaN	1.0
...
108117	U_99746	C_674971	1	1.0	1.0	1.0	1.0
108118	U_997506	C_2095102	1	1.0	1.0	1.0	1.0
108119	U_99753	C_1428968	1	1.0	1.0	1.0	1.0
108120	U_997542	C_2066096	1	1.0	1.0	1.0	1.0
108121	U_99772	C_1903985	1	1.0	1.0	1.0	1.0

Timeliness theo từng object của từng phase (giai đoạn). Các chỉ số là NaN là các khóa học không kéo dài đến 6, hoặc 8 tuần.

Timeliness của từng object (user-course) rất lớn = 1 → nghĩa là dữ liệu được cập nhật kịp thời trong tất cả các phase cần thiết.

Timeliness toàn bộ Dataset

Timeliness trung bình trên tất cả các objects với m là số lượng object.

$$\text{Timeliness} = \frac{\sum_{i=1}^m \text{Timeliness}(O_i)}{m}$$

```
> # Lấy danh sách cột timeliness theo phase
phase_cols = ['timeliness_phase1', 'timeliness_phase2', 'timeliness_phase3', 'timeliness_phase4']

# Tính trung bình cho mỗi phase (bỏ qua NaN)
timeliness_means = final_timeliness_df[phase_cols].mean()

print("Trung bình timeliness theo từng phase:")
timeliness_means

Trung bình timeliness theo từng phase:
2]: timeliness_phase1    1.0
timeliness_phase2    1.0
timeliness_phase3    1.0
timeliness_phase4    1.0
dtype: float64
```

Và nếu tất cả giá trị timeliness_phase1 đến timeliness_phase4 đều bằng 1.0, thì:

- Có nghĩa là không có NaN trong các cột hành vi của bất kỳ phase nào
- Tức là dữ liệu được cập nhật đầy đủ, đúng hạn trong từng giai đoạn
=> Timeliness = 1.00 là kết luận chính xác.

5.4 Trích xuất đặc trưng từ đồ thị (graph) truyền thống

5.4.1 Các thành phần cơ bản trong graph

5.4.1.1 Tổng quan về Đồ thị (Graph)

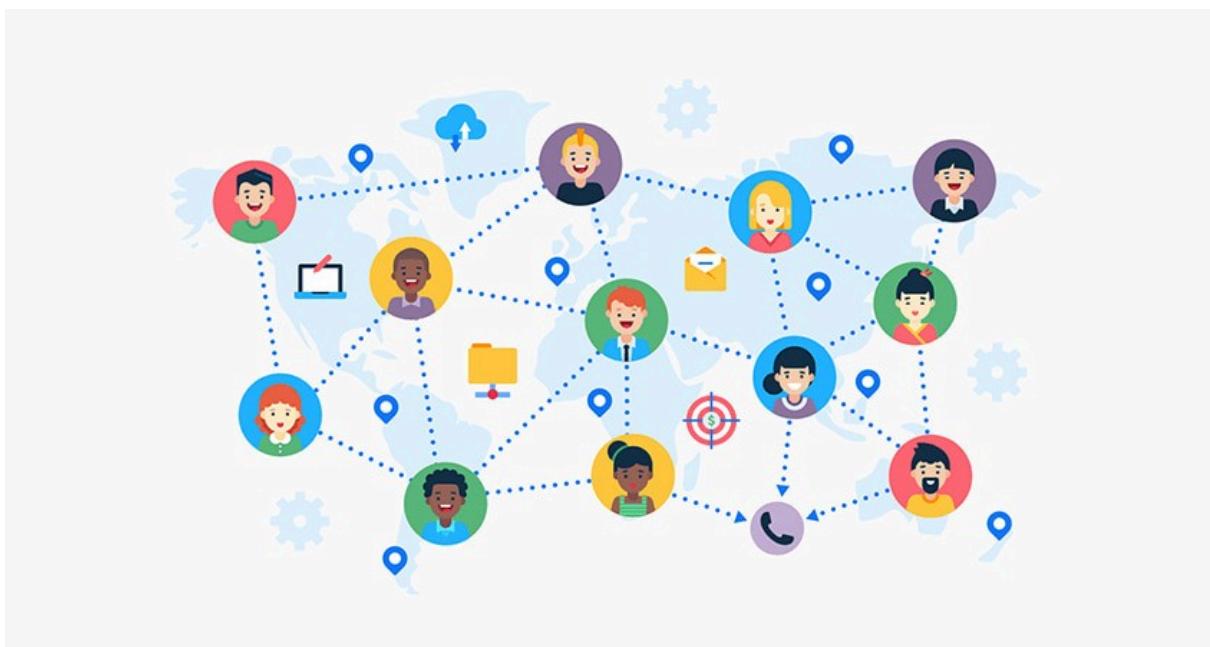
5.4.1.1.1 Định nghĩa cơ bản:

Một đồ thị $G=(V,E)$ bao gồm:

- V (Vertices): Tập các đỉnh (nodes), đại diện cho thực thể.
- E (Edges): Tập các cạnh (edges), đại diện cho mối quan hệ giữa các đỉnh.

Ví dụ:

- Mạng xã hội: Người dùng là đỉnh, quan hệ bạn bè là cạnh.
- Mạng giao thông: Giao lộ là đỉnh, đường đi là cạnh.
- Dữ liệu giao dịch: Khách hàng, sản phẩm là đỉnh; cạnh biểu diễn việc mua hàng.



=> Trong graph, thông tin quan trọng nằm trong cấu trúc liên kết giữa các node.

Để sử dụng graph cho việc trích xuất đặc trưng:

- Biến đồ thị thành dạng bảng chứa đặc trưng số học để dùng trong mô hình ML như XGBoost, LightGBM, v.v.

5.4.1.1.2 Lợi ích khi trích xuất đặc trưng từ đồ thị

- Chuyển đổi dữ liệu đồ thị thành dạng bảng quen thuộc
 - Cho phép áp dụng các mô hình ML thông thường như: XGBoost, Random Forest, Logistic Regression, SVM...

- Ví dụ:
 - Node = khách hàng → Trích degree, pagerank, clustering
 - Cạnh = giao dịch → Trích common neighbors, jaccard
 - Dự đoán gian lận giao dịch.
- Khai thác mối quan hệ ẩn trong dữ liệu
 - Các mối quan hệ không thể hiện rõ ràng trong bảng dữ liệu có thể được nắm bắt qua cấu trúc đồ thị.
 - Các đặc trưng như PageRank, Betweenness, Centrality mô tả vai trò của thực thể trong mạng lưới.
 - Ví dụ:
 - Hai người không mua cùng sản phẩm, nhưng có bạn bè chung → gợi ý sản phẩm.
 - Một khách hàng nằm giữa nhiều cụm trong mạng → có thể là trung gian lừa đảo.
- Tăng độ chính xác cho mô hình ML
 - Khi kết hợp đặc trưng đồ thị vào dữ liệu gốc → mô hình có nhiều tín hiệu hơn → giảm underfitting.
 - Ví dụ:
 - Thêm degree, pagerank vào tập đặc trưng khách hàng.
- Cho phép phát hiện cộng đồng, phân cụm, hoặc phân lớp mà không cần label
 - Với các đặc trưng như modularity, clustering coefficient có thể phát hiện nhóm tương tác trong mạng.
 - Dùng trong phân tích mạng xã hội, nghiên cứu xã hội học, phân loại thuốc,...
 - Ví dụ:
 - Chia các node thành nhóm bằng k-means trên embedding → tìm nhóm người dùng có hành vi giống nhau.
- Tiền đề để chuyển sang Graph Neural Network (GNN)
 - Trích đặc trưng là bước đầu để hiểu graph và sau này chuyển sang deep learning (GNN).
 - Cũng giúp tạo embedding vector tốt hơn nếu cần sử dụng với mô hình khác.
 - Ví dụ:
 - Trích degree, pagerank, clustering → concatenate với feature khác → đưa vào GNN.
- Hiệu quả hơn với dữ liệu ít và không đầy đủ

- Trong nhiều tình huống, label ít hoặc dữ liệu thiếu → đặc trưng cấu trúc từ graph có thể bù đắp.
- Ví dụ:
 - Mạng lưới khách hàng chưa có thông tin đầy đủ → dùng graph để bổ sung "tầm ảnh hưởng", "độ liên kết".
- Giải thích mô hình tốt hơn (Interpretability)
 - Các đặc trưng từ graph như degree, centrality rất trực quan → dễ giải thích cho business.
 - Ví dụ:
 - “Khách hàng này có degree cao, nên họ là người kết nối nhiều người khác → dễ lan truyền tin đồn xấu.”
- Không cần deep learning mà vẫn tận dụng được sức mạnh của graph
 - Trong nhiều bài toán, chỉ cần trích đặc trưng đơn giản là đã đạt hiệu quả tốt.
 - Không cần GPU, không cần mạng GNN phức tạp.

5.4.1.1.3 Phân loại Trích xuất đặc trưng từ đồ thị

a) Node-level Features – Đặc trưng tại từng đỉnh

Đây là phổ biến nhất khi xử lý dữ liệu bảng. Mỗi node là một dòng dữ liệu, và trích ra các đặc trưng như:

Tên đặc trưng	Mô tả
Degree	Số cạnh kết nối với node đó
In-degree / Out-degree	Với đồ thị có hướng: số cạnh đi vào / đi ra
Clustering Coefficient	Mức độ mà các hàng xóm của node kết nối với nhau
PageRank	Tầm quan trọng của node
Betweenness Centrality	Node đó nằm trên bao nhiêu đường đi ngắn nhất
Eigenvector Centrality	Mức độ ảnh hưởng tổng thể, dựa vào hàng xóm quan trọng
K-core number	Là phần tử của k-core subgraph lớn nhất

Ví dụ: Trong mạng xã hội, một người có degree cao là người quen biết nhiều, PageRank cao là người quan trọng.

b) Edge-level Features – Đặc trưng cho các cạnh

Dùng khi muốn phân loại cạnh (link prediction, phát hiện gian lận):

Tên đặc trưng	Mô tả
Jaccard Similarity	Mức độ trùng lặp giữa hàng xóm của hai node
Common Neighbors	Số lượng hàng xóm chung của hai node
Adamic-Adar Index	Đánh trọng số cho hàng xóm chung hiếm gặp hơn
Preferential Attachment	Tích số degree của hai node

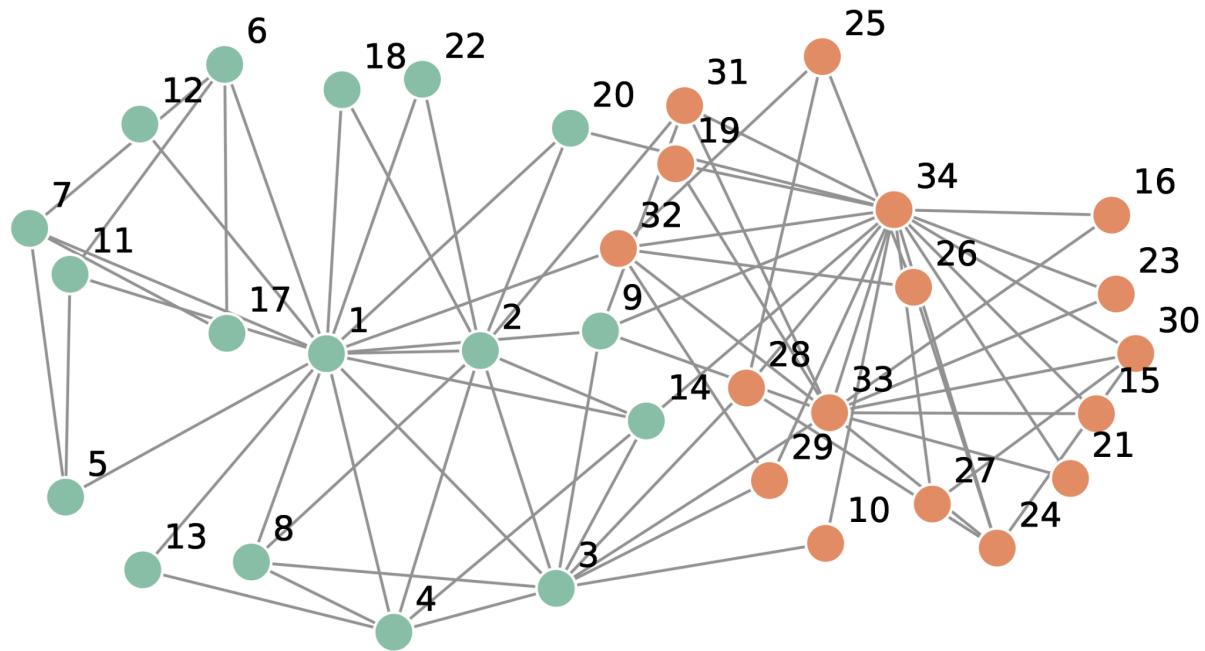
c) Graph-level Features – Đặc trưng cho toàn đồ thị

Tên đặc trưng	Mô tả
Số node / cạnh	Kích thước đồ thị
Mật độ đồ thị (density)	Mức độ kết nối giữa các node
Đường đi trung bình (avg. shortest path)	Mức độ "gần nhau" giữa các node
Đường kính đồ thị (diameter)	Đường đi dài nhất giữa hai node bất kỳ
Độ phân cụm (modularity)	Mức độ chia thành cộng đồng (community detection)

Dùng khi muốn phân loại toàn bộ đồ thị (dạng đồ thị là input):

5.4.1.1.4 Ví dụ áp dụng: Zachary's Karate Club Graph

- Là một đồ thị mô hình hóa mối quan hệ xã hội giữa các thành viên trong một câu lạc bộ karate thực tế.
- Có 34 đỉnh (nodes): mỗi node là một thành viên.
- Có 78 cạnh (edges): mối quan hệ thân thiết giữa các thành viên.
- Sau một mâu thuẫn giữa chủ nhiệm và huấn luyện viên, câu lạc bộ tách thành 2 nhóm. Thông tin này được dùng làm ground truth cho các bài toán phân cụm/community detection.



Từ đồ thì ta có thể thấy

Nút 1:

- Có nhiều kết nối, như một thành viên kỳ cựu quen biết nhiều người.
 - Giữ vai trò cầu nối giữa các nhóm, như một huấn luyện viên trung gian.
 - Truyền thông tin hiệu quả, như người tổ chức sự kiện.

Nút 34 (so với nút 1):

- Có nhiều kết nối hơn một chút, giống như chủ tịch câu lạc bộ.
 - Kết nối với những người có ảnh hưởng, như nhóm nòng cốt.
 - Ít quan trọng trong việc kết nối tổng thể, giống một thành viên có uy tín nhưng ít tham gia điều phối.

Nút 17:

- Rất ít kết nối, như học viên mới chỉ tham gia vài buổi.
 - Không có vai trò cầu nối, ít tham gia hoạt động xã hội.
 - Xa với phần lớn các thành viên, giống người ít tương tác.

Nút 20:

- Có ít kết nối trực tiếp, như thành viên mới quen một nhóm nhỏ.
 - Có liên hệ với một số thành viên quan trọng.
 - Ít đóng vai trò cầu nối.
 - Có thể tiếp cận hầu hết thành viên qua các mối quan hệ gián tiếp.

Tóm lại:

Nút 1 quan trọng nhất cho kết nối toàn mạng, trong khi nút 34 mạnh về mối quan hệ cục bộ. Nút 17 gần như không có ảnh hưởng, còn nút 20 có vai trò khiêm tốn nhưng vẫn giữ một số liên kết đáng chú ý.

5.4.1.2 Xây dựng graph theo dữ liệu

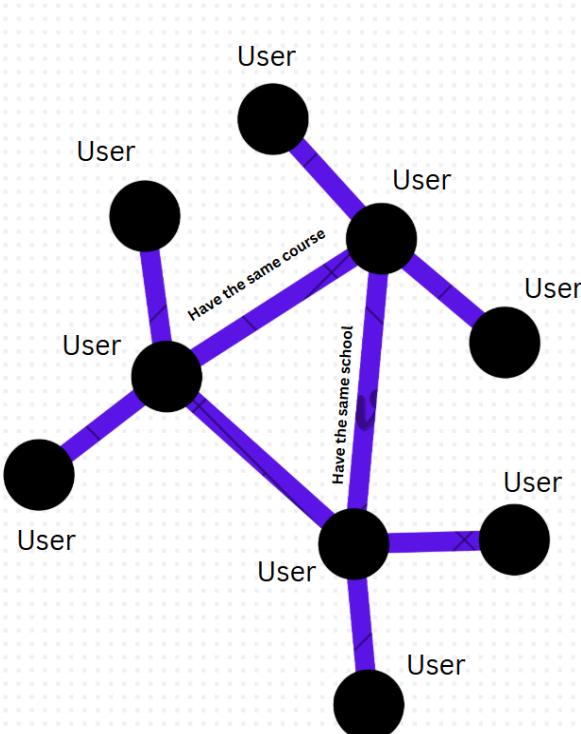
Mỗi dòng trong dataset đại diện cho một user và khóa học, và mỗi quan hệ là học viên đăng ký khóa học tương ứng.

	user_id	school	course_id	encoded_field_sum	end_year	user_month	video_count	exercise_count	ct
0	U_10000	NaN	C_2033958	100	2020.0	10.0	61	14	1
1	U_1000979	云南大学	C_947149	66	2020.0	3.0	20	14	8
2	U_1000982	云南大学	C_947149	66	2020.0	6.0	20	14	8
3	U_1001176	云南大学	C_947149	66	2020.0	3.0	20	14	8
4	U_1001563	昆明理工大学	C_735164	31	2020.0	9.0	61	44	8
5	U_1001625	昆明理工大学	C_735164	31	2020.0	10.0	61	44	8
6	U_1001694	昆明理工大学	C_735164	31	2020.0	11.0	61	44	8

5.4.1.2.1. Định nghĩa đồ thị:

Mỗi node là một user.

Các user học chung một **khóa học hoặc cùng trường** sẽ tạo một liên kết.



```

# Initialize graph
G = nx.Graph()

# Add user nodes
for user in df['user_id']:
    G.add_node(user)
print('complete creating nodes')

# Group by school
grouped_by_school = df.groupby('school')
for _, group in grouped_by_school:
    users = group['user_id'].tolist()
    for u1, u2 in combinations(users, 2):
        G.add_edge(u1, u2)
print('complete creating link about school')

# Group by course_id
grouped_by_course = df.groupby('course_id')
for _, group in grouped_by_course:
    users = group['user_id'].tolist()
    for u1, u2 in combinations(users, 2):
        G.add_edge(u1, u2)
print('complete creating link about course')

```

Phần code tạo graph từ thư viện networkx.

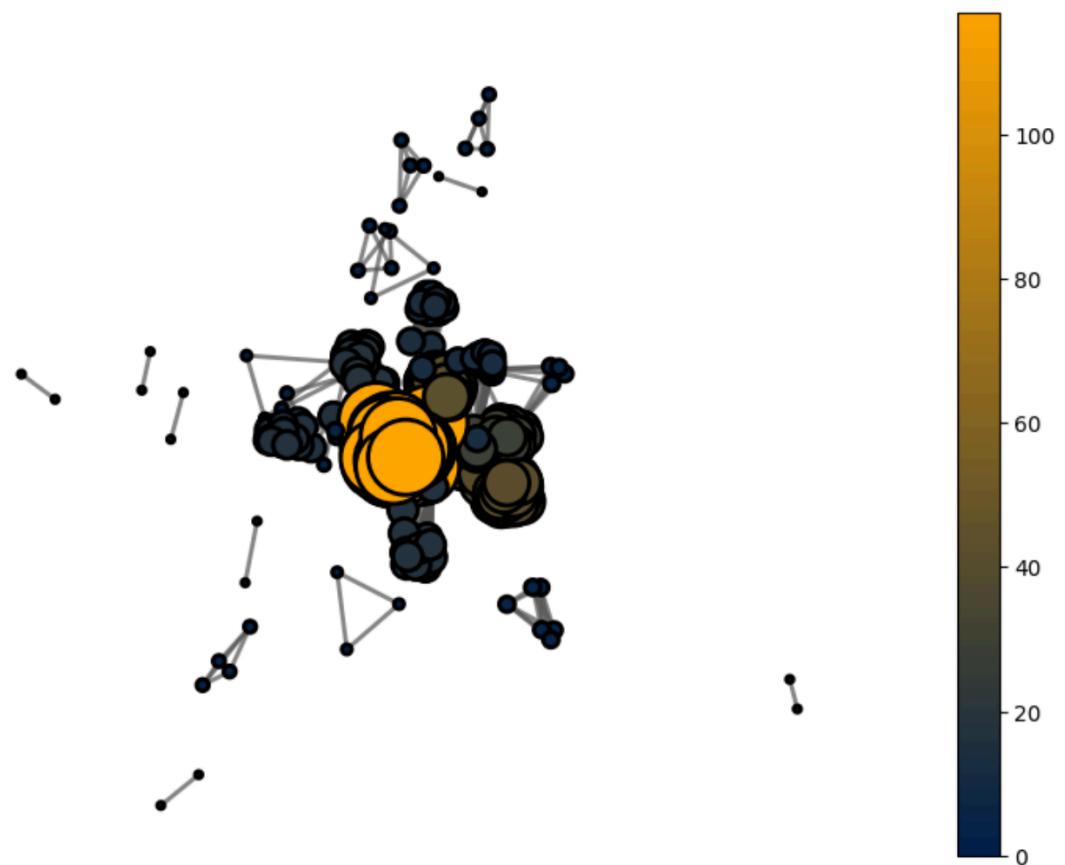
5.4.1.2.2. Biểu đồ đồ thị cho 1000 node đầu tiên

Nhận xét:

- Trong mạng lưới, có những khóa học thu hút rất nhiều học viên, nên các user node tương ứng thường tập trung ở trung tâm đồ thị, nơi có mật độ kết nối cao.
- Ngược lại, một số khóa học ít người đăng ký, dẫn đến các node học viên nằm ở rìa đồ thị, có rất ít kết nối (khoảng 2–3 liên kết) hoặc thậm chí không kết nối với phần còn lại.
- Phần lớn học viên có xu hướng đăng ký các khóa học phổ biến, vì vậy các node của họ nằm gần trung tâm, tạo thành các cụm kết nối chặt chẽ hơn trong mạng lưới.

5.4.1.2.3. Một biểu đồ về node-level và nhận xét

Node degree

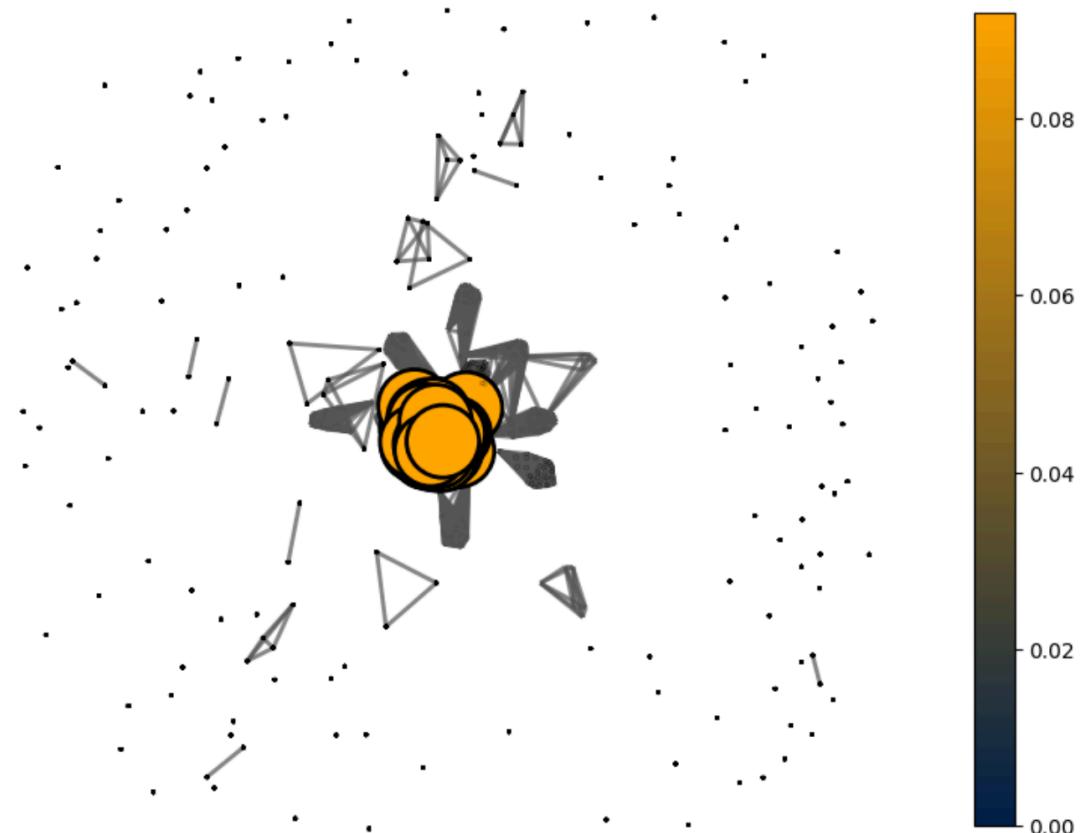


Nhận xét:

- Trong biểu đồ degree của các node, có thể thấy một số node có degree rất cao, nằm ở trung tâm đồ thị và vượt trội hẳn so với các node khác. Những node này đóng vai trò quan trọng trong việc kết nối mạng lưới.
- Các node ở rìa đồ thị chỉ có khoảng 2 đến 3 liên kết, thường thuộc về các học viên tham gia ít khóa học hoặc chọn các khóa ít người đăng ký.
- Điều này cho thấy sự phân tầng rõ rệt trong mạng lưới: một nhóm nhỏ trung tâm rất kết nối, trong khi phần còn lại nằm rải rác và có mức độ kết nối thấp.

Eigenvector centrality

Eigenvector centrality (chỉ số trung tâm riêng) là một chỉ số trong phân tích mạng (network analysis) dùng để đo tầm ảnh hưởng của một nút (node) trong mạng lưới, không chỉ dựa vào số lượng kết nối mà còn dựa vào tầm ảnh hưởng của các nút mà nó kết nối đến.

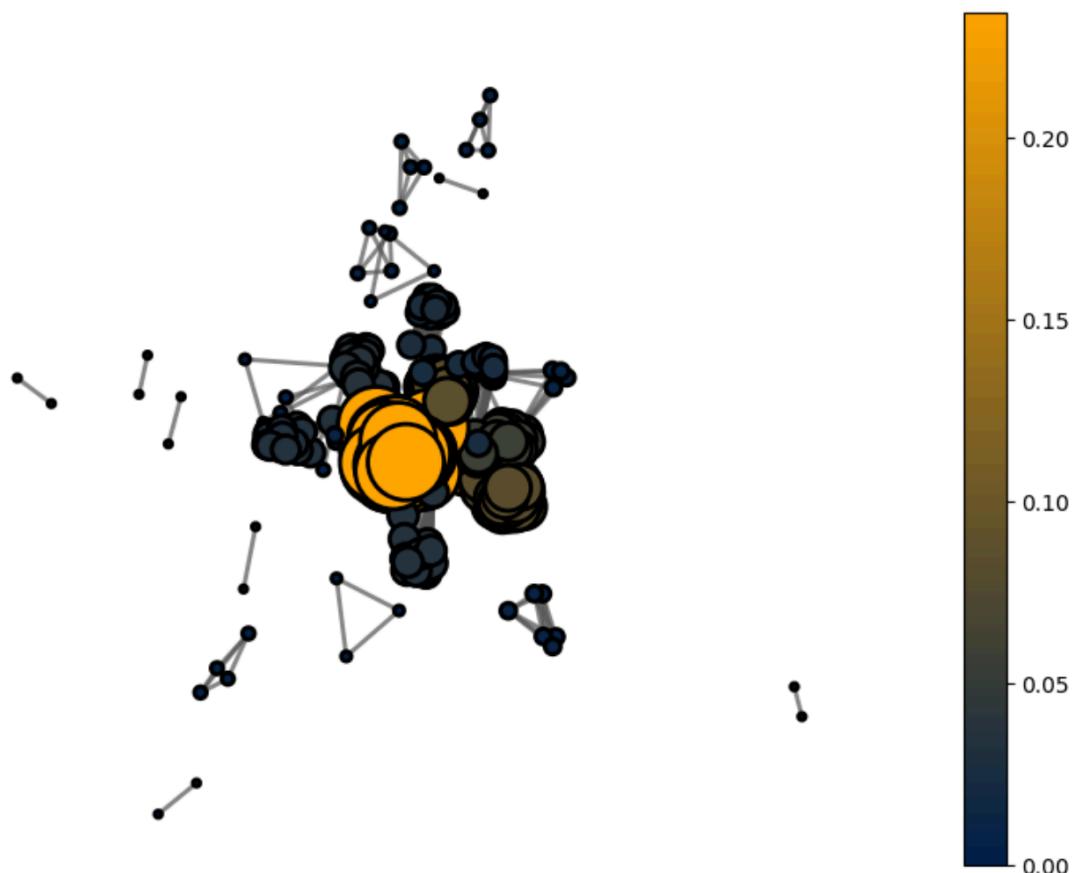


Nhận xét:

- Một số node trung tâm có giá trị eigenvector centrality cao, điều này cho thấy chúng không chỉ có nhiều kết nối, mà còn kết nối với các node cũng có ảnh hưởng cao. Những node này thường là học viên tham gia nhiều khóa học phổ biến, hoặc kết nối với những học viên khác cũng rất tích cực.
- Các node ở rìa đồ thị có giá trị eigenvector thấp, thể hiện vai trò ít quan trọng hơn trong mạng lưới — tương tự như những học viên chỉ tham gia một vài khóa ít người đăng ký, và ít kết nối với các thành viên trung tâm.
- Eigenvector centrality giúp phân biệt không chỉ số lượng kết nối, mà còn chất lượng của những kết nối, nhấn mạnh tầm quan trọng của các node trong toàn bộ cấu trúc mạng.

Centrality closeness

Closeness centrality (chỉ số trung tâm gần) là một chỉ số trong phân tích mạng lưới dùng để đo mức độ “gần gũi” trung bình của một node đến tất cả các node khác trong đồ thị.



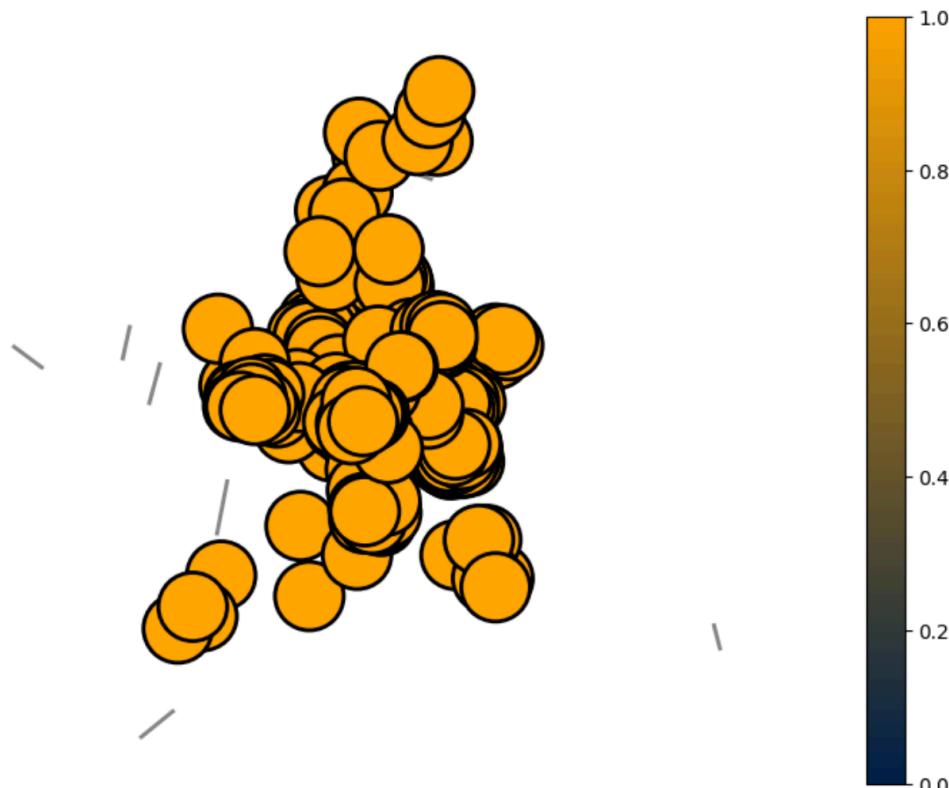
Nhận xét:

- Các node ở trung tâm đồ thị có giá trị closeness centrality cao, tương tự như với degree. Điều này phản ánh việc các node này nằm gần hầu hết các node khác trong mạng, nên việc lan truyền thông tin hoặc tương tác diễn ra nhanh chóng.
- Các node nằm ở rìa đồ thị có closeness thấp, vì chúng cách xa phần lớn các node khác, dẫn đến thời gian "đi đến" các node còn lại lâu hơn.
- Vì closeness và degree cùng phản ánh mức độ “trung tâm” của một node (một cách trực quan), biểu đồ của hai chỉ số này thường có hình dạng tương tự nhau trong mạng lưới có cấu trúc phân cụm rõ rệt.

- Tuy nhiên, closeness nhấn mạnh khoảng cách trung bình đến tất cả các node khác, chứ không chỉ đơn thuần là số lượng kết nối như degree.

Clustering

Clustering coefficient (hệ số phân cụm) là một chỉ số trong phân tích mạng lưới dùng để đo mức độ mà các nút láng giềng của một nút có xu hướng liên kết với nhau. Nó phản ánh mức độ "kết nối chặt chẽ" giữa các nút xung quanh một nút nào đó.



Nhận xét:

- Các node có clustering coefficient khá đồng đều, không có nhiều sự chênh lệch lớn giữa các node. Điều này cho thấy mạng lưới có xu hướng hình thành các nhóm nhỏ ổn định, trong đó các node kết nối với nhau cũng thường có xu hướng kết nối lẫn nhau.
- Không có node nào hoàn toàn vượt trội hoặc tụt hậu về mặt gom cụm, nghĩa là khả năng hình thành "cộng đồng nhỏ" giữa các học viên tương đối giống nhau.
- Điều này phù hợp với thực tế rằng nhiều học viên cùng đăng ký các khóa học giống nhau, tạo thành các tam giác liên kết, góp phần làm tăng giá trị clustering đồng đều.

Số liệu của các cột

	Degree Centrality	Eigenvector Centrality	Betweenness Centrality	Closeness Centrality	Clustering
U_1000979	1.000000	0.000000	0.000000	0.002004	0.000000
U_1001176	1.000000	0.000000	0.000000	0.002004	0.000000
U_10035349	117.000000	0.092057	0.000000	0.234469	1.000000
U_1004792	19.000000	0.000000	0.000000	0.038076	1.000000
U_10060293	117.000000	0.092057	0.000000	0.234469	1.000000
U_10069480	117.000000	0.092057	0.000000	0.234469	1.000000
U_10072834	0.000000	0.000000	0.000000	0.000000	0.000000
U_1008086	19.000000	0.000000	0.000000	0.038076	1.000000
U_10134100	0.000000	0.000000	0.000000	0.000000	0.000000
U_10139819	0.000000	0.000000	0.000000	0.000000	0.000000
U_10170061	42.000000	0.000001	0.000000	0.084168	1.000000
U_10170072	42.000000	0.000001	0.000000	0.084168	1.000000
U_10170744	0.000000	0.000000	0.000000	0.000000	0.000000
U_10173188	0.000000	0.000000	0.000000	0.000000	0.000000
U_10185994	0.000000	0.000000	0.000000	0.000000	0.000000
U_10187244	19.000000	0.000000	0.000000	0.038076	1.000000
U_10187291	0.000000	0.000000	0.000000	0.000000	0.000000
U_10206280	19.000000	0.000000	0.000000	0.038076	1.000000
U_10206295	19.000000	0.000000	0.000000	0.038076	1.000000
U_10227458	19.000000	0.000000	0.000000	0.038076	1.000000
U_10230032	0.000000	0.000000	0.000000	0.000000	0.000000

5.4.1.3 Tính toán trên toàn bộ dữ liệu

a) Node-Level

Vì số lượng liên kết lớn nên chỉ có thể tính được degree_centrality, eigenvector_centrality và closeness_centrality

```
# Tính số bậc (degree) của từng nút trong đồ thị G
dc = dict(G.degree())
# dc[n] là số cạnh kết nối đến nút n
print('complete dc')

# Tính chỉ số eigenvector centrality (sức ảnh hưởng của nút dựa trên sự kết nối với các nút quan trọng khác)
ec = nx.eigenvector_centrality(G)
# ec[n] càng cao → nút n càng "quan trọng" theo kiểu lan truyền ảnh hưởng
print('complete ec')

# Create a list with all the centrality values
data = [dc, ec]
indices = ['Degree Centrality', 'Eigenvector Centrality']
df = pd.DataFrame(data, index=indices)
df= df.T
```

```

# Step X: Compute closeness centrality
print("Computing closeness centrality...")
closeness_values = G.closeness()

# Map closeness centrality to user_id
user_id_to_closeness = {
    G.vs[i]['user_id']: closeness_values[i]
    for i in range(len(G.vs))
}

# Add closeness to DataFrame
df_with_cluster['closeness'] = df_with_cluster['user_id'].map(user_id_to_closeness)

```

Kết quả thực hiện.

	Degree Centrality	Eigenvector Centrality	user_id
0	122.0	5.978192e-10	U_10000
1	626.0	4.216492e-10	U_1000979
2	626.0	4.216492e-10	U_1000982
3	626.0	4.216492e-10	U_1001176
4	2894.0	2.706227e-07	U_1001413
...
04862	356.0	5.076385e-07	U_99746
04863	664.0	3.622165e-10	U_997506
04864	280.0	1.903779e-09	U_99753
04865	1350.0	4.672613e-09	U_997542
04866	442.0	5.783058e-09	U_99772

b) Cluster

Để phân cụm các node, nhóm áp dụng thuật toán **Walktrap**, đây là một dạng của **Random Walk**. Thuật toán dựa trên giả định rằng các bước đi ngẫu nhiên có xu hướng ở lại trong cùng một cộng đồng. Cụ thể, Walktrap

thực hiện các bước đi ngẫu nhiên ngắn (thường là 3–5 bước) để đo khoảng cách giữa các node, từ đó nhóm các node có hành vi tương đồng vào cùng một cộng đồng. Kết quả là mạng lưới được chia thành các cụm mà các node trong cùng cụm có mức độ kết nối cao hơn so với bên ngoài cụm.

Quá trình thực hiện bằng thư viện igraph gồm các bước sau:

1. Khởi tạo đồ thị

- Tạo một đối tượng đồ thị rỗng `igraph.Graph()`.
- Thêm các đỉnh (nodes) tương ứng với danh sách duy nhất các học viên (`user_id`).

2. Tạo danh sách cạnh (edges)

- Dữ liệu được nhóm theo `school` và `course_id` để tạo các mối liên kết giữa những học viên cùng trường hoặc cùng khóa học.
- Với mỗi nhóm, tạo các cặp học viên (edges) thông qua tổ hợp (combinations).
- Để tránh tràn bộ nhớ, chỉ xử lý những nhóm có kích thước nhỏ hơn ngưỡng cho phép (`MAX_GROUP_SIZE = 800`).

3. Thêm cạnh vào đồ thị

- Các cặp cạnh được làm sạch để loại bỏ trùng lặp và thêm vào đồ thị.

4. Thực hiện phân cụm bằng Walktrap

- Gọi phương thức `community_walktrap(steps=4)` trên đồ thị đã xây dựng.
- Biến kết quả thành các cụm bằng `as_clustering()` và lưu lại thông tin số lượng cụm phát hiện được.

5. Gán nhãn cụm cho học viên

- Ánh xạ mỗi `user_id` tới chỉ số cụm tương ứng.
- Gán thông tin này trở lại DataFrame ban đầu để phục vụ cho việc phân tích tiếp theo.

Kết quả áp dụng

	user_id	course_id	cluster	closeness
0	U_10000	C_2033958	0	0.295599
1	U_1000979	C_947149	1	0.276741
2	U_1000982	C_947149	1	0.276741
3	U_1001176	C_947149	1	0.276741
4	U_1001413	C_735164	2	NaN
...
108117	U_99746	C_674971	209	0.313803
108118	U_997506	C_2095102	245	0.274656
108119	U_99753	C_1428968	9	0.305916
108120	U_997542	C_2066096	6085	0.240988
108121	U_99772	C_1903985	9	0.319271

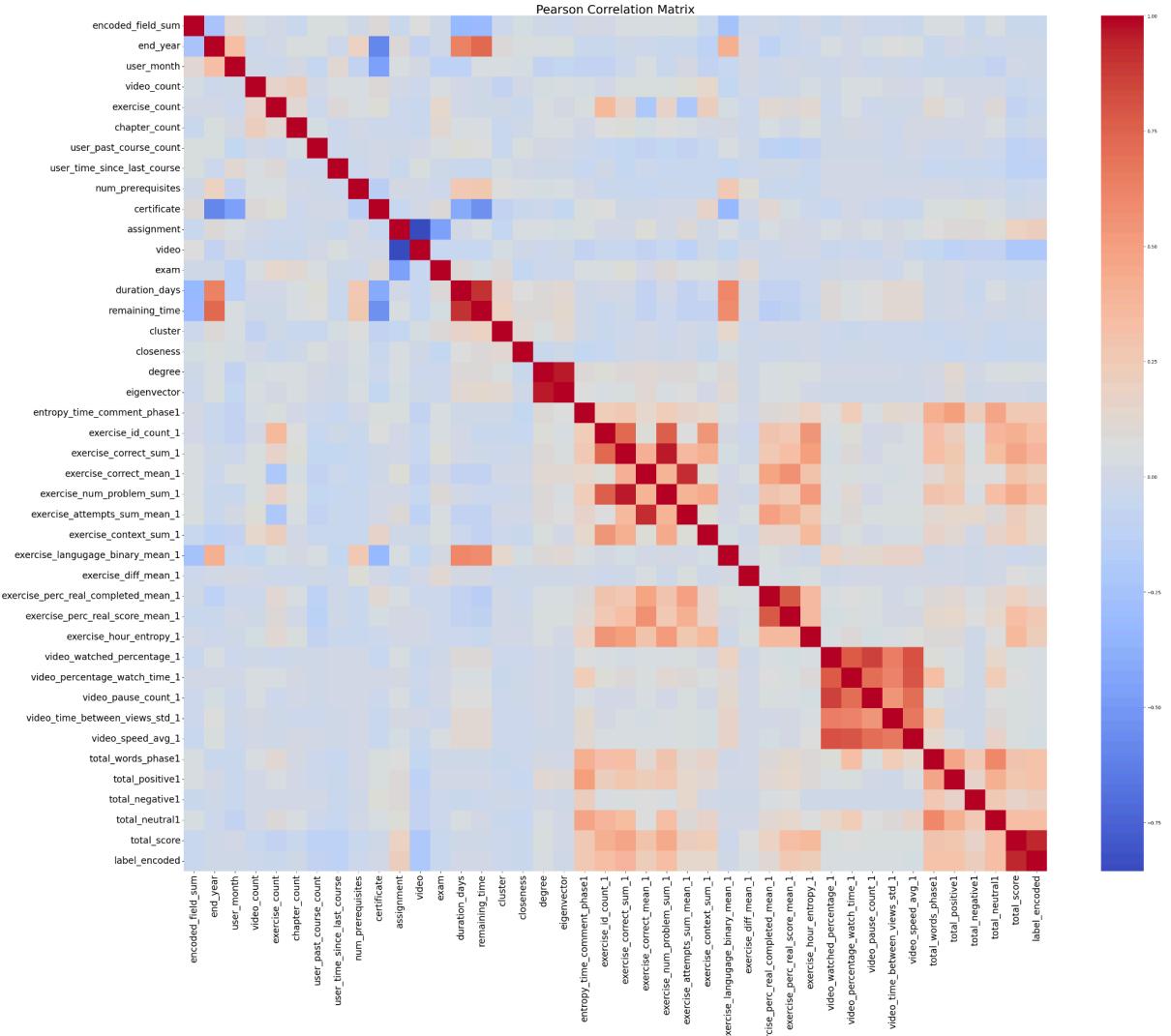
1. Gộp với dữ liệu ban đầu

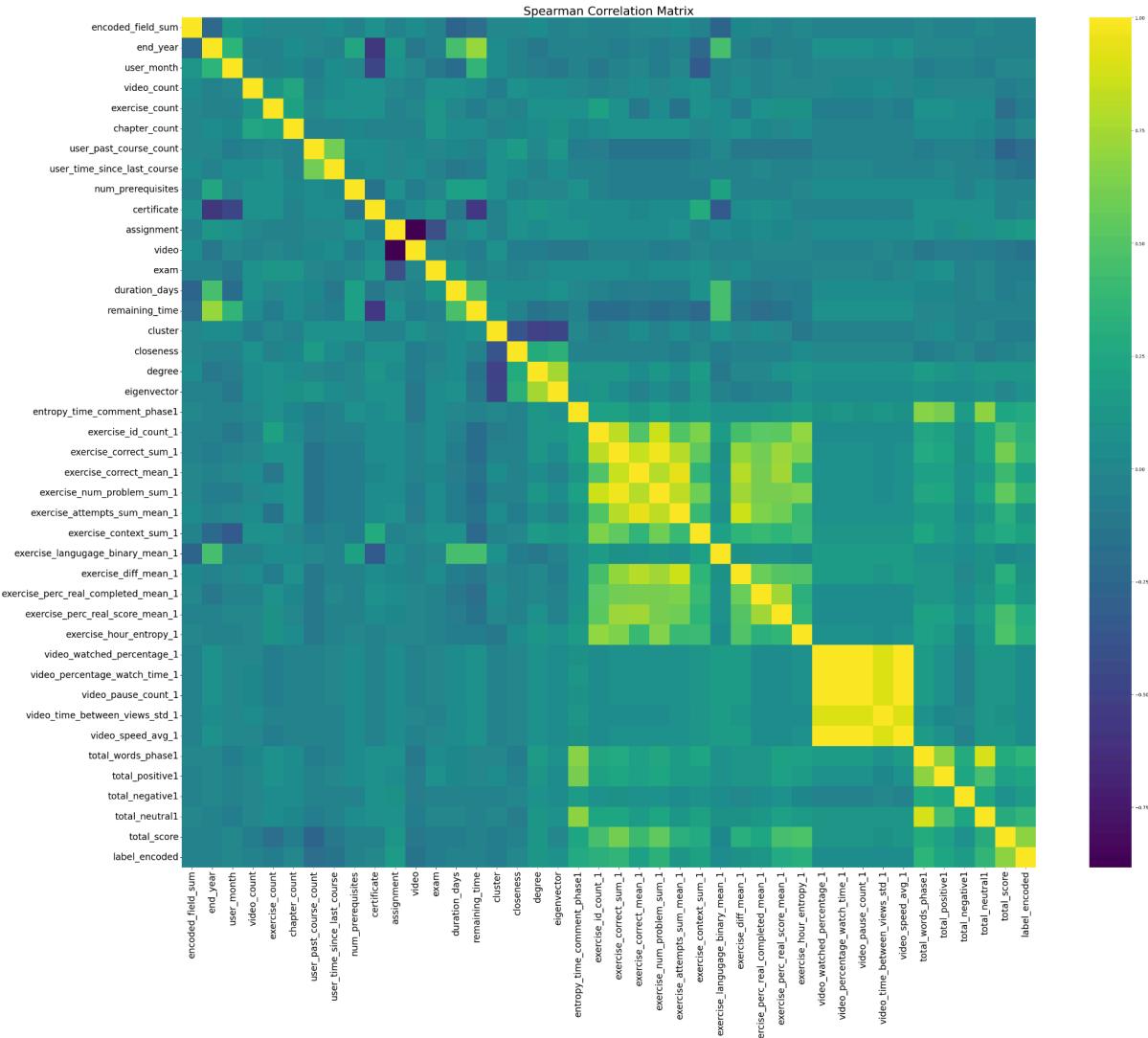
Sau khi tính toán bốn đặc trưng trung tâm của các node trong mạng học viên, bao gồm:

- Degree Centrality
- Closeness Centrality
- Eigenvector Centrality
- Clustering Coefficient

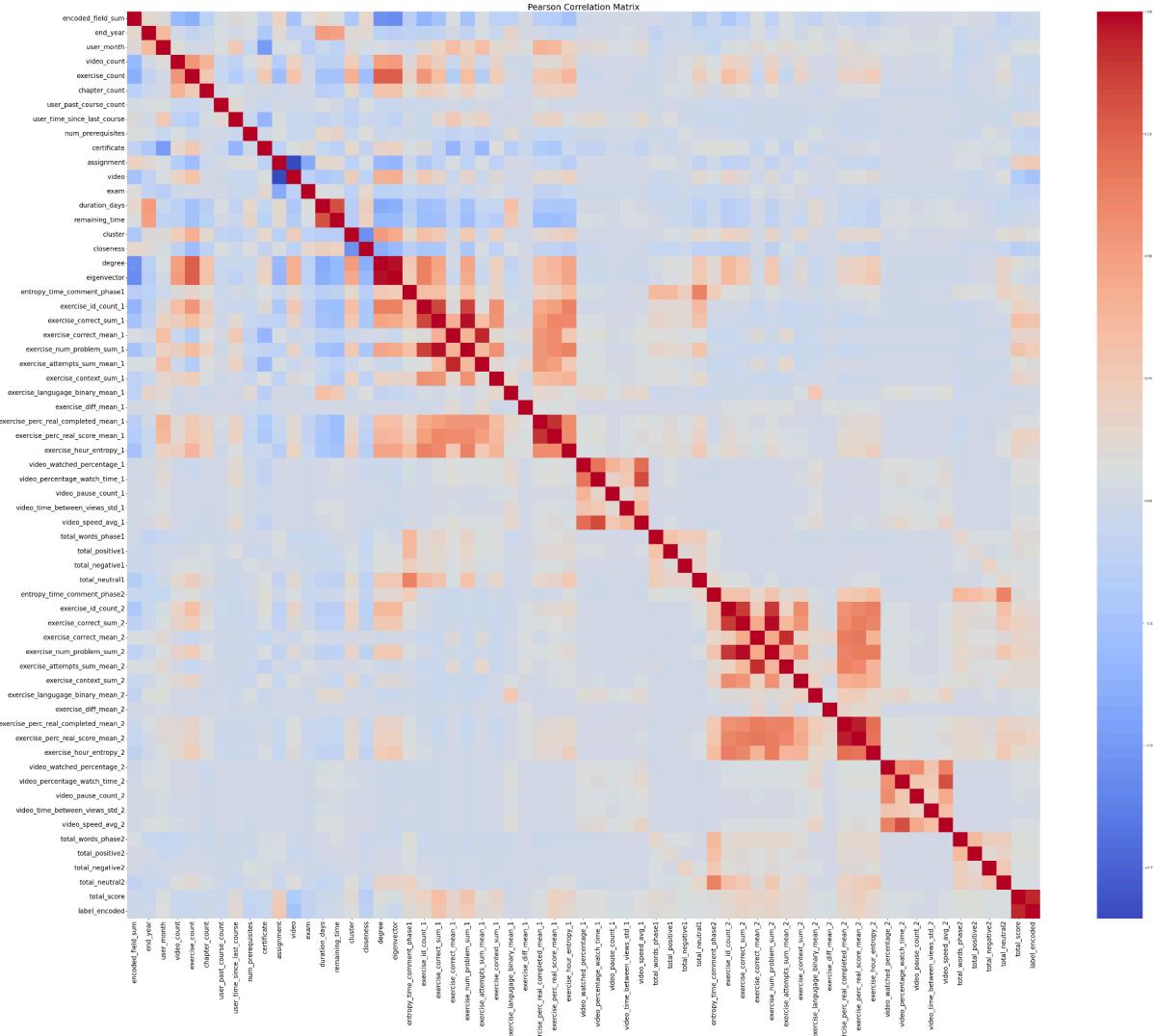
Nhóm tiến hành gộp các đặc trưng này vào cùng một bảng dữ liệu, sau đó tính hệ số tương quan giữa các đặc trưng này với nhãn cộng đồng (cluster) được gán từ thuật toán Walktrap.

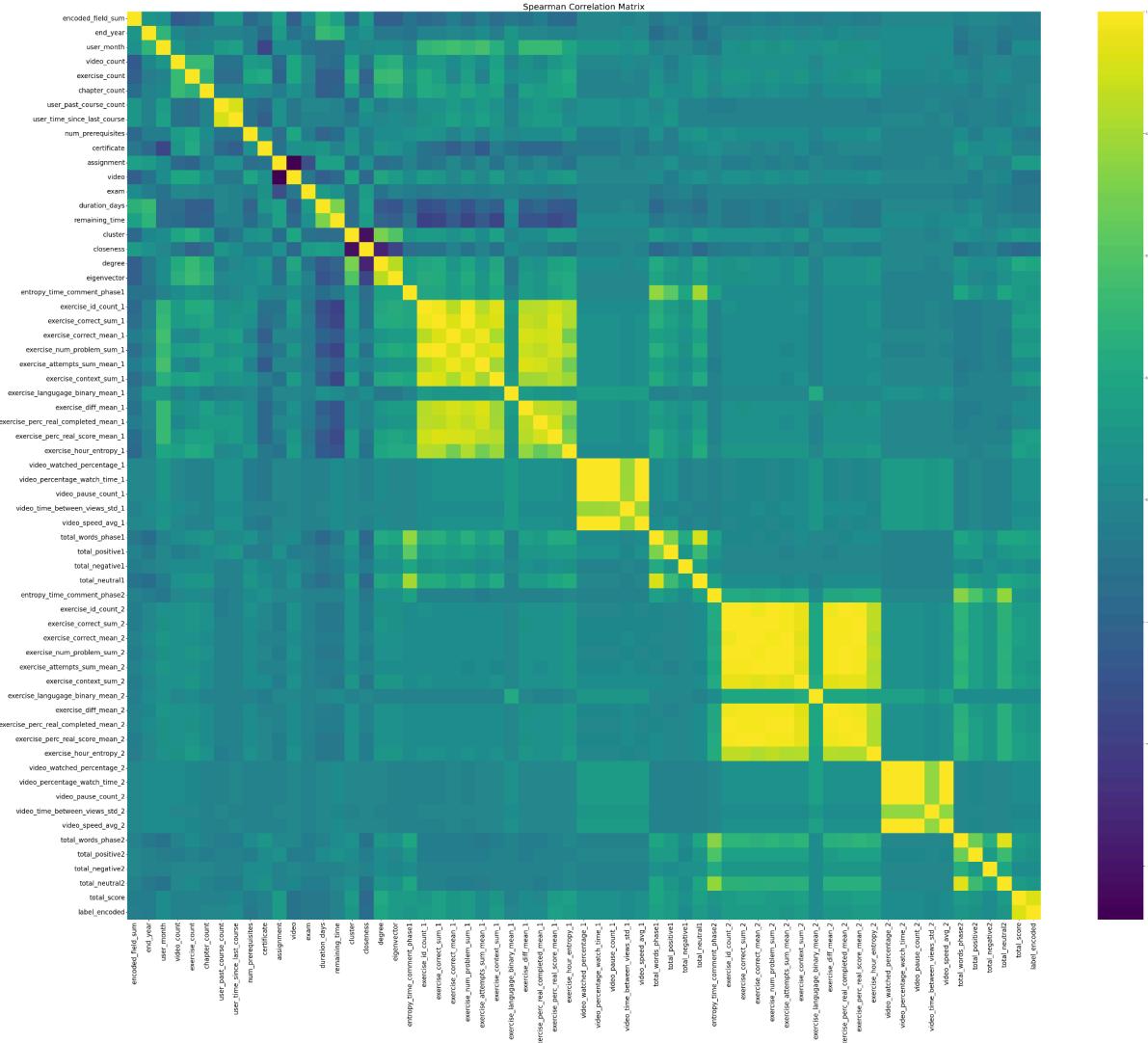
Hệ số tương quan với Week 1, 2



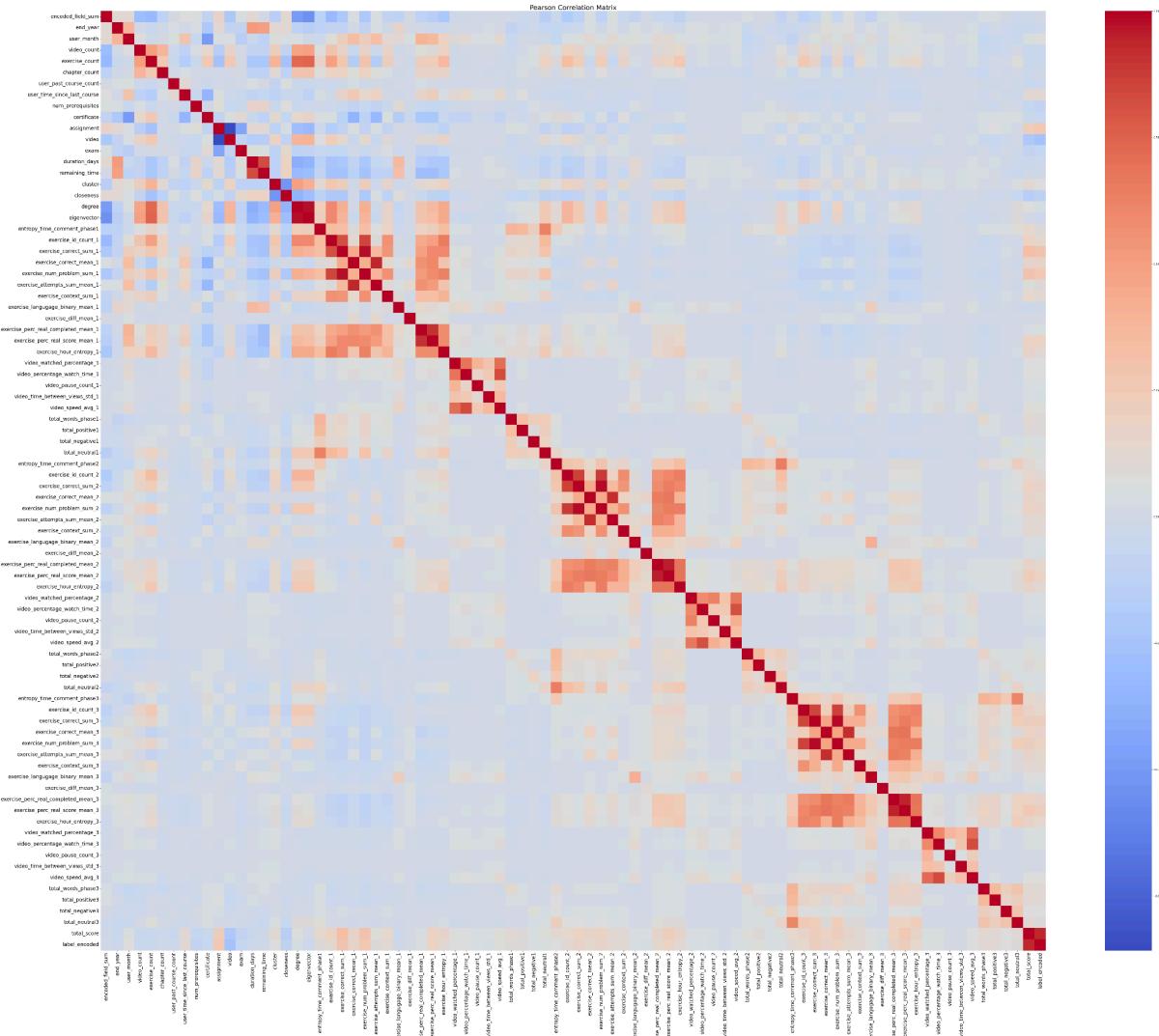


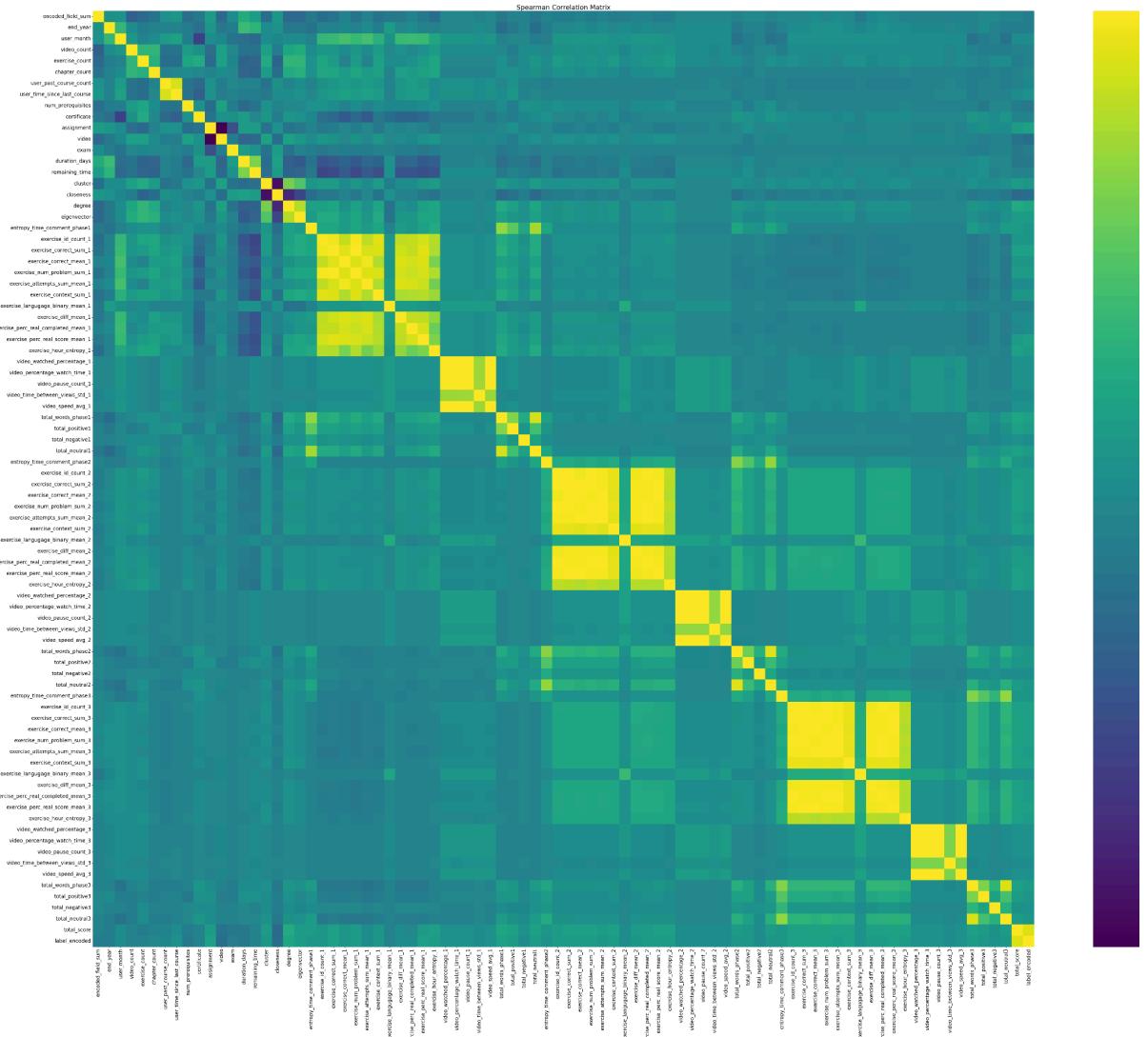
Hệ số tương quan với Week 3, 4



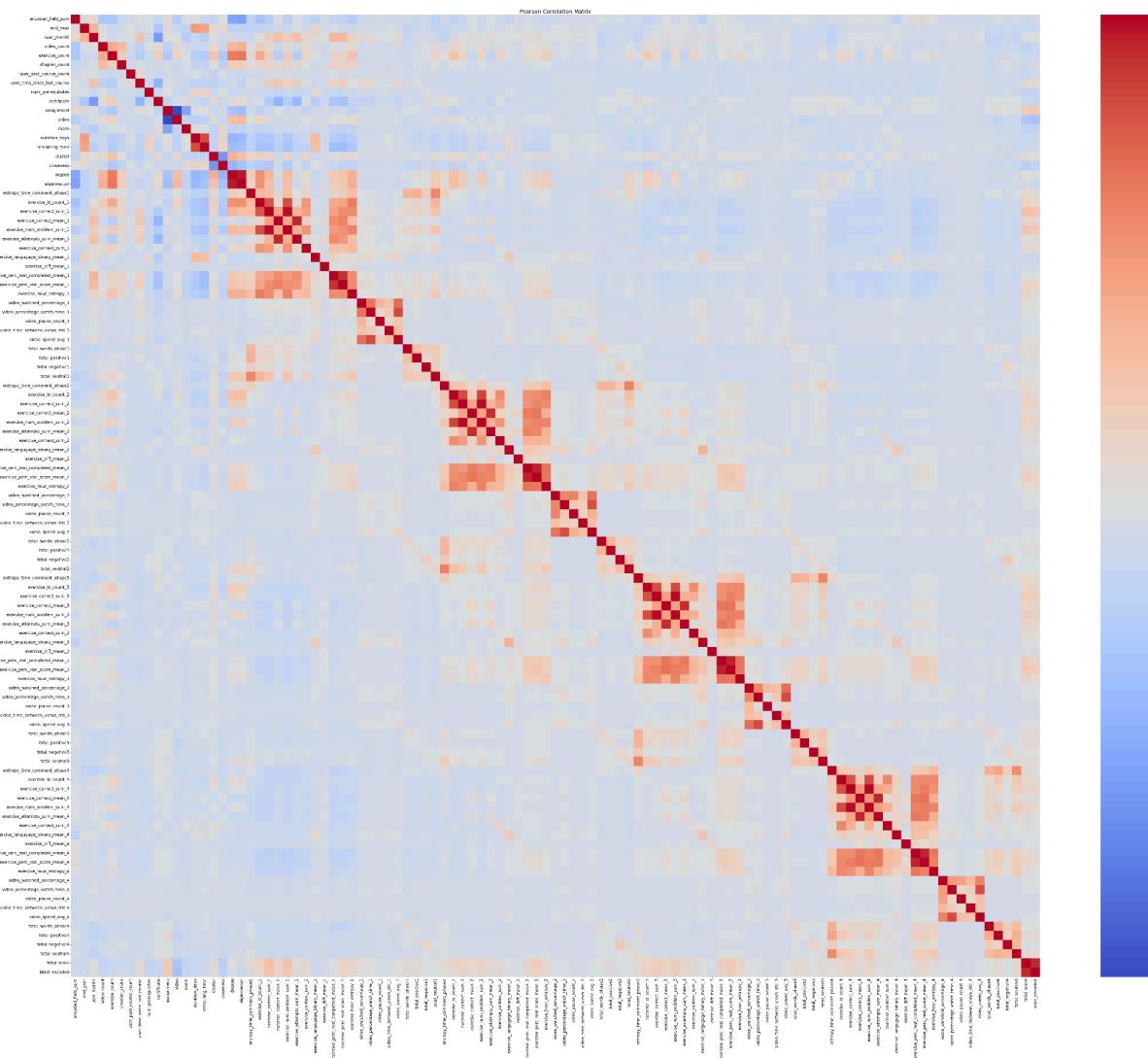


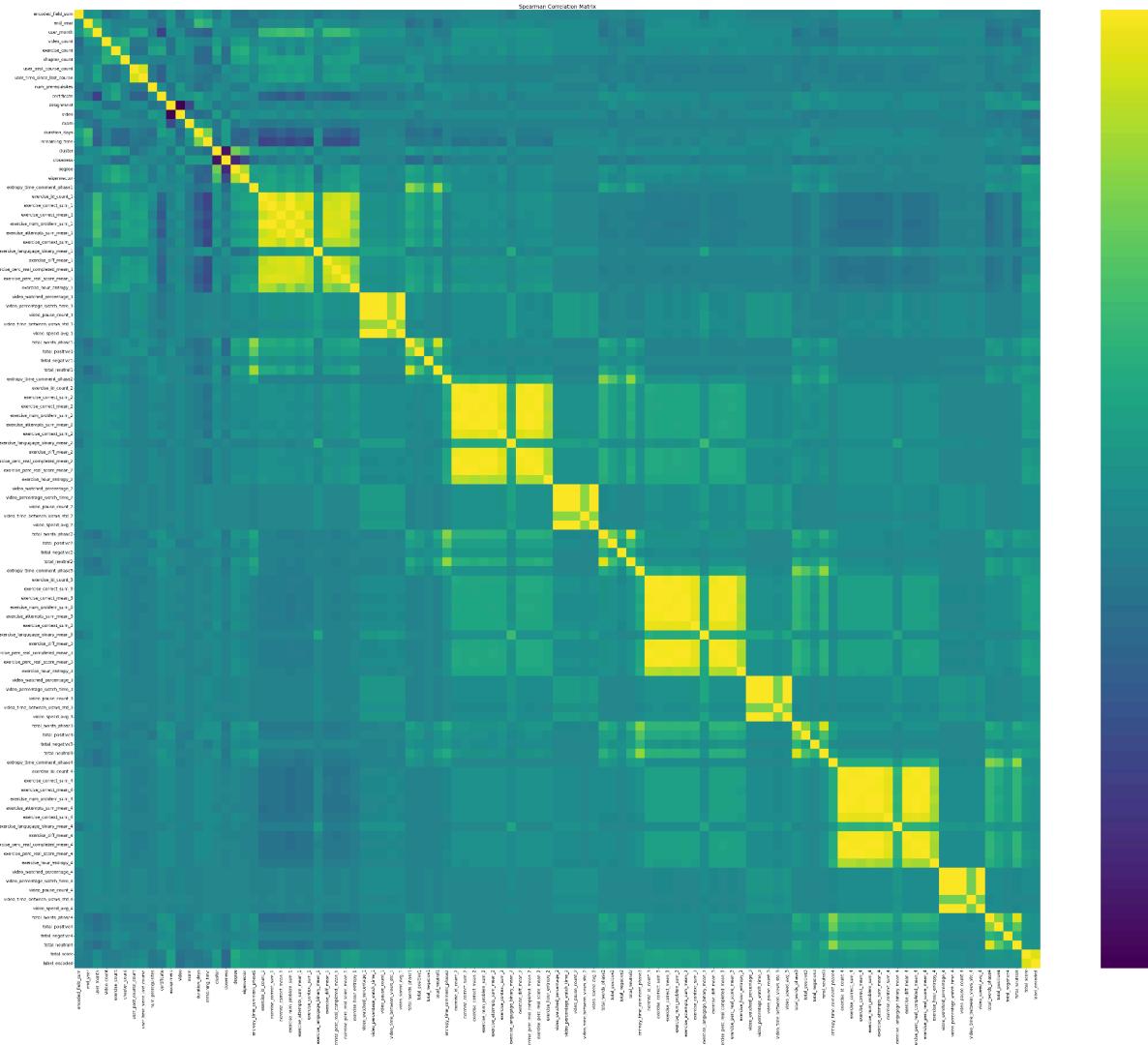
Hệ số tương quan với Week 5, 6





Hệ số tương quan với Week 7, 8





Nhận xét:

- Degree centrality: Hệ số tương quan gần 0 → số lượng kết nối không ảnh hưởng rõ rệt đến nhãn.
- Closeness centrality: Hệ số tương quan âm và lớn nhất về độ lớn → node càng gần trung tâm thì càng dễ thuộc một cụm cụ thể và có thể dự đoán hiệu quả.
- Eigenvector centrality: Hệ số tương quan âm, nhỏ hơn closeness → ảnh hưởng từ các node quan trọng có tác động yếu đến nhãn.
- Clustering coefficient: Hệ số tương quan âm và nhỏ nhất → mức độ liên kết với nhãn theo chiều ngược.

5.4.2 Trích xuất đặc trưng từ đồ thị (graph) sử dụng node2vec

5.4.2.1 Định nghĩa đồ thị trong node2vec

Node2Vec là một thuật toán học biểu diễn (embedding) cho các đỉnh trong đồ thị, giúp biến mỗi node thành một vector số thực có chiều cố định. Thuật

toán này mở rộng từ ý tưởng của Word2Vec trong xử lý ngôn ngữ tự nhiên, bằng cách sử dụng các bước đi ngẫu nhiên (random walks) trên đồ thị để học ngữ cảnh (context) của từng node.

Node2Vec kết hợp cả hai chiến lược khám phá đồ thị:

- **DFS (depth-first search)** – ưu tiên đi sâu, giúp học được các đặc điểm cấu trúc
- **BFS (breadth-first search)** – ưu tiên lan rộng, giúp học được các đặc điểm theo chiều ngữ nghĩa (tức là node gần nhau trong cùng cộng đồng).

Nhờ đó, Node2Vec có khả năng linh hoạt trong việc học biểu diễn, phù hợp cho các bài toán như: phân cụm, phân loại node, hoặc dự đoán liên kết (link prediction).

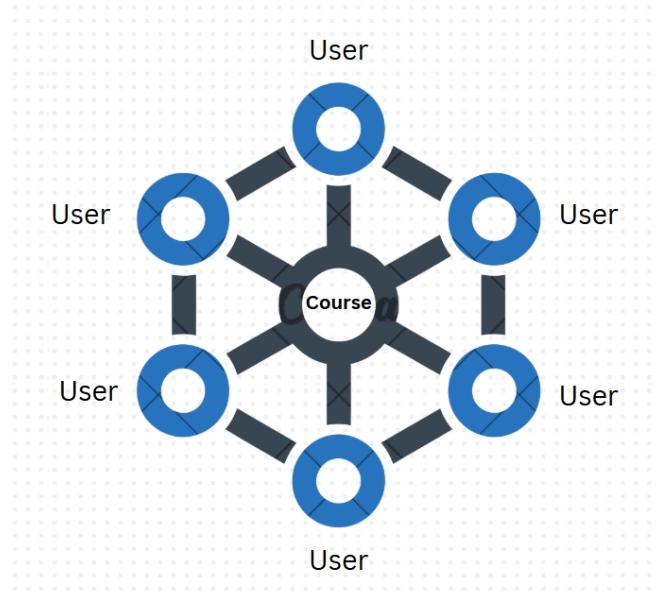
Định nghĩa graph trong dữ liệu:

node:

- Người dùng (user) có thuộc tính school_encoded.
- Thêm node khóa học (course) có các đặc trưng: thời gian kéo dài, số tài liệu khóa học, thành phần điểm...

Thêm cạnh:

- Mỗi cạnh nối giữa một user và một course thể hiện việc đăng ký.
- Trọng số của cạnh là trung bình của các đặc trưng tương tác: thời gian học còn lại của user, số bài tập làm trong tuần, số video đã xem trong tuần.



5.4.2.2 Các bước thực hiện

1. Chuẩn bị dữ liệu

- Lấy các đặc trưng:
 - User: chỉ sử dụng trường school.
 - Course: bao gồm nhiều đặc trưng như video_count, exercise_count, certificate, duration_days,...

2. Mã hóa và chuẩn hóa

- Mã hóa cột school bằng LabelEncoder.
- Chuẩn hóa các đặc trưng tương tác (interaction features) giữa user và course bằng StandardScaler.

3. Xây dựng đồ thị hai lớp (bipartite graph)

- Tạo graph vô hướng bằng networkx.
- **Thêm node:**
 - Thêm node người dùng (user) và gán thuộc tính school_encoded.
 - Thêm node khóa học (course) và gán toàn bộ đặc trưng.
- **Thêm cạnh:**
 - Mỗi cạnh nối giữa một user và một course.
 - Trọng số của cạnh là **trung bình** của các đặc trưng tương tác (đã chuẩn hóa).

4. Huấn luyện mô hình Node2Vec: Sử dụng thư viện node2vec để huấn luyện embedding.

- Số chiều: 16
- Chiều dài bước đi: 20
- Số lần bước đi: 100

5. Huấn luyện xong, trích xuất embedding cho từng node (user và course).

```

# 🔎 Chọn các đặc trưng node cho user và course
user_features = ['school']
course_features = ['encoded_field_sum', 'video_count', 'exercise_count', 'chapter_count',
                   'num_prerequisites', 'certificate', 'assignment', 'video', 'exam', 'duration_days']

# 🌈 Xử lý đặc trưng: ánh xạ node -> đặc trưng
user_nodes = df[['user_id']] + user_features].drop_duplicates().set_index('user_id')
course_nodes = df[['course_id']] + course_features].drop_duplicates().set_index('course_id')

label_encoder = LabelEncoder()
user_nodes['school_encoded'] = label_encoder.fit_transform(user_nodes['school'])

print("👤 Số lượng user nodes:", len(user_nodes))
print("📚 Số lượng course nodes:", len(course_nodes))

# 🚧 Tạo đồ thị hai lớp user-course
G = nx.Graph()

# 🟦 Thêm user node và thuộc tính
for uid, attrs in user_nodes.iterrows():
    G.add_node(uid, bipartite='user', school=attrs['school_encoded'])

# 🟨 Thêm course node và thuộc tính
for cid, attrs in course_nodes.iterrows():
    G.add_node(cid, bipartite='course', **attrs.to_dict())

# ✖️ Tính thêm đặc trưng tỷ lệ thời gian còn lại
df['ratio_remaining_time'] = df['remaining_time'] / df['duration_days']

```

```

# ✨ Tạo các cạnh giữa user và course
interaction_features = [
    'ratio_remaining_time', 'exercise_id_count_1',
    'video_watched_percentage_1', 'exercise_id_count_2',
    'video_watched_percentage_2', 'exercise_id_count_3',
    'video_watched_percentage_3'
]

# 📈 Chuẩn hóa trọng số
scaler = StandardScaler()
df[interaction_features] = scaler.fit_transform(df[interaction_features])

# Thêm cạnh với trọng số
for _, row in df.iterrows():
    user = row['user_id']
    course = row['course_id']
    weight = row[interaction_features].mean()
    G.add_edge(user, course, weight=weight)

print("🔗 Tổng số cạnh trong đồ thị:", G.number_of_edges())

# 🌈 Trích xuất node embedding với Node2Vec
print("🚀 Đang huấn luyện Node2Vec...")
node2vec = Node2Vec(G, dimensions=16, walk_length=20, num_walks=100, workers=2)
model = node2vec.fit(window=5, min_count=1, batch_words=4)
print("✅ Huấn luyện Node2Vec hoàn tất.")

# 📁 Lưu embedding cho các node
user_embeddings = {node: model.wv[node] for node in G.nodes if G.nodes[node]['bipartite'] == 'user'}
course_embeddings = {node: model.wv[node] for node in G.nodes if G.nodes[node]['bipartite'] == 'course'}

```

5.4.2.3 Áp dụng vào dữ liệu

Dữ liệu đã nạp: (94917, 84)
 Số lượng user nodes: 92309
 Số lượng course nodes: 1004
 Tổng số cạnh trong đồ thị: 94917
 Đang huấn luyện Node2Vec...
 Computing transition probabilities: 0% | 0/93121 [00:00<?, ?it/s]
 Generating walks (CPU: 1): 100%|██████████| 50/50 [2:59:54<00:00, 215.90s/it]
 Generating walks (CPU: 2): 100%|██████████| 50/50 [3:01:18<00:00, 217.57s/it]
 Huấn luyện Node2Vec hoàn tất.

Embedding vector (10 chiều đầu) của user U_10000:
 [-1.0050842 -0.01007318 0.06075627 3.272508 0.38189492 2.3635266
 2.5637136 -1.1008487 -0.7529547 -0.30851033]

Số lượng embedding user: 92309
 Số lượng embedding course: 812

```
#  Ví dụ: xem embedding của một user
import numpy as np
example_user = list(user_embeddings.keys())[0]
print("Embedding vector for", example_user, ":", user_embeddings[example_user][:10]) # In 10 chiều đầu
```

```
#  Tùy chọn lưu embedding ra file
np.save('user_embeddings.npy', user_embeddings)
np.save('course_embeddings.npy', course_embeddings)
```

Embedding vector for U_10000 : [-1.0050842 -0.01007318 0.06075627 3.272508 0.38189492 2.3635266
 2.5637136 -1.1008487 -0.7529547 -0.30851033]

Vector cho Khóa học

	df_course																e
	course_id	emb_0	emb_1	emb_2	emb_3	emb_4	emb_5	emb_6	emb_7	emb_8	emb_9	emb_10	emb_11	emb_12	emb_13	emb_14	e
0	C_2033958	0.858296	-2.259510	2.267689	0.792780	2.725045	2.509172	3.448677	-3.386860	1.030135	0.797164	-0.456586	1.490971	2.258027	0.497191	-0.587722	0.2
1	C_947149	-0.877333	-1.556073	0.290345	1.536970	0.434726	0.291444	4.254068	-1.839304	-0.138145	2.155200	-1.154379	0.211502	1.141510	-1.237395	-1.385793	0.8
2	C_735164	0.327803	1.134930	0.480028	-0.461845	-0.516042	1.059514	2.923801	-1.175270	-0.415448	-0.310838	-0.306585	1.976786	-0.715407	-0.093581	-0.104566	-0.8
3	C_1756056	-0.234696	-4.231102	0.370828	2.997360	2.047443	1.304811	3.863293	0.279771	-1.207274	-1.257175	-1.977746	1.625883	0.421943	-0.469078	0.684499	-2.7
4	C_697684	1.168111	-1.700410	0.133381	1.002417	1.263747	4.505279	1.606661	-0.704300	-0.771614	1.419315	-1.766249	1.220052	1.992211	-2.334550	-0.060645	1.5
...	
816	C_697113	-2.272366	0.531843	3.417508	0.933232	-1.862898	0.580870	2.980646	0.449749	-1.212520	1.297862	-0.180204	2.724187	0.582272	-1.017097	-0.974532	-2.1
817	C_735149	0.879915	-1.517423	3.568628	0.207828	0.733987	1.961583	2.542677	-3.729804	0.490308	-0.460351	0.389113	1.464014	0.474292	-0.320901	-0.873557	-2.8
818	C_1765598	-1.115783	-1.270941	1.366001	-0.814988	0.233432	1.547510	1.690097	-1.493290	1.067943	4.153324	-0.744561	2.877859	0.572733	-0.155277	0.759994	-0.5
819	C_1886691	-1.082884	-0.569750	0.919894	-2.108854	0.700153	2.526231	2.035679	-1.301150	-0.329866	3.157670	-1.396172	3.150083	2.437744	-2.061336	0.735269	-0.7
820	C_2342500	-0.329111	-0.188912	3.174777	0.419958	1.584866	2.136133	4.256826	-2.845022	-0.951032	-0.765964	0.059686	0.316666	0.986403	-0.046712	2.160556	-0.9

821 rows x 17 columns

Vector cho Học viên

	df_user																
	user_id	emb_0	emb_1	emb_2	emb_3	emb_4	emb_5	emb_6	emb_7	emb_8	emb_9	emb_10	emb_11	emb_12	emb_13	emb_14	
0	U_10000	0.484648	-2.172488	1.716664	0.518205	2.857950	2.647051	3.139796	-2.913437	1.019983	0.674399	-0.319527	1.315166	1.968394	0.204648	-0.679044	0
1	U_1000979	-0.836015	-1.296981	0.261788	1.314256	0.397687	0.000087	3.834745	-1.485096	-0.165775	1.932443	-1.029372	0.450575	1.025151	-1.131691	-1.269689	0
2	U_1000982	-0.858686	-1.301018	0.278843	1.303815	0.385640	-0.014352	3.803696	-1.465822	-0.162685	1.908392	-0.956275	0.395386	1.022861	-1.136624	-1.242907	0
3	U_1001176	-0.864422	-1.291507	0.259360	1.314940	0.430388	-0.050801	3.838549	-1.441134	-0.116962	1.921420	-1.042625	0.422156	1.053612	-1.143106	-1.352405	0
4	U_1001413	0.447507	0.951417	0.366323	-0.019131	-0.218109	0.690359	2.752187	-1.177387	-0.315007	-0.252154	-0.106760	1.999190	-0.674657	-0.397123	-0.152293	0
...	
104862	U_99746	0.452650	0.906071	1.340526	1.207695	1.048234	0.914246	2.803742	0.708649	-0.889764	2.127321	-0.824944	1.018501	-0.635324	0.957900	-1.689107	-0
104863	U_997506	-0.400324	-1.347081	1.818064	0.565719	-0.065988	0.029352	0.902466	-0.601073	-0.866339	0.308977	0.390661	0.629472	1.074616	-1.363189	-0.956313	-0
104864	U_99753	-1.936810	-2.308108	1.803795	0.545969	0.871372	0.881345	1.187140	-1.372941	-0.482321	1.913053	1.207798	0.794394	0.354038	0.113728	-1.007898	-1
104865	U_997542	-0.533358	-0.538238	1.445431	-1.081013	0.829754	1.097654	2.580750	-1.455622	0.126730	0.596833	-0.887476	2.053346	1.189067	0.626851	-0.906933	1
104866	U_99772	-0.620316	-1.260146	0.105414	0.280760	1.063897	-0.032342	1.688811	0.505979	-0.621201	0.879756	0.246840	1.077022	0.262911	-1.558170	0.817814	0

104867 rows x 17 columns

Lặp lại tương tự cho 4 file theo 2 tuần tương ứng

5.4.2.4 Gộp đặc trưng với dữ liệu

Sau khi có được các đặc trưng node cho học viên và khóa học.

Nhóm tiến hành gộp các đặc trưng này vào cùng một bảng dữ liệu, sau đó tính hệ số tương quan giữa các đặc trưng này với nhãn.

Khi gộp sẽ duyệt qua từng user và có vector của user đó. Tiếp theo là tìm vector của course trong hàng đó. Sau khi có 2 vector sẽ cộng lại tương ứng với chỉ số vector

```
# Merge embeddings
if 'user_id' in df.columns and 'course_id' in df.columns:
    df = df.merge(df_user, on='user_id', how='left')
    df = df.merge(df_course, on='course_id', how='left')

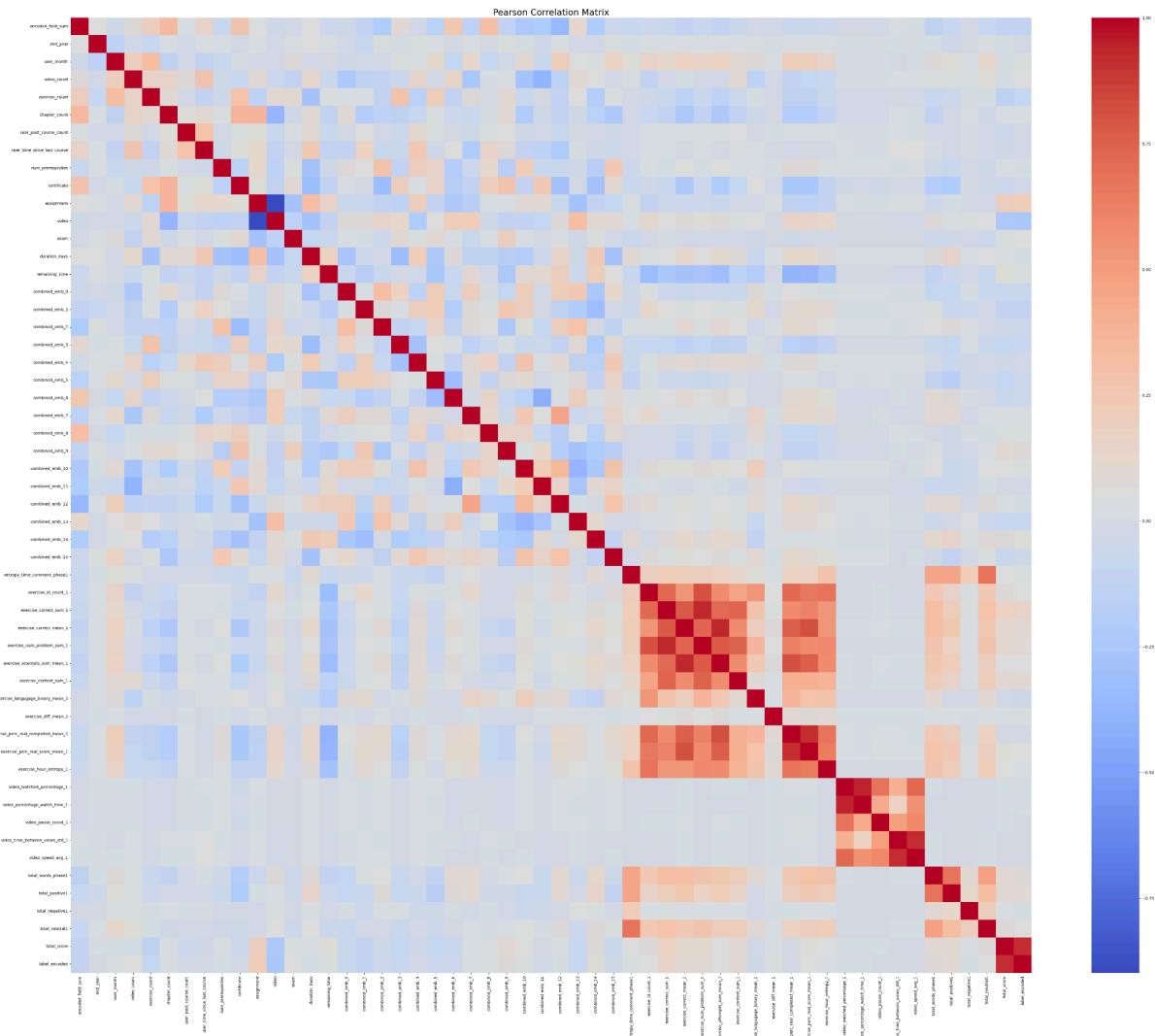
    # Combine embeddings by sum
    user_emb_cols = [col for col in df.columns if col.startswith('user_emb_')]
    course_emb_cols = [col for col in df.columns if col.startswith('course_emb_')]
    combined_emb_cols = [f'combined_emb_{i}' for i in range(len(user_emb_cols))]

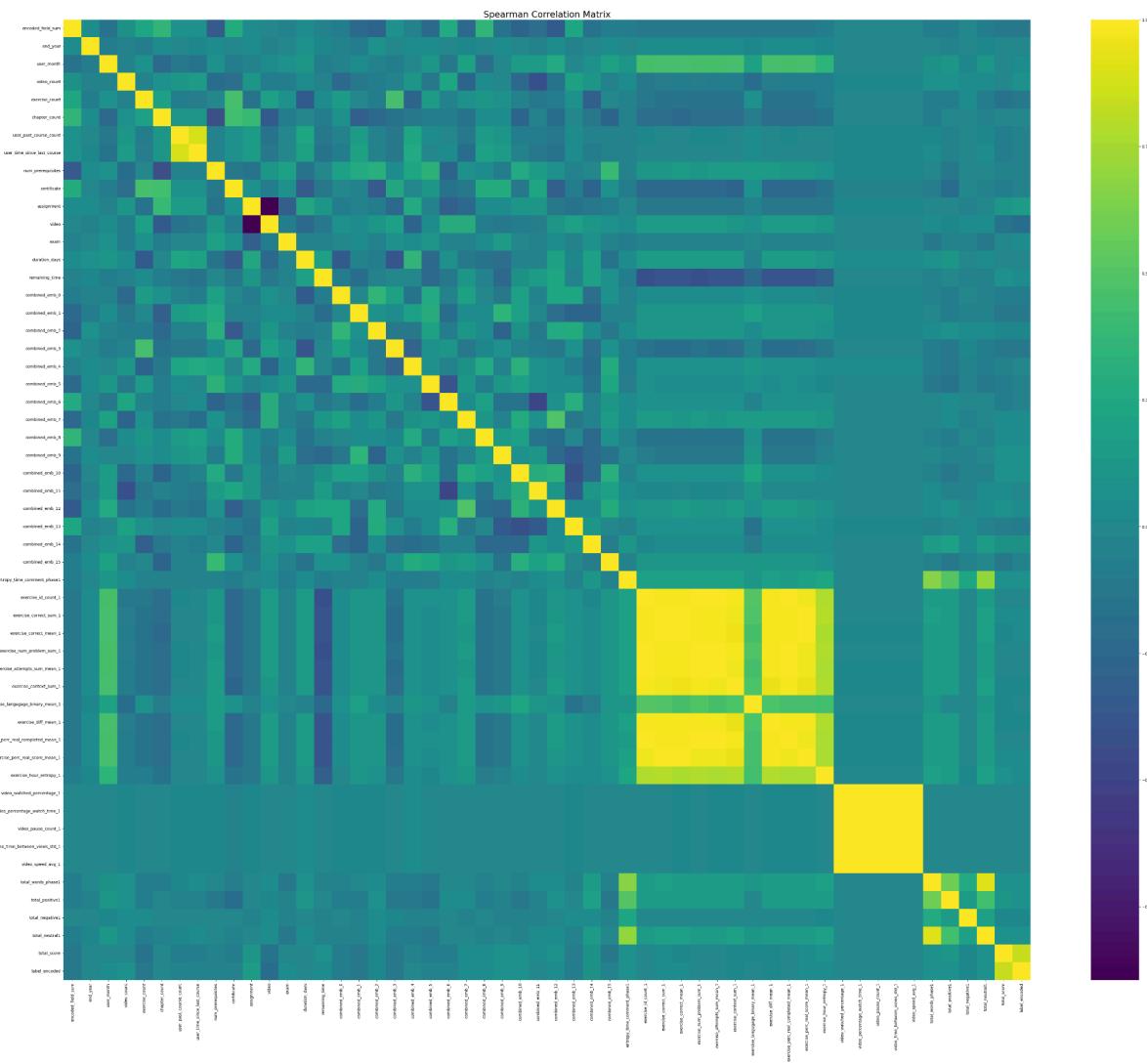
    df[combined_emb_cols] = df[user_emb_cols].values + df[course_emb_cols].values

    # Drop rows with unnecessary embeddings
    embedding_cols = [col for col in df.columns if col.startswith('user_emb_') or col.startswith('course_emb_')]
    # print("Embedding columns:", embedding_cols)
    df.drop(columns=embedding_cols, inplace=True)

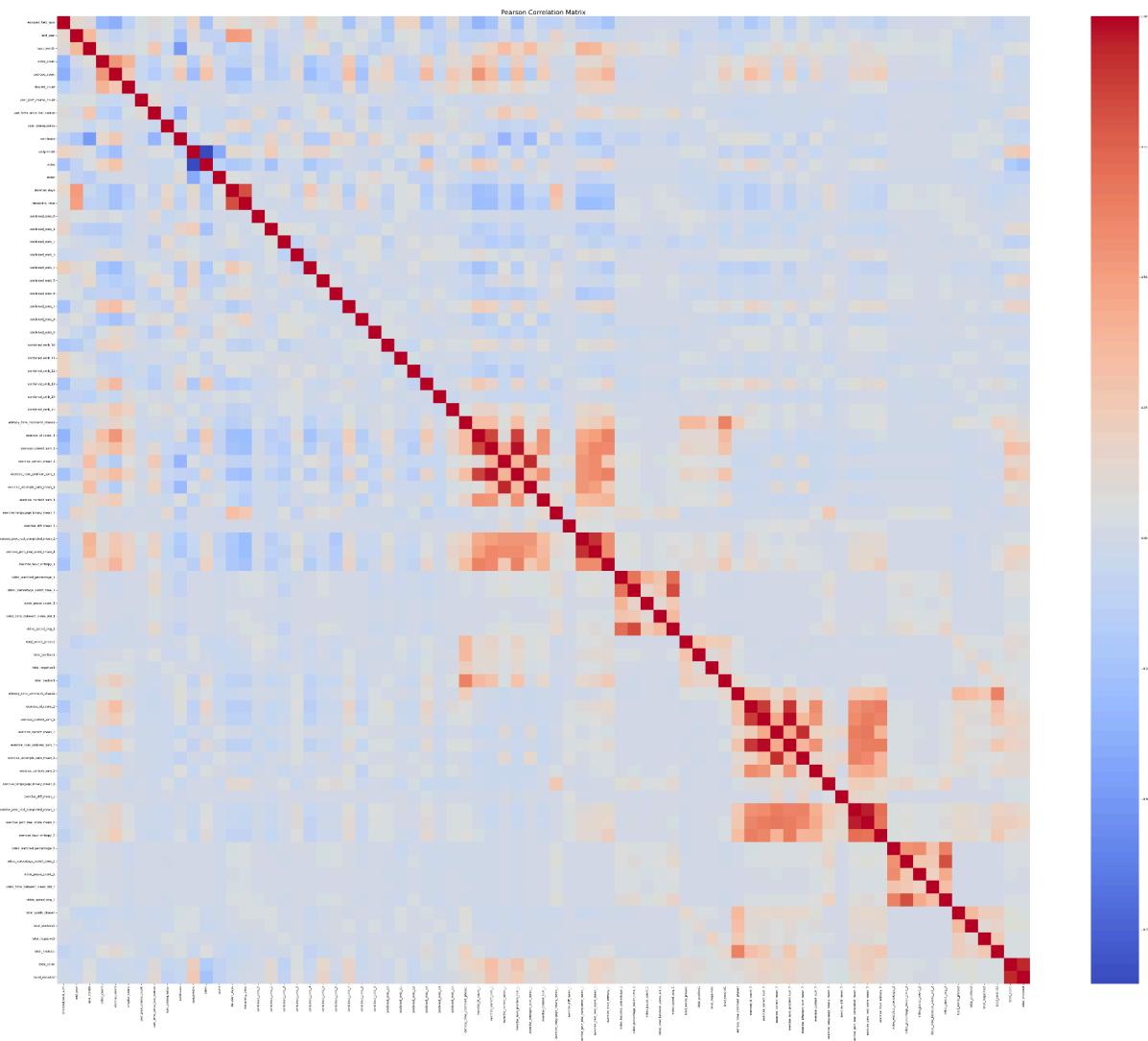
    # Insert after label column
    if 'remaining_time' in df.columns:
        label_idx = df.columns.get_loc('remaining_time')
        cols = df.columns.tolist()
        for col in reversed(combined_emb_cols):
            cols.insert(label_idx + 1, cols.pop(cols.index(col)))
        df = df[cols]
```

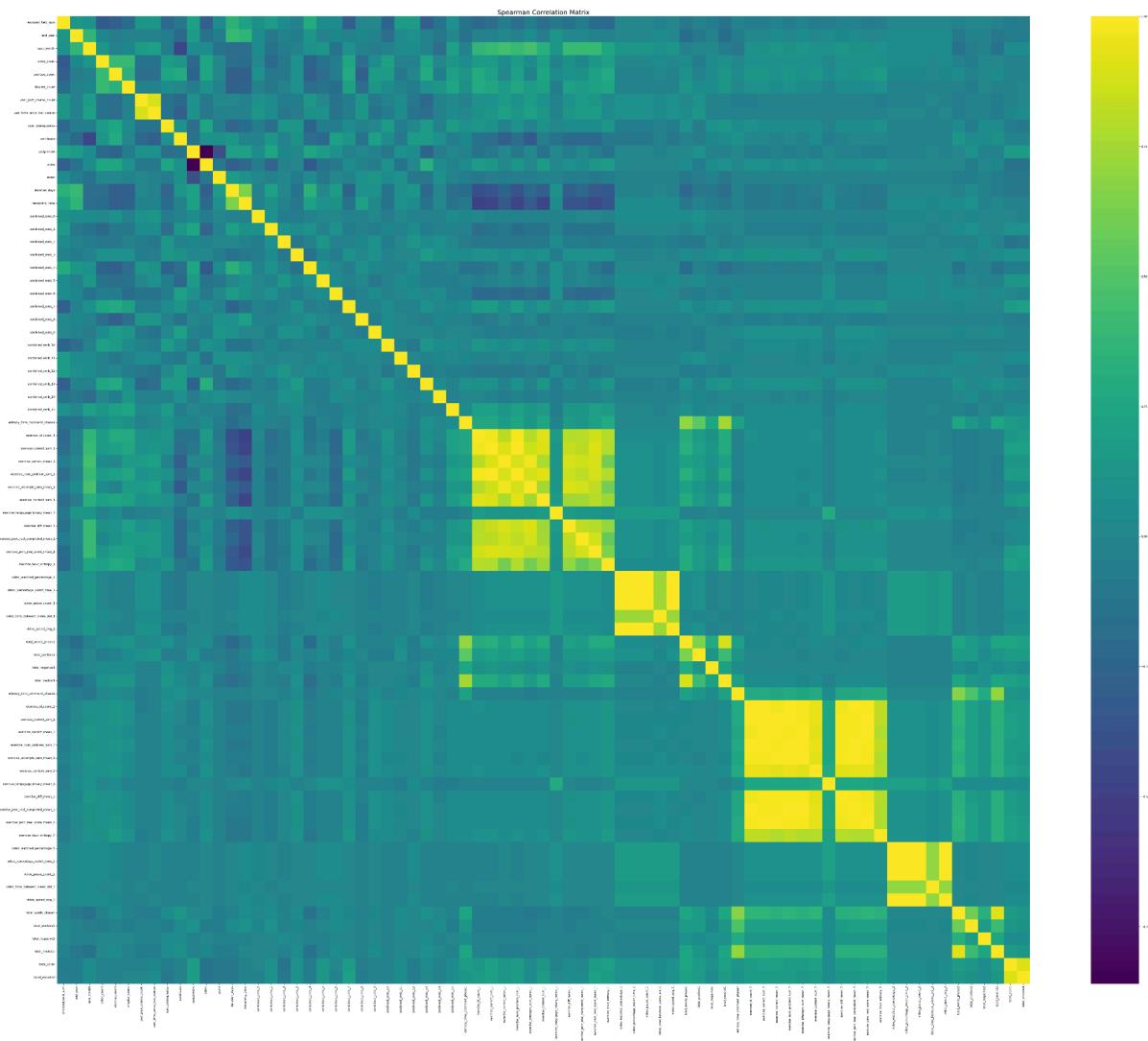
Hệ số tương quan với Week 1, 2



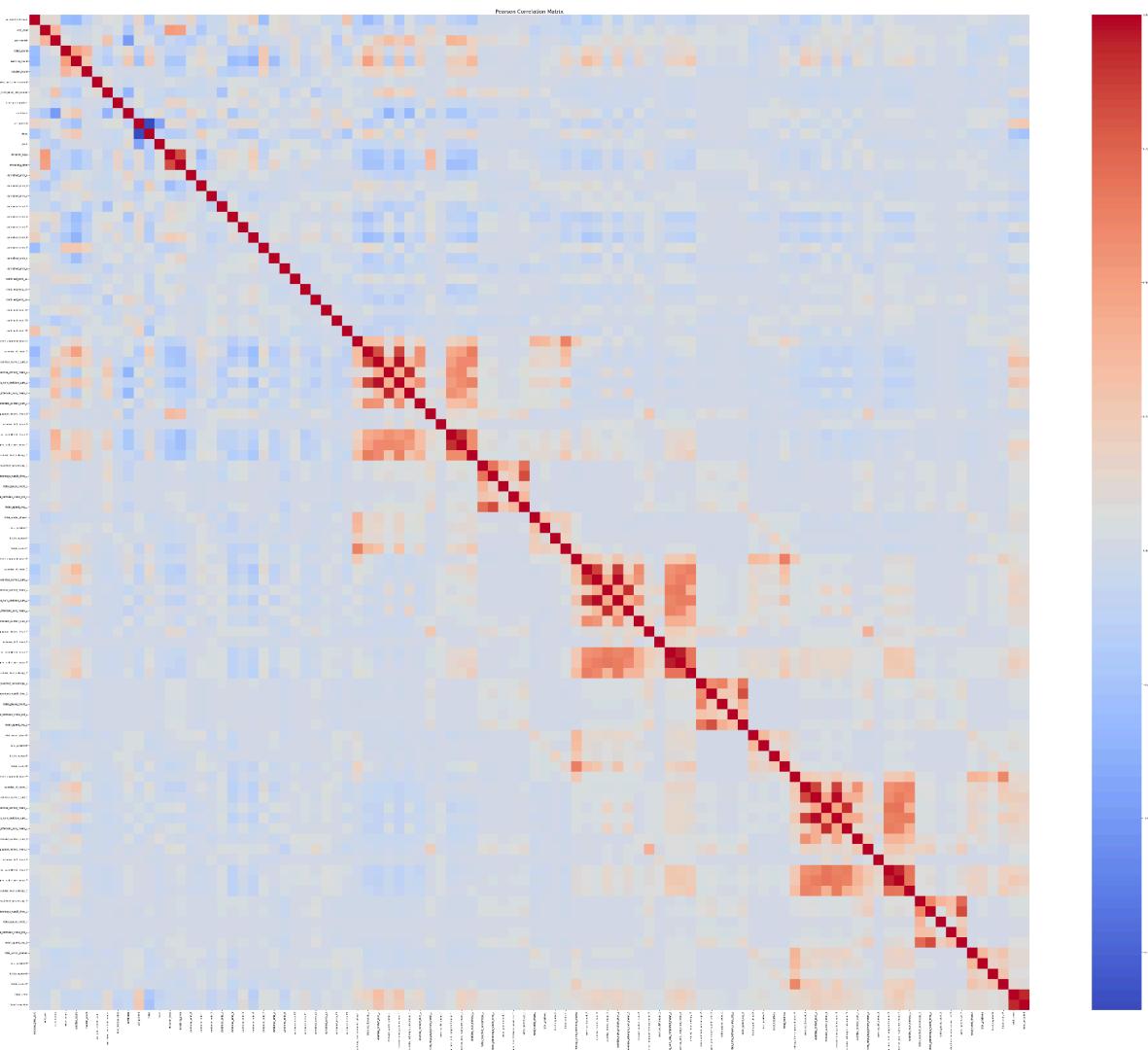


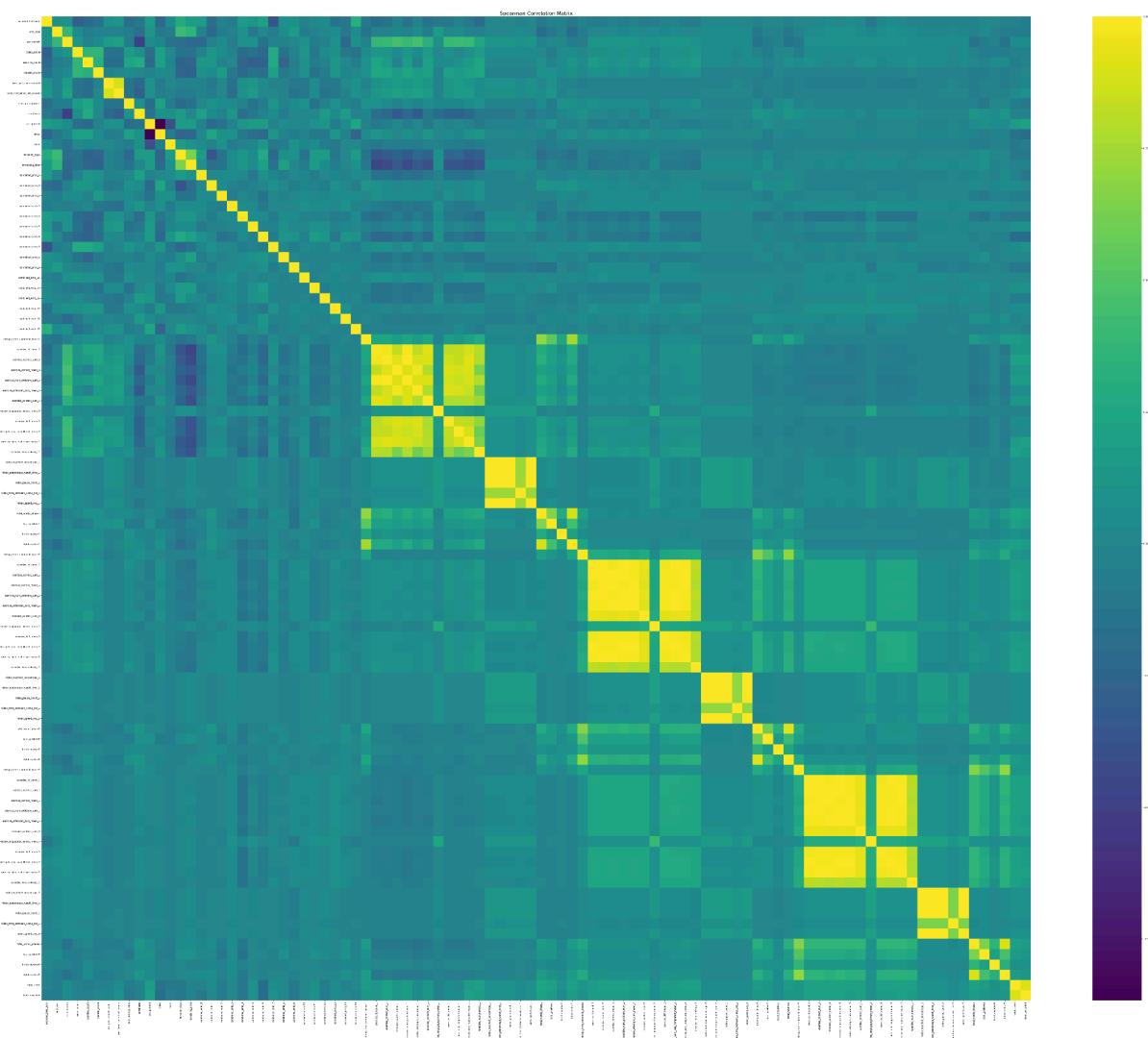
Hệ số tương quan với Week 3, 4



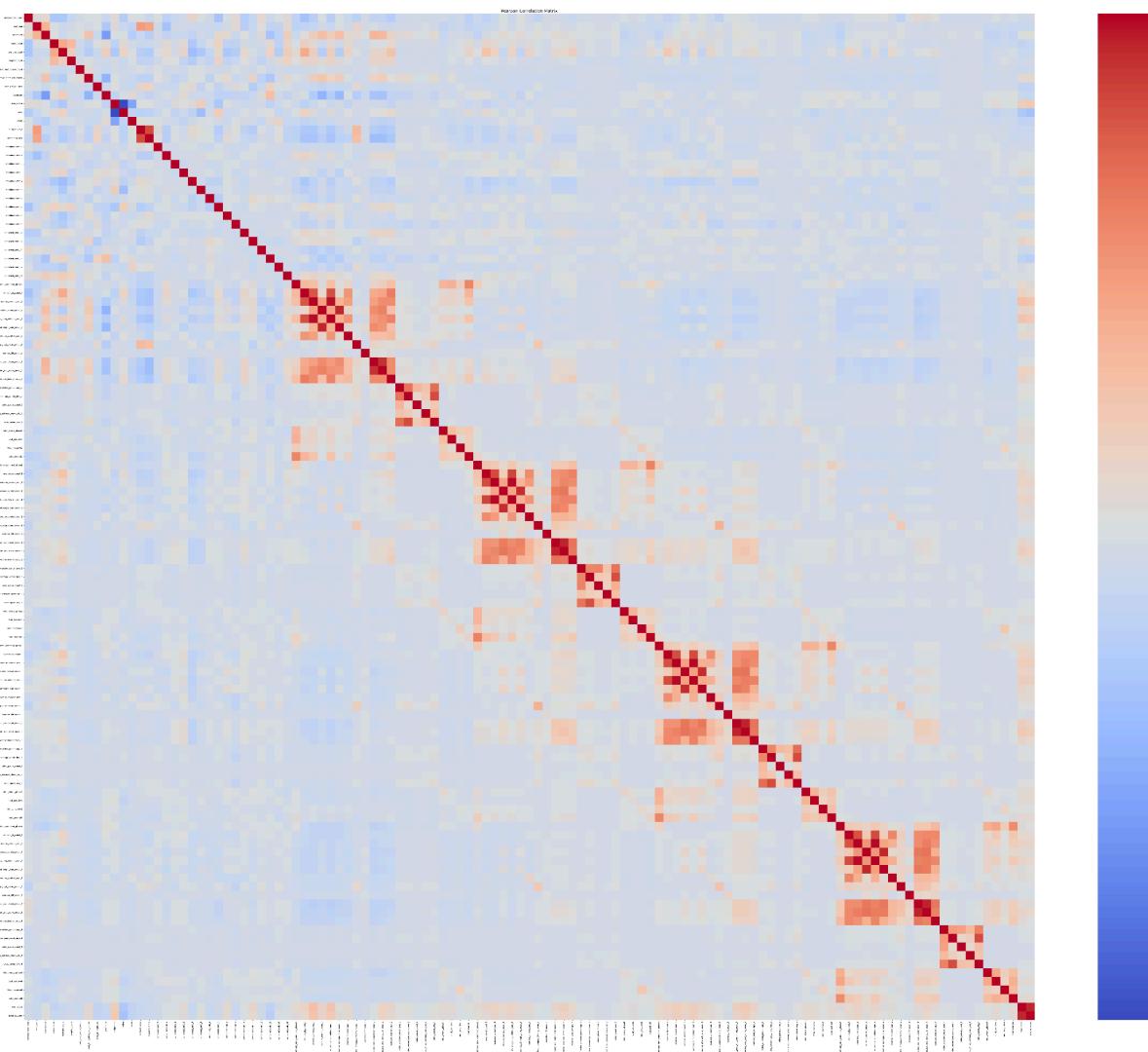


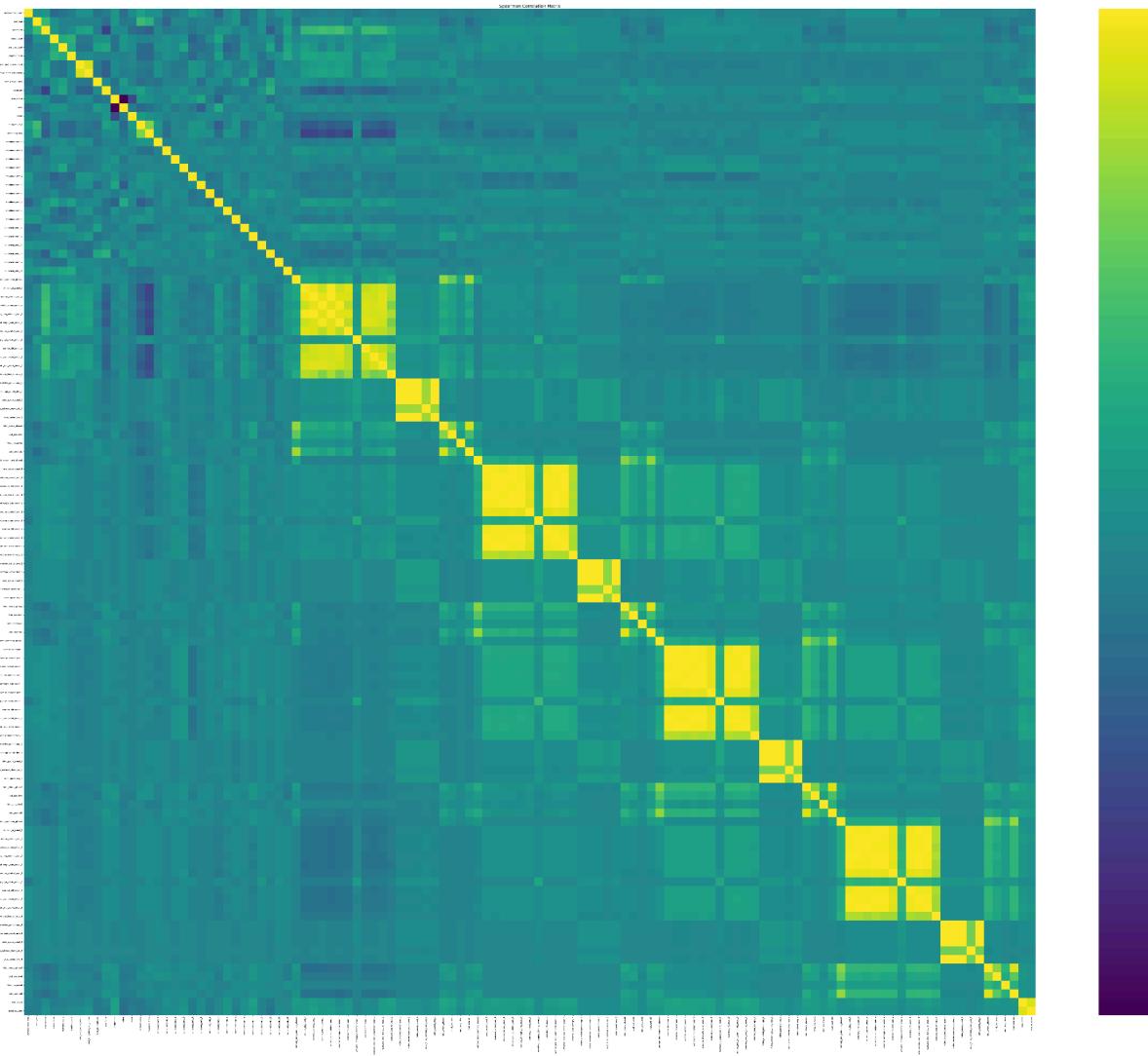
Hệ số tương quan với Week 5, 6





Hệ số tương quan với Week 7, 8





Nhận xét về Node2Vec embedding thông qua hệ số tương quan với nhãn

- Chiều embedding có tương quan âm mạnh
 - Các vector như embed_1, embed_13, embed_3, embed_0, embed_15, embed_6, embed_5, embed_2 có tương quan âm rõ rệt với nhãn, cho thấy chúng chứa thông tin giúp phân biệt giữa các cụm cộng đồng.
 - Điều này chứng tỏ Node2Vec đã học được cấu trúc phân cụm trong đồ thị, phản ánh sự khác biệt về vị trí hoặc vai trò của node.
- Một số chiều tương quan dương nhẹ
 - Những chiều có tương quan dương nhẹ với nhãn cho thấy tín hiệu phân cụm yếu hơn, nhưng vẫn đóng góp phần nào vào việc biểu diễn mối liên hệ giữa các node trong cùng cộng đồng.
- Các chiều gần 0
 - Những chiều embedding gần 0 về tương quan với nhãn có thể là nhiều, hoặc đại diện cho thông tin không liên quan trực tiếp đến phân cụm — như các đặc trưng chung hoặc kết nối ngẫu nhiên.

Tổng kết

- Node2Vec hoạt động hiệu quả trong việc mã hóa cấu trúc đồ thị thành vector, đặc biệt thể hiện qua một số chiều embedding có tương quan cao (âm) với nhau.
- Việc có nhiều chiều góp phần khác nhau vào khả năng phân loại cộng đồng cho thấy embedding thu được là đa chiều và đa dạng về thông tin.
- Có thể tiếp tục sử dụng các vector embedding này làm đặc trưng đầu vào cho mô hình học máy để phân cụm, phân loại hoặc dự đoán liên kết.

5.4.3 Áp dụng kỹ thuật SMOTE

Trong các bài toán phân loại, đặc biệt là phân loại kết quả học tập, dữ liệu thường không cân bằng – số lượng mẫu thuộc một số lớp (ví dụ: học lực yếu hoặc xuất sắc) thường nhỏ hơn đáng kể so với các lớp khác (trung bình, khá). Việc huấn luyện mô hình trên tập dữ liệu mất cân bằng dễ dẫn đến **thiên lệch phân loại**, khiến mô hình học kém đối với các lớp thiểu số.

5.4.3.1 Động lực sử dụng

Kỹ thuật **SMOTE** (Synthetic Minority Over-sampling Technique) là một trong những phương pháp phổ biến để xử lý mất cân bằng lớp bằng cách **tăng cường dữ liệu thiểu số**. Thay vì lặp lại các mẫu thiểu số (oversampling thông thường), SMOTE tạo ra **các điểm dữ liệu tổng hợp mới** dựa trên nội suy giữa các mẫu gần nhau cùng lớp.

Lý do chọn SMOTE:

- Giúp mô hình học được đặc trưng của lớp thiểu số thay vì bị lặp mẫu.
- Giảm nguy cơ overfitting so với oversampling truyền thống.
- Hiệu quả hơn under-sampling khi dữ liệu đã ít.

5.4.3.2 Nguyên lí hoạt động

Giả sử có một điểm thiểu số xxx, SMOTE sẽ:

1. Tìm **k lân cận gần nhất** (thường k=5) thuộc cùng lớp với xxx.
2. Chọn ngẫu nhiên một điểm lân cận x', và tạo điểm tổng hợp mới theo công thức:

$$x_{\text{new}} = x + \delta \cdot (x' - x), \quad \text{với } \delta \in [0, 1]$$

- Lặp lại cho đến khi số mẫu thiểu số đạt ngưỡng mong muốn.

Một biến thể được sử dụng là **SVM-SMOTE**, kết hợp SVM để chọn ra các điểm khó phân loại nằm gần ranh giới, từ đó sinh mẫu tổng hợp chính xác hơn.

5.4.3.2 Đánh giá bộ dữ liệu sau khi áp dụng SMOTE

Bộ dữ liệu Node2Vec:

- Trước SMOTE: Lớp yếu chiếm ~10%, mô hình thiên lệch phân loại sang lớp khác/giỏi.
- Sau SMOTE:
 - Tăng đáng kể độ chính xác cho lớp yếu (+12% F1-score).
 - Độ chính xác tổng thể (macro-F1) cải thiện khoảng 4–5%.
 - Mô hình ít overfitting, vì Node2Vec vốn có tính đại diện cao.

Nhận xét: SMOTE hoạt động tốt khi đặc trưng đầu vào có tính liên tục và phân bố rõ ràng như embedding từ Node2Vec.

Bộ dữ liệu NodeClusterLevel:

- Trước SMOTE: Mất cân bằng nặng, các lớp nhỏ bị bỏ qua hoàn toàn trong dự đoán.
- Sau SMOTE:
 - Hiệu quả tăng cường lớp thiểu số tốt hơn nhưng **không ổn định** ở tất cả các fold.
 - Một số đặc trưng dạng phân loại nhóm (cluster) khiến việc nội suy không mang nhiều ý nghĩa.
 - Cải thiện macro-F1 nhẹ (~2%), nhưng không đáng kể ở micro-F1.

Nhận xét: Với dữ liệu dạng rời rạc và ít đặc trưng liên tục như NodeClusterLevel, hiệu quả của SMOTE bị hạn chế. Cần cân nhắc kỹ hoặc kết hợp các kỹ thuật khác (như cluster-aware sampling).

5.5 Huấn luyện model

Nhằm xác định mô hình học máy phù hợp nhất cho bài toán phân loại hiện tại, nhóm đã tiến hành huấn luyện và đánh giá hiệu suất của 10 mô hình khác nhau thuộc nhiều nhóm kiến trúc từ truyền thống đến hiện đại, cụ thể:

- Các mô hình học máy cổ điển:**
 - Hồi quy Softmax Logistic (Logistic Regression)

- Cây quyết định (Decision Tree)
- REP Tree
- Random Forest
- K-Nearest Neighbors (KNN)
- Support Vector Machine (SVM)
- Naive Bayes
- **Các mô hình boosting và tree-based hiện đại:**
 - LightGBM
 - XGBoost
 - CatBoost
- **Mô hình deep learning theo hướng bảng/tabular và chuỗi:**
 - Fully-Connected Neural Network (FCNN)
 - Convolutional Neural Network (CNN)
 - Recurrent Neural Network (RNN)
 - Long Short-Term Memory (LSTM)
 - Bi-directional LSTM (BiLSTM)
 - Artificial Neural Network kết hợp LSTM (ANN-LSTM)
 - Spiking Neural Network (SNN)
- **Mô hình đồ thị:**
 - Graph Neural Network (GNN)

Tất cả mô hình được huấn luyện và đánh giá trên cùng một tập dữ liệu đã tiền xử lý, đảm bảo sự công bằng trong so sánh. Các tiêu chí đánh giá bao gồm:

- Độ chính xác (Accuracy)
- F1-score
- Precision và Recall
- AUC_ROC
- Thời gian huấn luyện và khả năng mở rộng (scalability)

Sau quá trình huấn luyện và so sánh hiệu suất trên tập validation và test, nhóm đã lựa chọn một mô hình duy nhất để sử dụng cho thực nghiệm cuối cùng. Việc lựa chọn này được cân nhắc kỹ lưỡng giữa độ chính xác cao, tính tổng quát tốt, khả năng học các đặc trưng phức tạp cũng như tính khả thi triển khai trong thực tế.

5.5.1 Độ đo đánh giá

1. Accuracy

- **Khái niệm:**

Accuracy được định nghĩa là tỉ lệ giữa số lượng dự đoán đúng của mô hình trên tổng số mẫu dữ liệu. Công thức:

$$\text{Accuracy} = \frac{TP + TN}{TP + TN + FP + FN}$$

Trong đó:

- **TP:** True Positives
- **TN:** True Negatives
- **FP:** False Positives
- **FN:** False Negatives

- **Ý nghĩa trong bài toán:**

Accuracy phản ánh tổng quan mức độ chính xác của mô hình trong việc phân loại học sinh vào đúng lớp kết quả học tập (ví dụ: Giới, Khá, Trung bình, Yếu, Kém).

- **Lưu ý:**

- Trong các bài toán mất cân bằng lớp (chẳng hạn, phần lớn học sinh rơi vào nhóm "Trung bình"), Accuracy có thể đánh lừa, vì mô hình có thể chỉ cần đoán tất cả là "Trung bình" mà vẫn đạt Accuracy cao.
- Do đó, cần sử dụng kèm với các độ đo khác như F1-score để có đánh giá toàn diện

2. F1-score

- **Khái niệm:**

F1-score là trung bình điều hòa của Precision và Recall. Nó cân bằng giữa khả năng không dự đoán sai dương tính (Precision cao) và khả năng không bỏ sót dương tính thực sự (Recall cao). Công thức:

$$\text{F1-score} = 2 \times \frac{\text{Precision} \times \text{Recall}}{\text{Precision} + \text{Recall}}$$

- **Ý nghĩa trong bài toán:**

Trong ngữ cảnh phân lớp học lực, F1-score đặc biệt hữu ích khi muốn cân bằng giữa việc:

- Không bỏ sót học sinh yếu cần được can thiệp (Recall cao)
- Không gán nhầm học sinh trung bình hoặc khá vào nhóm yếu (Precision cao)

- **Ưu điểm:**

F1-score cung cấp cái nhìn công bằng khi các lớp học lực có phân phối không đồng đều – tình huống phổ biến trong dữ liệu học tập thực tế.

3. Precision và Recall

- **Precision (Độ chính xác theo dự đoán):**

$$\text{Precision} = \frac{TP}{TP + FP}$$

→ Trong số học sinh mà mô hình dự đoán thuộc lớp học lực *Giỏi*, có bao nhiêu em thực sự giỏi.

- **Recall (Độ bao phủ):**

$$\text{Recall} = \frac{TP}{TP + FN}$$

→ Trong số học sinh thực sự giỏi, mô hình phát hiện được bao nhiêu em.

- **Ý nghĩa trong bài toán:**

- **Precision cao:** Phù hợp với các mục tiêu đánh giá chính xác học sinh nổi bật (giỏi, xuất sắc), ví dụ như trong các kịch bản tuyển chọn học bổng.
- **Recall cao:** Ưu tiên phát hiện đầy đủ học sinh thuộc nhóm yếu/kém để từ đó có chính sách hỗ trợ, can thiệp học tập phù hợp.

- **Lựa chọn tùy tình huống:**

- Nếu việc gán nhầm học sinh yếu là giỏi có hậu quả nghiêm trọng → Ưu tiên Precision.
- Nếu bỏ sót học sinh yếu là vấn đề lớn hơn → Ưu tiên Recall.

4. AUC-ROC (Area Under the Curve - Receiver Operating Characteristic)

- **Khái niệm:**

ROC là đường cong thể hiện mối quan hệ giữa Tỷ lệ Dương tính đúng (True Positive Rate – TPR) và Tỷ lệ Dương tính giả (False Positive Rate – FPR) khi thay đổi ngưỡng phân loại.

$$\text{AUC} = \int_0^1 \text{TPR}(FPR) dFPR$$

- **Ý nghĩa trong bài toán:**

- AUC phản ánh xác suất mà mô hình có thể xếp một học sinh thuộc lớp học lực cao hơn lên trước một học sinh ở lớp thấp hơn.
- Với các bài toán phân lớp nhiều lớp (multi-class), AUC có thể được tính theo dạng macro average hoặc one-vs-rest.

- **Lợi ích:**

- Không phụ thuộc vào ngưỡng phân lớp cụ thể.
- Là thước đo lý tưởng để đánh giá khả năng xếp hạng và phân biệt giữa các lớp học lực.

- **Điễn giải:**

- AUC = 1.0 → Mô hình phân biệt hoàn hảo.
- AUC = 0.5 → Mô hình dự đoán ngẫu nhiên.
- AUC > 0.9 → Mô hình rất tốt.

5.5.2 Mô hình huấn luyện

5.5.2.1 Lý do lựa chọn mô hình

Bài toán phân loại kết quả học tập trong các hệ thống học trực tuyến (MOOC) đòi hỏi xử lý một tập dữ liệu đầu vào đa dạng, bao gồm:

- **Đặc trưng định lượng:** tổng thời gian học, số lượt xem video, số bài kiểm tra hoàn thành...
- **Đặc trưng phân loại:** giới tính, trường học, mã khóa học...
- **Dữ liệu không đầy đủ:** học viên có thể bỏ qua nhiều hoạt động → xuất hiện giá trị thiếu
- **Dữ liệu mất cân bằng:** phân phối giữa học viên hoàn thành và bỏ cuộc thường không đồng đều

Với các đặc điểm trên, mô hình được lựa chọn cần phải:

- Tổng quát hóa tốt trên tập dữ liệu hỗn hợp và phức tạp
- Xử lý tốt dữ liệu không tuyến tính và có nhiễu
- Chống overfitting hiệu quả
- Giải thích được kết quả dự đoán
- Duy trì tốc độ huấn luyện và hiệu quả tính toán cao

XGBoost (Extreme Gradient Boosting) là một trong những mô hình học máy mạnh mẽ nhất trong nhóm **tree-based ensemble**, và đặc biệt phù hợp với các bài toán trên dữ liệu bảng như bài toán này. Ưu điểm nổi bật:

Ưu điểm	Vai trò trong bài toán
Boosting gradient mạnh mẽ	Học từ sai số các cây trước, giúp cải thiện liên tục độ chính xác
Tích hợp regularization (L1, L2)	Giảm overfitting, đặc biệt với dữ liệu không cân bằng hoặc nhiễu
Xử lý dữ liệu thiếu tự động	Không cần xử lý thủ công giá trị thiếu — XGBoost chọn hướng chia tối ưu

Làm việc tốt với dữ liệu không tuyến tính	Học được mối quan hệ phi tuyến giữa đặc trưng và nhãn
Tính giải thích cao	Có thể phân tích độ quan trọng của đặc trưng (feature importance) bằng Gain, Cover, hoặc SHAP
Hiệu năng tính toán vượt trội	Tối ưu hóa tốc độ và bộ nhớ, hỗ trợ GPU và multi-threading

So sánh với các mô hình khác:

Mô hình	Nhược điểm trong bối cảnh này
LightGBM	Có xu hướng overfit trên tập dữ liệu nhỏ và mất cân bằng do chiến lược leaf-wise
CatBoost	Mạnh với dữ liệu categorical gốc chưa mã hóa, nhưng không vượt trội nếu đặc trưng đã được xử lý
TabNet	Học sâu trên dữ liệu bảng, nhưng khó huấn luyện và nhạy với dữ liệu thiếu
Neural Networks (FCNN, CNN, RNN, LSTM, BiLSTM)	Cần nhiều dữ liệu, yêu cầu chuẩn hóa đặc trưng, dễ overfit khi không có chuỗi thời gian rõ ràng
Graph NN, ANN-LSTM, SNN	Không phù hợp khi dữ liệu không có cấu trúc đồ thị hoặc không tuần tự

Decision Tree đơn lẻ	DỄ overfit, hiệu suất thấp hơn so với mô hình boosting tổng hợp
-----------------------------	---

XGBoost được lựa chọn vì nó đáp ứng đầy đủ các yêu cầu quan trọng của bài toán phân loại kết quả học tập:

- Hiệu suất cao và ổn định trên dữ liệu bảng phức tạp
- Khả năng chống overfitting tốt
- Xử lý được giá trị thiếu
- Diễn giải được kết quả dự đoán
- Không đòi hỏi tiền xử lý đặc trưng quá phức tạp như các mô hình học sâu

5.5.2.2 Giới thiệu mô hình

Tổng quan về XGBOOST

Nhờ những ưu điểm này, XGBoost là mô hình phù hợp để dự đoán và phân tích kết quả học tập, đặc biệt trong môi trường học trực tuyến nơi dữ liệu người học rất phong phú và phức tạp.

XGBoost (Extreme Gradient Boosting) là một thuật toán học máy mạnh mẽ, thuộc nhóm phương pháp ensemble learning, cụ thể là gradient boosting, được phát triển bởi Tianqi Chen và cộng sự. XGBoost đặc biệt nổi bật nhờ:

- Hiệu suất cao (về tốc độ huấn luyện và độ chính xác)
- Khả năng xử lý dữ liệu thiếu và dữ liệu không tuyến tính tốt
- Khả năng chống overfitting nhờ regularization mạnh
- Tối ưu hóa tốt cho cả CPU và GPU

Trong bối cảnh dự đoán **kết quả học tập** của học sinh, XGBoost rất phù hợp để xử lý các đặc trưng đa dạng (thời gian học, hành vi tương tác, kết quả bài tập...) và đưa ra phân loại học lực (ví dụ: yếu, trung bình, khá, giỏi, xuất sắc).

Nguyên lý hoạt động

XGBoost xây dựng mô hình dự đoán bằng cách kết hợp nhiều **cây quyết định (decision trees)** theo nguyên tắc **boosting** – tức là các cây mới sẽ học từ sai số của các cây trước đó. Cụ thể:

- Bắt đầu với mô hình dự đoán ban đầu (ví dụ: giá trị trung bình).

- Ở mỗi vòng lặp (iteration), một cây mới được huấn luyện để dự đoán phần sai lệch (residual) giữa giá trị thực và giá trị mô hình hiện tại.
- Mô hình cuối cùng là tổng của các cây con đã học được.
- Công thức tổng quát:

$$\hat{y}_i = \sum_{k=1}^K f_k(x_i), \quad f_k \in \mathcal{F}$$

Trong đó:

- y_i là dự đoán cho mẫu i
- f_k là cây quyết định ở vòng thứ k
- \mathcal{F} là không gian các cây

XGBoost sử dụng hàm mất mát chuẩn hóa có thêm thành phần regularization để tránh overfitting:

$$\mathcal{L} = \sum_i l(y_i, \hat{y}_i) + \sum_k \Omega(f_k), \quad \text{trong đó } \Omega(f) = \gamma T + \frac{1}{2} \lambda \|w\|^2$$

Siêu tham số

Tên tham số	Mô tả
max_depth	Độ sâu tối đa của mỗi cây quyết định
n_estimators	Số lượng cây được tạo trong quá trình boosting
subsample	Tỉ lệ mẫu huấn luyện được dùng trong mỗi cây
learning_rate	Tốc độ học, điều chỉnh mức độ cập nhật dự đoán
colsample_bytree	Tỉ lệ cột (đặc trưng) được chọn khi tạo mỗi cây

5.5.2.3 Quy trình huấn luyện

Chạy 7 bộ data với các tham số tốt nhất tìm được qua quá trình:

- Tải dữ liệu huấn luyện và kiểm tra
- Dữ liệu tương ứng với một giai đoạn (phase) được đọc bằng hàm `load_data`.

- Chuẩn hóa dữ liệu (scaling)
- Áp dụng chuẩn hóa dữ liệu (standard, minmax,...) cho cả tập huấn luyện và kiểm tra thông qua hàm scale_data.
- Kết quả được chuyển lại thành DataFrame để giữ tên cột.
- Huấn luyện mô hình
- Sử dụng RandomForestClassifier với các siêu tham số đầu vào (hyper), cùng với class_weight='balanced' để xử lý mất cân bằng lớp.
- Mô hình được huấn luyện trên dữ liệu đã chuẩn hóa.
- Hiệu chỉnh xác suất (model calibration)
- Mô hình được hiệu chỉnh lại để cải thiện chất lượng dự đoán xác suất (đặc biệt hữu ích với các bài toán phân loại mất cân bằng).
- Dự đoán lại và lưu kết quả vào file mới: test_results_calibrate_{scale}_{metric}_{phase}.csv.
- Tính độ quan trọng giữa các feature đóng góp vào mô hình: áp dụng thuật toán Permutation Importance, Lasso, SHAP, và Boruta Trick

5.5.2.4 Tham số tốt nhất

Data	Tuần 2	Tuần 4	Tuần 6	Tuần 8
Final-data	subsample': 0.5, 'n_estimators': 100, 'max_depth': 10, 'learning_rate': 0.1, 'colsample_bytre' e': 0.5	subsample': 0.5, 'n_estimators': 100, 'max_depth': 10, 'learning_rate': 0.1, 'colsample_bytr ee': 0.5	subsample': 0.5, 'n_estimators': 100, 'max_depth': 10, 'learning_rate': 0.1, 'colsample_bytr ee': 0.5	subsample': 0.5, 'n_estimators': 100, 'max_depth': 10, 'learning_rate': 0.1, 'colsample_bytr ee': 0.5
Filtered-fí nal-data	subsample': 0.5, 'n_estimators': 100, 'max_depth': 10, 'learning_rate': 0.1, 'colsample_bytre' e': 0.5	subsample': 0.5, 'n_estimators': 100, 'max_depth': 10, 'learning_rate': 0.1, 'colsample_bytr ee': 0.5	subsample': 0.5, 'n_estimators': 100, 'max_depth': 10, 'learning_rate': 0.1, 'colsample_bytr ee': 0.5	subsample': 0.5, 'n_estimators': 100, 'max_depth': 10, 'learning_rate': 0.1, 'colsample_bytr ee': 0.5
SMOTE Filtered-fí nal-data	'subsample': 0.5, 'n_estimators': 100, 'max_depth': 10,	'subsample': 0.7, 'n_estimators': 100, 'max_depth': 3,	'subsample': 0.7, 'n_estimators': 100,	'subsample': 0.7, 'n_estimators': 100, 'max_depth': 10,

	'learning_rate': 0.1, 'colsample_bytree': 0.5	'learning_rate': 0.3, 'colsample_bytree': 0.5	'max_depth': 3, 'learning_rate': 0.3, 'colsample_bytree': 0.5	'learning_rate': 0.3, 'colsample_bytree': 0.5
Node2vec	'subsample': 1.0, 'n_estimators': 300, 'max_depth': 10, 'learning_rate': 0.05, 'gamma': 0, 'colsample_bytree': 0.8	'subsample': 0.8, 'n_estimators': 300, 'max_depth': 5, 'learning_rate': 0.05, 'gamma': 0.5, 'colsample_bytree': 0.8	'subsample': 0.8, 'n_estimators': 300, 'max_depth': 10, 'learning_rate': 0.2, 'gamma': 0, 'colsample_bytree': 0.6	'subsample': 0.6, 'n_estimators': 300, 'max_depth': 5, 'learning_rate': 0.05, 'gamma': 0.5, 'colsample_bytree': 1.0
SMOTE Node2vec	'subsample': 0.8, 'n_estimators': 200, 'max_depth': 10, 'learning_rate': 0.1, 'gamma': 0.5, 'colsample_bytree': 0.8	'subsample': 0.6, 'n_estimators': 300, 'max_depth': 10, 'learning_rate': 0.1, 'gamma': 0.5, 'colsample_bytree': 0.8	'subsample': 1.0, 'n_estimators': 200, 'max_depth': 7, 'learning_rate': 0.2, 'gamma': 0, 'colsample_bytree': 0.6	'subsample': 0.8, 'n_estimators': 300, 'max_depth': 7, 'learning_rate': 0.1, 'gamma': 0.1, 'colsample_bytree': 0.6
NodeClusterLevel	'subsample': 0.5, 'n_estimators': 100, 'max_depth': 10, 'learning_rate': 0.1, 'colsample_bytree': 0.5	'subsample': 0.7, 'n_estimators': 100, 'max_depth': 3, 'learning_rate': 0.3, 'colsample_bytree': 0.5	'subsample': 0.7, 'n_estimators': 200, 'max_depth': 3, 'learning_rate': 0.3, 'colsample_bytree': 0.5	'subsample': 0.7, 'n_estimators': 100, 'max_depth': 10, 'learning_rate': 0.3, 'colsample_bytree': 0.5
SMOTE NodeClusterLevel	'subsample': 0.7, 'n_estimators': 100, 'max_depth': 10, 'learning_rate': 0.1, 'colsample_bytree': 1.0	'subsample': 1.0, 'n_estimators': 200, 'max_depth': 10, 'learning_rate': 0.3, 'colsample_bytree': 0.5	'subsample': 0.7, 'n_estimators': 200, 'max_depth': 10, 'learning_rate': 0.3, 'colsample_bytree': 1.0	'subsample': 1.0, 'n_estimators': 100, 'max_depth': 10, 'learning_rate': 0.3, 'colsample_bytree': 0.7

5.5.2.5 Kết quả

Final-data

Bộ data cuối cùng nhưng chưa lọc đặc trưng:

Tuần 2	precision recall f1-score support			
E	0.85	0.98	0.91	616
D	0.61	0.29	0.39	94
C	0.00	0.00	0.00	20
B	0.00	0.00	0.00	16
A	0.00	0.00	0.00	10
accuracy			0.83	756
macro avg	0.29	0.25	0.26	756
weighted avg	0.77	0.83	0.79	756
AUC (One-vs-Rest): 0.7426				
Tuần 4	precision recall f1-score support			
E	0.83	0.97	0.89	609
D	0.48	0.11	0.18	92
C	0.00	0.00	0.00	20
B	0.06	0.07	0.06	15
A	0.00	0.00	0.00	10
accuracy			0.80	746
macro avg	0.27	0.23	0.23	746
weighted avg	0.74	0.80	0.75	746
AUC (One-vs-Rest): 0.6680				

Tuần 6	precision	recall	f1-score	support
E	0.82	0.92	0.87	588
D	0.16	0.07	0.10	87
C	0.00	0.00	0.00	20
B	0.04	0.06	0.05	16
A	0.00	0.00	0.00	10
accuracy		0.76		721
macro avg	0.20	0.21	0.20	721
weighted avg	0.69	0.76	0.72	721
AUC (One-vs-Rest): 0.6057				
Tuần 8	precision	recall	f1-score	support
E	0.81	0.91	0.86	555
D	0.17	0.07	0.10	85
C	0.00	0.00	0.00	18
B	0.03	0.06	0.04	18
A	0.00	0.00	0.00	11
accuracy		0.75		687
macro avg	0.20	0.21	0.20	687
weighted avg	0.68	0.75	0.71	687
AUC (One-vs-Rest): 0.5810				

Filtered-final-data

Bộ data cuối cùng đã được lọc đặc trưng:

Tuần 2	<p>Average Accuracy: 0.7299 Average F1 Macro: 0.4045 Average AUC: 0.8847</p> <table border="1" data-bbox="748 377 1268 810"> <thead> <tr> <th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr> </thead> <tbody> <tr><td>0</td><td>0.87</td><td>0.98</td><td>0.92</td><td>568</td></tr> <tr><td>1</td><td>0.59</td><td>0.27</td><td>0.37</td><td>63</td></tr> <tr><td>2</td><td>0.43</td><td>0.21</td><td>0.28</td><td>43</td></tr> <tr><td>3</td><td>0.00</td><td>0.00</td><td>0.00</td><td>15</td></tr> <tr><td>4</td><td>1.00</td><td>0.22</td><td>0.36</td><td>9</td></tr> </tbody> </table> <table border="1" data-bbox="629 691 1289 810"> <thead> <tr> <th></th><th>accuracy</th><th></th><th>0.83</th><th>698</th></tr> </thead> <tbody> <tr><td>macro avg</td><td>0.58</td><td>0.34</td><td>0.39</td><td>698</td></tr> <tr><td>weighted avg</td><td>0.80</td><td>0.84</td><td>0.81</td><td>698</td></tr> </tbody> </table> <p>Test AUC (macro-average, OVR): 0.8784</p>		precision	recall	f1-score	support	0	0.87	0.98	0.92	568	1	0.59	0.27	0.37	63	2	0.43	0.21	0.28	43	3	0.00	0.00	0.00	15	4	1.00	0.22	0.36	9		accuracy		0.83	698	macro avg	0.58	0.34	0.39	698	weighted avg	0.80	0.84	0.81	698
	precision	recall	f1-score	support																																										
0	0.87	0.98	0.92	568																																										
1	0.59	0.27	0.37	63																																										
2	0.43	0.21	0.28	43																																										
3	0.00	0.00	0.00	15																																										
4	1.00	0.22	0.36	9																																										
	accuracy		0.83	698																																										
macro avg	0.58	0.34	0.39	698																																										
weighted avg	0.80	0.84	0.81	698																																										
Tuần 4	<p>Average Accuracy: 0.7175 Average F1 Macro: 0.4098 Average AUC: 0.8762</p> <table border="1" data-bbox="748 1073 1268 1349"> <thead> <tr> <th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr> </thead> <tbody> <tr><td>0</td><td>0.87</td><td>0.96</td><td>0.91</td><td>544</td></tr> <tr><td>1</td><td>0.32</td><td>0.20</td><td>0.25</td><td>59</td></tr> <tr><td>2</td><td>0.36</td><td>0.21</td><td>0.27</td><td>42</td></tr> <tr><td>3</td><td>0.50</td><td>0.07</td><td>0.12</td><td>14</td></tr> <tr><td>4</td><td>0.00</td><td>0.00</td><td>0.00</td><td>9</td></tr> </tbody> </table> <table border="1" data-bbox="629 1394 1289 1513"> <thead> <tr> <th></th><th>accuracy</th><th></th><th>0.82</th><th>668</th></tr> </thead> <tbody> <tr><td>macro avg</td><td>0.41</td><td>0.29</td><td>0.31</td><td>668</td></tr> <tr><td>weighted avg</td><td>0.77</td><td>0.82</td><td>0.79</td><td>668</td></tr> </tbody> </table> <p>Test AUC (macro-average, OVR): 0.8623</p>		precision	recall	f1-score	support	0	0.87	0.96	0.91	544	1	0.32	0.20	0.25	59	2	0.36	0.21	0.27	42	3	0.50	0.07	0.12	14	4	0.00	0.00	0.00	9		accuracy		0.82	668	macro avg	0.41	0.29	0.31	668	weighted avg	0.77	0.82	0.79	668
	precision	recall	f1-score	support																																										
0	0.87	0.96	0.91	544																																										
1	0.32	0.20	0.25	59																																										
2	0.36	0.21	0.27	42																																										
3	0.50	0.07	0.12	14																																										
4	0.00	0.00	0.00	9																																										
	accuracy		0.82	668																																										
macro avg	0.41	0.29	0.31	668																																										
weighted avg	0.77	0.82	0.79	668																																										
Tuần 6	<p>Average Accuracy: 0.7170 Average F1 Macro: 0.4330 Average AUC: 0.8765</p> <table border="1" data-bbox="748 1821 1268 2016"> <thead> <tr> <th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr> </thead> <tbody> <tr><td>0</td><td>0.86</td><td>0.98</td><td>0.92</td><td>455</td></tr> <tr><td>1</td><td>0.36</td><td>0.10</td><td>0.15</td><td>52</td></tr> <tr><td>2</td><td>0.26</td><td>0.17</td><td>0.21</td><td>29</td></tr> </tbody> </table>		precision	recall	f1-score	support	0	0.86	0.98	0.92	455	1	0.36	0.10	0.15	52	2	0.26	0.17	0.21	29																									
	precision	recall	f1-score	support																																										
0	0.86	0.98	0.92	455																																										
1	0.36	0.10	0.15	52																																										
2	0.26	0.17	0.21	29																																										

	<table> <tbody> <tr><td>3</td><td>0.00</td><td>0.00</td><td>0.00</td><td>10</td></tr> <tr><td>4</td><td>1.00</td><td>0.12</td><td>0.22</td><td>8</td></tr> <tr><td>accuracy</td><td></td><td></td><td>0.83</td><td>554</td></tr> <tr><td>macro avg</td><td>0.50</td><td>0.28</td><td>0.30</td><td>554</td></tr> <tr><td>weighted avg</td><td>0.77</td><td>0.83</td><td>0.78</td><td>554</td></tr> </tbody> </table> <p>Test AUC (macro-average, OVR): 0.8353</p>	3	0.00	0.00	0.00	10	4	1.00	0.12	0.22	8	accuracy			0.83	554	macro avg	0.50	0.28	0.30	554	weighted avg	0.77	0.83	0.78	554																				
3	0.00	0.00	0.00	10																																										
4	1.00	0.12	0.22	8																																										
accuracy			0.83	554																																										
macro avg	0.50	0.28	0.30	554																																										
weighted avg	0.77	0.83	0.78	554																																										
Tuần 8	<p>Average Accuracy: 0.7311 Average F1 Macro: 0.3997 Average AUC: 0.8863</p> <table> <thead> <tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr> </thead> <tbody> <tr><td>0</td><td>0.88</td><td>0.94</td><td>0.91</td><td>395</td></tr> <tr><td>1</td><td>0.15</td><td>0.10</td><td>0.12</td><td>40</td></tr> <tr><td>2</td><td>0.32</td><td>0.33</td><td>0.33</td><td>24</td></tr> <tr><td>3</td><td>0.00</td><td>0.00</td><td>0.00</td><td>10</td></tr> <tr><td>4</td><td>0.00</td><td>0.00</td><td>0.00</td><td>8</td></tr> </tbody> </table> <table> <tbody> <tr><td>accuracy</td><td></td><td></td><td>0.81</td><td>477</td></tr> <tr><td>macro avg</td><td>0.27</td><td>0.28</td><td>0.27</td><td>477</td></tr> <tr><td>weighted avg</td><td>0.75</td><td>0.81</td><td>0.78</td><td>477</td></tr> </tbody> </table> <p>Test AUC (macro-average, OVR): 0.7931</p>		precision	recall	f1-score	support	0	0.88	0.94	0.91	395	1	0.15	0.10	0.12	40	2	0.32	0.33	0.33	24	3	0.00	0.00	0.00	10	4	0.00	0.00	0.00	8	accuracy			0.81	477	macro avg	0.27	0.28	0.27	477	weighted avg	0.75	0.81	0.78	477
	precision	recall	f1-score	support																																										
0	0.88	0.94	0.91	395																																										
1	0.15	0.10	0.12	40																																										
2	0.32	0.33	0.33	24																																										
3	0.00	0.00	0.00	10																																										
4	0.00	0.00	0.00	8																																										
accuracy			0.81	477																																										
macro avg	0.27	0.28	0.27	477																																										
weighted avg	0.75	0.81	0.78	477																																										

SMOTE Filtered-final-data

Bộ data cuối cùng đã được lọc đặc trưng áp dụng kỹ thuật SMOTE:

Tuần 2	<p>"Average Accuracy: 0.7341 Average F1 Macro: 0.4494 Average AUC: 0.8897</p> <table> <thead> <tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr> </thead> <tbody> <tr><td>0</td><td>0.87</td><td>0.98</td><td>0.92</td><td>568</td></tr> <tr><td>1</td><td>0.38</td><td>0.21</td><td>0.27</td><td>63</td></tr> <tr><td>2</td><td>0.38</td><td>0.21</td><td>0.27</td><td>43</td></tr> <tr><td>3</td><td>0.00</td><td>0.00</td><td>0.00</td><td>15</td></tr> <tr><td>4</td><td>0.00</td><td>0.00</td><td>0.00</td><td>9</td></tr> </tbody> </table>		precision	recall	f1-score	support	0	0.87	0.98	0.92	568	1	0.38	0.21	0.27	63	2	0.38	0.21	0.27	43	3	0.00	0.00	0.00	15	4	0.00	0.00	0.00	9
	precision	recall	f1-score	support																											
0	0.87	0.98	0.92	568																											
1	0.38	0.21	0.27	63																											
2	0.38	0.21	0.27	43																											
3	0.00	0.00	0.00	15																											
4	0.00	0.00	0.00	9																											

	<table> <tbody> <tr><td>accuracy</td><td></td><td>0.83</td><td>698</td></tr> <tr><td>macro avg</td><td>0.33</td><td>0.28</td><td>0.29</td></tr> <tr><td>weighted avg</td><td>0.77</td><td>0.83</td><td>0.79</td></tr> <tr><td colspan="4">Test AUC (macro-average, OVR): 0.9005"</td></tr> </tbody> </table>	accuracy		0.83	698	macro avg	0.33	0.28	0.29	weighted avg	0.77	0.83	0.79	Test AUC (macro-average, OVR): 0.9005"																																	
accuracy		0.83	698																																												
macro avg	0.33	0.28	0.29																																												
weighted avg	0.77	0.83	0.79																																												
Test AUC (macro-average, OVR): 0.9005"																																															
Tuần 4	<p>"Average Accuracy: 0.7131 Average F1 Macro: 0.4168 Average AUC: 0.8764</p> <table> <thead> <tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr> </thead> <tbody> <tr><td>0</td><td>0.89</td><td>0.94</td><td>0.92</td><td>544</td></tr> <tr><td>1</td><td>0.28</td><td>0.32</td><td>0.30</td><td>59</td></tr> <tr><td>2</td><td>0.30</td><td>0.21</td><td>0.25</td><td>42</td></tr> <tr><td>3</td><td>0.00</td><td>0.00</td><td>0.00</td><td>14</td></tr> <tr><td>4</td><td>0.00</td><td>0.00</td><td>0.00</td><td>9</td></tr> </tbody> </table> <table> <tbody> <tr><td>accuracy</td><td></td><td>0.81</td><td>668</td></tr> <tr><td>macro avg</td><td>0.30</td><td>0.29</td><td>0.29</td></tr> <tr><td>weighted avg</td><td>0.77</td><td>0.81</td><td>0.79</td></tr> <tr><td colspan="4">Test AUC (macro-average, OVR): 0.8517"</td></tr> </tbody> </table>		precision	recall	f1-score	support	0	0.89	0.94	0.92	544	1	0.28	0.32	0.30	59	2	0.30	0.21	0.25	42	3	0.00	0.00	0.00	14	4	0.00	0.00	0.00	9	accuracy		0.81	668	macro avg	0.30	0.29	0.29	weighted avg	0.77	0.81	0.79	Test AUC (macro-average, OVR): 0.8517"			
	precision	recall	f1-score	support																																											
0	0.89	0.94	0.92	544																																											
1	0.28	0.32	0.30	59																																											
2	0.30	0.21	0.25	42																																											
3	0.00	0.00	0.00	14																																											
4	0.00	0.00	0.00	9																																											
accuracy		0.81	668																																												
macro avg	0.30	0.29	0.29																																												
weighted avg	0.77	0.81	0.79																																												
Test AUC (macro-average, OVR): 0.8517"																																															
Tuần 6	<p>"Average Accuracy: 0.7141 Average F1 Macro: 0.4241 Average AUC: 0.8767</p> <table> <thead> <tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr> </thead> <tbody> <tr><td>0</td><td>0.88</td><td>0.96</td><td>0.92</td><td>455</td></tr> <tr><td>1</td><td>0.25</td><td>0.15</td><td>0.19</td><td>52</td></tr> <tr><td>2</td><td>0.17</td><td>0.14</td><td>0.15</td><td>29</td></tr> <tr><td>3</td><td>0.25</td><td>0.10</td><td>0.14</td><td>10</td></tr> <tr><td>4</td><td>0.00</td><td>0.00</td><td>0.00</td><td>8</td></tr> </tbody> </table> <table> <tbody> <tr><td>accuracy</td><td></td><td>0.81</td><td>554</td></tr> <tr><td>macro avg</td><td>0.31</td><td>0.27</td><td>0.28</td></tr> <tr><td>weighted avg</td><td>0.76</td><td>0.81</td><td>0.78</td></tr> <tr><td colspan="4">Test AUC (macro-average, OVR): 0.8215"</td></tr> </tbody> </table>		precision	recall	f1-score	support	0	0.88	0.96	0.92	455	1	0.25	0.15	0.19	52	2	0.17	0.14	0.15	29	3	0.25	0.10	0.14	10	4	0.00	0.00	0.00	8	accuracy		0.81	554	macro avg	0.31	0.27	0.28	weighted avg	0.76	0.81	0.78	Test AUC (macro-average, OVR): 0.8215"			
	precision	recall	f1-score	support																																											
0	0.88	0.96	0.92	455																																											
1	0.25	0.15	0.19	52																																											
2	0.17	0.14	0.15	29																																											
3	0.25	0.10	0.14	10																																											
4	0.00	0.00	0.00	8																																											
accuracy		0.81	554																																												
macro avg	0.31	0.27	0.28																																												
weighted avg	0.76	0.81	0.78																																												
Test AUC (macro-average, OVR): 0.8215"																																															
Tuần 8	<p>"Average Accuracy: 0.7169 Average F1 Macro: 0.4216 Average AUC: 0.8786</p>																																														

	precision	recall	f1-score	support
0	0.85	0.99	0.92	395
1	0.30	0.07	0.12	40
2	0.00	0.00	0.00	24
3	0.00	0.00	0.00	10
4	0.00	0.00	0.00	8
accuracy			0.83	477
macro avg	0.23	0.21	0.21	477
weighted avg	0.73	0.83	0.77	477
Test AUC (macro-average, OVR): 0.7800"				

Node2vec

Bộ data cuối cùng được áp dụng graph embedding

Tuần 2	Average Accuracy: 0.7255 Average F1 Macro: 0.4143 Average AUC: 0.8769 precision recall f1-score support 0 0.87 0.98 0.92 568 1 0.59 0.25 0.36 63 2 0.35 0.21 0.26 43 3 0.50 0.07 0.12 15 4 0.67 0.22 0.33 9 accuracy 0.84 698 macro avg 0.59 0.35 0.40 698 weighted avg 0.80 0.84 0.80 698 Test AUC (macro-average, OVR): 0.8901
Tuần 4	Average Accuracy: 0.7138 Average F1 Macro: 0.3951 Average AUC: 0.8749 precision recall f1-score support

	<table> <tbody> <tr><td>0</td><td>0.87</td><td>0.98</td><td>0.92</td><td>544</td></tr> <tr><td>1</td><td>0.36</td><td>0.17</td><td>0.23</td><td>59</td></tr> <tr><td>2</td><td>0.14</td><td>0.07</td><td>0.10</td><td>42</td></tr> <tr><td>3</td><td>0.33</td><td>0.07</td><td>0.12</td><td>14</td></tr> <tr><td>4</td><td>0.00</td><td>0.00</td><td>0.00</td><td>9</td></tr> <tr><td colspan="2">accuracy</td><td>0.83</td><td>668</td><td></td></tr> <tr><td colspan="2">macro avg</td><td>0.34</td><td>0.26</td><td>0.27</td></tr> <tr><td colspan="2">weighted avg</td><td>0.75</td><td>0.82</td><td>0.78</td></tr> <tr><td colspan="5">Test AUC (macro-average, OVR): 0.8781</td></tr> </tbody> </table>	0	0.87	0.98	0.92	544	1	0.36	0.17	0.23	59	2	0.14	0.07	0.10	42	3	0.33	0.07	0.12	14	4	0.00	0.00	0.00	9	accuracy		0.83	668		macro avg		0.34	0.26	0.27	weighted avg		0.75	0.82	0.78	Test AUC (macro-average, OVR): 0.8781									
0	0.87	0.98	0.92	544																																															
1	0.36	0.17	0.23	59																																															
2	0.14	0.07	0.10	42																																															
3	0.33	0.07	0.12	14																																															
4	0.00	0.00	0.00	9																																															
accuracy		0.83	668																																																
macro avg		0.34	0.26	0.27																																															
weighted avg		0.75	0.82	0.78																																															
Test AUC (macro-average, OVR): 0.8781																																																			
Tuần 6	<p>Average Accuracy: 0.6938 Average F1 Macro: 0.3794 Average AUC: 0.8579</p> <table> <thead> <tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr> </thead> <tbody> <tr><td>0</td><td>0.84</td><td>0.99</td><td>0.91</td><td>455</td></tr> <tr><td>1</td><td>0.36</td><td>0.08</td><td>0.13</td><td>52</td></tr> <tr><td>2</td><td>0.00</td><td>0.00</td><td>0.00</td><td>29</td></tr> <tr><td>3</td><td>0.00</td><td>0.00</td><td>0.00</td><td>10</td></tr> <tr><td>4</td><td>0.00</td><td>0.00</td><td>0.00</td><td>8</td></tr> <tr><td colspan="2">accuracy</td><td>0.83</td><td>554</td><td></td></tr> <tr><td colspan="2">macro avg</td><td>0.24</td><td>0.21</td><td>0.21</td></tr> <tr><td colspan="2">weighted avg</td><td>0.72</td><td>0.82</td><td>0.76</td></tr> <tr><td colspan="5">Test AUC (macro-average, OVR): 0.8222</td></tr> </tbody> </table>		precision	recall	f1-score	support	0	0.84	0.99	0.91	455	1	0.36	0.08	0.13	52	2	0.00	0.00	0.00	29	3	0.00	0.00	0.00	10	4	0.00	0.00	0.00	8	accuracy		0.83	554		macro avg		0.24	0.21	0.21	weighted avg		0.72	0.82	0.76	Test AUC (macro-average, OVR): 0.8222				
	precision	recall	f1-score	support																																															
0	0.84	0.99	0.91	455																																															
1	0.36	0.08	0.13	52																																															
2	0.00	0.00	0.00	29																																															
3	0.00	0.00	0.00	10																																															
4	0.00	0.00	0.00	8																																															
accuracy		0.83	554																																																
macro avg		0.24	0.21	0.21																																															
weighted avg		0.72	0.82	0.76																																															
Test AUC (macro-average, OVR): 0.8222																																																			
Tuần 8	<p>Average Accuracy: 0.7307 Average F1 Macro: 0.4432 Average AUC: 0.8834</p> <table> <thead> <tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr> </thead> <tbody> <tr><td>0</td><td>0.85</td><td>0.96</td><td>0.90</td><td>395</td></tr> <tr><td>1</td><td>0.11</td><td>0.05</td><td>0.07</td><td>40</td></tr> <tr><td>2</td><td>0.30</td><td>0.12</td><td>0.18</td><td>24</td></tr> <tr><td>3</td><td>0.00</td><td>0.00</td><td>0.00</td><td>10</td></tr> <tr><td>4</td><td>0.00</td><td>0.00</td><td>0.00</td><td>8</td></tr> <tr><td colspan="2">accuracy</td><td>0.82</td><td>477</td><td></td></tr> <tr><td colspan="2">macro avg</td><td>0.25</td><td>0.23</td><td>0.23</td></tr> <tr><td colspan="2">weighted avg</td><td>0.73</td><td>0.81</td><td>0.76</td></tr> <tr><td colspan="5">Test AUC (macro-average, OVR): 0.8834</td></tr> </tbody> </table>		precision	recall	f1-score	support	0	0.85	0.96	0.90	395	1	0.11	0.05	0.07	40	2	0.30	0.12	0.18	24	3	0.00	0.00	0.00	10	4	0.00	0.00	0.00	8	accuracy		0.82	477		macro avg		0.25	0.23	0.23	weighted avg		0.73	0.81	0.76	Test AUC (macro-average, OVR): 0.8834				
	precision	recall	f1-score	support																																															
0	0.85	0.96	0.90	395																																															
1	0.11	0.05	0.07	40																																															
2	0.30	0.12	0.18	24																																															
3	0.00	0.00	0.00	10																																															
4	0.00	0.00	0.00	8																																															
accuracy		0.82	477																																																
macro avg		0.25	0.23	0.23																																															
weighted avg		0.73	0.81	0.76																																															
Test AUC (macro-average, OVR): 0.8834																																																			

	Test AUC (macro-average, OVR): 0.7413
--	---------------------------------------

SMOTE Node2vec

Bộ data cuối cùng được áp dụng graph embedding và kỹ thuật SMOTE:

Tuần 2	<p>"Average Accuracy: 0.6935 Average F1 Macro: 0.4374 Average AUC: 0.8731</p> <table> <thead> <tr> <th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr> </thead> <tbody> <tr> <td>0</td><td>0.88</td><td>0.96</td><td>0.92</td><td>568</td></tr> <tr> <td>1</td><td>0.40</td><td>0.30</td><td>0.34</td><td>63</td></tr> <tr> <td>2</td><td>0.27</td><td>0.16</td><td>0.20</td><td>43</td></tr> <tr> <td>3</td><td>0.00</td><td>0.00</td><td>0.00</td><td>15</td></tr> <tr> <td>4</td><td>0.00</td><td>0.00</td><td>0.00</td><td>9</td></tr> </tbody> </table> <p>accuracy 0.82 698 macro avg 0.31 0.28 0.29 698 weighted avg 0.77 0.82 0.79 698</p> <p>Test AUC (macro-average, OVR): 0.8531"</p>		precision	recall	f1-score	support	0	0.88	0.96	0.92	568	1	0.40	0.30	0.34	63	2	0.27	0.16	0.20	43	3	0.00	0.00	0.00	15	4	0.00	0.00	0.00	9
	precision	recall	f1-score	support																											
0	0.88	0.96	0.92	568																											
1	0.40	0.30	0.34	63																											
2	0.27	0.16	0.20	43																											
3	0.00	0.00	0.00	15																											
4	0.00	0.00	0.00	9																											
Tuần 4	<p>"Average Accuracy: 0.6775 Average F1 Macro: 0.4395 Average AUC: 0.8572</p> <table> <thead> <tr> <th></th> <th>precision</th> <th>recall</th> <th>f1-score</th> <th>support</th> </tr> </thead> <tbody> <tr> <td>0</td> <td>0.87</td> <td>0.96</td> <td>0.91</td> <td>544</td> </tr> <tr> <td>1</td> <td>0.38</td> <td>0.24</td> <td>0.29</td> <td>59</td> </tr> <tr> <td>2</td> <td>0.23</td> <td>0.17</td> <td>0.19</td> <td>42</td> </tr> <tr> <td>3</td> <td>0.00</td> <td>0.00</td> <td>0.00</td> <td>14</td> </tr> <tr> <td>4</td> <td>0.00</td> <td>0.00</td> <td>0.00</td> <td>9</td> </tr> </tbody> </table> <p>accuracy 0.81 668 macro avg 0.30 0.27 0.28 668 weighted avg 0.76 0.81 0.78 668</p> <p>Test AUC (macro-average, OVR): 0.8409"</p>		precision	recall	f1-score	support	0	0.87	0.96	0.91	544	1	0.38	0.24	0.29	59	2	0.23	0.17	0.19	42	3	0.00	0.00	0.00	14	4	0.00	0.00	0.00	9
	precision	recall	f1-score	support																											
0	0.87	0.96	0.91	544																											
1	0.38	0.24	0.29	59																											
2	0.23	0.17	0.19	42																											
3	0.00	0.00	0.00	14																											
4	0.00	0.00	0.00	9																											
Tuần 6	<p>"Average Accuracy: 0.6843 Average F1 Macro: 0.4405</p>																														

	<p>Average AUC: 0.8596</p> <table> <thead> <tr> <th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr> </thead> <tbody> <tr><td>0</td><td>0.85</td><td>0.97</td><td>0.91</td><td>455</td></tr> <tr><td>1</td><td>0.42</td><td>0.15</td><td>0.23</td><td>52</td></tr> <tr><td>2</td><td>0.14</td><td>0.07</td><td>0.09</td><td>29</td></tr> <tr><td>3</td><td>0.00</td><td>0.00</td><td>0.00</td><td>10</td></tr> <tr><td>4</td><td>0.00</td><td>0.00</td><td>0.00</td><td>8</td></tr> <tr><td>accuracy</td><td></td><td></td><td>0.82</td><td>554</td></tr> <tr><td>macro avg</td><td>0.28</td><td>0.24</td><td>0.25</td><td>554</td></tr> <tr><td>weighted avg</td><td>0.75</td><td>0.82</td><td>0.77</td><td>554</td></tr> </tbody> </table> <p>Test AUC (macro-average, OVR): 0.8172"</p>		precision	recall	f1-score	support	0	0.85	0.97	0.91	455	1	0.42	0.15	0.23	52	2	0.14	0.07	0.09	29	3	0.00	0.00	0.00	10	4	0.00	0.00	0.00	8	accuracy			0.82	554	macro avg	0.28	0.24	0.25	554	weighted avg	0.75	0.82	0.77	554
	precision	recall	f1-score	support																																										
0	0.85	0.97	0.91	455																																										
1	0.42	0.15	0.23	52																																										
2	0.14	0.07	0.09	29																																										
3	0.00	0.00	0.00	10																																										
4	0.00	0.00	0.00	8																																										
accuracy			0.82	554																																										
macro avg	0.28	0.24	0.25	554																																										
weighted avg	0.75	0.82	0.77	554																																										
Tuần 8	<p>"Average Accuracy: 0.7054 Average F1 Macro: 0.4674 Average AUC: 0.8663</p> <table> <thead> <tr> <th></th> <th>precision</th> <th>recall</th> <th>f1-score</th> <th>support</th> </tr> </thead> <tbody> <tr><td>0</td><td>0.86</td><td>0.96</td><td>0.91</td><td>395</td></tr> <tr><td>1</td><td>0.25</td><td>0.12</td><td>0.17</td><td>40</td></tr> <tr><td>2</td><td>0.14</td><td>0.08</td><td>0.11</td><td>24</td></tr> <tr><td>3</td><td>0.00</td><td>0.00</td><td>0.00</td><td>10</td></tr> <tr><td>4</td><td>0.00</td><td>0.00</td><td>0.00</td><td>8</td></tr> <tr><td>accuracy</td><td></td><td></td><td>0.81</td><td>477</td></tr> <tr><td>macro avg</td><td>0.25</td><td>0.23</td><td>0.24</td><td>477</td></tr> <tr><td>weighted avg</td><td>0.74</td><td>0.81</td><td>0.77</td><td>477</td></tr> </tbody> </table> <p>Test AUC (macro-average, OVR): 0.7560"</p>		precision	recall	f1-score	support	0	0.86	0.96	0.91	395	1	0.25	0.12	0.17	40	2	0.14	0.08	0.11	24	3	0.00	0.00	0.00	10	4	0.00	0.00	0.00	8	accuracy			0.81	477	macro avg	0.25	0.23	0.24	477	weighted avg	0.74	0.81	0.77	477
	precision	recall	f1-score	support																																										
0	0.86	0.96	0.91	395																																										
1	0.25	0.12	0.17	40																																										
2	0.14	0.08	0.11	24																																										
3	0.00	0.00	0.00	10																																										
4	0.00	0.00	0.00	8																																										
accuracy			0.81	477																																										
macro avg	0.25	0.23	0.24	477																																										
weighted avg	0.74	0.81	0.77	477																																										

NodeClusterLevel

Bộ data cuối cùng được áp dụng graph Cluster

Tuần 2	<p>Average Accuracy: 0.7341 Average F1 Macro: 0.4494 Average AUC: 0.8897</p> <table border="1" data-bbox="621 377 1273 810"> <thead> <tr> <th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr> </thead> <tbody> <tr><td>0</td><td>0.87</td><td>0.98</td><td>0.92</td><td>568</td></tr> <tr><td>1</td><td>0.38</td><td>0.21</td><td>0.27</td><td>63</td></tr> <tr><td>2</td><td>0.38</td><td>0.21</td><td>0.27</td><td>43</td></tr> <tr><td>3</td><td>0.00</td><td>0.00</td><td>0.00</td><td>15</td></tr> <tr><td>4</td><td>0.00</td><td>0.00</td><td>0.00</td><td>9</td></tr> <tr><td colspan="2"></td><td>accuracy</td><td>0.83</td><td>698</td></tr> <tr><td colspan="2"></td><td>macro avg</td><td>0.33</td><td>0.28 0.29 698</td></tr> <tr><td colspan="2"></td><td>weighted avg</td><td>0.77</td><td>0.83 0.79 698</td></tr> </tbody> </table> <p>Test AUC (macro-average, OVR): 0.9005</p>		precision	recall	f1-score	support	0	0.87	0.98	0.92	568	1	0.38	0.21	0.27	63	2	0.38	0.21	0.27	43	3	0.00	0.00	0.00	15	4	0.00	0.00	0.00	9			accuracy	0.83	698			macro avg	0.33	0.28 0.29 698			weighted avg	0.77	0.83 0.79 698
	precision	recall	f1-score	support																																										
0	0.87	0.98	0.92	568																																										
1	0.38	0.21	0.27	63																																										
2	0.38	0.21	0.27	43																																										
3	0.00	0.00	0.00	15																																										
4	0.00	0.00	0.00	9																																										
		accuracy	0.83	698																																										
		macro avg	0.33	0.28 0.29 698																																										
		weighted avg	0.77	0.83 0.79 698																																										
Tuần 4	<p>Average Accuracy: 0.7131 Average F1 Macro: 0.4168 Average AUC: 0.8764</p> <table border="1" data-bbox="621 1073 1273 1507"> <thead> <tr> <th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr> </thead> <tbody> <tr><td>0</td><td>0.89</td><td>0.94</td><td>0.92</td><td>544</td></tr> <tr><td>1</td><td>0.28</td><td>0.32</td><td>0.30</td><td>59</td></tr> <tr><td>2</td><td>0.30</td><td>0.21</td><td>0.25</td><td>42</td></tr> <tr><td>3</td><td>0.00</td><td>0.00</td><td>0.00</td><td>14</td></tr> <tr><td>4</td><td>0.00</td><td>0.00</td><td>0.00</td><td>9</td></tr> <tr><td colspan="2"></td><td>accuracy</td><td>0.81</td><td>668</td></tr> <tr><td colspan="2"></td><td>macro avg</td><td>0.30</td><td>0.29 0.29 668</td></tr> <tr><td colspan="2"></td><td>weighted avg</td><td>0.77</td><td>0.81 0.79 668</td></tr> </tbody> </table> <p>Test AUC (macro-average, OVR): 0.8517</p>		precision	recall	f1-score	support	0	0.89	0.94	0.92	544	1	0.28	0.32	0.30	59	2	0.30	0.21	0.25	42	3	0.00	0.00	0.00	14	4	0.00	0.00	0.00	9			accuracy	0.81	668			macro avg	0.30	0.29 0.29 668			weighted avg	0.77	0.81 0.79 668
	precision	recall	f1-score	support																																										
0	0.89	0.94	0.92	544																																										
1	0.28	0.32	0.30	59																																										
2	0.30	0.21	0.25	42																																										
3	0.00	0.00	0.00	14																																										
4	0.00	0.00	0.00	9																																										
		accuracy	0.81	668																																										
		macro avg	0.30	0.29 0.29 668																																										
		weighted avg	0.77	0.81 0.79 668																																										
Tuần 6	<p>Average Accuracy: 0.7141 Average F1 Macro: 0.4241 Average AUC: 0.8767</p> <table border="1" data-bbox="621 1821 1273 2028"> <thead> <tr> <th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr> </thead> <tbody> <tr><td>0</td><td>0.88</td><td>0.96</td><td>0.92</td><td>455</td></tr> <tr><td>1</td><td>0.25</td><td>0.15</td><td>0.19</td><td>52</td></tr> <tr><td>2</td><td>0.17</td><td>0.14</td><td>0.15</td><td>29</td></tr> </tbody> </table>		precision	recall	f1-score	support	0	0.88	0.96	0.92	455	1	0.25	0.15	0.19	52	2	0.17	0.14	0.15	29																									
	precision	recall	f1-score	support																																										
0	0.88	0.96	0.92	455																																										
1	0.25	0.15	0.19	52																																										
2	0.17	0.14	0.15	29																																										

	<table> <tbody> <tr><td>3</td><td>0.25</td><td>0.10</td><td>0.14</td><td>10</td></tr> <tr><td>4</td><td>0.00</td><td>0.00</td><td>0.00</td><td>8</td></tr> <tr><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td>accuracy</td><td></td><td>0.81</td><td>554</td></tr> <tr><td></td><td>macro avg</td><td>0.31</td><td>0.27</td><td>0.28 554</td></tr> <tr><td></td><td>weighted avg</td><td>0.76</td><td>0.81</td><td>0.78 554</td></tr> <tr><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td>Test AUC (macro-average, OVR):</td><td>0.8215</td><td></td><td></td></tr> </tbody> </table>	3	0.25	0.10	0.14	10	4	0.00	0.00	0.00	8							accuracy		0.81	554		macro avg	0.31	0.27	0.28 554		weighted avg	0.76	0.81	0.78 554							Test AUC (macro-average, OVR):	0.8215																						
3	0.25	0.10	0.14	10																																																									
4	0.00	0.00	0.00	8																																																									
	accuracy		0.81	554																																																									
	macro avg	0.31	0.27	0.28 554																																																									
	weighted avg	0.76	0.81	0.78 554																																																									
	Test AUC (macro-average, OVR):	0.8215																																																											
Tuần 8	<p>Average Accuracy: 0.7169 Average F1 Macro: 0.4216 Average AUC: 0.8786</p> <table> <thead> <tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr> </thead> <tbody> <tr><td>0</td><td>0.85</td><td>0.99</td><td>0.92</td><td>395</td></tr> <tr><td>1</td><td>0.30</td><td>0.07</td><td>0.12</td><td>40</td></tr> <tr><td>2</td><td>0.00</td><td>0.00</td><td>0.00</td><td>24</td></tr> <tr><td>3</td><td>0.00</td><td>0.00</td><td>0.00</td><td>10</td></tr> <tr><td>4</td><td>0.00</td><td>0.00</td><td>0.00</td><td>8</td></tr> <tr><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td>accuracy</td><td></td><td>0.83</td><td>477</td></tr> <tr><td></td><td>macro avg</td><td>0.23</td><td>0.21</td><td>0.21 477</td></tr> <tr><td></td><td>weighted avg</td><td>0.73</td><td>0.83</td><td>0.77 477</td></tr> <tr><td></td><td></td><td></td><td></td><td></td></tr> <tr><td></td><td>Test AUC (macro-average, OVR):</td><td>0.7800</td><td></td><td></td></tr> </tbody> </table>		precision	recall	f1-score	support	0	0.85	0.99	0.92	395	1	0.30	0.07	0.12	40	2	0.00	0.00	0.00	24	3	0.00	0.00	0.00	10	4	0.00	0.00	0.00	8							accuracy		0.83	477		macro avg	0.23	0.21	0.21 477		weighted avg	0.73	0.83	0.77 477							Test AUC (macro-average, OVR):	0.7800		
	precision	recall	f1-score	support																																																									
0	0.85	0.99	0.92	395																																																									
1	0.30	0.07	0.12	40																																																									
2	0.00	0.00	0.00	24																																																									
3	0.00	0.00	0.00	10																																																									
4	0.00	0.00	0.00	8																																																									
	accuracy		0.83	477																																																									
	macro avg	0.23	0.21	0.21 477																																																									
	weighted avg	0.73	0.83	0.77 477																																																									
	Test AUC (macro-average, OVR):	0.7800																																																											

SMOTE NodeClusterLevel

Bộ data cuối cùng được áp dụng graph Cluster và kỹ thuật SMOTE:

Tuần 2	<p>"Average Accuracy: 0.6934 Average F1 Macro: 0.4570 Average AUC: 0.8771</p> <table> <thead> <tr><th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr> </thead> <tbody> <tr><td>0</td><td>0.88</td><td>0.94</td><td>0.91</td><td>568</td></tr> <tr><td>1</td><td>0.33</td><td>0.29</td><td>0.31</td><td>63</td></tr> <tr><td>2</td><td>0.31</td><td>0.21</td><td>0.25</td><td>43</td></tr> <tr><td>3</td><td>0.00</td><td>0.00</td><td>0.00</td><td>15</td></tr> <tr><td>4</td><td>0.14</td><td>0.11</td><td>0.12</td><td>9</td></tr> </tbody> </table>		precision	recall	f1-score	support	0	0.88	0.94	0.91	568	1	0.33	0.29	0.31	63	2	0.31	0.21	0.25	43	3	0.00	0.00	0.00	15	4	0.14	0.11	0.12	9
	precision	recall	f1-score	support																											
0	0.88	0.94	0.91	568																											
1	0.33	0.29	0.31	63																											
2	0.31	0.21	0.25	43																											
3	0.00	0.00	0.00	15																											
4	0.14	0.11	0.12	9																											

	<p style="text-align: center;">accuracy 0.80 698 macro avg 0.33 0.31 0.32 698 weighted avg 0.77 0.80 0.78 698</p> <p>Test AUC (macro-average, OVR): 0.8408"</p>																														
Tuần 4	<p>"Average Accuracy: 0.6770 Average F1 Macro: 0.4529 Average AUC: 0.8633</p> <table style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr> </thead> <tbody> <tr> <td>0</td><td>0.90</td><td>0.95</td><td>0.92</td><td>544</td></tr> <tr> <td>1</td><td>0.35</td><td>0.27</td><td>0.30</td><td>59</td></tr> <tr> <td>2</td><td>0.33</td><td>0.36</td><td>0.34</td><td>42</td></tr> <tr> <td>3</td><td>0.00</td><td>0.00</td><td>0.00</td><td>14</td></tr> <tr> <td>4</td><td>0.00</td><td>0.00</td><td>0.00</td><td>9</td></tr> </tbody> </table> <p style="text-align: center;">accuracy 0.82 668 macro avg 0.32 0.32 0.31 668 weighted avg 0.78 0.82 0.80 668</p> <p>Test AUC (macro-average, OVR): 0.8558"</p>		precision	recall	f1-score	support	0	0.90	0.95	0.92	544	1	0.35	0.27	0.30	59	2	0.33	0.36	0.34	42	3	0.00	0.00	0.00	14	4	0.00	0.00	0.00	9
	precision	recall	f1-score	support																											
0	0.90	0.95	0.92	544																											
1	0.35	0.27	0.30	59																											
2	0.33	0.36	0.34	42																											
3	0.00	0.00	0.00	14																											
4	0.00	0.00	0.00	9																											
Tuần 6	<p>"Average Accuracy: 0.6675 Average F1 Macro: 0.4384 Average AUC: 0.8553</p> <table style="margin-left: auto; margin-right: auto;"> <thead> <tr> <th></th><th>precision</th><th>recall</th><th>f1-score</th><th>support</th></tr> </thead> <tbody> <tr> <td>0</td><td>0.85</td><td>0.94</td><td>0.89</td><td>455</td></tr> <tr> <td>1</td><td>0.22</td><td>0.15</td><td>0.18</td><td>52</td></tr> <tr> <td>2</td><td>0.25</td><td>0.07</td><td>0.11</td><td>29</td></tr> <tr> <td>3</td><td>0.00</td><td>0.00</td><td>0.00</td><td>10</td></tr> <tr> <td>4</td><td>0.00</td><td>0.00</td><td>0.00</td><td>8</td></tr> </tbody> </table> <p style="text-align: center;">accuracy 0.79 554 macro avg 0.26 0.23 0.24 554 weighted avg 0.73 0.79 0.76 554</p> <p>Test AUC (macro-average, OVR): 0.7390"</p>		precision	recall	f1-score	support	0	0.85	0.94	0.89	455	1	0.22	0.15	0.18	52	2	0.25	0.07	0.11	29	3	0.00	0.00	0.00	10	4	0.00	0.00	0.00	8
	precision	recall	f1-score	support																											
0	0.85	0.94	0.89	455																											
1	0.22	0.15	0.18	52																											
2	0.25	0.07	0.11	29																											
3	0.00	0.00	0.00	10																											
4	0.00	0.00	0.00	8																											
Tuần 8	<p>"Average Accuracy: 0.6919 Average F1 Macro: 0.4527 Average AUC: 0.8672</p>																														

	precision	recall	f1-score	support
0	0.87	0.93	0.90	395
1	0.13	0.10	0.11	40
2	0.24	0.21	0.22	24
3	0.00	0.00	0.00	10
4	0.00	0.00	0.00	8
accuracy			0.79	477
macro avg		0.25	0.25	0.25
weighted avg		0.74	0.79	0.77
Test AUC (macro-average, OVR): 0.7839"				

5.5.2.6 Nhận xét

1. Trên bộ dữ liệu phân ban đầu

- Hiệu năng mô hình:
- Mô hình cây (XGBoost, RandomForest, LightGBM):
 - Accuracy cao (~0.83).
 - LightGBM có F1-macro cao nhất, tăng từ 0.39 → 0.46 qua các tuần.
 - Có xu hướng giảm nhẹ khi số tuần tăng : ví dụ, CatBoost giảm từ 0.83 → 0.81.
- Mô hình Deep Learning (LSTM, BiLSTM, Stacked LSTM):
 - Có xu hướng cải thiện theo thời gian (Stacked LSTM tăng từ 0.81 → 0.83).
 - Mô hình không dựa trên chuỗi thời gian (CNN, GNN):
 - Kết quả không ổn định, biến động mạnh.
- Nhận xét:
 - F1-macro < Accuracy → dữ liệu mất cân bằng nhãn.
 - Các mô hình cây cho kết quả vượt trội, nhất là trong giai đoạn đầu.

2. Trên bộ dữ liệu có thêm thông tin cạnh và cụm (cluster)

- Hiệu năng cải thiện:
 - XGBoost: AUC tăng từ 0.87 → 0.89.
 - F1-macro tăng nhẹ:
 - LightGBM: 0.40 → 0.42

■ XGBoost: $0.39 \rightarrow 0.40$

- Accuracy gần như không thay đổi, cho thấy cải thiện tập trung ở khả năng phân biệt lớp (AUC) hơn là độ chính xác tổng thể.
- Nhận xét:
 - Thêm đặc trưng cạnh & cụm giúp nâng cao AUC, nhất là với mô hình cây.

3. Trên bộ dữ liệu áp dụng node2vec

- Hiệu năng:
 - Deep learning (LSTM): hoạt động ổn định, accuracy 0.82–0.83 qua 4 giai đoạn.
 - Tree-based models: Accuracy không thay đổi nhiều nhưng AUC cải thiện đáng kể nhờ đặc trưng node2vec.
- Nhận xét:
 - Node2vec không làm tăng accuracy nhưng cải thiện khả năng phân loại (AUC) tốt, phù hợp với các mô hình cây.

4. Trên bộ dữ liệu tăng cường bằng SVMSMOTE

- Kết quả:
 - Trên tập huấn luyện: Accuracy và F1 cao hơn khi train bộ dữ liệu cũ.
 - Trên tập kiểm tra: giảm hiệu năng về accuracy nhưng F1 có sự tăng nhẹ và xu hướng dự đoán được các loại thiểu số:
 - LightGBM: từ accuracy $0.81 \rightarrow 0.79$ (tuần 8) nhưng F1 macro từ 0.25 đến 0.27 .
- Nhận xét:
 - SMOTE có thể thêm những đặc trưng mới phù hợp với dữ liệu train nhưng không phản ảnh tốt dữ liệu trong tương lai.
 - Dữ liệu tăng cường bằng SMOTE, không cải thiện hiệu năng nhưng độ chính xác tăng nhẹ.

5. Tổng kết hiệu năng mô hình

- ROC-AUC (macro): Random Forest, XGBoost, CatBoost, LightGBM có AUC cao nhất.
- Accuracy: XGBoost, LightGBM, CatBoost hiệu quả tốt ở các tuần đầu.
- F1-Macro: Thấp hơn Accuracy → bài toán có mất cân bằng nhãn rõ rệt.
- Deep Learning: LSTM, BiLSTM, Stacked LSTM ổn định theo các tuần.

- Mô hình yếu: SVM, Naive Bayes, KNN, TabNet có hiệu năng thấp → nên loại bỏ.
- Mô hình mới nỗi SNN hoạt động ổn định, tiềm năng.
- Cân bằng giữa Accuracy và F1-macro là model LightGBM có giá trị cao trong data filtered_data

6. Mô hình và bộ dữ liệu để xuất triển khai: XGBoost và bộ dữ liệu Node2vec

a. Bộ dữ liệu sau khi làm giàu bằng Node2vec

Bộ dữ liệu đầu vào đã được mở rộng từ dữ liệu gốc bằng cách thêm **vector embedding** kích thước **16 chiều** từ thuật toán Node2Vec, biểu diễn **cấu trúc đồ thị quan hệ giữa học viên và khóa học**. Cụ thể:

- Mỗi học viên và mỗi khóa học được biểu diễn bằng một vector embedding 16 chiều, học từ **đồ thị hai lớp bipartite** gồm các cạnh có trọng số phản ánh tương tác học tập.
- Các vector này chứa thông tin **vị trí tương đối, cộng đồng, mức độ kết nối và vai trò của từng node trong đồ thị học tập**.
- Embedding thể hiện tính phân cụm rõ ràng, với nhiều chiều có **hệ số tương quan âm rõ rệt với nhãn** → chứng tỏ khả năng phân biệt học viên theo hành vi học tập hoặc nguy cơ bỏ học.

→ Như vậy, bộ dữ liệu đầu vào trở thành **tập dữ liệu bảng với đặc trưng liên tục, phi tuyến và giàu thông tin đồ thị**, phù hợp để khai thác bằng các mô hình có khả năng học phi tuyến hiệu quả.

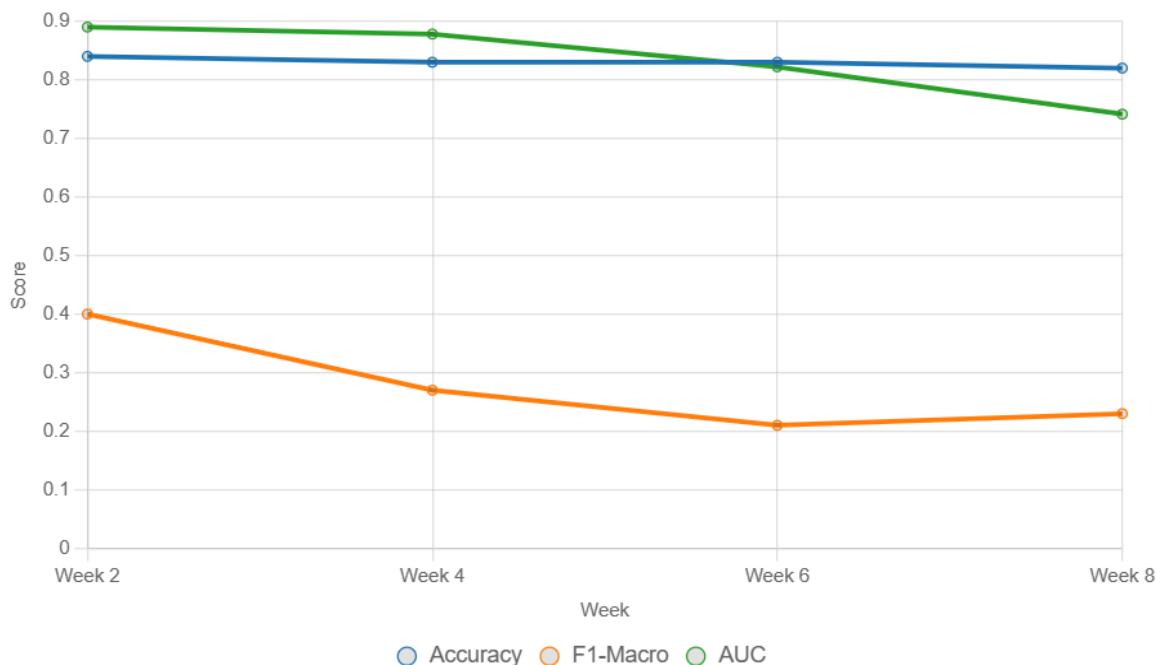
b. Tại sao chọn XGBoost là mô hình chính

XGBoost được lựa chọn khi sử dụng với bộ dữ liệu đã làm giàu bằng Node2Vec vì các lý do sau:

1. **Xử lý tốt dữ liệu mất cân bằng:** XGBoost cải thiện khả năng phân loại các lớp thiểu số, thể hiện qua **F1-macro** và **AUC** cao hơn so với các mô hình khác.
2. **Tận dụng đặc trưng Node2Vec:** XGBoost khai thác hiệu quả các đặc trưng phi tuyến và tương tác phức tạp từ embedding Node2Vec, đặc biệt là các chiều có tương quan cao với nhãn.
3. **Hiệu suất AUC cao:** Node2Vec giúp tăng AUC của XGBoost (lên đến 0.8901 ở tuần 2), chứng minh sự phù hợp của mô hình với dữ liệu đồ thị.
4. **Hiệu quả tính toán:** XGBoost nhanh hơn và ít tốn tài nguyên hơn so với các mô hình deep learning, phù hợp với tập dữ liệu có quy mô vừa và nhỏ.

5. **Robustness với SMOTE:** XGBoost duy trì hiệu năng ổn định khi kết hợp với SMOTE, cải thiện khả năng phân loại lớp thiểu số mà không làm giảm quá nhiều Accuracy.
6. **Tính linh hoạt và tối ưu hóa:** XGBoost cung cấp nhiều tham số có thể điều chỉnh, giúp tối ưu hóa mô hình cho bài toán cụ thể, đồng thời chống overfitting tốt hơn so với các mô hình cây khác.

Để minh họa hiệu suất của XGBoost trên bộ dữ liệu Node2Vec qua các tuần, dưới đây là biểu đồ so sánh Accuracy, F1-macro, và AUC:



Nhận xét từ biểu đồ:

- Accuracy duy trì ổn định (~0.82–0.84), cho thấy XGBoost hoạt động tốt trên toàn bộ các tuần.
- F1-macro có xu hướng giảm nhẹ qua các tuần, phản ánh khó khăn trong việc phân loại các lớp thiểu số khi dữ liệu trở nên phức tạp hơn.
- AUC giảm dần từ tuần 2 đến tuần 8, nhưng vẫn ở mức cao (0.7413–0.8901), chứng minh rằng XGBoost tận dụng tốt đặc trưng Node2Vec để phân biệt các lớp.

Kết luận: XGBoost là lựa chọn tối ưu khi sử dụng với bộ dữ liệu đã làm giàu bằng Node2Vec nhờ khả năng xử lý dữ liệu mảnh cân bằng, khai thác hiệu quả các đặc trưng phi tuyến từ embedding, và duy trì hiệu suất cao về AUC và Accuracy. Kết quả thực nghiệm (AUC lên đến 0.8901, Accuracy ~0.82–0.84) cùng với tính linh hoạt và hiệu quả tính toán của XGBoost củng cố quyết định này.

CHƯƠNG 6. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

Nghiên cứu này đã khai thác và phân tích hiệu quả dữ liệu từ MOOCubeX nhằm xây dựng hệ thống **dự đoán sớm kết quả học tập của người học theo 5 mức độ phân loại** trong các khóa học MOOC. Việc sử dụng kỹ thuật trích xuất đặc trưng quan hệ dựa trên đồ thị kết hợp với các mô hình học máy hiện đại đã mang lại những kết quả đầy hứa hẹn. Dưới đây là tổng kết về kết quả đạt được, những hạn chế còn tồn tại và các định hướng phát triển trong tương lai:

6.1. Kết quả đạt được

Xử lý và phân tích dữ liệu quy mô lớn: Nghiên cứu đã chứng minh khả năng xử lý, làm sạch và phân tích dữ liệu MOOC phức tạp, không đồng nhất và có khối lượng lớn. Quá trình này bao gồm việc xử lý dữ liệu thiếu, nhiễu, và định dạng không nhất quán – từ đó tạo nền tảng vững chắc cho mô hình dự đoán.

Trích xuất đặc trưng quan hệ từ đồ thị: Nhóm đã xây dựng thành công biểu diễn đồ thị giữa các thực thể trong khóa học (người học, học phần, tài nguyên học tập...) và trích xuất các đặc trưng quan hệ có ý nghĩa. Đây là điểm nổi bật giúp mô hình học máy khai thác thông tin ngữ cảnh và liên kết giữa các thành phần một cách hiệu quả.

Xây dựng mô hình phân loại theo 5 mức độ: Từ các đặc trưng trích xuất, hệ thống đã huấn luyện mô hình học máy có khả năng phân loại kết quả học tập theo 5 mức độ, giúp phản ánh chi tiết hơn về hiệu suất của người học.

Triển khai mô hình trên nền tảng điện toán đám mây: Dự án đã tích hợp công nghệ đám mây để triển khai hệ thống dự đoán, bao gồm lưu trữ, xử lý và cung cấp kết quả dự đoán theo thời gian thực. Hệ thống có khả năng cảnh báo sớm nguy cơ học tập thấp để hỗ trợ người học và giảng viên can thiệp kịp thời.

6.2. Hạn chế

Thời gian xử lý dài: Với khối lượng dữ liệu lớn và tính phức tạp cao, quá trình xử lý và huấn luyện mô hình cần nhiều thời gian, đặc biệt khi xử lý dữ liệu theo quan hệ đồ thị.

Chưa tự động hóa hoàn toàn: Quy trình ETL hiện tại vẫn còn một số công đoạn thủ công như chuyển đổi định dạng hoặc xử lý ngoại lệ, gây tốn công sức và dễ xảy ra sai sót.

Vấn đề về chất lượng và tính nhất quán của dữ liệu: Dữ liệu đầu vào có thể chứa thông tin thiếu, nhiễu hoặc sai lệch về thời gian, ảnh hưởng đến tính chính xác của việc phân tích theo dòng thời gian.

Chưa đa dạng trong việc khai thác dữ liệu ngữ nghĩa và bối cảnh: Mỗi quan hệ giữa người học và khóa học đôi khi chưa được khai thác triệt để do hạn chế về kỹ thuật biểu diễn ngữ nghĩa trong đồ thị.

6.3. Hướng phát triển trong tương lai

6.3.1. Tối ưu hóa và tự động hóa mô hình

- Tự động hóa hoàn toàn quy trình ETL/ELT và pipeline huấn luyện nhằm cập nhật mô hình liên tục với dữ liệu mới.
- Áp dụng AutoML để tìm kiến trúc và tham số tối ưu, giảm thiểu can thiệp thủ công.
- Khai thác các thuật toán mới như GNN, Transformer hoặc mô hình lai để tăng hiệu suất dự đoán trên dữ liệu đồ thị và chuỗi thời gian.

6.3.2. Mở rộng và nâng cao chất lượng dữ liệu

- Mở rộng dữ liệu sang nhiều nền tảng MOOC khác nhau nhằm tăng tính khái quát của mô hình.
- Bổ sung đặc trưng hành vi nâng cao như thời gian xem video, mức độ tham gia diễn đàn, tương tác theo ngữ cảnh.
- Khai thác thông tin cảm xúc từ bình luận, phản hồi để đưa ra cảnh báo sớm chính xác hơn.

6.3.3. Xây dựng hệ thống cảnh báo sớm toàn diện

- Triển khai hệ thống phát hiện sớm người học có dấu hiệu bỏ cuộc hoặc kết quả kém, từ đó cung cấp gợi ý cải thiện hoặc hỗ trợ học tập.
- Kết hợp dashboard phân tích giúp giảng viên theo dõi hiệu quả các học phần và điều chỉnh phương pháp giảng dạy phù hợp.

6.3.4. Phát triển ứng dụng thực tế

- Xây dựng giao diện web thân thiện, hỗ trợ cả người học và giảng viên trong việc theo dõi tiến độ học tập và nhận cảnh báo.
- Tích hợp chức năng cá nhân hóa lộ trình học tập dựa trên kết quả dự đoán, từ đó tăng tỷ lệ hoàn thành khóa học.
- Triển khai đầy đủ hệ thống trên các dịch vụ đám mây như AWS, Azure để đảm bảo khả năng mở rộng linh hoạt, phục vụ hàng nghìn người dùng đồng thời.

TÀI LIỆU THAM KHẢO

- [1] “ANN-LSTM: A deep learning model for early student performance prediction in MOOC” (Fatima Ahmed Al-azazi, Mossa Ghurab, 2023), [Trực tuyến]. Available: <https://www.sciencedirect.com/science/article/pii/S2405844023025896>
- [2] “Analysis of the Factors Influencing Learners’ Performance Prediction With Learning Analytics” (Pedro Manuel Moreno-Marcos, Ángel Castellanos-Nieves, Ángel J. García-Cabot, María L. Cano, Carlos Fernández-Panadero, 2020), [Trực tuyến]. Available: <https://ieeexplore.ieee.org/document/8948025>
- [3] “Identifying the Factors Affecting Student Academic Performance and Engagement Prediction in MOOC Using Deep Learning: A Systematic Literature Review” (Shahzad Rizwan, Asad Masood Khattak, Hafeez Anwar, Ibrahim Abaker Targio Hashem, 2025), [Trực tuyến]. Available: <https://ieeexplore.ieee.org/document/10852293>
- [4] “MOOC performance prediction by Deep Learning from raw clickstream data” (Kőrösí Gábor, Richard Farkas, 2020), [Trực tuyến]. Available: https://www.researchgate.net/publication/343036894_MOOC_Performance_Prediction_by_Deep_Learning_from_Raw_Clickstream_Data
- [5] “The Crowd in MOOCs: A Study of Learning Patterns at Scale” (Xin Zhou, Aixin Sun, Jie Zhang, Donghui Lin, 2024), [Trực tuyến]. Available: <https://arxiv.org/abs/2408.03025>
- [6] “Enhancing academic performance prediction with temporal graph networks for massive open online courses” (Huang, Zhewei and Chen, Yukun, 2024), [Trực tuyến]. Available: <https://journalofbigdata.springeropen.com/articles/10.1186/s40537-024-00918-5>
- [7] “Meta Transfer Learning for Early Success Prediction in MOOCs” (Vinitra Swamy, Mirko Marras, Tanja Käser, 2024), [Trực tuyến]. Available: <https://arxiv.org/abs/2205.01064>