

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



MÔN MẠNG XÃ HỘI
BÁO CÁO ĐỒ ÁN CUỐI KỲ

**Đề tài: DỰ ĐOÁN KHẢ NĂNG BỎ HỌC CỦA HỌC VIÊN ĐỐI
VỚI KHÓA HỌC**

GVHD: ThS. Nguyễn Thị Anh Thư

Nhóm sinh viên thực hiện: Nhóm 11

- | | |
|---------------------|----------------|
| 1. Đoàn Bảo Long | MSSV: 21520332 |
| 2. Cao Thiên An | MSSV: 22520008 |
| 3. Cao Quốc Kiệt | MSSV: 22520714 |
| 4. Lê Thiên Kim | MSSV: 22520728 |
| 5. Thạch Minh Luân | MSSV: 22520827 |
| 6. Đỗ Nguyên Phương | MSSV: 22521159 |

LỜI CẢM ƠN

Chúng em xin gửi lời cảm ơn chân thành và sâu sắc nhất đến Ths Nguyễn Thị Anh Thư, người đã tận tình hướng dẫn và chỉ dẫn chúng em trong suốt quá trình thực hiện đề tài. Sự tận tâm và nhiệt huyết của cô không chỉ giúp chúng em hoàn thành tốt công việc mà còn truyền cảm hứng và động lực để chúng em không ngừng học hỏi và phát triển.

Cô đã dành nhiều thời gian và công sức để giải đáp mọi thắc mắc, hướng dẫn chi tiết và chia sẻ những kinh nghiệm quý báu. Nhờ sự chỉ dẫn tận tình của cô, chúng em đã có thể vượt qua những khó khăn và đạt được những kết quả nhất định.

Một lần nữa, chúng em xin chân thành cảm ơn cô và mong rằng sẽ tiếp tục nhận được sự hỗ trợ và chỉ dẫn từ cô trong những chặng đường học tập và nghiên cứu tiếp theo.

MỤC LỤC

Chương 1. TỔNG QUAN	5
1.1Giới thiệu	5
1.2Định nghĩa bài toán.....	5
1.2.1 Loại bài toán	5
1.2.2 Ngữ cảnh.....	5
1.2.3 Đầu vào bài toán	6
1.2.4 Đầu ra bài toán.....	6
1.3Ứng dụng.....	6
1.4Phạm vi thực hiện	7
Chương 2. TỔNG QUAN VỀ BỘ DỮ LIỆU	8
2.1Giới thiệu bộ dữ liệu đã sử dụng.....	8
2.2Mô tả chi tiết bộ dữ liệu.....	8
2.2.1 Tracking Log.....	8
2.2.2 Dropout Prediction Dataset.....	9
2.2.3 User Profile	9
2.2.4 Course Info	10
2.3Khám phá (EDA) và trực quan hóa bộ dữ liệu	11
2.3.1 course_info.csv	11
2.3.2 23	
2.3.3 user_info.csv	23
2.3.4 train_truth.csv - dữ liệu về thông tin bỏ học.....	30
2.3.5 train_log.csv - dữ liệu nhật ký hoạt động	32
Chương 3. QUÁ TRÌNH THỰC NGHIỆM.....	41
3.1Xử lý dữ liệu	41
3.1.1 Tập dữ liệu course_info	41
3.1.2 Tập dữ liệu user_info	48
3.1.3 Tập dữ liệu prediction_log.....	53
3.2Trích xuất đặc trưng.....	54
3.2.1 Đặc trưng khoá học.....	54
3.2.2 Đặc trưng học viên.....	57

3.2.3	Đặc trưng hoạt động	62
3.3	Huấn luyện mô hình và đánh giá	65
3.4	Xây dựng đồ thị mạng.....	66
Chương 5.	KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN.....	70
5.1	Kết luận	70
5.2	Hướng phát triển	70
TÀI LIỆU THAM KHẢO	71

Chương 1. TỔNG QUAN

1.1 Giới thiệu

Trong kỷ nguyên số hóa hiện nay, học trực tuyến đã trở thành một phần quan trọng trong giáo dục. Hình thức này mang lại nhiều lợi ích như sự linh hoạt và tiện lợi, nhưng việc duy trì sự tham gia của học viên trong các khóa học trực tuyến vẫn là một thách thức đối với giáo viên và các nhà quản lý. Một vấn đề then chốt là làm thế nào để dự báo và ngăn chặn tình trạng bỏ học một cách hiệu quả trước khi nó xảy ra.

MOOCs (Massive Open Online Courses) là những nền tảng học trực tuyến quy mô lớn, thu hút hàng triệu người tham gia trên khắp thế giới. Các nền tảng này cung cấp nhiều tài liệu học tập, bài giảng video, bài tập, và cơ hội giao tiếp trực tuyến. Tuy nhiên, một vấn đề nghiêm trọng được đặt ra cho MOOCs là làm thế nào để duy trì động lực học tập và ngăn chặn việc bỏ học.

Nghiên cứu này đề xuất áp dụng các phương pháp học máy và phân tích dữ liệu để dự báo khả năng bỏ học của học viên dựa trên dữ liệu thu thập từ các hoạt động như xem video, hoàn thành bài tập, tương tác với giáo viên, và góp ý kiến. Việc phát triển mô hình dự báo sẽ giúp nhà quản lý giáo dục có được những đề xuất hợp lý để can thiệp kịp thời, nâng cao hiệu quả học tập.

1.2 Định nghĩa bài toán

1.2.1 Loại bài toán

Bài toán này thuộc loại bài toán phân loại nhị phân (Binary classification), thuộc nhóm học máy có giám sát (supervised machine learning).

1.2.2 Ngữ cảnh

Học trực tuyến thông qua các nền tảng MOOCs đang trở thành xu hướng phổ biến, thu hút học viên với các mục tiêu và lý do học tập đa dạng. Đối tượng tham gia bao gồm:

- Học sinh và sinh viên: Tìm kiếm kiến thức bổ sung hoặc kỹ năng để hỗ trợ việc học chính quy.
- Người đi làm: Mong muốn nâng cao kỹ năng hoặc chuyển đổi nghề nghiệp.
- Người học tự do: Quan tâm đến việc học vì đam mê hoặc phát triển bản thân.

MOOCs cung cấp cơ hội học tập không giới hạn về địa lý và thời gian. Tuy nhiên, tỷ lệ bỏ học cao là một trong những vấn đề lớn, ảnh hưởng trực tiếp đến chất lượng giáo dục và trải nghiệm học viên. Việc phân tích hành vi học tập và dự báo khả năng bỏ học là cần thiết để:

- Xác định nguyên nhân dẫn đến việc bỏ học.
- Đưa ra biện pháp hỗ trợ kịp thời nhằm cải thiện tỷ lệ hoàn thành khóa học.
- Tối ưu hóa tài nguyên và nội dung giáo dục.

1.2.3 Đầu vào bài toán

Dữ liệu đầu vào của bài toán bao gồm:

Dữ liệu thông tin của học viên: thông tin cơ bản của học viên trong file `user_info.csv` như `id`, `giới tính`, `trình độ học vấn`, `năm sinh`.

Thông tin khóa học: thông tin cơ bản của khóa học trong file `course_info.csv` như `id` (được sử dụng trong tập dữ liệu dự đoán bỏ học), `course_id` (được sử dụng trong nhật kí theo dõi), `start` (thời gian bắt đầu khóa học), `end` (thời gian kết thúc khóa học), `course type` (0: khóa học theo nhịp độ của người hướng dẫn, 1: khóa học theo nhịp độ riêng), `category` (loại khóa học).

Tình hình hoạt động và học tập của học viên: trích xuất thông tin hành vi của học viên từ log như:

- `Session`: Số lần truy cập khóa học của học viên.
- Tổng thời gian các `session`: Tổng thời gian học viên dành cho khóa học.
- Số lần mở Courseware (mục lục khóa học).

1.2.4 Đầu ra bài toán

Đầu ra bài toán là dự đoán nhãn là 0 (không bỏ học) hoặc 1 (bỏ học) cho từng học viên tham gia khóa học. Kết quả này giúp nhận diện các học viên có nguy cơ bỏ học cao để đưa ra biện pháp hỗ trợ.

1.3 Ứng dụng

Mô hình dự báo khả năng bỏ học có thể được áp dụng trong các nền tảng giáo dục trực tuyến, trong nghiên cứu này là MOOCs. Các ứng dụng cụ thể bao gồm:

- Hỗ trợ giáo viên: Nhận diện học viên có nguy cơ bỏ học để đưa ra hỗ trợ phù hợp.

- Quản lý học tập: Phân bổ nguồn lực giáo dục một cách hiệu quả.
- Nâng cao trải nghiệm học viên: Tối ưu hóa nội dung khóa học và cải thiện trải nghiệm học tập.
- Đánh giá chương trình: Xác định yếu tố ảnh hưởng đến tỷ lệ bỏ học để cải thiện thiết kế chương trình học.

1.4 Phạm vi thực hiện

Đối tượng của nghiên cứu là các học viên tham gia vào các khóa học trực tuyến trên các nền tảng MOOCs. Họ có thể là học sinh, sinh viên, hoặc người học muốn nâng cao kỹ năng cá nhân.

Phạm vi nghiên cứu tập trung vào phân tích và dự báo khả năng bỏ học dựa trên dữ liệu lịch sử hoạt động học tập. Dữ liệu này bao gồm các thông tin chi tiết về hoạt động học tập, kết quả học tập, và mức độ tham gia vào khóa học. Các yếu tố như nội dung khóa học, mức độ khó, và cách thức đánh giá cũng được xem xét.

Về giới hạn, nghiên cứu này sẽ không xem xét các yếu tố bên ngoài như thay đổi cá nhân, hoàn cảnh xã hội, hoặc các biến động kinh tế. Đồng thời, nghiên cứu cũng không phân tích tác động của việc thay đổi nội dung hoặc cấu trúc khóa học đến tỷ lệ bỏ học.

Chương 2. TỔNG QUAN VỀ BỘ DỮ LIỆU

2.1 Giới thiệu bộ dữ liệu đã sử dụng

Bộ dữ liệu được sử dụng trong đề tài này là MOOCData - User Activity

MOOCCubeX được duy trì bởi Nhóm Kỹ thuật Tri thức của Đại học Thanh Hoa và được hỗ trợ bởi XuetangX, một trong những trang web MOOC lớn nhất tại Trung Quốc. Kho dữ liệu này bao gồm 4.216 khóa học, 230.263 video, 358.265 bài tập, 637.572 khái niệm chi tiết và hơn 296 triệu dữ liệu hành vi từ 3.330.294 sinh viên, nhằm hỗ trợ nghiên cứu các chủ đề về học tập thích ứng trong MOOCs.

Các tệp trong bộ dữ liệu MOOCData - User Activity được sử dụng trong bài báo "Hiểu về Bỏ học trong MOOC" trong AAAI 2019.

Các tệp nhật ký theo dõi (201508-201608, 201608-201708) bao gồm tất cả các hoạt động học tập của người dùng trên nền tảng XuetangX từ 201508 đến 201708. Những nhật ký này là dữ liệu hỗ trợ cho các phân tích trong bài báo. Bộ dữ liệu dự đoán bỏ học bao gồm tập huấn luyện và tập thử nghiệm được sử dụng trong bài báo cho nhiệm vụ dự đoán bỏ học. Hồ sơ người dùng là thông tin của người dùng XuetangX, bao gồm giới tính, năm sinh và trình độ học vấn. Thông tin khóa học bao gồm ngày bắt đầu khóa học, ngày kết thúc khóa học, danh mục khóa học và loại khóa học.

2.2 Mô tả chi tiết bộ dữ liệu

2.2.1 Tracking Log

Tên trường dữ liệu	Mô tả	Kiểu dữ liệu
course_id	The id (string) of course, which is used in tracking log.	String
user_id	The id (int) of the student.	Int
session_id	The id of the session.	A list of lists

2.2.2 Dropout Prediction Dataset

2.2.2.1 Train_log.csv và test_log.csv

Tên trường dữ liệu	Mô tả	Kiểu dữ liệu
enroll_id	the id of (user, course) pair	Int
name	the id of user	Int
course_id	the id of course	String
session_id	the id of session	String
action	the type of user activity	String
object	the corresponding object of the action	String
time	the occurrence time of the action	Datetime

2.2.2.2 Train_truth.csv và test_truth.csv

Tên trường dữ liệu	Mô tả	Kiểu dữ liệu	Miền giá trị
enroll_id	the id of (user, course) pair	Int	
truth	the label of user's dropout (1: dropout, 0: non-dropout)	Int	{0, 1}

2.2.3 User Profile

Thuộc tính	Mô tả	Kiểu dữ liệu	Miền giá trị
user_id	Mã định danh người dùng.	Integer hoặc String	Không giới hạn
gender	Giới tính của người dùng.	String	{"male", "female"}

education	Trình độ học vấn của người dùng.	String	{“Associate”, “Bachelor’s”, “Doctorate”, “High”, “Master’s”, “Middle”, “Primary”}
birth	Năm sinh của người dùng.	Integer	1894 - 2018

2.2.4 Course Info

Thuộc tính	Mô tả	Kiểu dữ liệu	Miền giá trị
id	Mã định danh khóa học, được sử dụng trong tập dữ liệu dự đoán bỏ học.	Integer hoặc String	Không giới hạn
course_id	Mã định danh khóa học (chuỗi), được sử dụng trong nhật ký theo dõi.	String	Không giới hạn
start	Thời gian bắt đầu của khóa học.	Datetime	“dd/mm/yyyy hh:mm:ss tt”
end	Thời gian kết thúc của khóa học.	Datetime	“dd/mm/yyyy hh:mm:ss tt”
course_type	Chế độ học của khóa học. 0 = Học theo lịch của giảng viên, 1 = Tự học theo lịch trình cá nhân.	Int	{0, 1}

category	Danh mục của khóa học, mô tả lĩnh vực chủ đề của khóa học.	String	{“art”, “biology”, “business”, “chemistry”, “computer”, “economics”, “education”, “electrical”, “engineering”, “environment”, “foreign language”, “history”, “literature”, “math”, “medicine”, “philosophy”, “physics”, “social science”}
----------	--	--------	---

2.3 Khám phá (EDA) và trực quan hóa bộ dữ liệu

2.3.1 *course_info.csv*

Dữ liệu về khóa học bao gồm 6410 mẫu và có 6 trường thông tin:

- id (số nguyên)
- course_id (dạng chuỗi)
- start (thời điểm bắt đầu)
- end (kết thúc khóa học)
- course_type (chế độ học)
- category (lĩnh vực của khóa học)

```
[ ] course_info = pd.read_csv('/content/drive/MyDrive/Nhóm 11/Đồ án môn học/Technology/raw_data/course_info.csv')
```

course_info.head(10)

	id	course_id	start	end	course_type	category
0	6561	course-v1:CPVS+CPVS-HDLSC001+20160901	2016-11-16 08:00:00	2016-12-31 23:30:00	0	NaN
1	5557	course-v1:SCUT+144282+201709	2016-09-01 00:00:00	2017-02-28 00:00:00	0	NaN
2	9433	course-v1:ZK+06093+J	2018-01-01 08:00:00	2020-01-01 00:00:00	0	NaN
3	8320	course-v1:nuist+001+2016-T1	2017-03-01 18:30:00	2017-07-01 23:30:00	0	NaN
4	231	FUDAN/CFD004/2014.9-2015.1	2014-09-10 08:00:00	2015-09-10 00:00:00	0	NaN
5	7645	course-v1:ANUx+EBM05x+3T2017	2017-09-18 08:00:00	2018-09-17 08:00:00	0	NaN
6	9953	course-v1:ChongqingUniversity+CQUMOOCMSE21202-...	2017-02-19 10:30:00	2017-05-15 00:00:00	0	NaN
7	7625	course-v1:JLUx+0000045603+SP	2016-09-26 08:00:00	NaN	1	NaN
8	8657	course-v1:TsinghuaX+00680082_1X_p1+sp	2016-12-01 08:00:00	NaN	1	philosophy
9	8833	course-v1:global_TsinghuaX+70120073X+sp	2016-09-12 10:00:00	2017-01-16 10:00:00	0	NaN

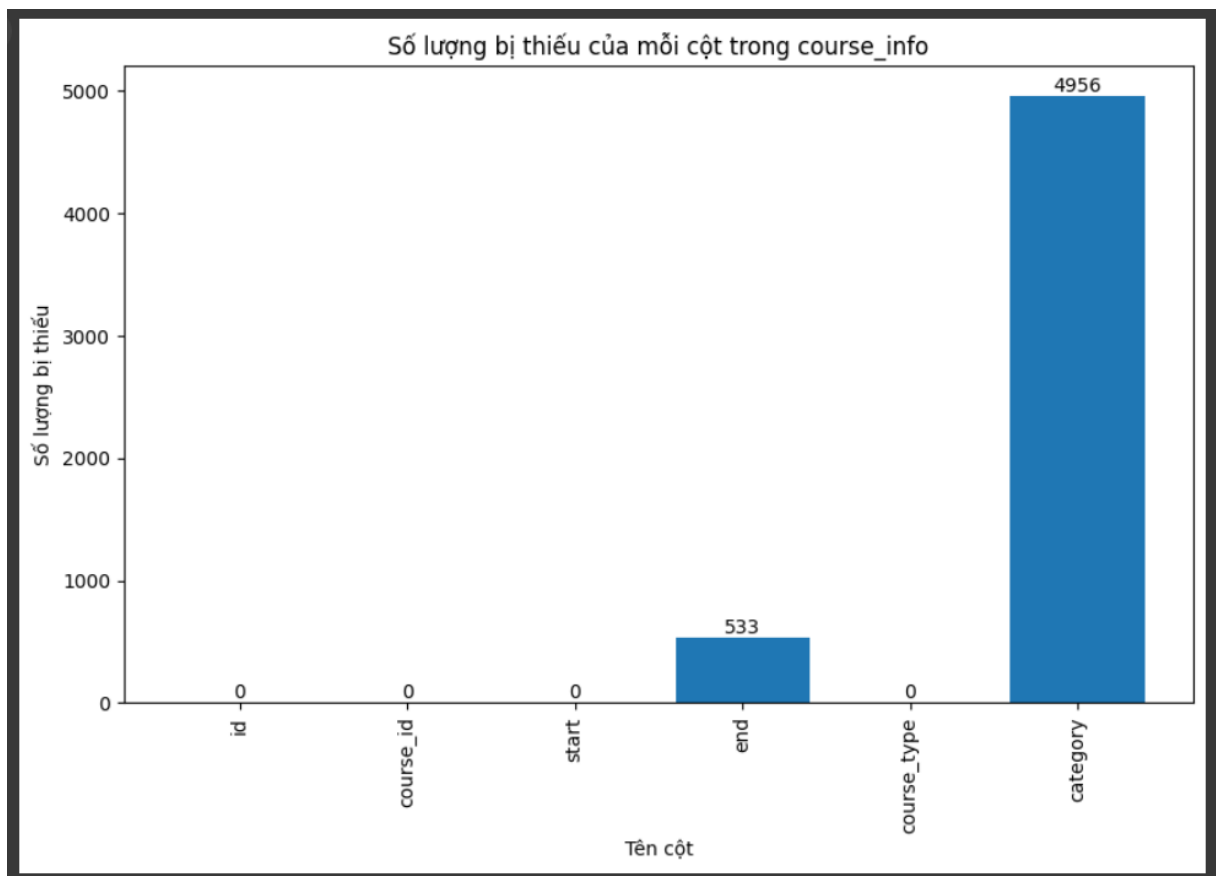
Sau khi kiểm tra thì không có dữ liệu trùng lặp. Các cột đều là biến phân loại ngoại trừ cột `course_type` là biến số.

```
[ ] course_info.isnull().sum()
```

	0
id	0
course_id	0
start	0
end	533
course_type	0
category	4956

dtype: int64

Thống kê số lượng mẫu bị thiếu của mỗi cột trong `course_info` thì nhận thấy chỉ có 2 cột bị thiếu thông tin là **end** (thời điểm kết thúc khóa học) và **category** (lĩnh vực khóa học). Trong khi cột `end` số lượng mẫu bị thiếu chỉ 533 nhưng đối với `category` thì nghiêm trọng hơn lên tới 4956 mẫu bị thiếu thông tin.

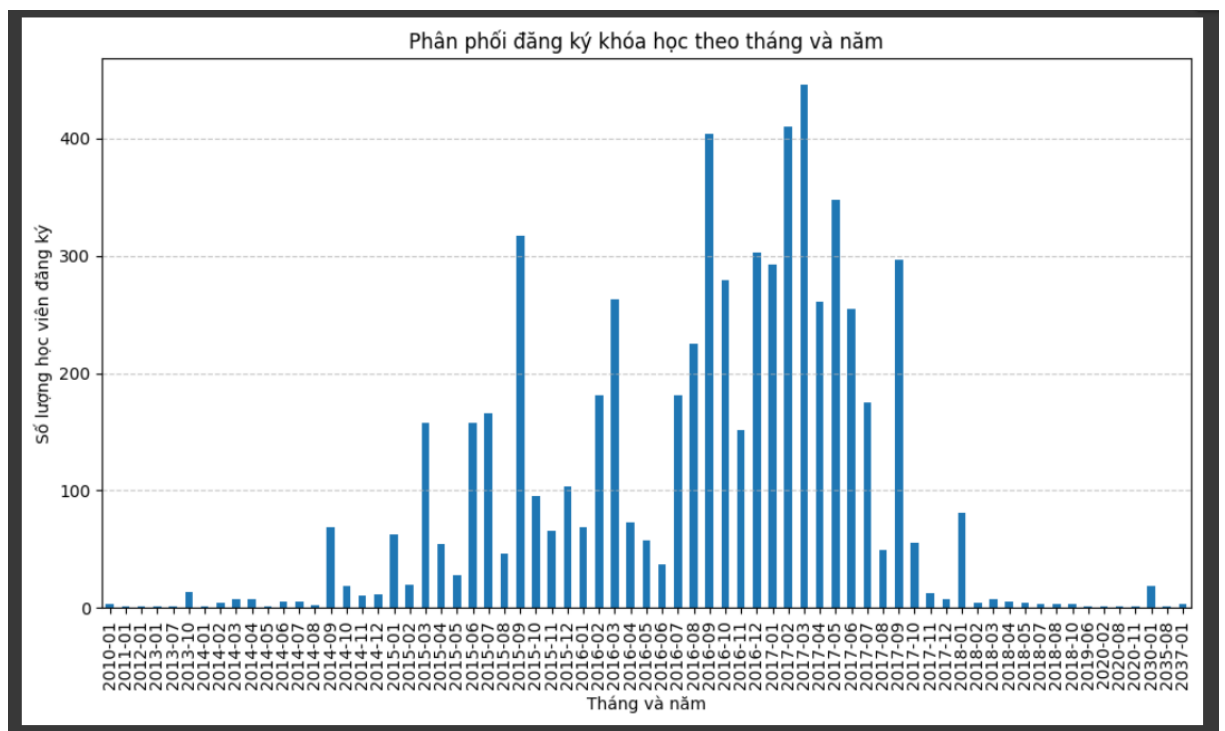


❖ **start- thời điểm bắt đầu khóa học**

```
[ ] course_info.describe()
```

	start	end	course_type
count	6410	5877	6410.000000
mean	2016-09-14 00:40:20.377535232	2017-04-09 06:50:41.807044352	0.083151
min	2010-01-01 08:00:00	2013-12-29 16:00:00	0.000000
25%	2016-02-22 09:00:00	2016-07-09 00:00:00	0.000000
50%	2016-10-31 08:00:00	2017-03-12 23:30:00	0.000000
75%	2017-03-31 08:00:00	2017-08-16 23:30:00	0.000000
max	2037-01-16 08:00:00	2066-01-01 00:00:00	1.000000
std	NaN	NaN	0.276132

Theo mô tả thì bộ dữ liệu này được thu thập từ 08/2015 đến 08/2017. Tuy nhiên, khi xem xét bảng tóm tắt, thì có thể nhận thấy có sự bất hợp lý trong các trường thông tin về ngày tháng. Cụ thể, giá trị lớn nhất của cột `start` là năm **2037**, nằm ngoài khoảng thời gian khảo sát.



Qua biểu đồ phân phối trên thì có thể nhận thấy rằng các khóa học được thu thập có khoảng thời gian bắt đầu từ 09/2014 - 01/2018.

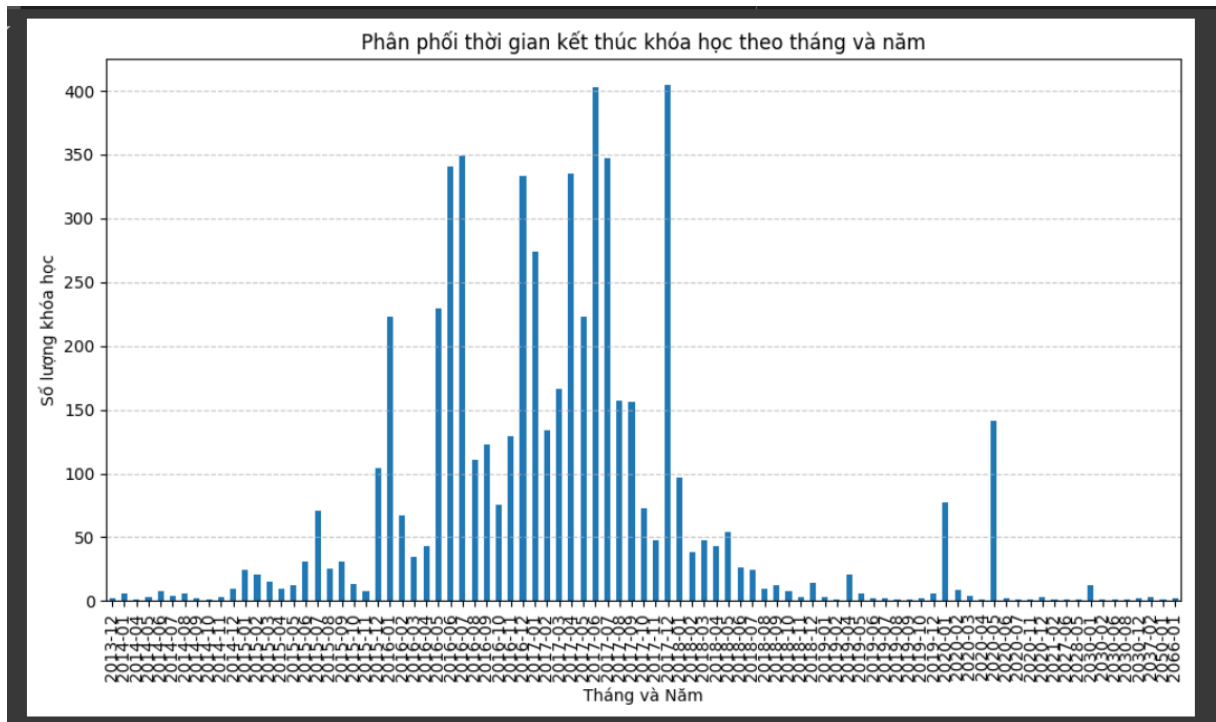
Đối với cột thông tin thời điểm bắt đầu khóa học “start”, nhóm chọn khoảng thời gian hợp lý từ 01/01/2015 đến 31/08/2017. Những giá trị nằm ngoài khoảng thời gian này sẽ được xử lý và làm sạch

❖ end - thời điểm kết thúc khóa học

```
[ ] course_info.describe()
```

	start	end	course_type
count	6410	5877	6410.000000
mean	2016-09-14 00:40:20.377535232	2017-04-09 06:50:41.807044352	0.083151
min	2010-01-01 08:00:00	2013-12-29 16:00:00	0.000000
25%	2016-02-22 09:00:00	2016-07-09 00:00:00	0.000000
50%	2016-10-31 08:00:00	2017-03-12 23:30:00	0.000000
75%	2017-03-31 08:00:00	2017-08-16 23:30:00	0.000000
max	2037-01-16 08:00:00	2066-01-01 00:00:00	1.000000
std	NaN	NaN	0.276132

Cột end chứa những giá trị bị thiếu như đã thống kê bên trên. Ngoài ra, cũng chứa các giá trị không đáng tin cậy (giá trị max ở cột end lên tới năm 2066).



Các khóa học được thu thập chủ yếu bắt đầu từ năm 2015 đến 2017, nhưng phân bố không đều, với một số giai đoạn tập trung nhiều khóa học hơn. Ngoài ra, cột "end" gặp hai vấn đề chính: giá trị Null và giá trị không hợp lý. Nhóm giả định các khóa học trực tuyến thường kéo dài không quá một năm, nên chọn khoảng thời gian hợp lý cho cột "end" là đến hết năm 2018. Các giá trị ngoài khoảng này được coi là bất thường và sẽ xử lý trong quá trình làm sạch dữ liệu.

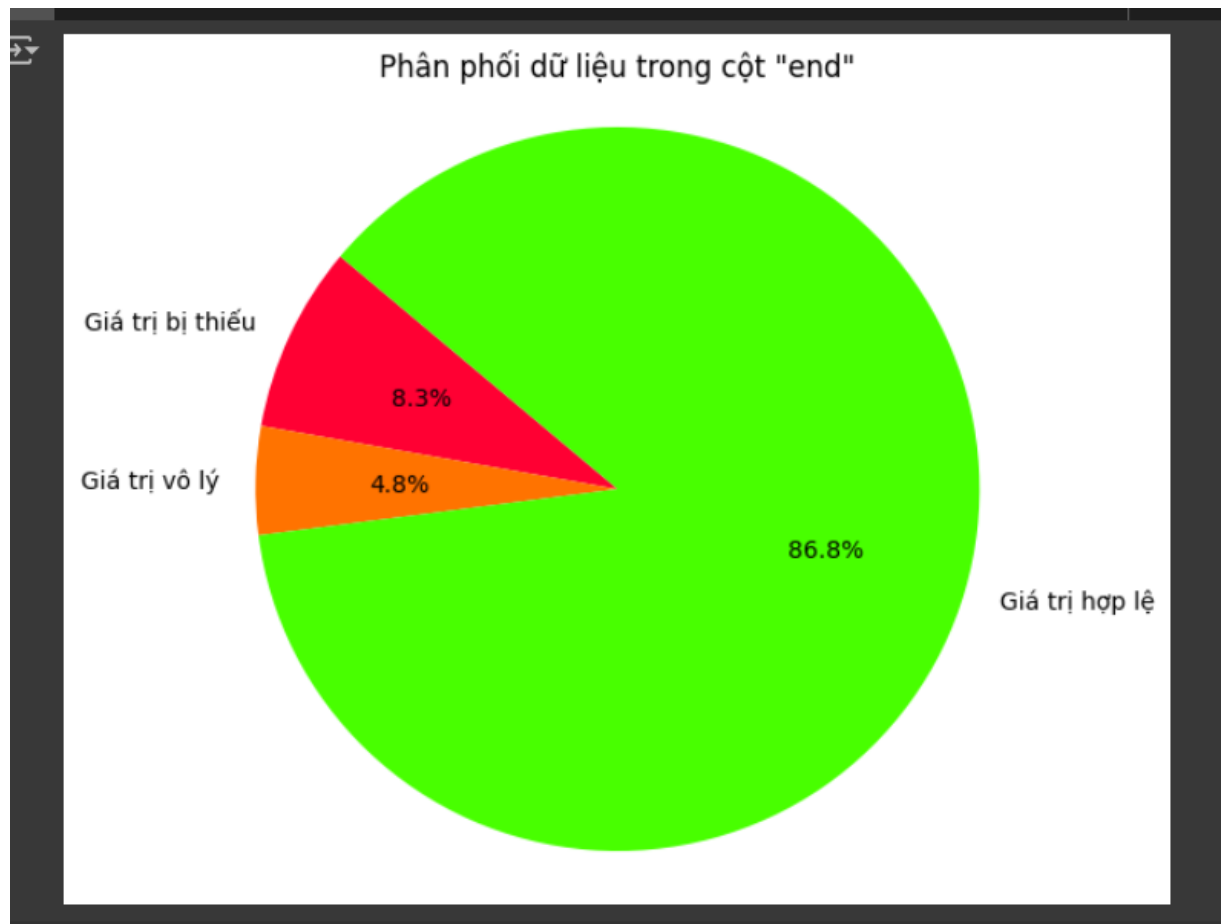
```
# Tính số lượng mẫu có giá trị trong cột 'end' lớn hơn năm 2018
count_end_greater_than_2018 = course_info[course_info['end'].dt.year > 2018]['end'].count()

# Tính số lượng giá trị null trong cột 'end'
count_null_values = course_info['end'].isnull().sum()

print("Số lượng mẫu có giá trị trong cột 'end' lớn hơn năm 2018 là:", count_end_greater_than_2018)
print("Số lượng giá trị null trong cột 'end' là:", count_null_values)
```

Số lượng mẫu có giá trị trong cột 'end' lớn hơn năm 2018 là: 310
Số lượng giá trị null trong cột 'end' là: 533

Nhóm đã thống kê được có tổng 310 mẫu có cột end lớn hơn năm 2018 và tổng cộng có 533 mẫu bị thiếu thông tin.



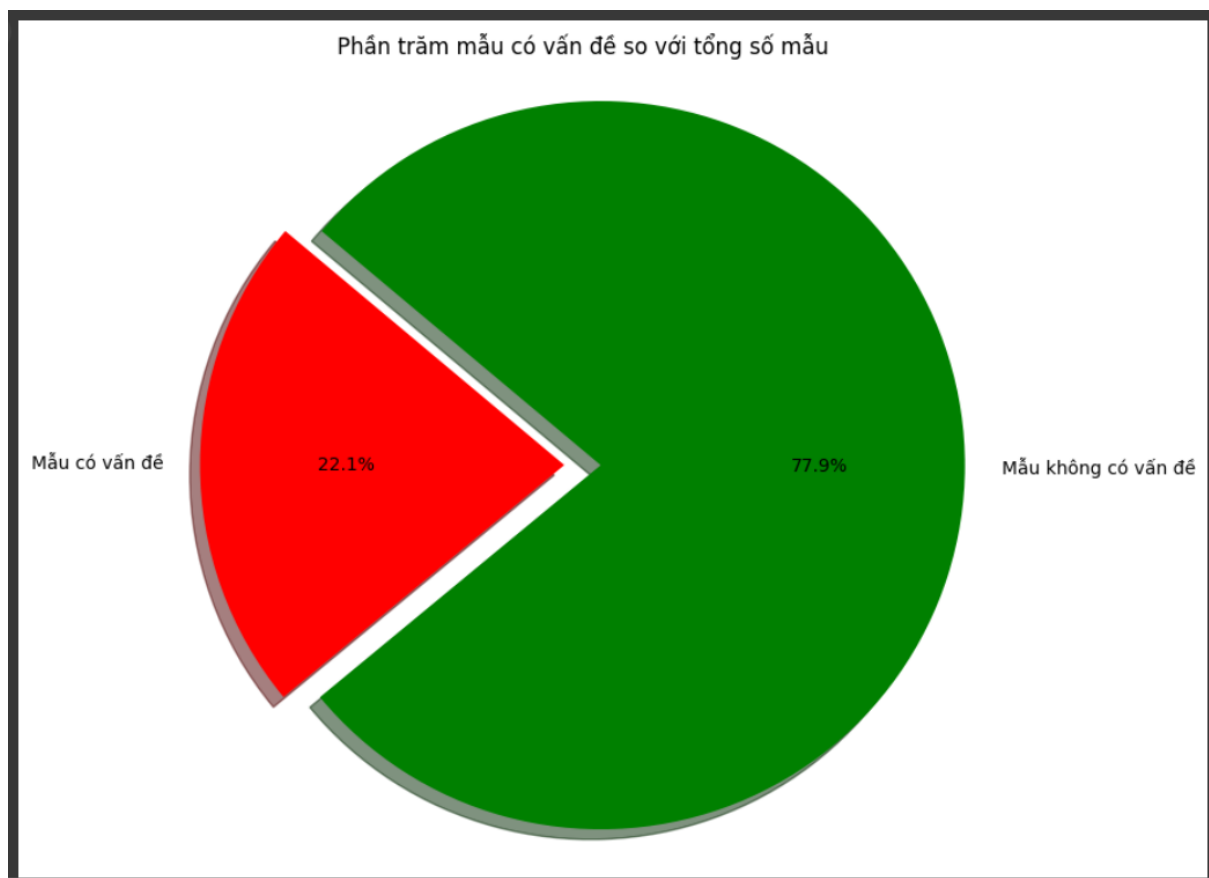
Vậy tổng giá trị không hợp lệ của cột 'end' là $8.3\% + 4.1\% = 12.4\%$. Đây là một con số đáng kể ảnh hưởng đến tính chính xác của mô hình \Rightarrow Cột end cần được xử lý trước khi sử dụng

❖ Kết hợp start và end

Nhóm đã thử tính số tháng kéo dài của mỗi khóa học dựa trên thời điểm bắt đầu (start) và kết thúc (end) và phát hiện một số vấn đề:

1. Chỉ chấp nhận giá trị "start" từ 01/01/2015 đến 31/08/2017.
2. Có nhiều giá trị "end" vượt quá năm 2018.
3. Một số giá trị "end" bị null.
4. Một số mẫu có thời gian "end" nhỏ hơn "start" (không tính các mẫu có cột 'end' là null).

Tổng hợp các điều kiện trên, nhóm xác định có 1417 mẫu gặp ít nhất một vấn đề. Điều này cho thấy cần xử lý để đảm bảo dữ liệu chính xác và đáng tin cậy.



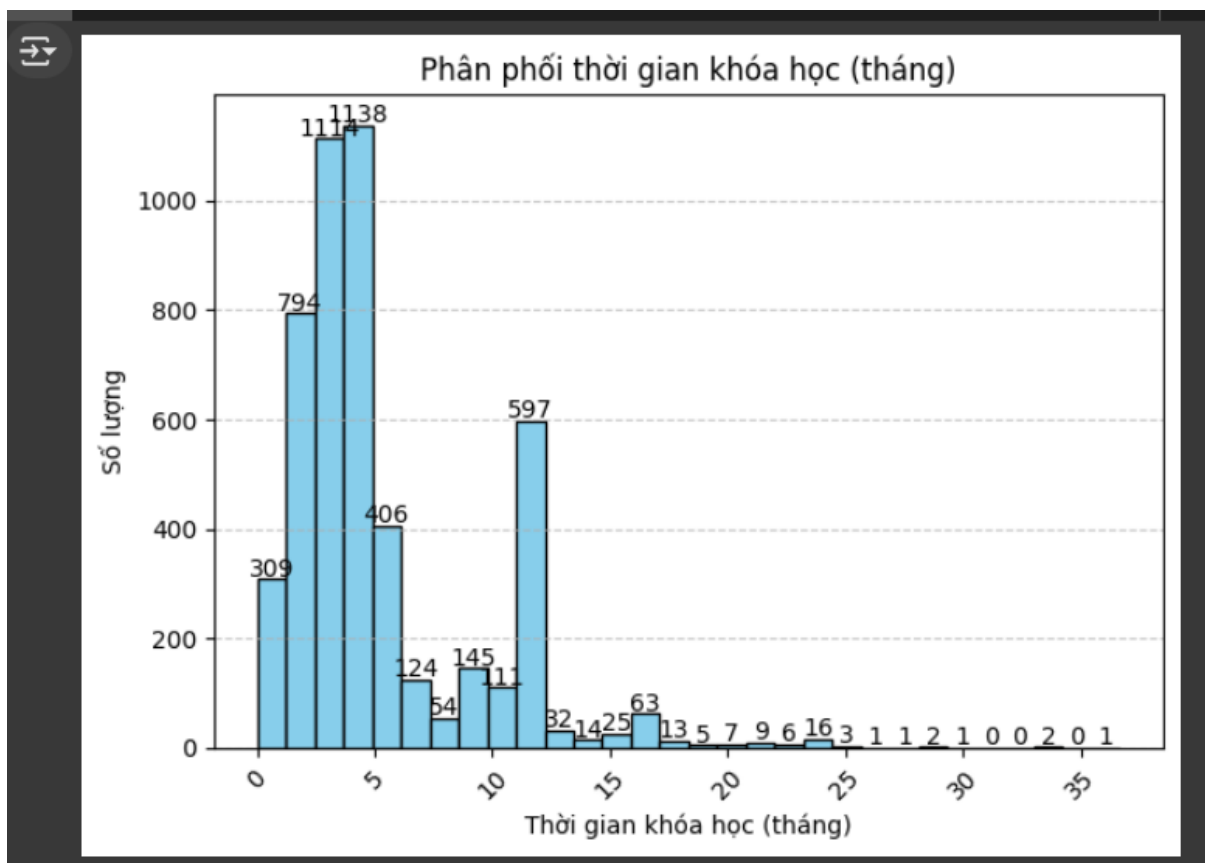
Nhận xét: Số lượng mẫu có vấn đề chiếm 22,1% tổng số, đây là tỷ lệ khá lớn. Do đó, việc xử lý dữ liệu cần thực hiện cẩn thận và chi tiết. Cần tập trung vào:

1. Xác định và sửa chữa giá trị không hợp lý.
2. Điền các giá trị thiếu một cách phù hợp.
3. Loại bỏ các mẫu không đáng tin cậy.

Những biện pháp này là cần thiết để đảm bảo dữ liệu chính xác và đáng tin cậy, từ đó hỗ trợ kết quả phân tích sau này đạt hiệu quả tốt nhất.

Nhóm đã tính thời lượng từng khóa học bằng cách lấy thời điểm kết thúc trừ thời điểm bắt đầu, sau đó quy đổi sang đơn vị tháng. Kết quả cho thấy thời lượng các khóa học có sự phân bố đa dạng.

Để đảm bảo tính chính xác, nhóm chỉ sử dụng 77,9% mẫu hợp lệ cho các phân tích tiếp theo. Điều này giúp đảm bảo các kết quả đáng tin cậy và phản ánh đúng phân phối thời lượng khóa học trên nền tảng.

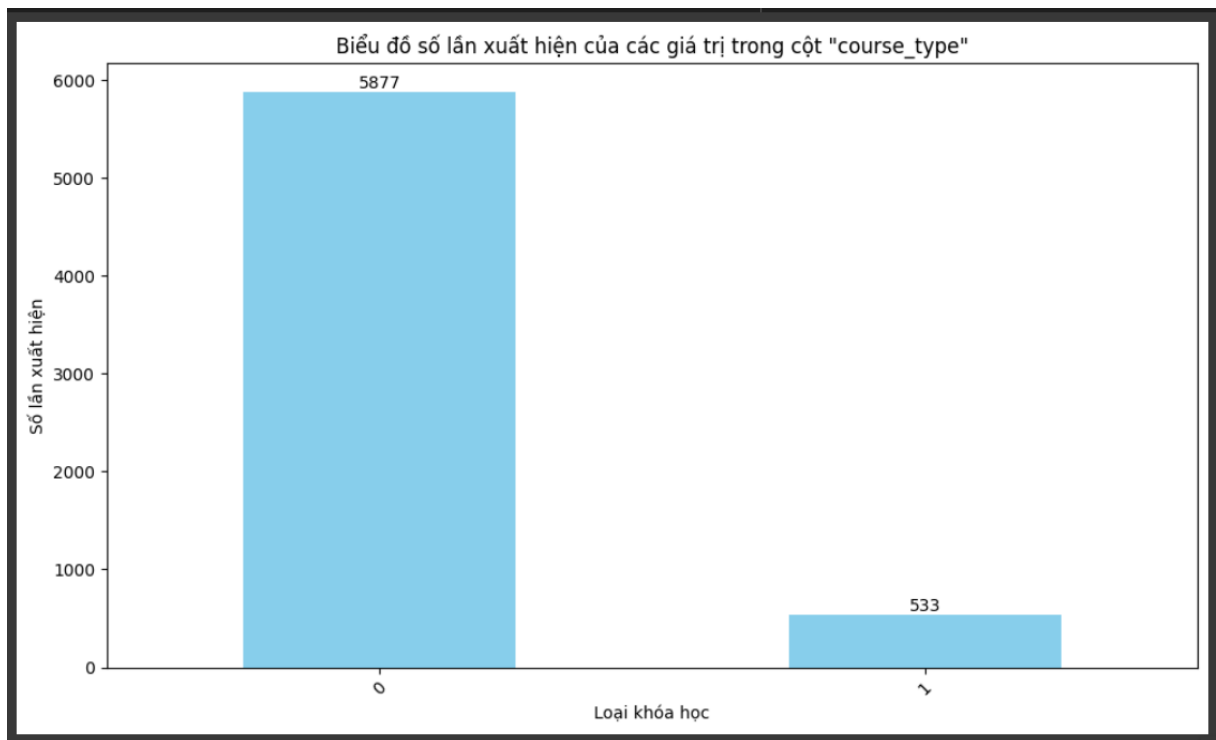


Các khóa học trên nền tảng có thời lượng phân bố rải rác, từ vài tháng đến hơn 35 tháng, phản ánh sự đa dạng lớn về độ dài. Tuy nhiên, phân phối không đồng đều.

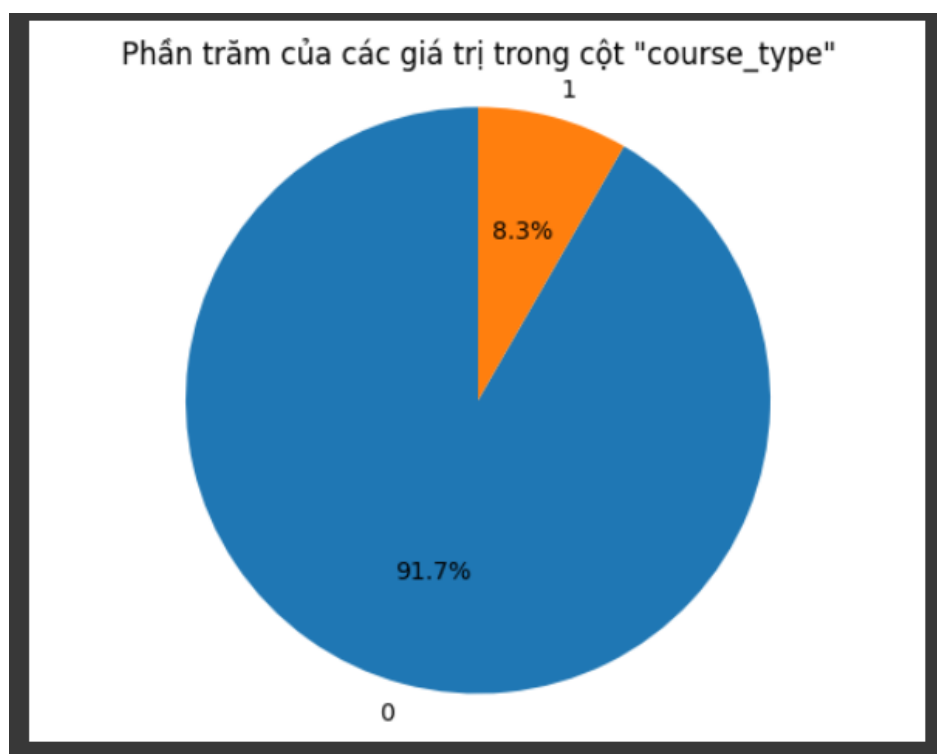
Thời lượng phổ biến nhất nằm trong khoảng 1–17 tháng, nhưng ngay trong khoảng này cũng có sự chênh lệch, với một số khóa học rất ngắn và một số khác kéo dài hơn. Điều này cho thấy sự biến động đáng kể về thời lượng các khóa học trên nền tảng.

❖ **course_type - chế độ khóa học**

Trường thông tin `course_type` là thông tin về chế độ học của khóa học. Nó chỉ bao gồm 2 giá trị số nguyên là 0 và 1 (với 0 là khóa học theo tốc độ của người hướng dẫn và 1 là khóa học theo nhịp độ riêng).



Nhận xét: Sự khác biệt giữa hai loại giá trị này là rất đáng kể: Giá trị 0 cao hơn giá trị 1 khoảng 10 lần. Điều này tác động mạnh mẽ đến quá trình phân tích và dự đoán, đặc biệt trong việc đánh giá khả năng bỏ học của học viên. Sự mất cân bằng này có thể khiến mô hình không học được đúng mối quan hệ giữa các biến và kết quả, từ đó làm giảm độ chính xác của dự đoán. Do đó, cần xem xét kỹ lưỡng để đảm bảo tính toàn diện và độ tin cậy của quá trình phân tích và dự đoán.

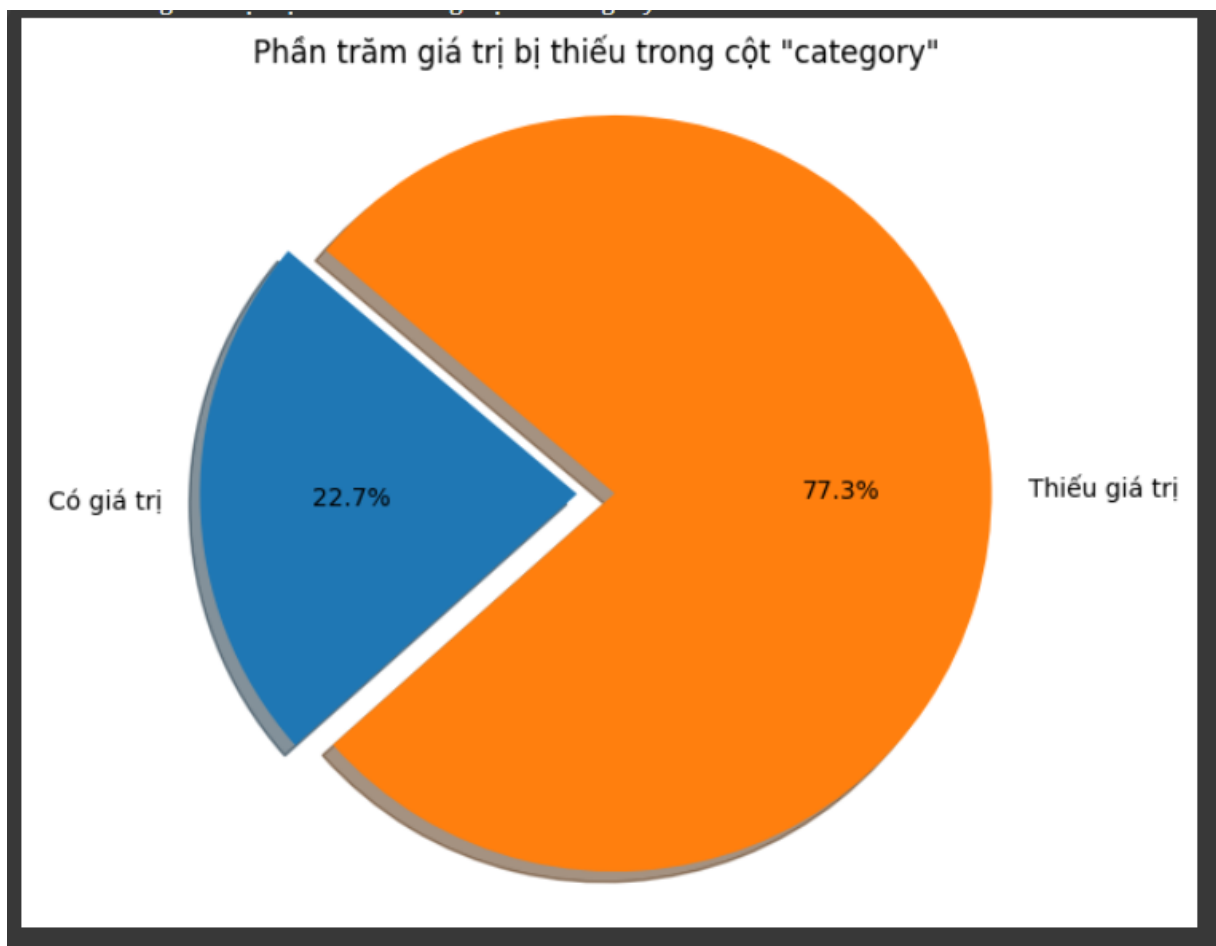


Nhận xét: Sự chênh lệch lớn giữa số lượng giá trị 0 và 1 trong trường "course_type" có thể gây ra:

- Thiên lệch dữ liệu: Mô hình khó học đúng mối quan hệ do nhóm có số lượng lớn (giá trị 0) lấn át.
- Mất cân bằng mô hình: Mô hình dễ dự đoán sai lệch theo lớp chiếm ưu thế, làm giảm hiệu quả với lớp thiểu số.

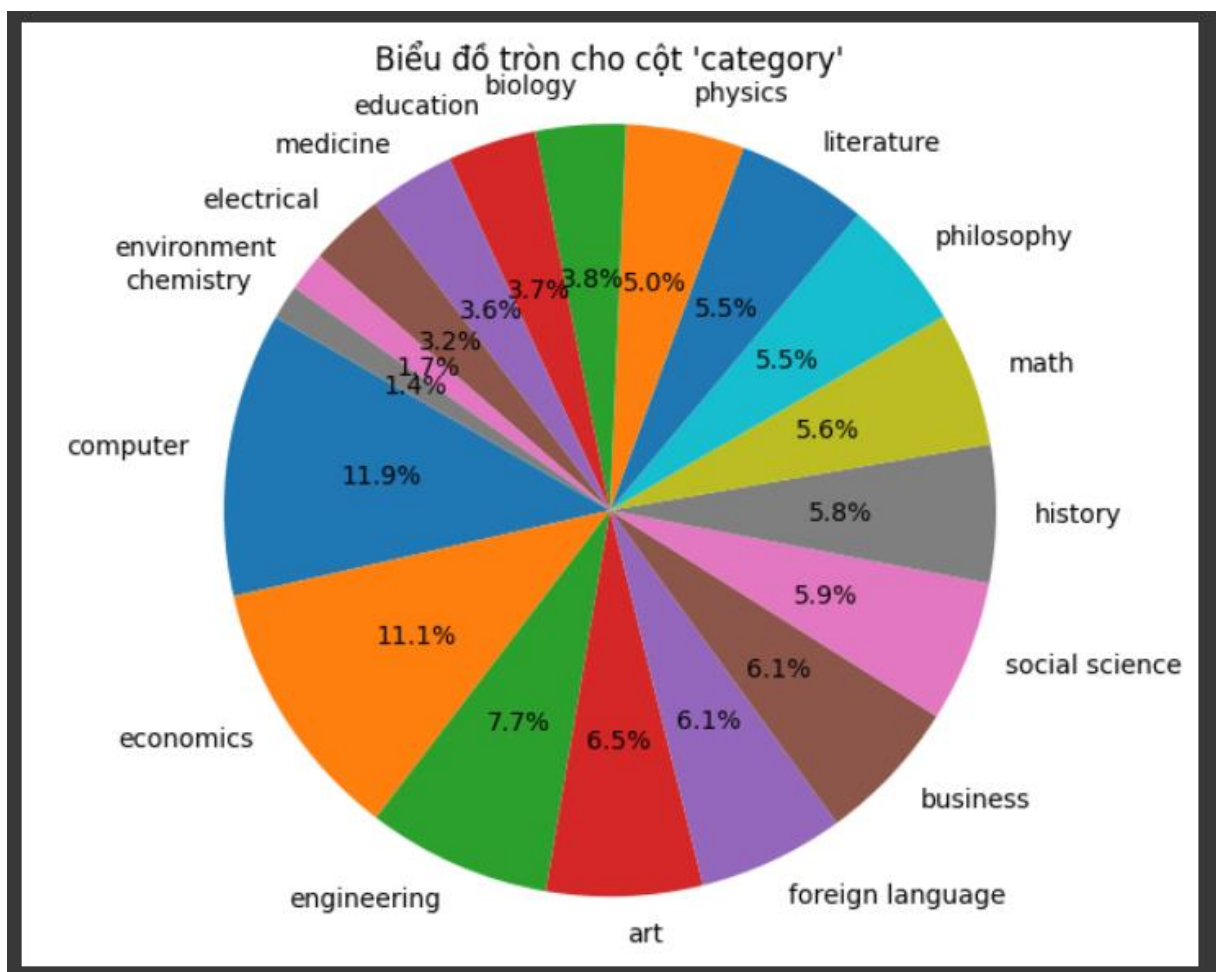
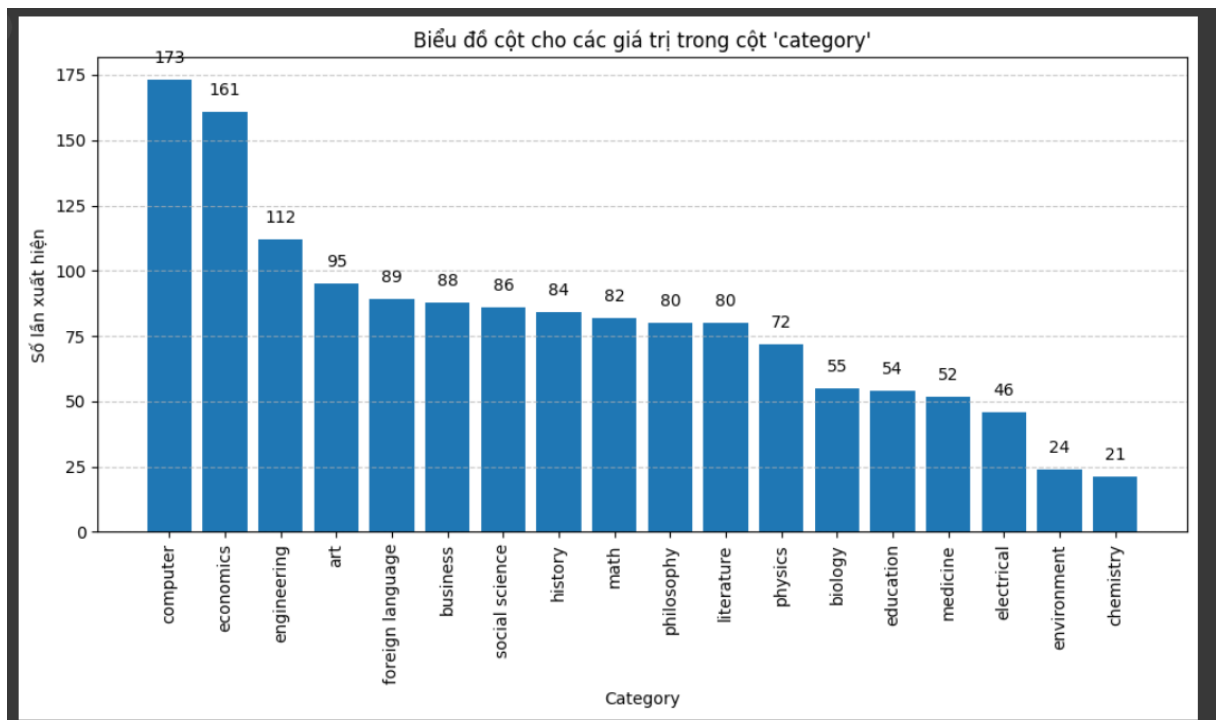
❖ **category - lĩnh vực khóa học**

Như đã thống kê trường thông tin này chiếm số lượng bị thiếu nhiều nhất trong course, lên đến 4956 mẫu.



Nhận xét: Cột "category" có tỷ lệ dữ liệu thiếu vượt gần ba lần so với dữ liệu đầy đủ, ảnh hưởng lớn đến tính toàn vẹn của bộ dữ liệu và quá trình dự đoán. Thông tin này rất quan trọng để hiểu sự tương tác giữa học viên và nội dung học, nên việc xử lý dữ liệu thiếu là cần thiết để đảm bảo độ chính xác và hiệu quả của mô hình dự đoán.

Cột category tồn tại tổng cộng 18 loại giá trị (không tính giá trị null)

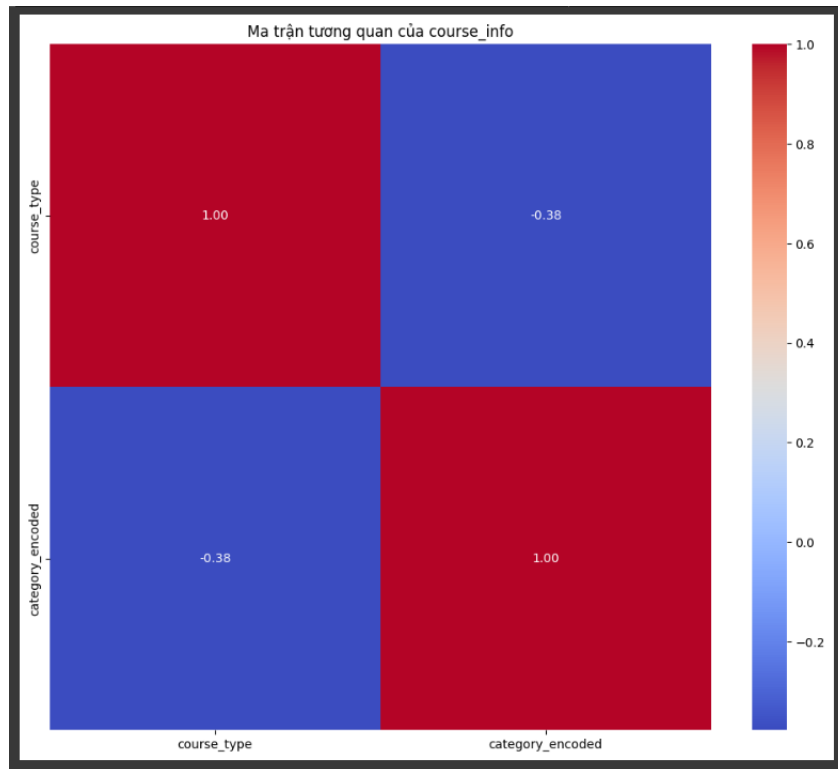


Nhận xét:

- Các lĩnh vực khóa học rất đa dạng, từ công nghệ thông tin (computer), kinh tế (economics), kỹ thuật (engineering) đến nghệ thuật, ngôn ngữ, và khoa học xã hội. Một số lĩnh vực như computer, economics, và engineering có số lượng khóa học vượt trội, phản ánh nhu cầu cao trên thị trường lao động. Ngược lại, các lĩnh vực như environment và chemistry có ít khóa học hơn, có thể do tính chuyên sâu và ít phổ biến. Phân bố không đồng đều này có thể ảnh hưởng đến việc phân tích và dự đoán khả năng bỏ học của học viên.
- Trường "category" có sự chênh lệch giữa các giá trị, nhưng không quá lớn. Giá trị phổ biến nhất, "computer," chiếm 11.9% tổng mẫu, trong khi giá trị thấp nhất chỉ chiếm 1.4% (21 mẫu). Dù chênh lệch này không đáng kể, nó vẫn có thể ảnh hưởng đến dự đoán khả năng bỏ học. Ngoài ra, việc thiếu dữ liệu ở trường này cũng ảnh hưởng đến độ chính xác của phân tích và dự đoán, đòi hỏi cần xử lý cẩn thận để đảm bảo kết quả đáng tin cậy.

❖ Ma trận tương quan:

Sử dụng Label Encoding để mã hóa cột "category" từ kiểu dữ liệu object sang int64, bằng cách gán một giá trị số duy nhất cho mỗi nhóm dữ liệu. Phương pháp này giúp dễ dàng tạo ma trận tương quan để phân tích mối quan hệ giữa các biến, đồng thời hỗ trợ hiệu quả cho việc xử lý dữ liệu và xây dựng mô hình dự đoán khả năng bỏ học của học viên.



Nhận xét:

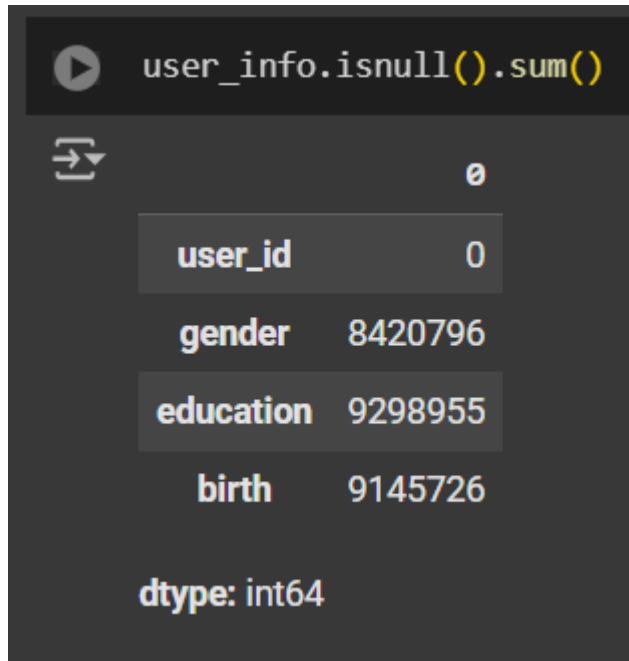
- Giá trị tương quan -0.38 giữa "course_type" và "category_encoded" cho thấy một mối tương quan âm vừa phải. "Course_type" phản ánh chế độ học, còn "category_encoded" biểu diễn lĩnh vực khóa học.
- Mức độ tương quan âm này gợi ý rằng một số chế độ học (ví dụ: tự học hoặc học với giảng viên) có thể được ưu tiên hơn trong một số lĩnh vực nhất định, phù hợp với đặc điểm của lĩnh vực đó.
- Tuy mức tương quan không quá mạnh, nhóm quyết định giữ lại cả hai biến để phân tích thêm và xây dựng mô hình dự đoán. Điều này đảm bảo khai thác đầy đủ thông tin và hỗ trợ đánh giá mối quan hệ giữa các yếu tố trong dữ liệu.

2.3.2

2.3.3 user_info.csv

Dữ liệu về học viên bao gồm 9627148 mẫu và có trường thông tin:

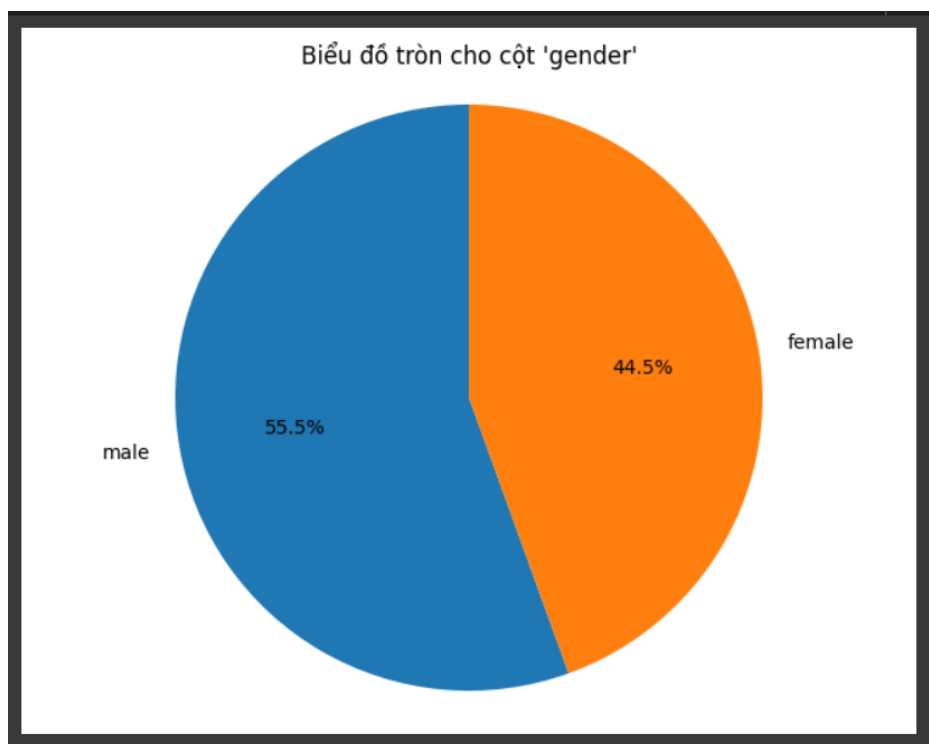
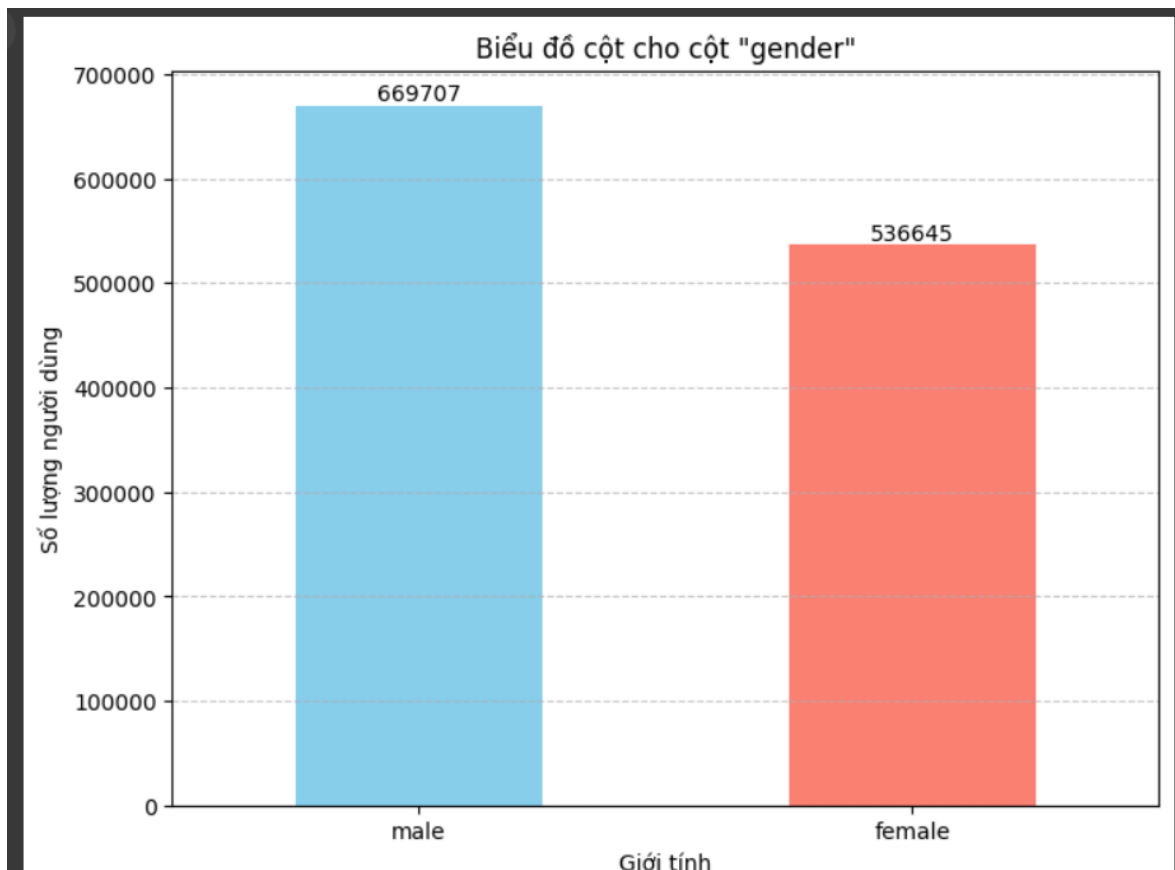
- user_id (số nguyên)
- gender (giới tính)
- education (học vấn)
- birth (năm sinh)



Thông kê số lượng mẫu bị thiếu của mỗi cột trong `user_info` thì nhận thấy ngoài cột `user_id` ra thì các cột còn lại đều thiếu thông tin rất nhiều. Cụ thể cột `gender` có số lượng mẫu bị thiếu là 8420796, cột `education` có số lượng mẫu bị thiếu là 9298955, cột `birth` có số lượng mẫu bị thiếu là 9145726.

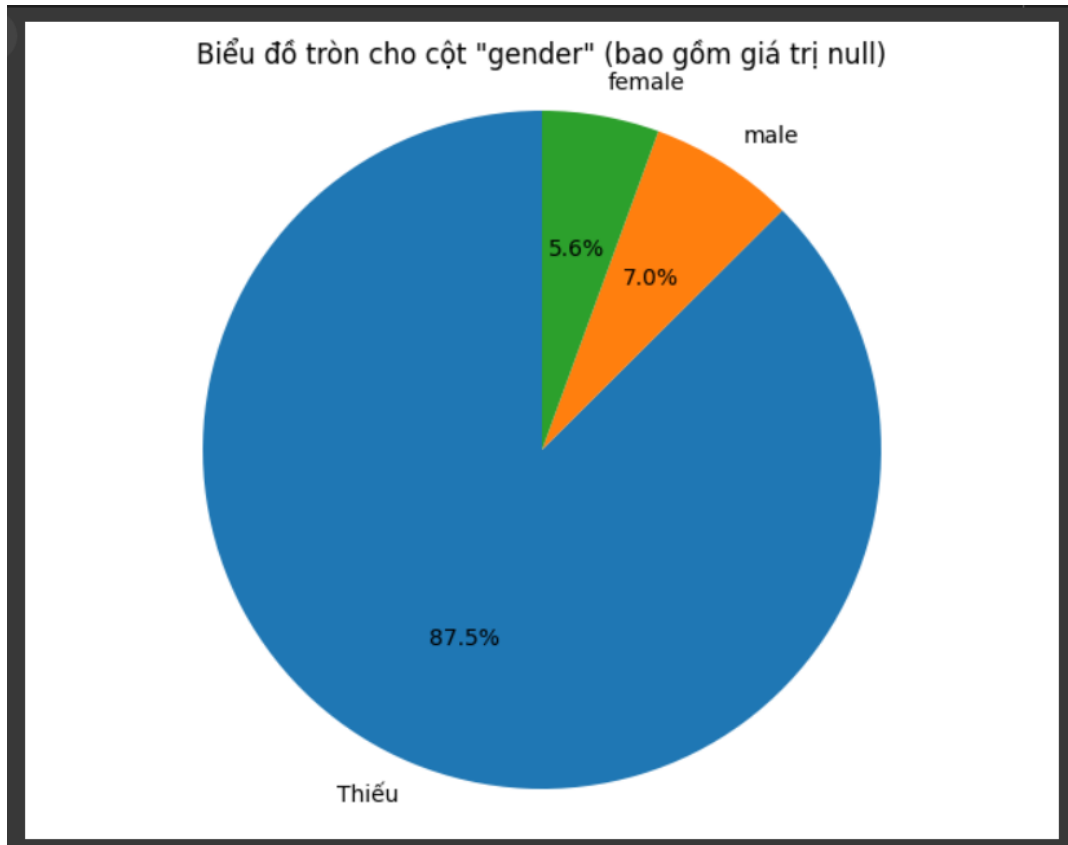
❖ **gender - thông tin về giới tính**

Thông tin về giới tính chỉ bao gồm hai loại giá trị: "male" và "female". Nhóm đã tiến hành thống kê phân phối số lượng và tỉ lệ của mỗi loại giá trị như sau:



Qua 2 biểu đồ trên, ta có thể thấy khoảng 55.5% là “male” và “female” chiếm khoảng 44.5%. Điều này tạo ra một sự phân phối khá cân đối giữa hai giới tính.

Tuy nhiên, trường thông tin này lại chứa rất nhiều các mẫu bị thiếu mất thông tin lên đến 8420796 mẫu.

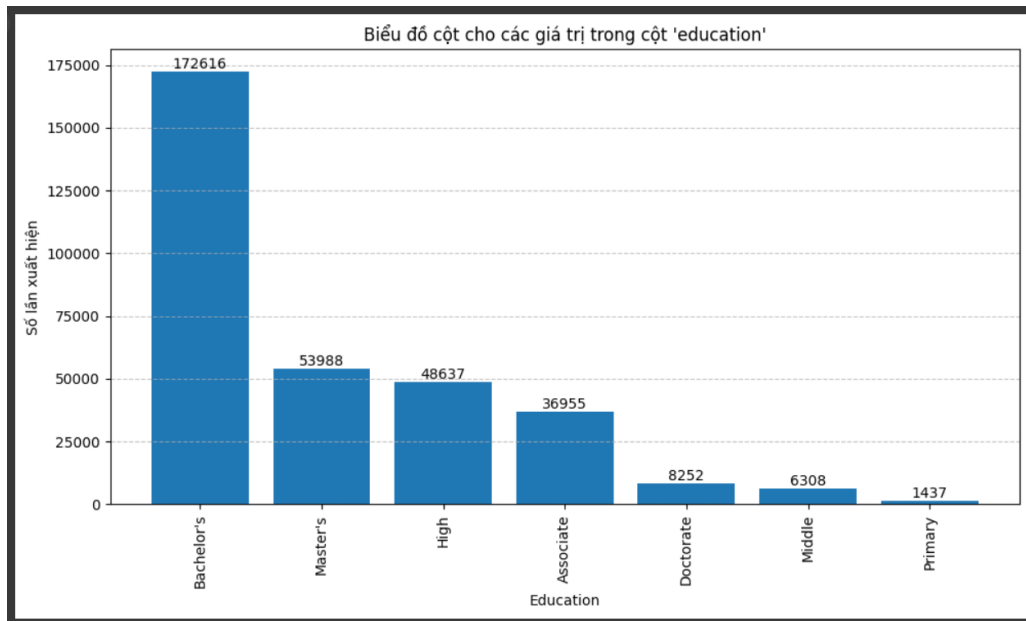


Nhận xét:

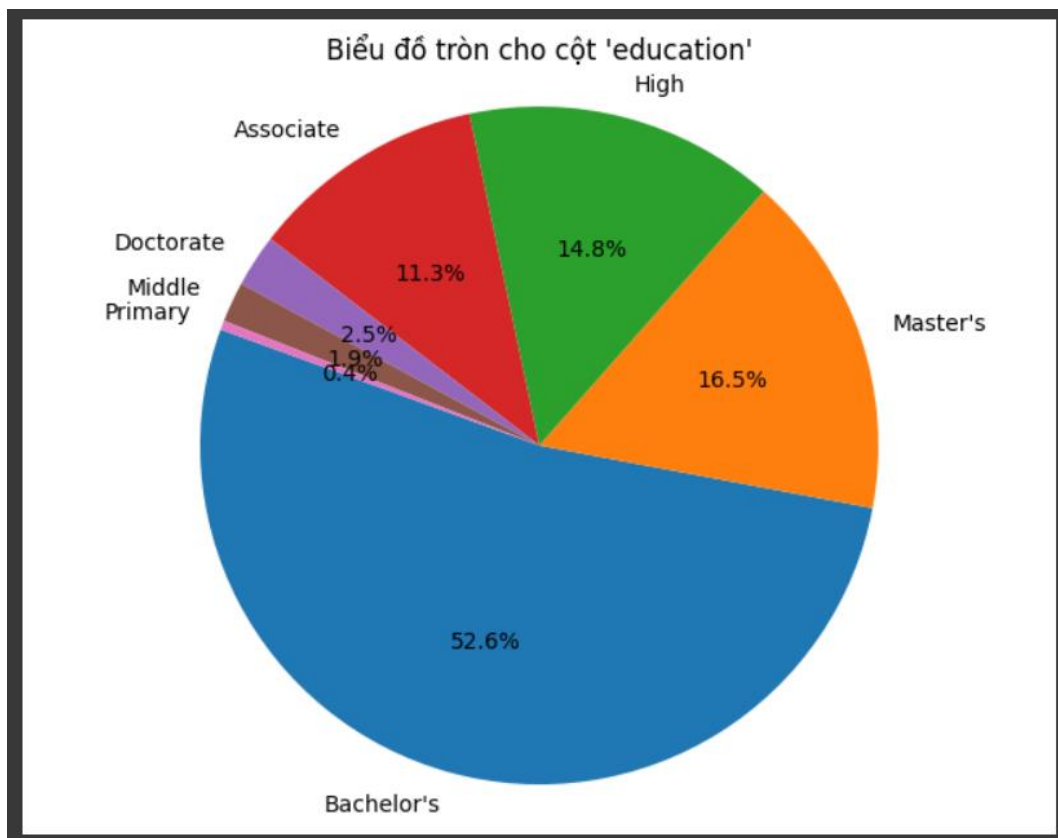
- Khi xem xét toàn bộ dữ liệu, 87.5% thuộc nhóm "Thiếu", nghĩa là phần lớn dữ liệu không có giá trị trong cột gender. Đây là tỷ lệ rất cao, cần được xem xét để xử lý, đặc biệt nếu giới tính là một biến quan trọng trong phân tích.
- Mặc dù cột gender có thể cung cấp thông tin quan trọng về phân phối giới tính trong mẫu dữ liệu, việc có một tỷ lệ lớn giá trị bị thiếu có thể giảm đi khả năng của cột này trong việc phân tích hoặc dự đoán.

❖ education - thông tin về trình độ

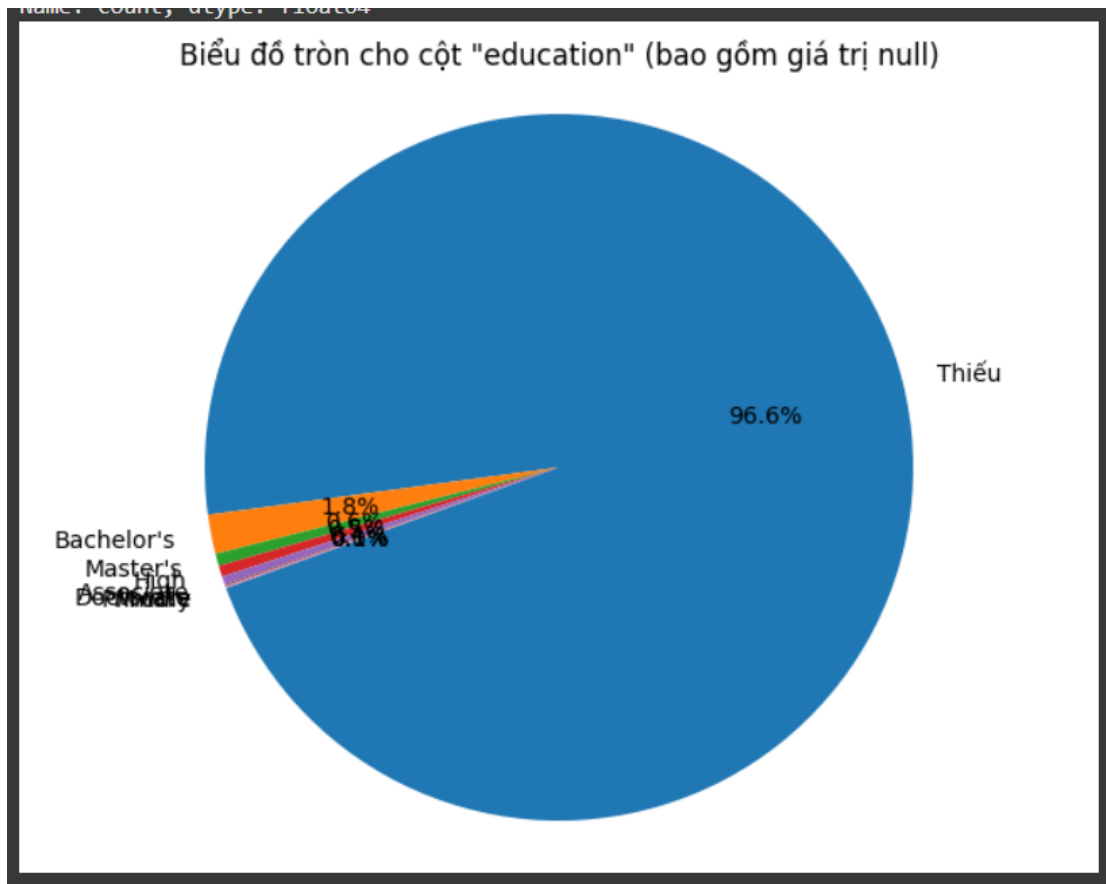
Trường thông tin này chứa lượng mẫu thiếu giá trị lên đến 9298955 mẫu. Đây là một con số rất lớn. Phân bố các giá trị của cột này như sau:



Tổng quan: Sự phân bố các giá trị là không đồng đều, rất mất cân bằng. Dữ liệu tập trung chủ yếu ở nhóm trình độ cao (Cử nhân, Thạc sĩ), cho thấy đây có thể là một tập dữ liệu liên quan đến các đối tượng có học vấn cao. Các nhóm như "Primary" và "Middle" rất nhỏ, điều này có thể phản ánh rằng nhóm đối tượng có học vấn thấp không được thu thập đầy đủ hoặc không thuộc trọng tâm phân tích.



Tuy nhiên khi tính thêm giá trị null thì kết quả cho ra vô cùng tệ như biểu đồ sau:



Nhận xét: Với 96.6% giá trị bị thiếu, cột "education" gần như không có tính khả thi và đáng tin cậy cho quá trình phân tích. Sự thiếu hụt nghiêm trọng này có thể làm giảm độ chính xác của phân tích và dự đoán, dẫn đến kết quả không đáng tin cậy.

❖ birth - thông tin về năm sinh

Trường thông tin này ghi nhận về năm sinh của học viên và chứa lượng mẫu thiếu giá trị lên đến 9145726 mẫu. Đây cũng là một con số rất lớn.

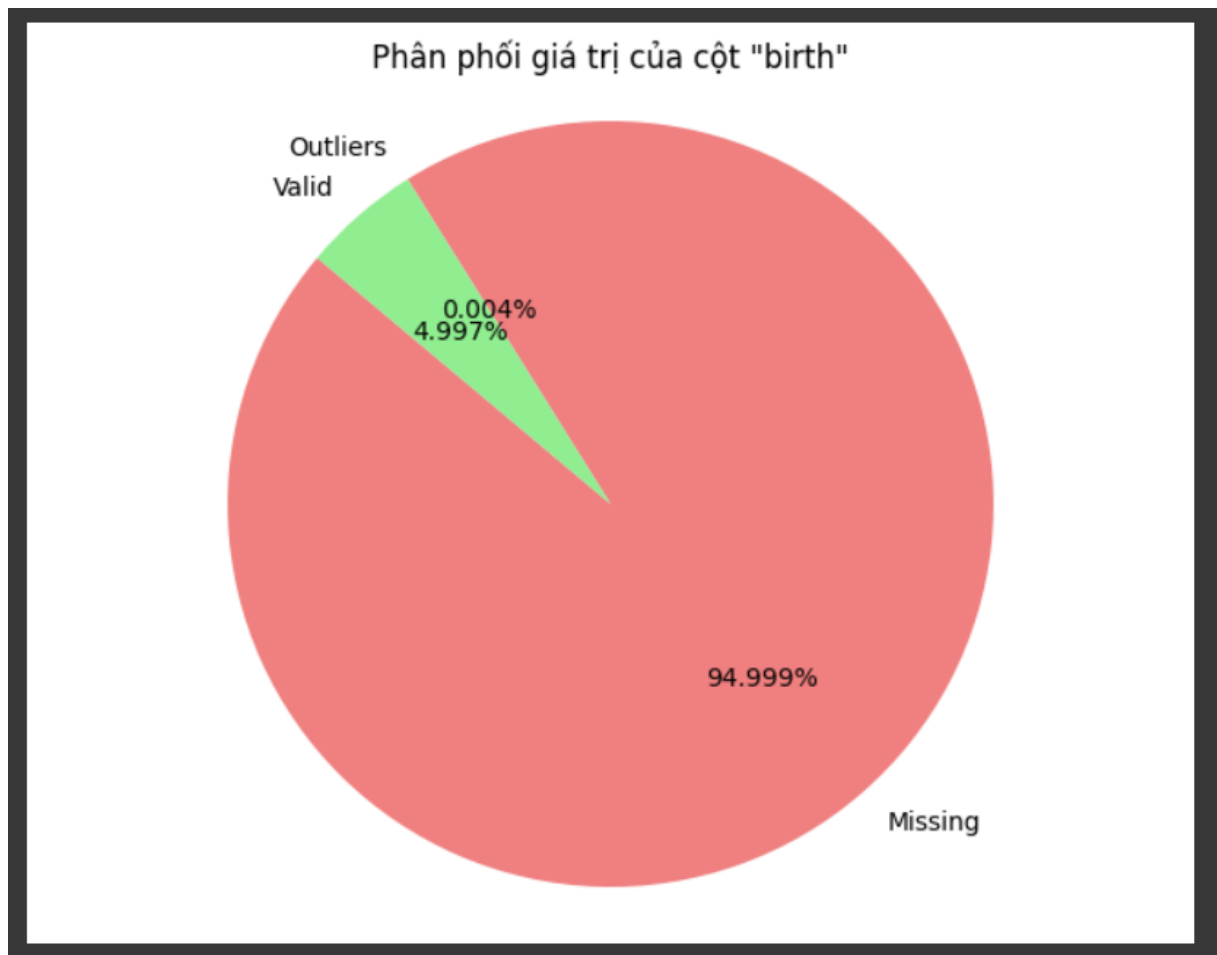
```
[ ] user_info.birth.describe()
```

	birth
count	481422.000000
mean	1990.901448
std	10.714057
min	996.000000
25%	1989.000000
50%	1993.000000
75%	1995.000000
max	7381.000000

dtype: float64

Nhận xét: Cột “birth” chứa các giá trị không hợp lệ với min là 996.0 và max là 7381.0. Những giá trị không hợp lệ này có thể xuất phát từ lỗi nhập liệu hoặc quá trình thu thập dữ liệu. Điều này có thể ảnh hưởng đến tính chính xác và khả năng khái quát của phân tích dữ liệu.

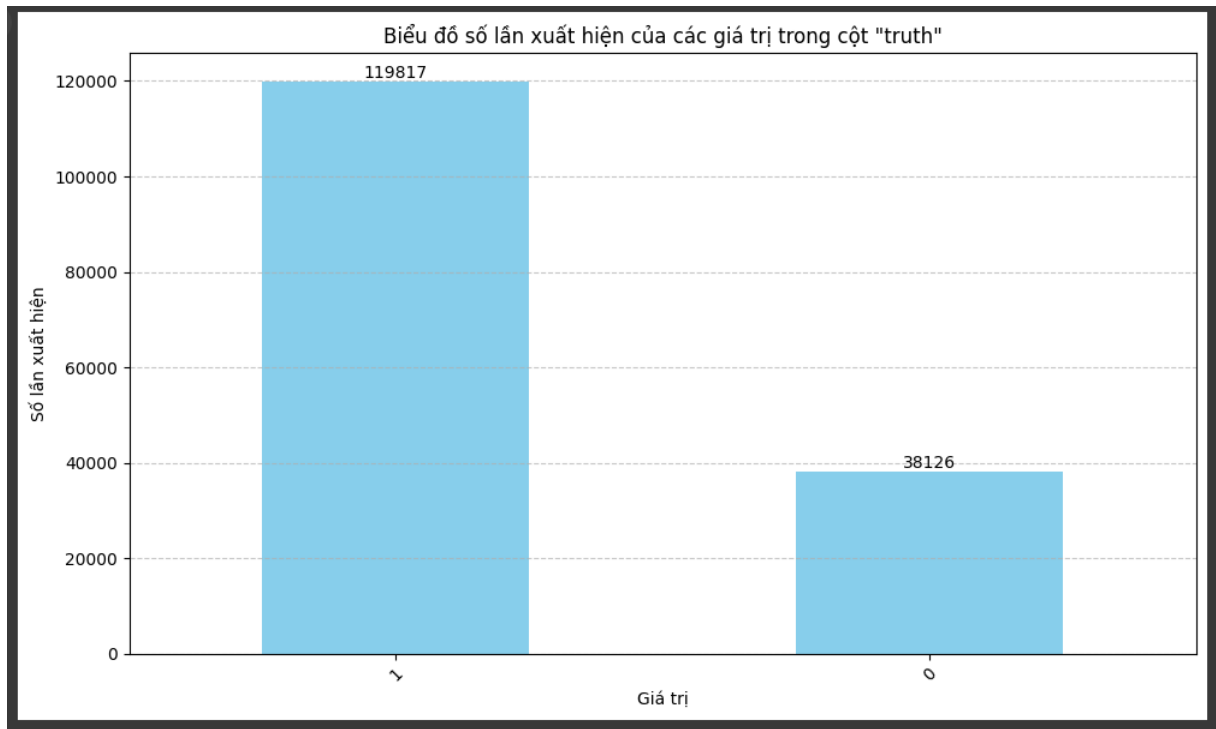
Nhóm quyết định lấy khoảng giá trị hợp lệ trong cột "birth" là từ 1920 đến 2015. Số lượng giá trị không hợp lệ là 345. Đây là một con số rất nhỏ so với tổng số lượng mẫu của bộ dữ liệu.



Nhận xét: Với 94.999% giá trị bị thiếu trong cột "birth," tỷ lệ này rất cao và có thể gây ra những thách thức lớn trong quá trình phân tích và dự đoán dựa trên dữ liệu. Cột này có thể ảnh hưởng nghiêm trọng đến quá trình dự đoán do số lượng mẫu rất lớn và tỷ lệ giá trị bị thiếu quá cao.

2.3.4 *train_truth.csv* - dữ liệu về thông tin bỏ học

Dữ liệu này ghi nhận thông tin về việc học viên có bỏ học hay không, cung cấp nhãn cho đề án phân loại. Bộ dữ liệu chứa 157943 mẫu và bao gồm 2 cột chính: mã định danh đăng ký khóa học của học viên (*enroll_id*) và thông tin về việc học viên có bỏ học hay không (*truth*). Nhóm sẽ chủ yếu khai thác thông tin từ cột "truth". Cột "truth" có 2 nhãn: nhãn “” là bỏ học và nhãn “0” là không bỏ học.

**Nhận xét:**

Sự phân bố của cột "truth" cho thấy một sự mất cân đối rõ rệt giữa hai giá trị. Với 75.9% cho giá trị 1, có sự chênh lệch đáng kể giữa hai lớp. Điều này có thể ảnh hưởng đến hiệu

suất của mô hình dự đoán, đặc biệt trong các bài toán phân loại như dự đoán khả năng bỏ học, vì mô hình có thể dễ bị chệch hướng hơn về dự đoán lớp có tỷ lệ lớn hơn.

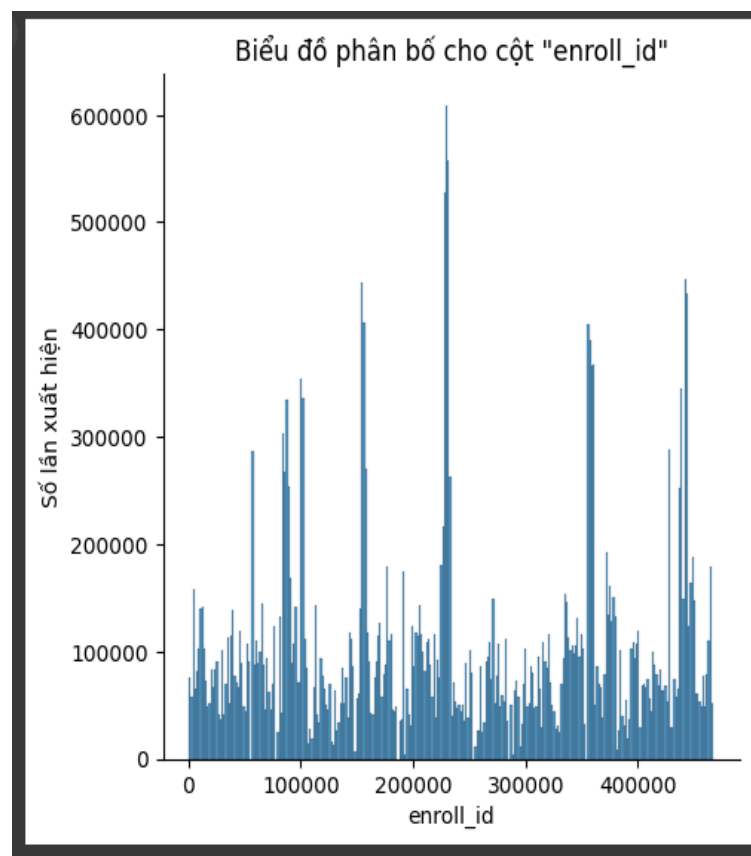
2.3.5 *train_log.csv* - dữ liệu nhật ký hoạt động

Dữ liệu này ghi nhận thông tin về nhật ký hoạt động học viên. Bộ dữ liệu chứa 29165540 mẫu và bao gồm 7 cột chính:

- mã định danh đăng ký khóa học của học viên (`enroll_id`)
- id của người dùng (`username`)
- id của khóa học (`course_id`)
- id của phiên học (`session_id`)
- hành động của người dùng (`action`)
- đối tượng (`object`)
- thời gian xảy ra hành động (`time`)

❖ Thông tin `enroll_id`

Đây là thông tin mã định danh đăng ký khóa học của mỗi học viên, do đó, trong bộ dữ liệu có thể có rất nhiều mẫu có cùng mã định danh đăng ký khóa học.



Qua biểu đồ cho thấy số lượng mẫu enroll_id phân bố không đều với sự chênh lệch lớn. Một số enroll_id xuất hiện rất nhiều lần, như enroll_id 191581 với 128,992 lần, trong khi nhiều enroll_id khác chỉ xuất hiện một vài lần.

Sự chênh lệch này có thể phản ánh mức độ tương tác cao từ những học viên chăm chỉ hoặc học lâu năm. Do đó, tần suất xuất hiện của enroll_id có thể là một tiêu chí hữu ích cho mô hình dự đoán khả năng bỏ học, nếu được xử lý cẩn thận. Việc này đòi hỏi các biện pháp tiền xử lý phù hợp để giảm thiểu thiên lệch và đảm bảo mô hình không bị ảnh hưởng bởi sự phân bố không đồng đều này.

❖ Thông tin username

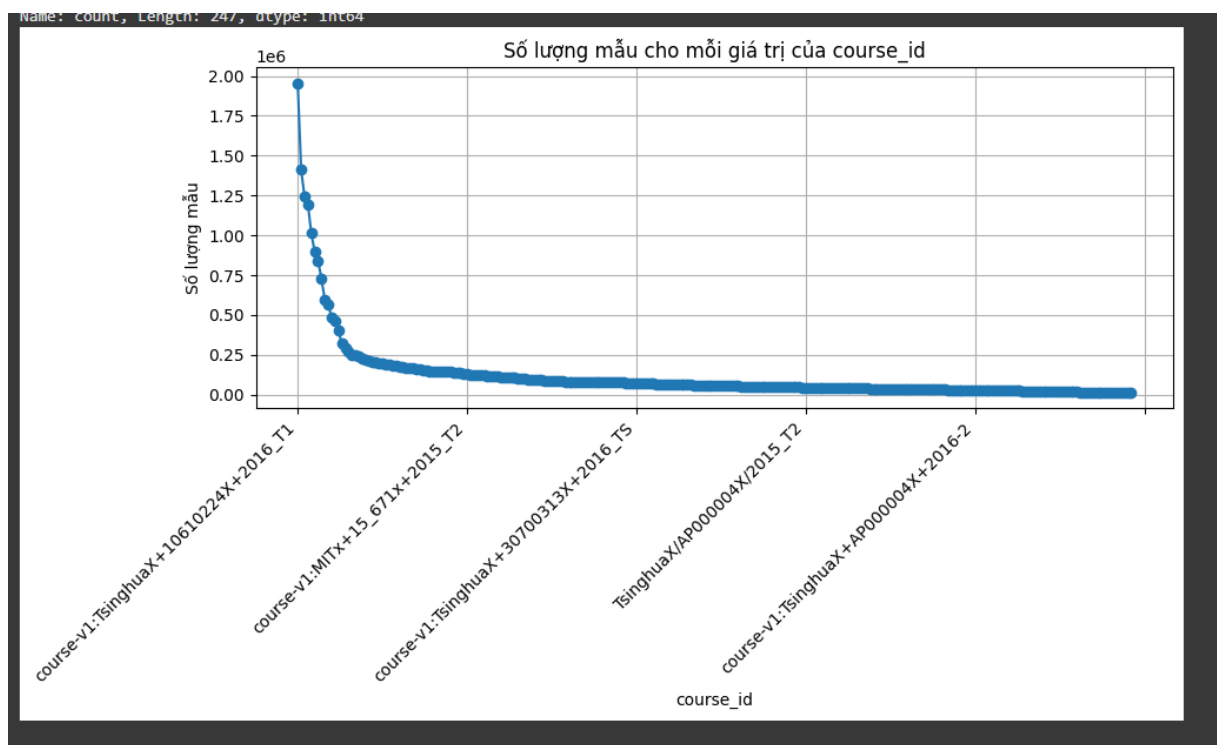
Cột username trong train_log.csv có ý nghĩa tương tự như user_id trong user_info.csv. Số lượng mã định danh học viên trong train_log.csv sẽ ít hơn vì user_info.csv bao gồm thông tin được ghi nhận qua nhiều năm, phản ánh một tập dữ liệu lớn hơn và toàn diện hơn về các học viên đã tham gia.

Sau khi kiểm tra thì tất cả các username trong train_log.csv đều có tồn tại trong user_info.csv

❖ Thông tin course_id

Đây là cũng chính là mã định danh trong course_info.csv, nhưng với số lượng ít hơn gồm 247 khóa học.

Thống kê số lượng mẫu trong train_log.csv của mỗi khóa học:



```

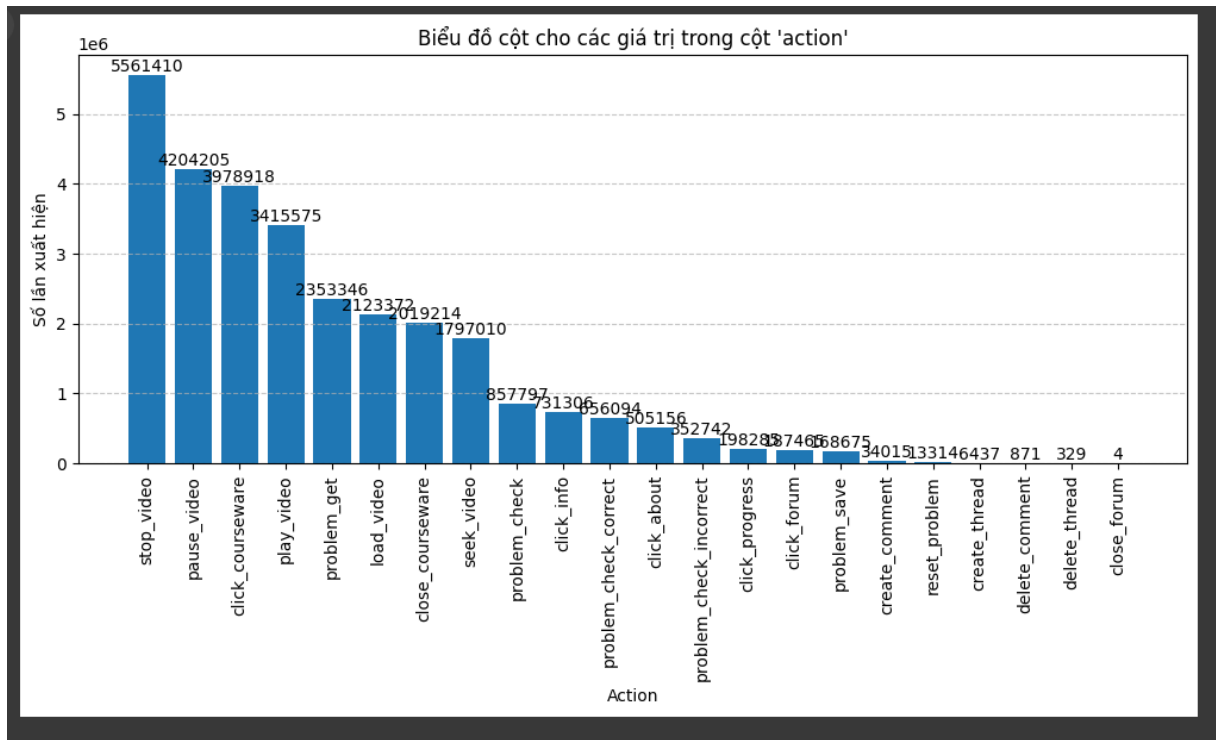
course_id
course-v1:TsinghuaX+10610224X+2016_T1      1955130
course-v1:TsinghuaX+10610183_2X+2016_T2     1410371
course-v1:TsinghuaX+30640014+2016_T2        1245965
course-v1:TsinghuaX+10610224X+2017_T1       1194954
course-v1:TsinghuaX+10610204X_2015_2+2015_T2 1013470
...
UQx/Write101_x/_                           13498
course-v1:HIT+13SC20301820+2015_T2          12347
course-v1:TsinghuaX+30150303X+2015_T2       12338
course-v1:TsinghuaX+40250074X+2016_T2       12042
TsinghuaX/THU00022X/2015_T1                 11909
Name: count, Length: 247, dtype: int64

```

Nhận xét: Sự chênh lệch lớn về số lượng tương tác giữa các khóa học được thể hiện rõ trên biểu đồ. Khóa học có số lượng tương tác cao nhất đạt 1,955,130, trong khi khóa học có số lượng tương tác thấp nhất chỉ có 11,909. Sự chênh lệch này phản ánh sự khác biệt về sự phổ biến hoặc tầm quan trọng của các khóa học trong hệ thống. Các yếu tố như chất lượng nội dung, sự phổ biến của chủ đề có thể ảnh hưởng đến mức độ tương tác của các khóa học này.

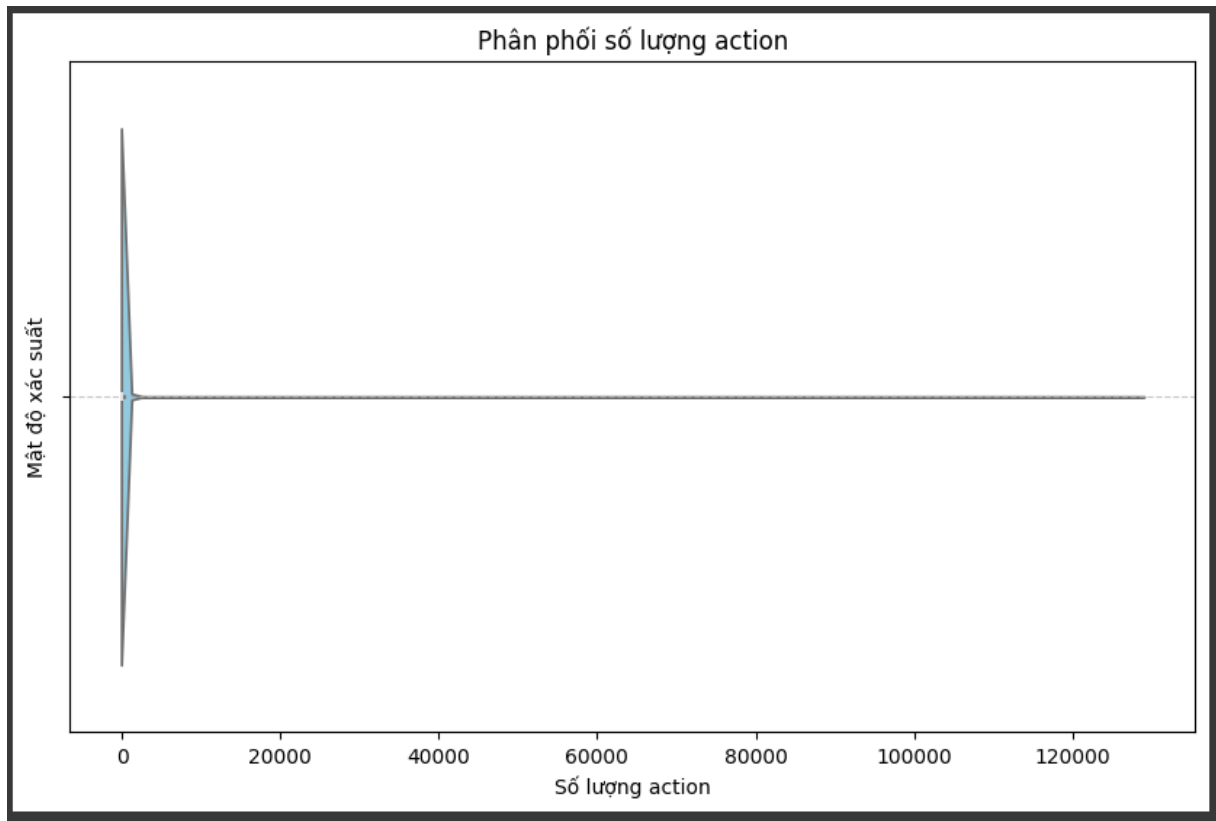
❖ Thông tin action

Đây là thông tin ghi lại tương tác với khóa học của mỗi học viên như stop_video, pause_video, click_courseware, ... Tổng cộng có tất cả 22 loại tương tác khác nhau.



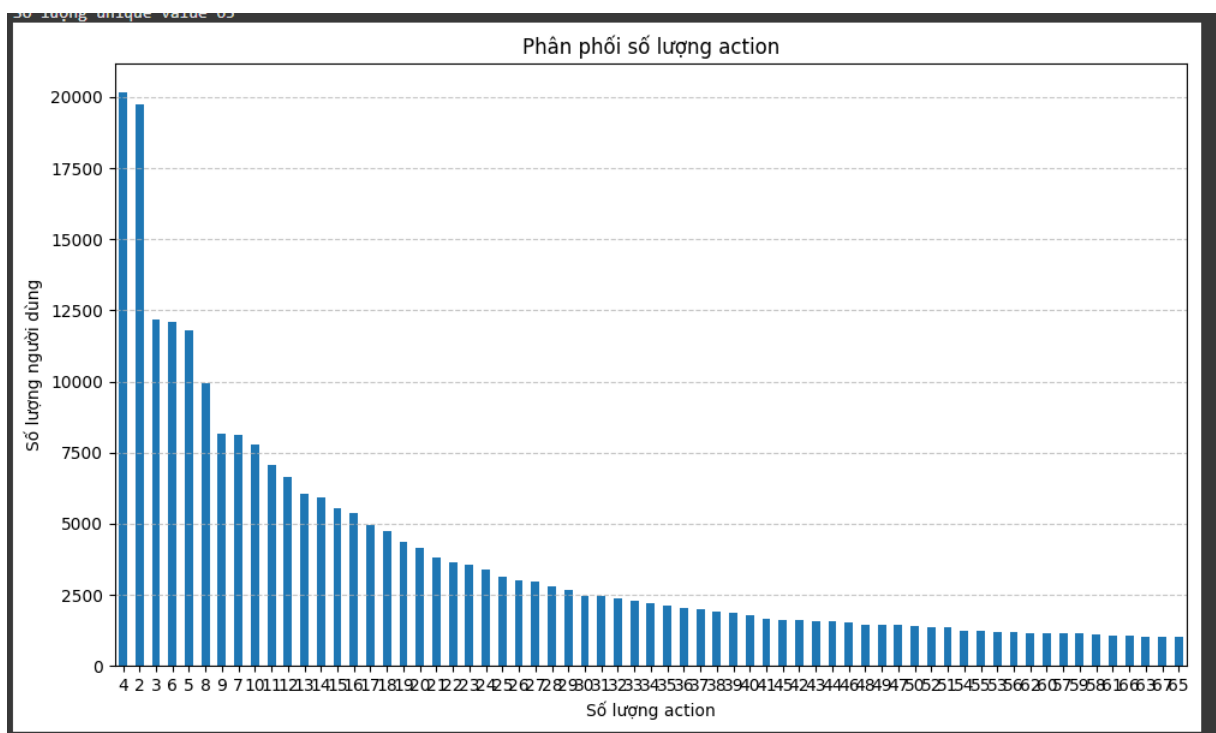
Nhận xét: Sự chênh lệch trong các hoạt động khóa học phản ánh sự đa dạng trong hành vi học tập của học viên. Các hoạt động phổ biến gồm xem video, tương tác nội dung và giải bài tập, trong khi các hoạt động như bình luận hoặc xóa chủ đề ít phổ biến hơn. Sự chênh lệch này cung cấp thông tin quan trọng để hiểu và dự đoán hành vi học tập của học viên, giúp cải thiện dự đoán về khả năng bỏ học.

Tiến hành vẽ biểu đồ violin plot để kiểm tra phân phối số lượng tương tác theo mỗi phiên học:



Nhận xét: Số lượng hành động trong mỗi phiên học rất khác nhau, nhưng phần lớn đều tập trung ở mức rất nhỏ. Từ đây có thể suy ra ở các phiên học học viên không tương tác quá nhiều với khóa học.

Để trực quan hơn thì nhóm chỉ xét những phiên học có số lượng hành động lớn hơn 1:

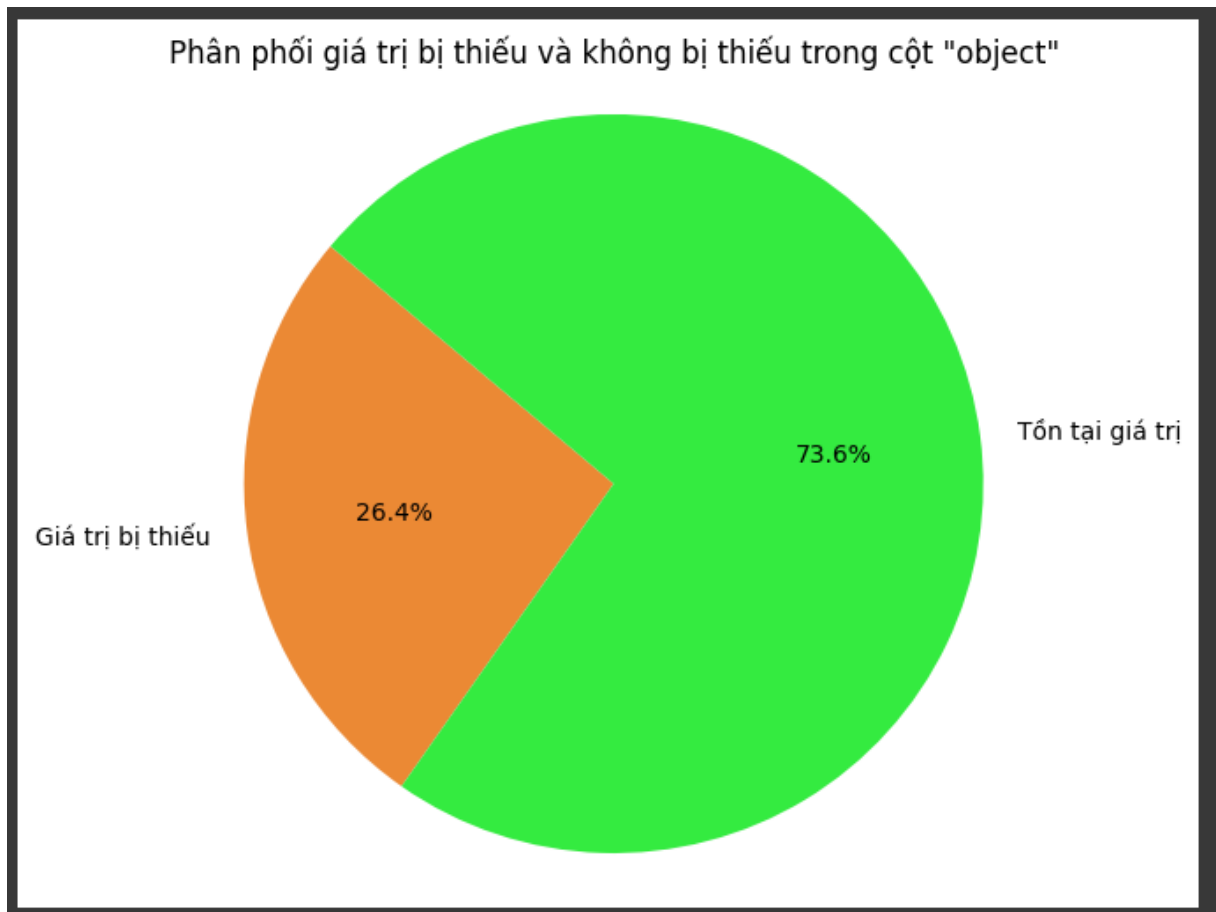


Nhận xét: Dựa vào biểu đồ cho thấy sự chênh lệch trong phân bố vẫn khá đáng kể, với một số giá trị xuất hiện tần suất cao hơn đáng kể so với các giá trị khác. Điều này cho thấy cần thiết phải thực hiện các biện pháp xử lý hiệu quả để khai thác thông tin từ dữ liệu một cách toàn diện và đáng tin cậy.

❖ Thông tin object

Trường thông tin này chứa thông tin về đối tượng mà hành động cụ thể liên quan đến. Ví dụ, nếu hành động là "click" thì đối tượng có thể là một video, tài liệu hoặc bất kỳ nội dung học tập nào mà người học tương tác.

Đây là cột duy nhất chứa giá trị Null trong file train_log.csv.

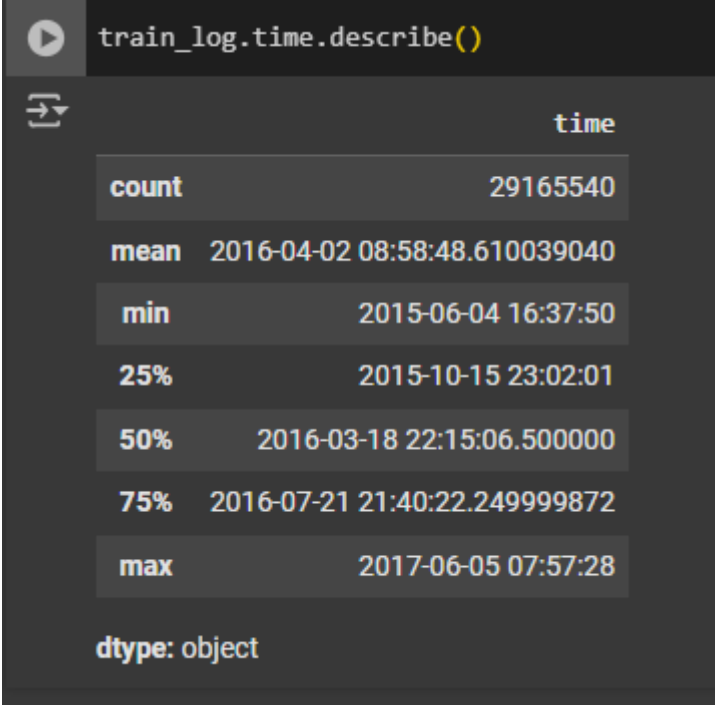


Cột "object" là cột ghi nhận sự phản hồi của học viên đối với action. Có 26.4% trong tổng số mẫu bị thiếu giá trị.

Nhận xét: Cột "object" có thể là tương tự với cột "course_id". Tuy nhiên nó bị thiếu giá trị. Với bản chất, chủ đề của các khóa học khác nhau thì việc phản hồi tương tác cũng sẽ khác nhau. Cần cân nhắc lại xem cột này có hữu ích xong quá trình xử lý và đưa ra dự đoán hay không?

❖ Thông tin time

Trường thông tin này ghi nhận thời điểm mà học viên tương tác với các thành phần của khóa học



```
train_log.time.describe()
```

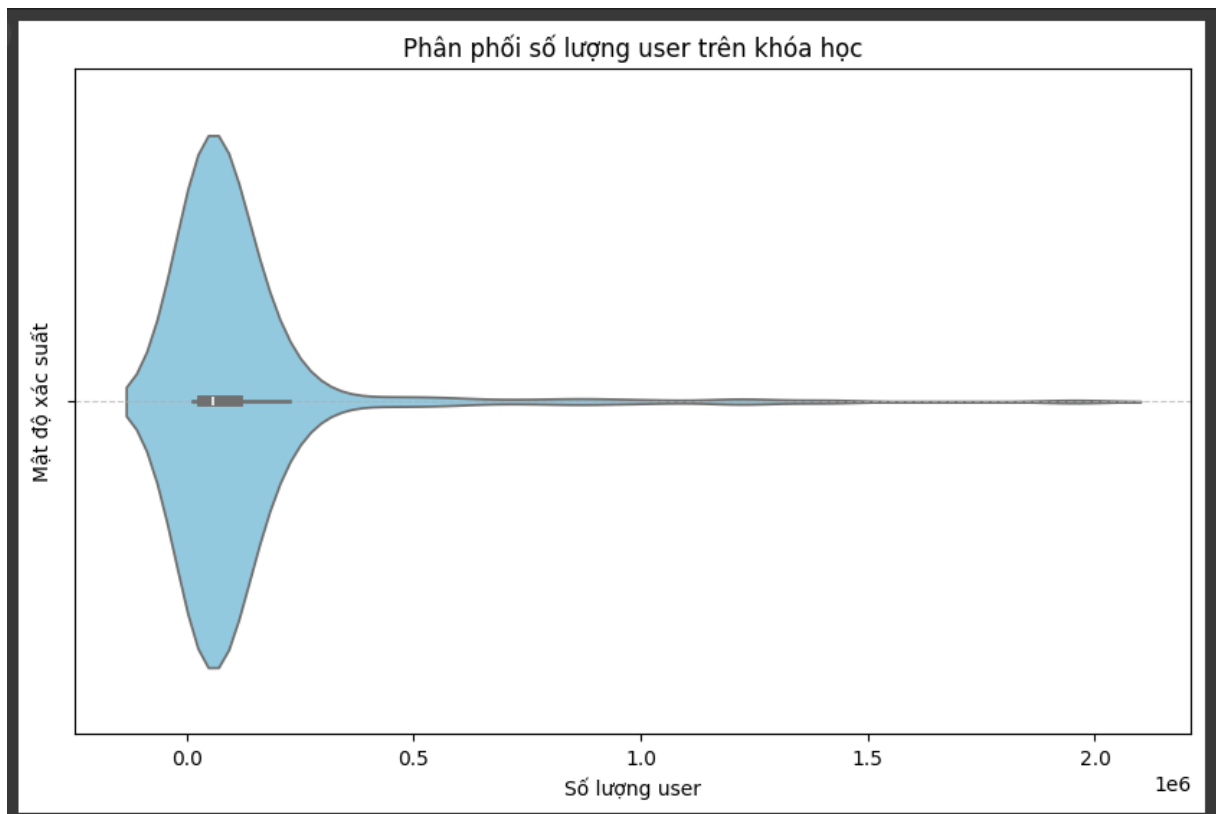
	time
count	29165540
mean	2016-04-02 08:58:48.610039040
min	2015-06-04 16:37:50
25%	2015-10-15 23:02:01
50%	2016-03-18 22:15:06.500000
75%	2016-07-21 21:40:22.249999872
max	2017-06-05 07:57:28

dtype: object

Theo bảng tóm tắt trên thì cột time phân bố giá trị khá hợp lý. (Giá trị min 04/06/2015 có thể tạm chấp nhận được mặc dù bộ dữ liệu được thu thập từ 08/2015)

❖ Thông tin số lượng user trên course_id

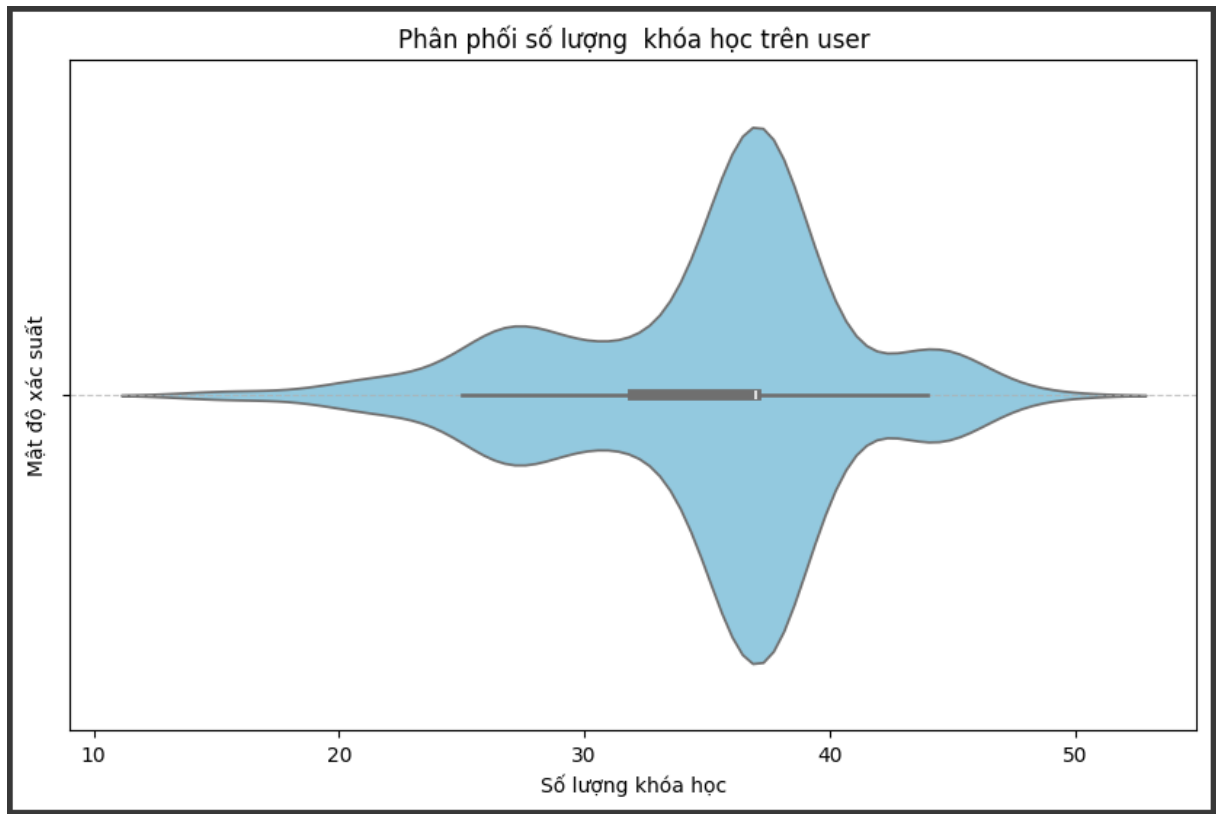
Tiến hành thống kê số lượng user trên từng khóa học để có thể hiểu ra hơn về sự phân phối của người dùng trên các khóa học cụ thể:



Nhận xét: Trục x của biểu đồ đo bằng triệu ($1e6$), với phần lớn giá trị dưới 0.5 triệu người dùng. Điều này cho thấy nhiều khóa học có ít người dùng, trong khi một số ít có nhiều người dùng, cho thấy sự phân bố không đồng đều về sự quan tâm và tham gia của người dùng.

❖ Thông tin số lượng course trên user

Tiến hành thống kê số lượng course trên từng người dùng để có thể hiểu ra hơn về mức độ tham gia khóa học của người dùng:



Nhận xét: Khoảng phổ biến nhất là từ 20 đến dưới 50 khóa học. Điều này cho thấy học viên tham gia nhiều khóa học để nâng cao kiến thức và kỹ năng. Thông tin này giúp hiểu rõ hơn về sự tham gia và cam kết của học viên, từ đó đưa ra các chiến lược cải thiện trải nghiệm học tập và hiệu quả khóa học.

Chương 3. QUÁ TRÌNH THỰC NGHIỆM

3.1 Xử lý dữ liệu

3.1.1 Tập dữ liệu *course_info*

3.1.1.1 Ngày bắt đầu khóa học “start”

Chuyển đổi 2 cột dữ liệu “start” và “end” sang kiểu dữ liệu datetime, với định dạng “YYYY-MM-DD”

- Tạo thêm 2 cột mới “start_date” và “end_date” và chuyển đổi 2 cột “start” và “end” sang kiểu date và gán vào 2 cột mới. Sau đó dùng hàm `.drop()` để xóa 2 cột “start” và “end”.

```

course_date = course_info.copy()
course_date['start_date'] = pd.to_datetime(course_date['start']).dt.normalize()
course_date['end_date'] = pd.to_datetime(course_date['end']).dt.normalize()
course_date.drop(columns=['start', 'end'], inplace=True)

course_date.info()
course_date

```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6410 entries, 0 to 6409
Data columns (total 6 columns):
#   Column          Non-Null Count  Dtype
---  ---
0   id               6410 non-null   object
1   course_id        6410 non-null   object
2   course_type      6410 non-null   int64
3   category         1454 non-null   object
4   start_date       6410 non-null   datetime64[ns]
5   end_date         5877 non-null   datetime64[ns]
dtypes: datetime64[ns](2), int64(1), object(3)
memory usage: 300.6+ KB

```

	id	course_id	course_type	category	start_date	end_date
0	6561	course-v1:CPVS+CPVS-HDLSC001+20160901	0	NaN	2016-11-16	2016-12-31
1	5557	course-v1:SCUT+144282+201709	0	NaN	2016-09-01	2017-02-28
2	9433	course-v1:ZK+06093+J	0	NaN	2018-01-01	2020-01-01
3	8320	course-v1:nuist+001+2016-T1	0	NaN	2017-03-01	2017-07-01
4	231	FUDAN/CFD004/2014.9-2015.1	0	NaN	2014-09-10	2015-09-10
...
6405	10493	course-v1:NBUX+lzu_MH001x+2017_T1	0	NaN	2017-04-10	2017-05-21
6406	11058	course-v1:Train+Train12+2017_T1	0	NaN	2017-05-01	2017-05-31
6407	4184	course-v1:nttec+10610204+2015_T2	0	NaN	2015-12-07	2016-12-07
6408	8333	course-v1:TsinghuaX+60610231+2016_T2_SP	1	philosophy	2016-08-25	NaT
6409	7848	course-v1:ncepubd+00510663X+2016_T2	0	NaN	2016-10-12	2016-12-30

6410 rows x 6 columns

Các khóa học có ngày bắt đầu không nằm trong khoảng hợp lệ thì tất cả sẽ được chuyển về ngày “2015-01-01”.

- Sử dụng hàm `.where()` để gán giá trị mới cho cột “start_date” dựa trên điều kiện giá trị trong cột “start_date” có nhỏ hơn “2015-01-01” hoặc lớn hơn “2017-08-31” hay không, nếu đúng thì cột “start_date” sẽ được gán là ngày “2015-01-01”

```

course_date['start_date'] = np.where(
    (course_date['start_date'] < pd.to_datetime('2015-01-01')) |
    (course_date['start_date'] > pd.to_datetime('2017-08-31')) ,
    pd.to_datetime('2015-01-01'),
    course_date['start_date']
)
course_date['start_date'] = pd.to_datetime(course_date['start_date'])

course_date.info()
course_date

```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 6410 entries, 0 to 6409
Data columns (total 6 columns):
#   Column          Non-Null Count  Dtype
---  ---
0   id               6410 non-null   object
1   course_id        6410 non-null   object
2   course_type      6410 non-null   int64
3   category         1454 non-null   object
4   start_date       6410 non-null   datetime64[ns]
5   end_date         5877 non-null   datetime64[ns]
dtypes: datetime64[ns](2), int64(1), object(3)
memory usage: 300.6+ KB

```

	id	course_id	course_type	category	start_date	end_date
0	6561	course-v1:CPVS+CPVS-HDLSC001+20160901	0	NaN	2016-11-16	2016-12-31
1	5557	course-v1:SCUT+144282+201709	0	NaN	2016-09-01	2017-02-28
2	9433	course-v1:ZK+06093+J	0	NaN	2015-01-01	2020-01-01
3	8320	course-v1:nuist+001+2016-T1	0	NaN	2017-03-01	2017-07-01
4	231	FUDAN/CFD004/2014.9-2015.1	0	NaN	2015-01-01	2015-09-10

3.1.1.2 Ngày kết thúc khóa học “end”

Các khóa học có ngày kết thúc khóa học là NULL, nhỏ hơn ngày bắt đầu và năm kết thúc lớn hơn 2018 sẽ được xử lý.

- Chọn các dòng theo các điều kiện trên và sử dụng hàm `.concat()` để kết hợp 3 DataFrame thành một DataFrame mới là `invalid_rows`.

```

invalid_rows = pd.concat([course_date[pd.DatetimeIndex(course_date['end_date']).year > 2018],
                           course_date[course_date['end_date'].isnull()],
                           course_date[course_date['end_date'] < course_date['start_date']]
                           ])

invalid_rows.info()
invalid_rows

```

```

<class 'pandas.core.frame.DataFrame'>
Index: 903 entries, 2 to 6180
Data columns (total 6 columns):
#   Column          Non-Null Count  Dtype
---  ---
0   id               903 non-null    object
1   course_id        903 non-null    object
2   course_type      903 non-null    int64
3   category         471 non-null    object
4   start_date       903 non-null    datetime64[ns]
5   end_date         370 non-null    datetime64[ns]
dtypes: datetime64[ns](2), int64(1), object(3)
memory usage: 49.4+ KB

```

	id	course_id	course_type	category	start_date	end_date
2	9433	course-v1:ZK+06093+J	0	NaN	2015-01-01	2020-01-01
14	10988	course-v1:SDJTU+lyh+2017_T2	0	NaN	2017-04-01	2030-08-31
46	9378	course-v1:ZK+00316+J	0	NaN	2015-01-01	2020-01-01
48	11495	course-v1:CQU+TSJYSSG00031+2017_T1	0	NaN	2017-05-01	2020-05-01
68	11384	course-v1:CQU+TSJYHG00062+2017_T1	0	NaN	2017-05-01	2020-05-01
...

Tính độ dài của khóa học đối với các khóa học “duration” có “start_date” và “end_date” hợp lệ.

- Sử dụng hàm `.isin()` để kiểm tra xem mỗi “id” trong `course_date` có nằm trong cột “id” của `invalid_rows` hay không. Tiếp theo sử dụng toán tử “~” để lấy giá trị ngược lại, tức là các dòng có cột “id” không nằm trong `invalid_rows`. Sử dụng hàm `.loc[]` để chọn các dòng từ `course_date` thỏa mãn điều kiện và lưu và `valid_rows`.

```
[ ] valid_rows = course_date.loc[~course_date['id'].isin(invalid_rows['id'])]
valid_rows.info()
valid_rows
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 5507 entries, 0 to 6409
Data columns (total 6 columns):
#   Column          Non-Null Count  Dtype
---  ---
0   id              5507 non-null   object
1   course_id       5507 non-null   object
2   course_type     5507 non-null   int64
3   category        983 non-null    object
4   start_date      5507 non-null   datetime64[ns]
5   end_date        5507 non-null   datetime64[ns]
dtypes: datetime64[ns](2), int64(1), object(3)
memory usage: 301.2+ KB
```

	id	course_id	course_type	category	start_date	end_date
0	6561	course-v1:CPVS+CPVS-HDLSC001+20160901	0	NaN	2016-11-16	2016-12-31
1	5557	course-v1:SCUT+144282+201709	0	NaN	2016-09-01	2017-02-28
3	8320	course-v1:nuist+001+2016-T1	0	NaN	2017-03-01	2017-07-01
4	231	FUDAN/CFD004/2014.9-2015.1	0	NaN	2015-01-01	2015-09-10
5	7645	course-v1:ANUx+EBM05x+3T2017	0	NaN	2015-01-01	2018-09-17
...

- Tạo cột mới “duration”, lấy “end_date” trừ “start_date” để lấy khoảng cách khóa học và gán vào cột “duration” vừa tạo.

```
[ ] valid_rows.loc[:, 'duration'] = (valid_rows['end_date'] - valid_rows['start_date']).dt.days
valid_rows.info()
valid_rows
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 5507 entries, 0 to 6409
Data columns (total 7 columns):
#   Column          Non-Null Count  Dtype
---  ---
0   id              5507 non-null   object
1   course_id       5507 non-null   object
2   course_type     5507 non-null   int64
3   category        983 non-null    object
4   start_date      5507 non-null   datetime64[ns]
5   end_date        5507 non-null   datetime64[ns]
6   duration        5507 non-null   int64
dtypes: datetime64[ns](2), int64(2), object(3)
memory usage: 344.2+ KB
```

	id	course_id	course_type	category	start_date	end_date	duration
0	6561	course-v1:CPVS+CPVS-HDLSC001+20160901	0	NaN	2016-11-16	2016-12-31	45
1	5557	course-v1:SCUT+144282+201709	0	NaN	2016-09-01	2017-02-28	180
3	8320	course-v1:nuist+001+2016-T1	0	NaN	2017-03-01	2017-07-01	122
4	231	FUDAN/CFD004/2014.9-2015.1	0	NaN	2015-01-01	2015-09-10	252
5	7645	course-v1:ANUx+EBM05x+3T2017	0	NaN	2015-01-01	2018-09-17	1355
...

Tiếp theo, tính trung bình độ dài khóa học của các khóa học hợp lệ

- Sử dụng hàm `.mean()` để lấy trung bình của cột “duration”

```
[ ] duration_mean = int(valid_rows['duration'].mean())
print(duration_mean)
```

↔ 232

Cuối cùng, tiến hành điền ngày kết thúc đối với các dòng dữ liệu có “end_date” không hợp lệ bằng cách lấy ngày bắt đầu cộng với độ dài trung bình của các khóa học hợp lệ. Và tạo thêm cột “duration” cho invalid_rows và gán giá trị bằng duration_mean.

```
invalid_rows['end_date'] = invalid_rows['start_date'] + pd.to_timedelta(duration_mean, unit="days")
invalid_rows['duration'] = duration_mean
invalid_rows.info()
invalid_rows
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 903 entries, 2 to 6180
Data columns (total 7 columns):
#   Column      Non-Null Count  Dtype
---  ---
0   id           903 non-null    object
1   course_id    903 non-null    object
2   course_type  903 non-null    int64
3   category     471 non-null    object
4   start_date   903 non-null    datetime64[ns]
5   end_date     903 non-null    datetime64[ns]
6   duration     903 non-null    int64
dtypes: datetime64[ns](2), int64(2), object(3)
memory usage: 56.4+ KB
```

	id	course_id	course_type	category	start_date	end_date	duration
2	9433	course-v1:ZK+06093+J	0	NaN	2015-01-01	2015-08-21	232
14	10988	course-v1:SDJTU+lyh+2017_T2	0	NaN	2017-04-01	2017-11-19	232
46	9378	course-v1:ZK+00316+J	0	NaN	2015-01-01	2015-08-21	232
48	11495	course-v1:CQU+TSJYSSG00031+2017_T1	0	NaN	2017-05-01	2017-12-19	232
68	11384	course-v1:CQU+TSJYHG00062+2017_T1	0	NaN	2017-05-01	2017-12-19	232
...

3.1.1.3 Điền khuyết dữ liệu cột “category”

Điền giá trị cột “category” bị thiếu với các khóa học có giá trị “category” là NULL.

- Sử dụng hàm `.notnull()` để lọc ra các dòng dữ liệu hợp lệ.

```
valid_rows = course_date[course_date['category'].notnull()]
valid_rows.info()
valid_rows
```

```
<class 'pandas.core.frame.DataFrame'>
Index: 1454 entries, 17 to 6180
Data columns (total 7 columns):
#   Column          Non-Null Count  Dtype
---  -
0   id               1454 non-null   object
1   course_id        1454 non-null   object
2   course_type      1454 non-null   int64
3   category         1454 non-null   object
4   start_date       1454 non-null   datetime64[ns]
5   end_date         1454 non-null   datetime64[ns]
6   duration         1454 non-null   int64
dtypes: datetime64[ns](2), int64(2), object(3)
memory usage: 90.9+ KB
```

	id	course_id	course_type	category	start_date	end_date	duration
17	577	TsinghuaX/70150023X/2015_T1	0	engineering	2015-03-02	2015-07-05	125
38	5735	course-v1:NJU+010101+2016_T1	0	social science	2016-03-28	2017-02-15	324
41	11612	course-v1:DYU+dyuglbbx+2017_T2	0	economics	2015-01-01	2017-12-25	1089
45	9290	course-v1:TsinghuaX+00690302_2+2017_T2	0	literature	2017-04-03	2017-05-31	58
59	2070	course-v1:TsinghuaX+00510663X+2015_T2	0	business	2015-10-12	2016-01-16	96
...

- Thống kê giá trị cột “category” trong valid_rows

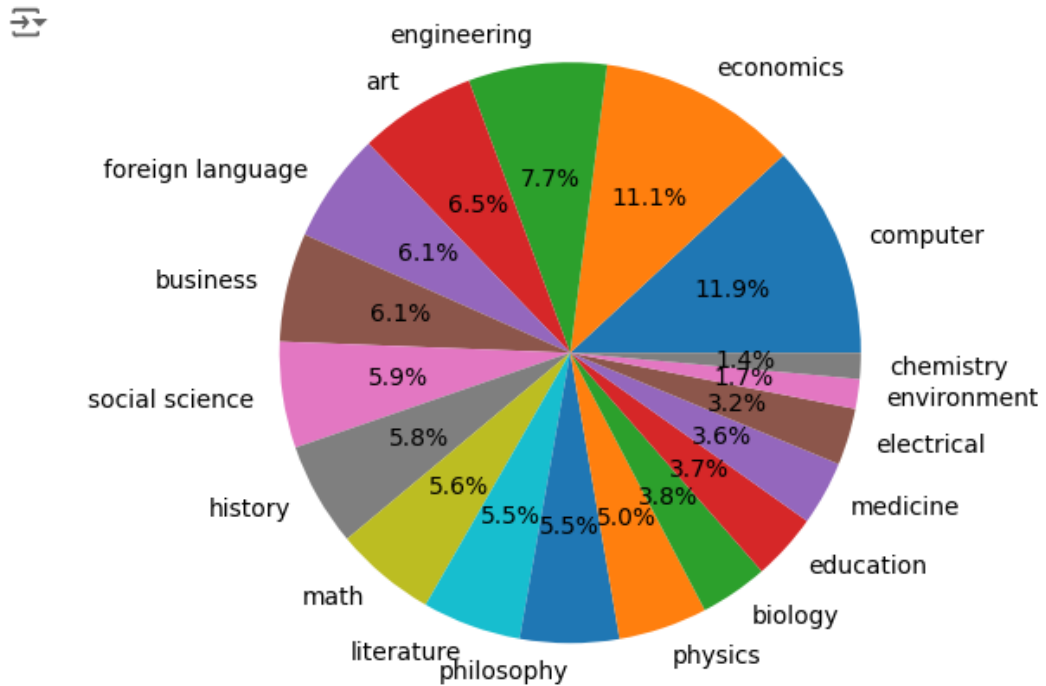
```

import matplotlib.pyplot as plt

valid_rows = course_date[course_date['category'].notnull()]
# Tạo dữ liệu
sizes = valid_rows['category'].value_counts()
labels = sizes.index

# Vẽ biểu đồ
plt.pie(sizes, labels=labels, autopct='%1.1f%%')
plt.axis('equal')
plt.show()

```



Tiến hành điền khuyết với các dòng có cột “category” theo tỉ lệ thống kê như trên

- Sử dụng hàm isnull() để lọc ra các dòng dữ liệu không hợp lệ. Hàm .shape[0] để trả về số lượng hàng trong cột “category”

```

invalid_rows = course_date[course_date['category'].isnull()]

# Tính số lượng hàng bị thiếu
num_missing = invalid_rows['category'].shape[0]

total = sum(sizes)
p = [size / total for size in sizes]

# Fill dữ liệu bị thiếu theo tỉ lệ
values = np.random.choice(labels, size=num_missing, p = p)
invalid_rows.loc[:, 'category'] = values
invalid_rows.info()
invalid_rows

```

```

<class 'pandas.core.frame.DataFrame'>
Index: 4956 entries, 0 to 6144
Data columns (total 7 columns):
#   Column      Non-Null Count  Dtype
---  -
0   id           4956 non-null   object
1   course_id    4956 non-null   object
2   course_type  4956 non-null   int64
3   category     4956 non-null   object
4   start_date   4956 non-null   datetime64[ns]
5   end_date     4956 non-null   datetime64[ns]
6   duration     4956 non-null   int64
dtypes: datetime64[ns](2), int64(2), object(3)
memory usage: 309.8+ KB

```

	id	course_id	course_type	category	start_date	end_date	duration
0	6561	course-v1:CPVS+CPVS-HDLSC001+20160901	0	education	2016-11-16	2016-12-31	45
1	5557	course-v1:SCUT+144282+201709	0	art	2016-09-01	2017-02-28	180
3	8320	course-v1:nuist+001+2016-T1	0	economics	2017-03-01	2017-07-01	122
4	231	FUDAN/CFD004/2014.9-2015.1	0	computer	2015-01-01	2015-09-10	252
5	7645	course-v1:ANUx+EBM05x+3T2017	0	environment	2015-01-01	2018-09-17	1355
...

3.1.2 Tập dữ liệu user_info

3.1.2.1 Điền khuyết các giá trị trong cột gender

Sau khi encode tập dữ liệu từ (male, female) thành (0,1) để dễ dàng tính toán. Chia tập dữ liệu thành 2 phần, valid_rows chứa các hàng có giá trị, invalid_rows chứa các hàng có dữ liệu bị khuyết.

- Trong tập invalid_rows, nhóm phân chia tỉ lệ giữa nam và nữ theo tỉ lệ 50/50, vì tỉ lệ giữa nam và nữ trong valid_rows là không chênh lệch quá nhiều.


```

▶ invalid_rows = user_info[user_info['gender'].isnull()]

# Tính số lượng hàng bị thiếu
num_missing = invalid_rows.shape[0]

# Fill dữ liệu bị thiếu theo tỉ lệ 50/50
values = np.random.choice([0,1], size=num_missing)
invalid_rows.loc[:, 'gender'] = values
invalid_rows

```




	user_id	gender	education	birth
4	7831	0	NaN	NaN
7	13631	1	NaN	NaN
20	46431	0	NaN	NaN
21	48231	1	NaN	NaN
27	64031	0	NaN	NaN
...
9627143	10435606	1	NaN	NaN
9627144	10437206	1	NaN	NaN
9627145	10438806	1	NaN	NaN
9627146	10440406	1	NaN	NaN
9627147	10442006	0	NaN	NaN

8420796 rows x 4 columns

- Sau đó gộp 2 tập dữ liệu lại.

```
user_info = pd.concat([valid_rows, invalid_rows], axis=0)
user_info
```



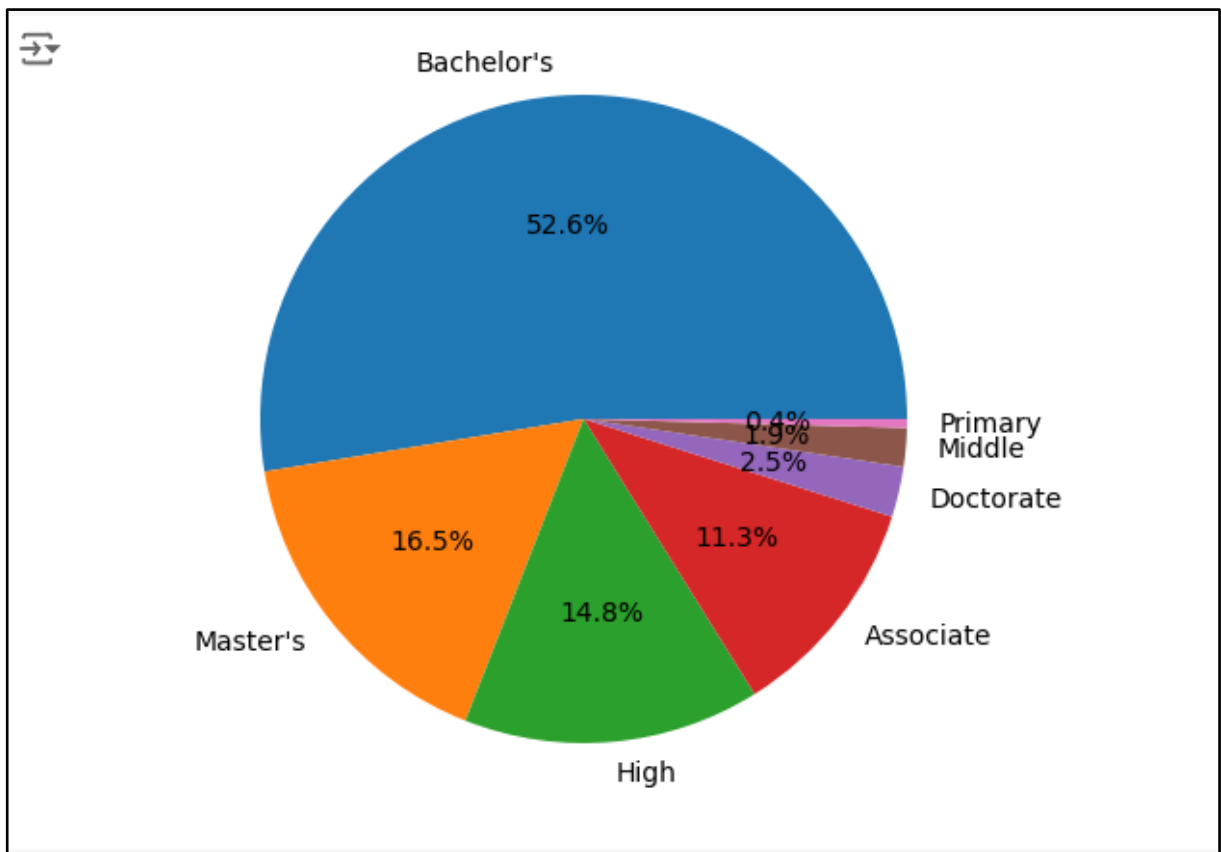
	user_id	gender	education	birth
0	631	0	High	1997.0
1	2631	0	Bachelor's	1990.0
2	4231	0	Associate	1991.0
3	6031	0	Bachelor's	1988.0
5	9631	0	Bachelor's	1992.0
...
9627143	10435606	1	NaN	NaN
9627144	10437206	1	NaN	NaN
9627145	10438806	1	NaN	NaN
9627146	10440406	1	NaN	NaN
9627147	10442006	0	NaN	NaN

9627148 rows x 4 columns

3.1.2.2 Điền khuyết các giá trị trong cột education

Chia tập dữ liệu thành 2 phần, `valid_rows` chứa các hàng có giá trị, `invalid_rows` chứa các hàng có dữ liệu bị khuyết.

- Trong tập `valid_rows`, nhóm tính tổng số lần xuất hiện của các giá trị và tỉ lệ xuất hiện của từng giá trị.



- Trong tập `invalid_rows`, từ tỉ lệ vừa tính được từ `valid_rows`, nhóm random các dữ liệu ở trên theo tỉ lệ đã tính toán để giữ được tỉ lệ ban đầu mà không bị quá nghiêng về một loại dữ liệu.

```

▶ invalid_rows = user_info[user_info['education'].isnull()]

# Tính số lượng hàng bị thiếu
num_missing = invalid_rows['education'].shape[0]

total = sum(sizes)
p = [size / total for size in sizes]

# Fill dữ liệu bị thiếu theo tỉ lệ
values = np.random.choice(labels, size=num_missing, p = p)
invalid_rows.loc[:, 'education'] = values
invalid_rows

```



	user_id	gender	education	birth
65	137231	0	High	1996.0
71	149031	0	Associate	1996.0
78	161231	0	Bachelor's	1998.0
88	179231	0	Bachelor's	1994.0
107	218831	1	Associate	1994.0
...
9627143	10435606	1	Bachelor's	NaN
9627144	10437206	1	High	NaN
9627145	10438806	1	Master's	NaN
9627146	10440406	1	Bachelor's	NaN
9627147	10442006	0	Associate	NaN

9298955 rows x 4 columns

3.1.2.3 Điền khuyết các giá trị trong cột birth

Chia tập dữ liệu thành 2 phần, `valid_rows` chứa các hàng có giá trị hợp lệ. Các giá trị trong `valid_rows` phải thoả mãn: không bị khuyết, có năm sinh phải lớn hơn 1920 và bé hơn 2015. Các giá trị trong `invalid_rows` là các hàng còn lại.

- Trong tập `invalid_rows`, nhóm chuyển tất cả các dữ liệu không hợp lệ thành NaN.
- Sau đó, gộp `invalid_rows` và `valid_rows` thành một tập dữ liệu. Dùng thuật toán `bfill()` để điền các dữ liệu bị thiếu bằng với các giá trị ở dưới nó, vì các nhóm học viên tham gia khoá học thường có chung độ tuổi.

```

not_null_rows = user_info[user_info['birth'].notnull()]

valid_rows = not_null_rows[(not_null_rows['birth'] > 1920) & (not_null_rows['birth'] < 2015)]

invalid_rows = not_null_rows[(not_null_rows['birth'] <= 1920) | (not_null_rows['birth'] >= 2015)]
invalid_rows.loc[:, 'birth'] = np.nan

invalid_rows = pd.concat([invalid_rows, user_info[user_info['birth'].isnull()]])

_ = pd.concat([invalid_rows, valid_rows], axis=0)
_['birth'] = _['birth'].bfill()
user_info = _
user_info

```

	user_id	gender	education	birth
72298	280849	1	Bachelor's	1997.0
202358	6776280	1	High	1997.0
208379	6775883	1	Bachelor's	1997.0
240417	10125494	1	Master's	1997.0
278448	3039301	0	High	1997.0
...
9548961	79056	0	Doctorate	2014.0
9579048	101553	1	Associate	1991.0
9585223	434554	0	Bachelor's	1989.0
9603149	168003	0	Associate	2014.0
9615301	409605	1	Bachelor's	1995.0

9627148 rows × 4 columns

3.1.3 Tập dữ liệu *prediction_log*

Chuyển đổi kiểu dữ liệu cột “time” từ kiểu chuỗi thành kiểu dữ liệu date cũng định dạng với các tập dữ liệu trên

```

[ ] train_date = train_log.copy()
train_date['time'] = pd.to_datetime(train_date['time']).dt.normalize()
train_date.info()
train_date

```

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 29165540 entries, 0 to 29165539
Data columns (total 7 columns):
#   Column      Dtype
---  ---
0   enroll_id   int64
1   username    int64
2   course_id   object
3   session_id  object
4   action      object
5   object      object
6   time        datetime64[ns]
dtypes: datetime64[ns](1), int64(2), object(4)
memory usage: 1.5+ GB

```

	enroll_id	username	course_id	session_id	action	object	time
0	772	5981	course-v1:TsinghuaX+70800232X+2015_T2	d8a9b787fa69063c34c73b9c29190b1c	click_about	NaN	2015-09-27
1	772	5981	course-v1:TsinghuaX+70800232X+2015_T2	d8a9b787fa69063c34c73b9c29190b1c	click_info	NaN	2015-09-27
2	773	1544995	course-v1:TsinghuaX+70800232X+2015_T2	2f02b86eb3ea2cbf0be11385a8dc62e5	pause_video	3dac5590435e43b3a65a9ae7426c16db	2015-10-19
3	773	1544995	course-v1:TsinghuaX+70800232X+2015_T2	2f02b86eb3ea2cbf0be11385a8dc62e5	load_video	3dac5590435e43b3a65a9ae7426c16db	2015-10-19
4	773	1544995	course-v1:TsinghuaX+70800232X+2015_T2	2f02b86eb3ea2cbf0be11385a8dc62e5	play_video	3dac5590435e43b3a65a9ae7426c16db	2015-10-19
...

```
test_log.info()
test_log
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12944862 entries, 0 to 12944861
Data columns (total 7 columns):
#   column      Dtype
---  -
0   enroll_id   int64
1   username    int64
2   course_id   object
3   session_id  object
4   action       object
5   object       object
6   time         object
dtypes: int64(2), object(5)
memory usage: 691.3+ MB
```

	enroll_id	username	course_id	session_id	action	object	time
0	775	1520977	course-v1:TsinghuaX+70800232X+2015_T2	5f421f644193c2d48c84df42aaf7e48b	load_video	3dac5590435e43b3a65a9ae7426c16db	2015-10-15T22:14:11
1	775	1520977	course-v1:TsinghuaX+70800232X+2015_T2	5f421f644193c2d48c84df42aaf7e48b	load_video	3169d758ee2d4262b07f0113df743c42	2015-10-15T22:43:35
2	775	1520977	course-v1:TsinghuaX+70800232X+2015_T2	5f421f644193c2d48c84df42aaf7e48b	play_video	3169d758ee2d4262b07f0113df743c42	2015-10-15T22:43:40
3	775	1520977	course-v1:TsinghuaX+70800232X+2015_T2	5f421f644193c2d48c84df42aaf7e48b	pause_video	3169d758ee2d4262b07f0113df743c42	2015-10-15T22:55:38
4	775	1520977	course-v1:TsinghuaX+70800232X+2015_T2	5f421f644193c2d48c84df42aaf7e48b	stop_video	3169d758ee2d4262b07f0113df743c42	2015-10-15T22:55:38

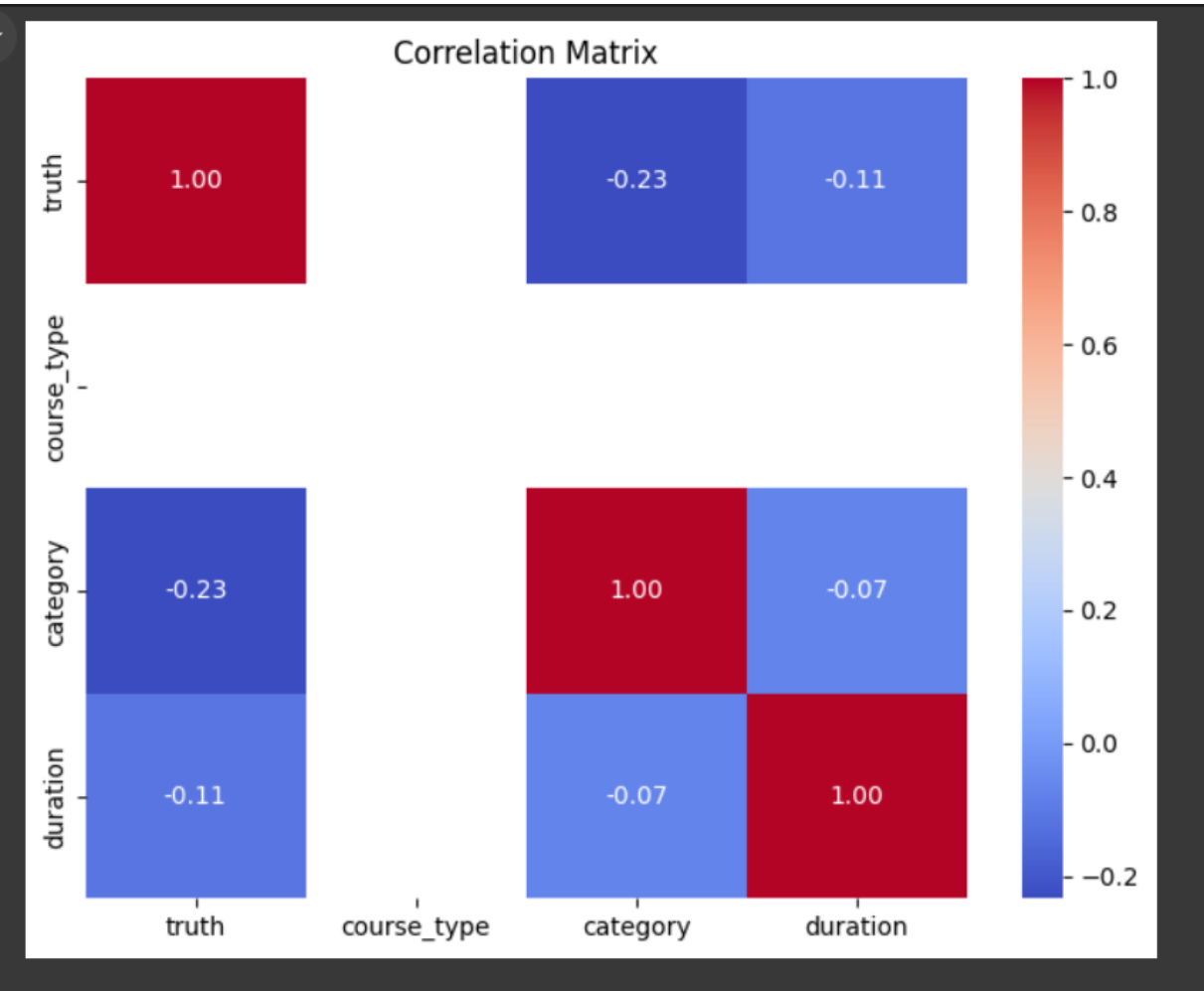
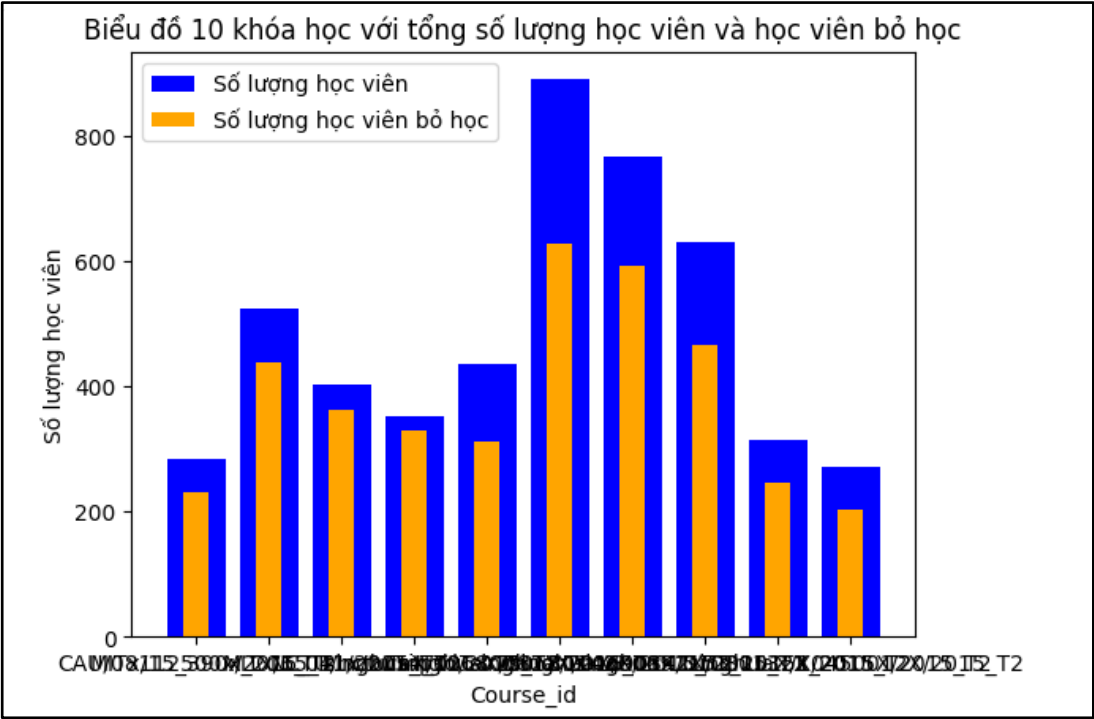
3.2 Trích xuất đặc trưng

3.2.1 Đặc trưng khoá học

3.2.1.1 Cơ sở trích xuất các đặc trưng

Để xác định trích xuất các đặc trưng nào nên trích xuất, sự ảnh hưởng của đặc trưng đó đến việc dự đoán. Để tránh trích xuất giảm tỷ lệ dự đoán, nhóm đã tìm các đặc trưng có thể ảnh hưởng đến việc dự đoán và phân tích, kiểm tra xem việc trích xuất đặc trưng đó các tác động như thế nào đến kết quả dự đoán.

- Số lượng học viên đã tham gia vào khóa học trước ngày đăng ký và số lượng học viên đã bỏ học trước ngày đăng ký: Nhóm đã tính tổng số lượng học viên đã tham gia vào khóa học và số học viên đã bỏ học khóa học để tính tỉ lệ giữa hai số liệu có ảnh hưởng đến việc bỏ học khóa học đó của học viên không. Sau khi có được biểu đồ như dưới, ta có thể thấy được với một khóa học chất lượng thì thường được nhiều học viên tham gia và tỉ lệ học viên bỏ học ở khóa học đó cũng sẽ giảm.



- Thời gian khoá học: Giá trị tương quan giữa duration và truth của một học viên là -0.11 và chỉ số này tuy thấp nhưng có phần ảnh hưởng đến kết quả dự đoán. Nó thể hiện được độ dài khoá học tỉ lệ nghịch đến việc bỏ học của học viên.

3.2.1.2 Cách thức trích xuất đặc trưng

❖ Số lượng học viên đã tham gia vào khóa học trước ngày đăng ký:

- Từ file train_log, tính được các latest_date - thời gian hoạt động mới nhất của học viên trong khoá học.
- Sau đó groupby theo course_id và dùng hàm cumsum() để đếm cộng dồn các học viên tham gia khoá học.

```
# Sort theo ngày tăng dần
course_features_train = course_features_train.sort_values(by=['course_id', 'latest_date'])
# Tạo cột đếm số lượng học viên trước thời điểm latest_date của mỗi enroll_id
course_features_train['previous_enroll_num'] = course_features_train.groupby('course_id').cumcount()
course_features_train
```

	enroll_id	latest_date	course_id	truth	previous_enroll_num
33047	96783	2015-09-08	CAU/08112500x/2015_T2	1	0
33096	96906	2015-09-08	CAU/08112500x/2015_T2	1	1
33153	97041	2015-09-08	CAU/08112500x/2015_T2	1	2
33220	97202	2015-09-08	CAU/08112500x/2015_T2	1	3
33259	97313	2015-09-08	CAU/08112500x/2015_T2	1	4
...
145102	432341	2016-03-28	course-v1:ustcX+USTC001+_	0	479
145107	432346	2016-03-28	course-v1:ustcX+USTC001+_	1	480
145155	432449	2016-03-28	course-v1:ustcX+USTC001+_	0	481
145163	432468	2016-03-28	course-v1:ustcX+USTC001+_	1	482
144834	431792	2016-03-29	course-v1:ustcX+USTC001+_	0	483

157943 rows x 5 columns

❖ Số lượng học viên đã bỏ học trước ngày đăng ký:

- Tương tự như tính số lượng học viên đã tham gia, tuy nhiên, sau khi cộng dồn kết hợp với truth để đếm học viên bỏ học. Sau đó, dùng hàm shift(1) để dịch giá trị về trước để xét số lượng học viên đã bỏ học (không tính đến học viên đang xét đến) và dùng fillna(0) để điền khuyết các giá trị đầu tiên sau khi bị dịch giá trị.


```

course_features_train = course_features_train.sort_values(by=['course_id', 'latest_date'])

# Tạo cột đếm số lượng học viên bỏ học trước thời điểm latest_date của mỗi enroll_id
# Sử dụng hàm cumcount() để đếm số lượng truth = 1 trước latest_date
course_features_train['previous_dropout_user_num'] = (
    course_features_train.groupby('course_id')['truth']
    .apply(lambda x: x.cumsum().shift(1).fillna(0))
    .reset_index(level=0, drop=True)
)
course_features_train['previous_dropout_user_num'] = course_features_train['previous_dropout_user_num'].astype(int)
course_features_train

```

	enroll_id	latest_date	course_id	truth	previous_enroll_num	previous_dropout_user_num
	33047	96783	2015-09-08	CAU/08112500x/2015_T2	1	0
	33096	96906	2015-09-08	CAU/08112500x/2015_T2	1	1
	33153	97041	2015-09-08	CAU/08112500x/2015_T2	1	2
	33220	97202	2015-09-08	CAU/08112500x/2015_T2	1	3
	33259	97313	2015-09-08	CAU/08112500x/2015_T2	1	4
...
	145102	432341	2016-03-28	course-v1:ustcX+USTC001+_	0	479
	145107	432346	2016-03-28	course-v1:ustcX+USTC001+_	1	480
	145155	432449	2016-03-28	course-v1:ustcX+USTC001+_	0	481
	145163	432468	2016-03-28	course-v1:ustcX+USTC001+_	1	482
	144834	431792	2016-03-29	course-v1:ustcX+USTC001+_	0	483

157943 rows x 6 columns

❖ Duration:

- Độ dài khoá học được tính bằng cách lấy thời điểm kết thúc trừ đi thời điểm bắt đầu khoá học. Và đã được tính trước trong quá trình tiền xử lý.

```

course_features_train = pd.merge(course_features_train, course_info[['course_id', 'duration']], on='course_id', how='left')
course_features_train

```

	enroll_id	course_id	previous_enroll_num	previous_dropout_user_num	duration
0	96783	CAU/08112500x/2015_T2	0	0	297
1	96906	CAU/08112500x/2015_T2	1	1	297
2	97041	CAU/08112500x/2015_T2	2	2	297
3	97202	CAU/08112500x/2015_T2	3	3	297
4	97313	CAU/08112500x/2015_T2	4	4	297
...
157938	432341	course-v1:ustcX+USTC001+_	479	422	63
157939	432346	course-v1:ustcX+USTC001+_	480	422	63
157940	432449	course-v1:ustcX+USTC001+_	481	423	63
157941	432468	course-v1:ustcX+USTC001+_	482	423	63
157942	431792	course-v1:ustcX+USTC001+_	483	424	63

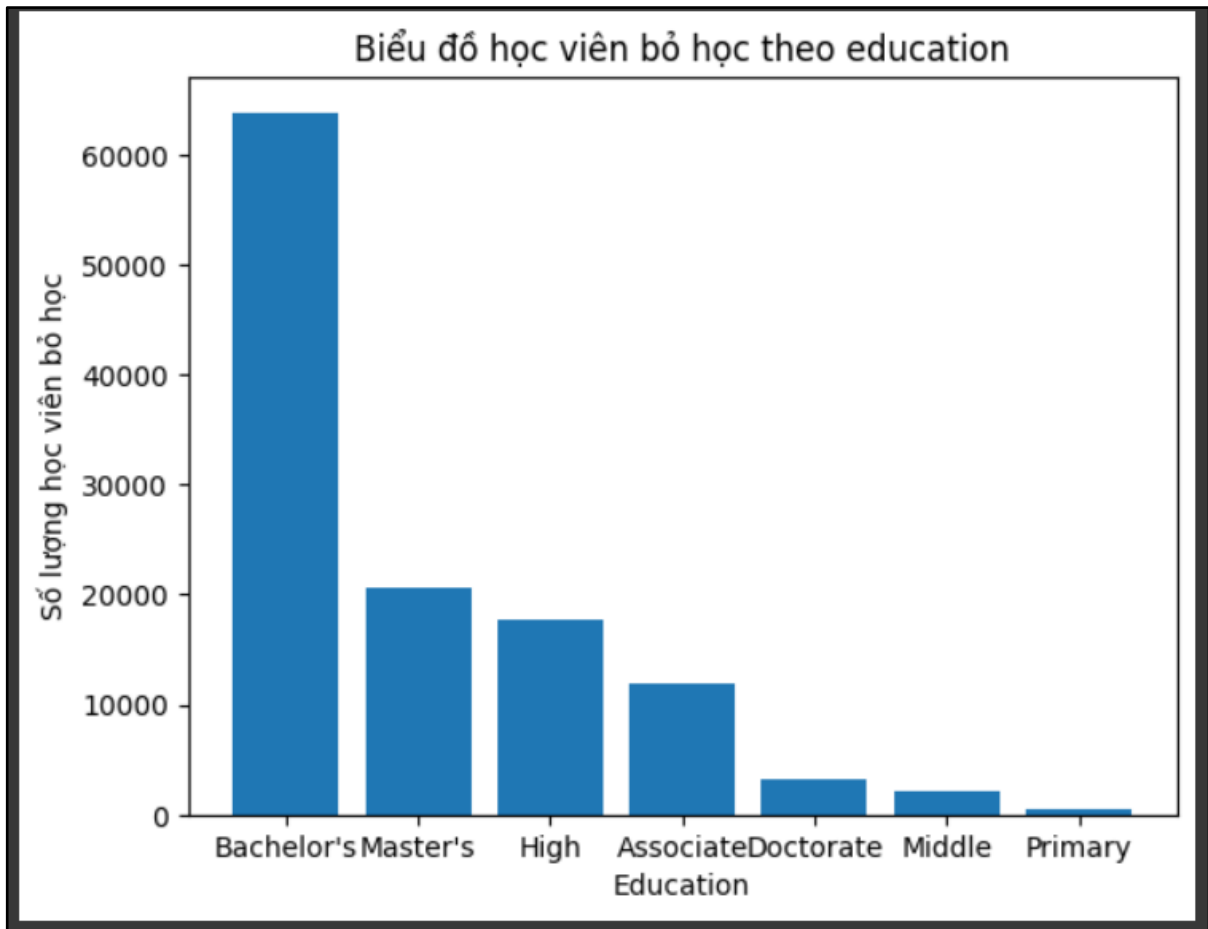
157943 rows x 5 columns

3.2.2 Đặc trưng học viên

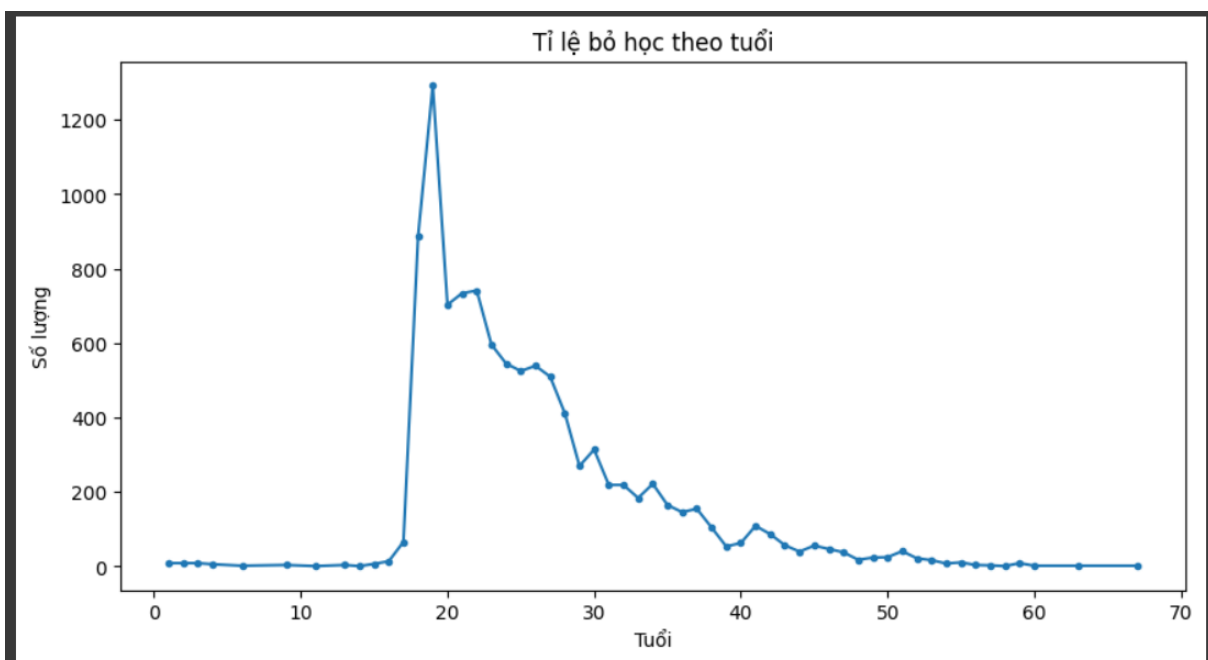
3.2.2.1 Cơ sở trích xuất các đặc trưng

- Sau khi phân tích các đặc trưng của học viên để trích xuất, thì các đặc trưng về age, education và số lượng cho thấy được sự ảnh hưởng đến việc bỏ học ở học

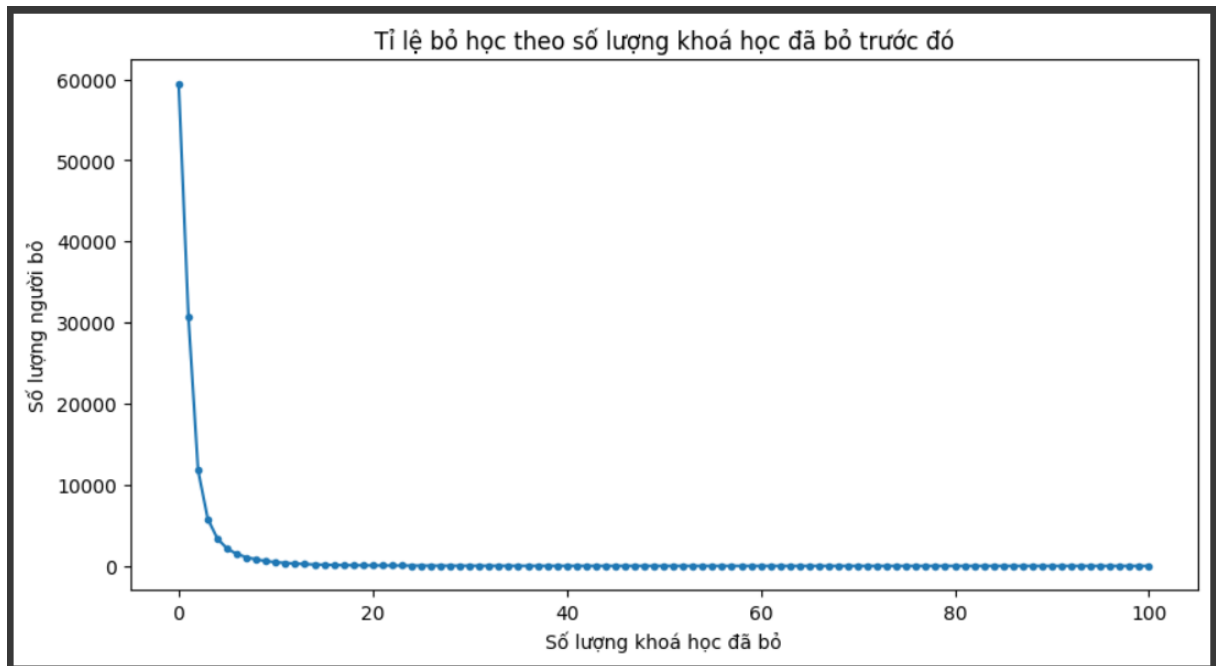
viên. Dưới đây là các biểu đồ thể hiện các đặc trưng của học viên dựa trên các học viên đã bỏ học.



=> Qua biểu đồ trên ta thấy, các trình độ học vấn và tỉ lệ bỏ học, từ đó ta có thể encode để tính toán.



=> Qua biểu đồ trên ta thấy, những học viên trong độ tuổi từ 15-40 có xu hướng bỏ học rất cao.



=> Qua biểu đồ trên ta thấy, xu hướng bỏ học thường có ở những học viên đã bỏ rất ít môn học hoặc chưa từng bỏ môn học nào.

3.2.2.2 Cách thức trích xuất

❖ Education

- Encode 'education' theo thứ tự từ 0 - 6: ["Bachelor's", "Master's", 'High', 'Associate', 'Doctorate', 'Middle', 'Primary']

Encode 'education' theo thứ tự từ 0 - 6: ['Bachelor's', 'Master's', 'High', 'Associate', 'Doctorate', 'Middle', 'Primary']

```
education_map = {
    "Bachelor's": 0,
    "Master's": 1,
    "High": 2,
    "Associate": 3,
    "Doctorate": 4,
    "Middle": 5,
    "Primary": 6
}

user_features_train = pd.merge(user_features_train, user_info, left_on='username', right_on='user_id', how='left')
user_features_train['education'] = user_features_train['education'].map(education_map)
user_features_train
```

	enroll_id	username	truth	Unnamed: 0	user_id	gender	education	birth
0	772	5981	1	613695	5981	0	1	1989.0
1	773	1544995	1	1878114	1544995	1	0	1997.0
2	774	1072798	1	1053475	1072798	0	0	1997.0
3	776	561867	0	7184420	561867	1	3	1981.0
4	777	1368125	1	5885315	1368125	0	0	1997.0
...
157938	466774	2588048	1	6084549	2588048	0	5	1999.0
157939	466776	2736225	1	4014797	2736225	0	1	1997.0
157940	466781	2830711	1	314440	2830711	1	1	1997.0
157941	466782	2680742	1	452751	2680742	1	2	1997.0
157942	466786	2659552	0	4796926	2659552	1	0	1997.0

157943 rows x 8 columns

❖ Age

- Từ file train_log, tính được các latest_date - thời gian hoạt động mới nhất của học viên trong khoá học.
- Sau đó lấy latest_date trừ cho birth của học viên để thu được độ tuổi học viên ngay tại thời điểm tham gia khoá học.

```

user_features_train = pd.merge(user_features_train, latest_train_log, on='enroll_id', how='inner')

def tinh_tuoi(row):
    row = pd.to_datetime(row)
    tuoi = row.dt.year
    return tuoi

user_features_train['age'] = tinh_tuoi(user_features_train['latest_date']) - user_features_train['birth']
user_features_train = user_features_train.drop('birth', axis=1)
user_features_train

```

	enroll_id	truth	user_id	gender	education	latest_date	age
0	772	1	5981	0	1	2015-09-27	26.0
1	773	1	1544995	1	0	2015-10-19	18.0
2	774	1	1072798	0	0	2015-10-29	18.0
3	776	0	561867	1	3	2015-10-25	34.0
4	777	1	1368125	0	0	2015-10-05	18.0
...
157938	466774	1	2588048	0	5	2016-03-27	17.0
157939	466776	1	2736225	0	1	2016-03-20	19.0
157940	466781	1	2830711	1	1	2016-04-02	19.0
157941	466782	1	2680742	1	2	2016-03-06	19.0
157942	466786	0	2659552	1	0	2016-04-01	19.0

157943 rows x 7 columns

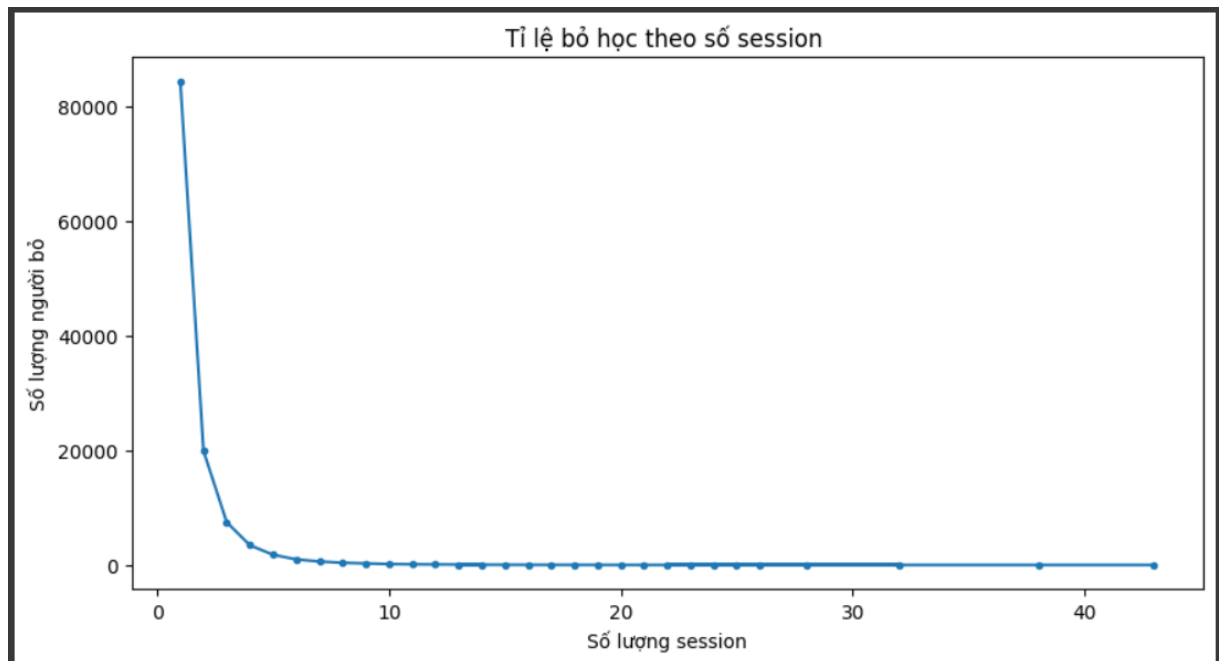
❖ Số khoá học đã bỏ trước đó

- Lấy ra cột latest_date - thời gian hoạt động mới nhất của học viên trong khóa học và sắp xếp tăng dần.
- Trong file train: Merge với cột truth để lấy giá trị học viên bỏ học hay không, sau đó dùng cumsum để cộng dồn các khóa học sinh viên đã bỏ học. Cuối cùng xét các giá trị có truth = 1 và lùi đi một giá trị để xét đến tổng các khóa học đã bỏ học trước đó(không tính đến khóa học hiện tại).
- Trong file test: Để bảo toàn dữ liệu, nhóm trích các khóa học mà học viên đã bỏ theo thời gian phù hợp từ file train. Còn các học viên mới chưa từng ghi nhận bỏ học thì gán giá trị bằng 0.

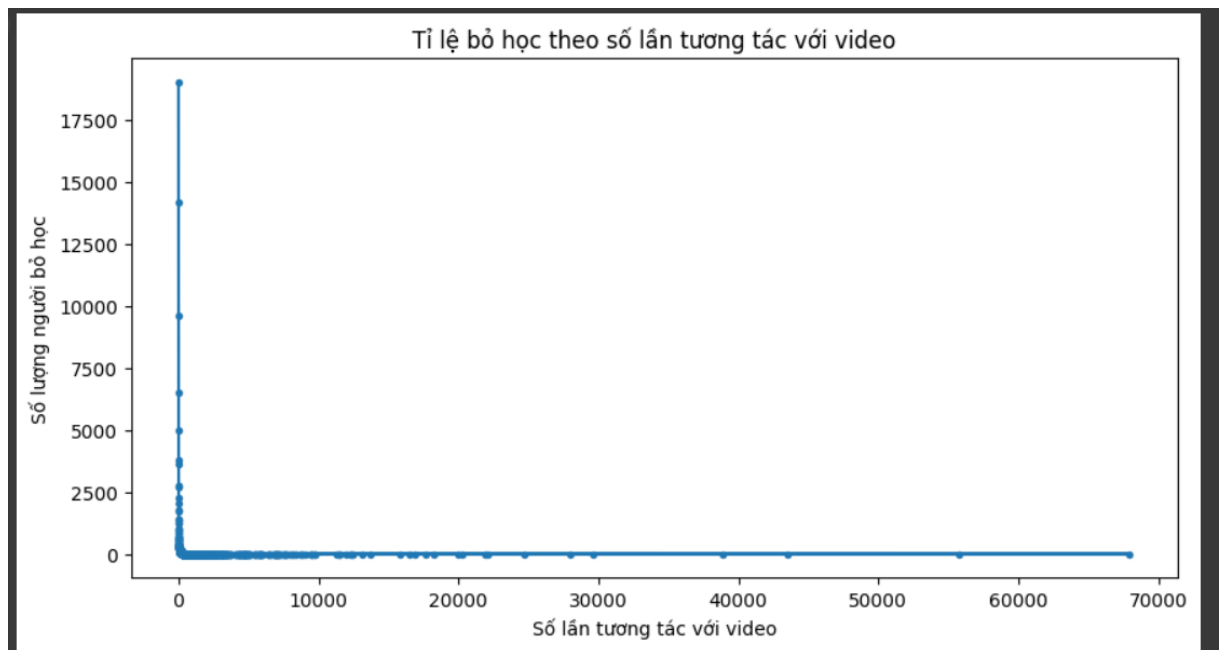
3.2.3 Đặc trưng hoạt động

3.2.3.1 Cơ sở trích xuất đặc trưng

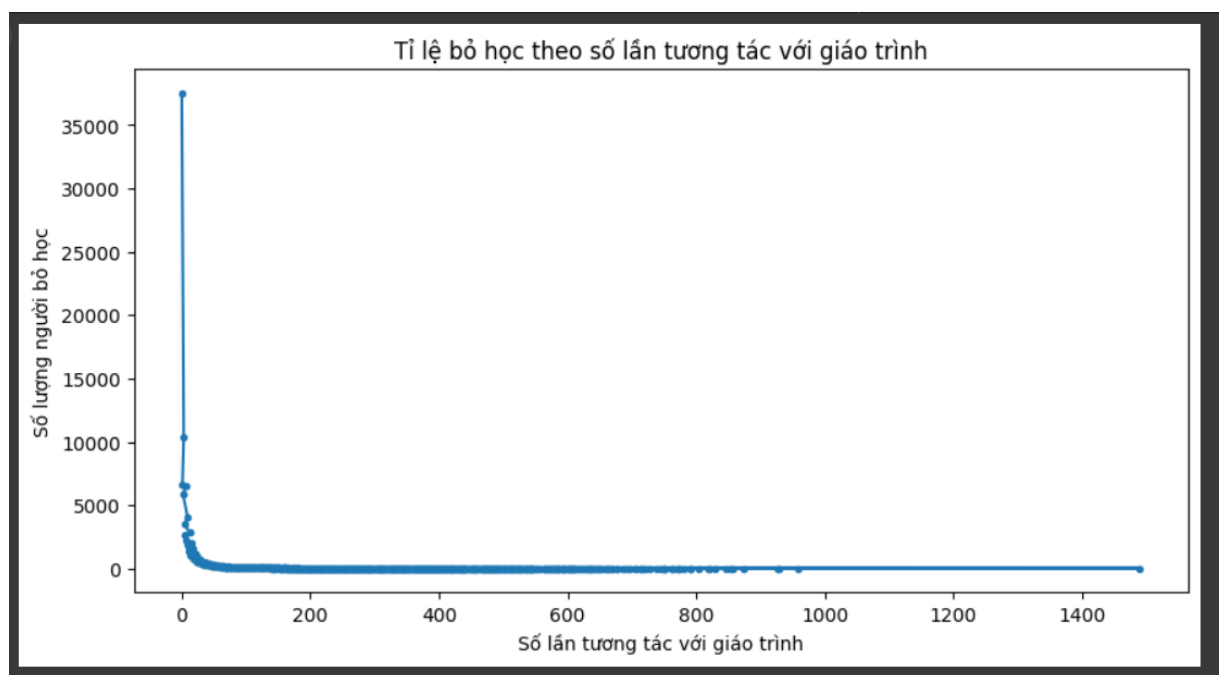
- Tổng số phần của khóa học mà học viên đã tham gia, tổng số lần tương tác với video của khóa học, tổng số lần tương tác với giáo trình (courseware) của khóa học. Nhóm đã thực hiện tính toán tỉ lệ giữa bỏ học dựa vào tổng số của session, video, courseware. Được thể hiện qua các biểu đồ sau:



=> Qua biểu đồ trên ta thấy, những học viên càng ít phiên hoạt động thì tỉ lệ bỏ học càng cao



=> Tương tự như trên, đối với biểu đồ này thì những học viên ít tương tác với video thì tỉ lệ bỏ càng cao



=> Cuối cùng với hành động tương tác với giáo trình cũng tương tự, học viên ít tương tác với giáo trình thì tỉ lệ bỏ học càng cao

3.2.3.2 Cách thức trích xuất

❖ **session_num**

```
[ ] # Đếm số lượng session_id
    session_num_train = train_log.groupby('enroll_id')['session_id'].nunique().reset_index()
    session_num_train.columns = ['enroll_id', 'session_num']
    session_num_train
```

Nhóm dữ liệu trong train_log dựa trên cột enroll_id. Tiếp theo, đếm số lượng giá trị duy nhất trong cột session_id cho từng nhóm.

❖ video_num

```
#Lọc ra các action có từ 'video'
video_actions_train = train_log[train_log['action'].str.contains('video')]
# Lấy ra số lần tương tác video theo từng enroll_id
video_num_train = video_actions_train.groupby('enroll_id').size().reset_index(name='video_num')

# Merge lại với enroll_id của train_log để lấy ra dc cái user không có action video và set giá trị bằng 0
enroll_ids_train = train_log[['enroll_id']].drop_duplicates()
video_num_train = pd.merge(enroll_ids_train, video_num_train, on='enroll_id', how='left').fillna(0)

video_num_train['video_num'] = video_num_train['video_num'].astype(int)
print(video_num_train)
```

- Lọc ra các dòng trong train_log mà cột action chứa từ “video”.
- Nhóm các dòng theo enroll_id và đếm số dòng (tương đương số lần tương tác video) cho từng người dùng
- Lấy danh sách duy nhất các enroll_id từ train_log.
- Kết hợp video_num_train với danh sách tất cả các enroll_id (enroll_ids_train) để đảm bảo mọi enroll_id đều có mặt trong kết quả. Gán giá trị 0 cho video_num của các enroll_id không có hành động liên quan đến video.

❖ courseware_num

```
#Lọc ra các action có từ 'courseware'
courseware_actions_train = train_log[train_log['action'].str.contains('courseware')]
# Lấy ra số lượng coi courseware theo từng enroll_id
courseware_num_train = courseware_actions_train.groupby('enroll_id').size().reset_index(name='courseware_num')

# Merge lại với enroll_id của train_log để lấy ra dc cái user không có action courseware và set giá trị bằng 0
courseware_num_train = pd.merge(enroll_ids_train, courseware_num_train, on='enroll_id', how='left').fillna(0)
courseware_num_train['courseware_num'] = courseware_num_train['courseware_num'].astype(int)

print(courseware_num_train)
```

- Lọc ra các dòng trong train_log mà cột action chứa từ ‘courseware’.
- Nhóm các dòng theo enroll_id và đếm số dòng (tương đương số lần tương tác với giáo trình) cho từng người dùng

- Lấy danh sách duy nhất các enroll_id từ train_log.
- Kết hợp courseware_num_train với danh sách tất cả các enroll_id (enroll_ids_train) để đảm bảo mọi enroll_id đều có mặt trong kết quả. Gán giá trị 0 cho courseware_num_train của các enroll_id không có hành động liên quan đến video.

3.3 Huấn luyện mô hình và đánh giá

Nhóm đã tiến hành thực nghiệm với các mô hình học máy có giám sát phổ biến như Decision Tree, Random Forest, K-Nearest Neighbors (KNN), Naive Bayes, Logistic Regression, Gradient Boosting, XGBoost và LightGBM. Các mô hình này được sử dụng rộng rãi cho các bài toán phân loại. Decision Tree dựa trên việc phân nhánh theo các thuộc tính, Random Forest kết hợp nhiều cây để giảm overfitting, còn KNN dự đoán dựa trên khoảng cách giữa các điểm dữ liệu. Naive Bayes sử dụng lý thuyết Bayes và giả định các thuộc tính độc lập điều kiện, Logistic Regression là một mô hình tuyến tính cho các bài toán phân loại, trong khi Gradient Boosting, XGBoost và LightGBM là các phương pháp ensemble learning theo kiểu boosting, giúp tăng cường độ chính xác thông qua việc kết hợp nhiều mô hình yếu.

Ngoài ra, nhóm tiến hành thực nghiệm với một mô hình học sâu Tabnet Classifier. Đây là một mô hình mạng nơ-ron học sâu có giám sát được tối ưu hóa đặc biệt để làm việc với dữ liệu dạng bảng.

Các mô hình được đánh giá dựa trên các độ đo: Precision, Recall, F1-Score, Accuracy. Dưới đây là kết quả của các mô hình sau khi dự đoán với tập test

	Model	Precision_0	Recall_0	F1_0	Precision_1	Recall_1	F1_1	Accuracy
0	Decision Tree	0.462323	0.426173	0.443513	0.821264	0.841765	0.831388	0.741193
1	Random Forest	0.796844	0.419215	0.549396	0.838947	0.965878	0.897949	0.833587
2	K-Nearest Neighbors	0.268633	0.167857	0.206612	0.762748	0.854100	0.805843	0.688031
3	Naive Bayes	0.794013	0.273637	0.407009	0.808228	0.977337	0.884774	0.807043
4	Logistic Regression	0.785765	0.408350	0.537414	0.836225	0.964456	0.895775	0.829879
5	Gradient Boosting	0.785419	0.452420	0.574129	0.846023	0.960539	0.899651	0.837575
6	XGBoost	0.767709	0.445218	0.563591	0.843826	0.956992	0.896853	0.833144
7	LightGBM	0.792743	0.462736	0.584368	0.848596	0.961377	0.901473	0.840707
8	Tabnet Classifier	0.790000	0.510000	0.620000	0.860000	0.960000	0.900000	0.850000

Nhận xét:

- Nhận thấy rằng các độ đo ở mô hình học sâu Tabnet Classifier cho kết quả tốt và cân bằng nhất.
- Các mô hình đều cho kết quả tốt.
- Các độ đo Precision và Recall ở nhãn 0 lại thấp hơn rất nhiều so với nhãn 1. Lý do chính: Sự mất cân bằng trong giữa các nhãn trong tập train và tập test.

Nhận xét chung: Tuy chỉ số Accuray và F1-Score cao ở các mô hình, các mô hình lại gặp khó khăn trong việc dự đoán đúng nhãn 0. Lý do là bởi vì có sự mất cân bằng về nhãn trong tập dữ liệu (tỉ lệ nhãn 0-1 là $\frac{1}{3}$). Cần xem xét một số phương pháp cân bằng dữ liệu để có thể huấn luyện mô hình tốt hơn.



3.4 Xây dựng đồ thị mạng

3.4.1 Khái niệm đồ thị mạng

Đồ thị mạng là một cấu trúc đồ thị dùng để biểu diễn mối quan hệ và sự kết nối giữa các phần tử trong một hệ thống, đặc biệt là trong bối cảnh mạng máy tính, mạng giao thông, hoặc các bài toán tối ưu hóa. Đồ thị mạng thường được sử dụng để biểu diễn các vấn đề liên quan đến dòng chảy, đường đi, và kết nối giữa các điểm trong một hệ thống.

Đối với bài toán dự đoán khả năng bỏ học của một học viên đối với một khóa học. Ta sẽ tiến hành xây dựng đồ thị bằng cách xét liên kết giữa các học viên học chung khóa học và đặc điểm của từng học viên.

3.4.2 Xây dựng đồ thị mạng cho bài toán dự đoán

3.4.2.1 Thư viện và công cụ hỗ trợ

- Thư viện `torch_geometric`: là một thư viện mở rộng của PyTorch, được thiết kế để hỗ trợ các bài toán học sâu trên đồ thị và dữ liệu không có cấu trúc (non-Euclidean data). Đây là một công cụ mạnh mẽ và linh hoạt, đặc biệt phù hợp với các bài toán về **Graph Neural Networks (GNNs)**, bao gồm phân loại nút, dự đoán liên kết, và phân loại đồ thị.
- Thư viện `Pandas`: được sử dụng để quản lý và xử lý dữ liệu đầu vào, bao gồm việc đọc dữ liệu từ các nguồn như CSV hoặc Excel, và thực hiện các thao tác trên `DataFrame` để chuẩn bị dữ liệu cho việc xây dựng đồ thị.

3.4.2.2 Tiến hành tạo đồ thị

Ta sẽ dựa vào dữ liệu `train_data`, `test_data`, `train_log` và `test_log` để tiến hành tạo đồ thị nhằm phục vụ cho việc dự đoán và huấn luyện mô hình **Graph Neural Networks (GNNs)**.

	enroll_id	course_id
0	775	course-v1:TsinghuaX+70800232X+2015_T2
1	778	course-v1:TsinghuaX+70800232X+2015_T2
2	784	course-v1:TsinghuaX+70800232X+2015_T2
3	788	course-v1:TsinghuaX+70800232X+2015_T2
4	797	course-v1:TsinghuaX+70800232X+2015_T2
...
135393	466770	course-v1:TsinghuaX+AP000001X+2016_T1
135394	466775	course-v1:TsinghuaX+AP000001X+2016_T1
135395	466777	course-v1:TsinghuaX+AP000001X+2016_T1
135396	466783	course-v1:TsinghuaX+AP000001X+2016_T1
135397	466785	course-v1:TsinghuaX+AP000001X+2016_T1
135398 rows × 2 columns		

Bên trên là dữ liệu `enroll_id` và `course_id` của `log_data`. Để thực hiện việc tạo ra các node và các cạnh của đồ thị ta sẽ dựa vào bảng trên. Các node sẽ ứng với các `enroll_id` và 2 nodes được nối với nhau nếu chúng cùng học chung ít nhất là 1 khóa học.

Sau khi đã có các node và các edge cho đồ thị, ta sẽ tiến hành thêm cho `node_feature` và `node_label` cho mỗi node bằng cách sử dụng dữ liệu của `user_data` (là

bộ dữ liệu sau khi gộp chung train_data và test_data, tuy nhiên thì cột ‘truth’ của test_data trước khi gộp sẽ được gán toàn bộ về -1 nhằm đánh dấu trong quá trình huấn luyện).

	enroll_id	gender	education	age	prev_dropout_num	previous_enroll_num	previous_dropout_user_num	duration	session_num	video_num	courseware_num	truth
0	772	0	1	26.0	1	25	25	103	1	0	0	1
1	773	1	0	18.0	1	561	523	103	1	14	3	1
2	774	0	0	18.0	1	850	711	103	3	42	26	1
3	776	1	3	34.0	1	716	637	103	1	9	6	0
4	777	0	0	18.0	1	223	212	103	1	0	2	1
...
225637	466770	1	0	19.0	1	20	20	120	2	33	9	-1
225638	466775	0	0	19.0	0	8	8	120	2	0	2	-1
225639	466777	0	0	16.0	0	22	22	120	2	6	6	-1
225640	466783	1	0	19.0	0	1	1	120	1	0	0	-1
225641	466785	0	0	19.0	0	37	34	120	2	9	6	-1

225642 rows x 12 columns

Node_feature cho một node sẽ là các giá trị ở các cột ứng với enroll_id của node đó

```
Index(['gender', 'education', 'age', 'prev_dropout_num', 'previous_enroll_num',
      'previous_dropout_user_num', 'duration', 'session_num', 'video_num',
      'courseware_num'],
      dtype='object')
```

Node_label của một node sẽ là giá trị cột ‘truth’ ứng với enroll_id của node đó. Đây cũng là giá trị mà mô hình GNNs sẽ dự đoán (dự đoán nhãn cho một node của một đồ thị).

Lưu ý: Vì vấn đề về phần cứng mà nhóm chỉ có thể xây mạng ở dạng có hướng và không tính đến edge_weight (trọng số của cạnh)

3.4.2.3 Thống kê số liệu của đồ thị

```
Thông tin cơ bản về đồ thị:
Data(x=[225642, 10], edge_index=[2, 60129559], y=[225642])

Số lượng nút: 225642
Số cạnh: 60129559
Danh sách các cạnh:
tensor([[172031, 172031, 172031, ..., 220160, 220160, 220161],
        [172032, 172033, 172034, ..., 220161, 220162, 220162]])

Nhãn của các node (y):
tensor([ 1.,  1.,  1., ..., -1., -1., -1.])
Đồ thị có hướng không? True
```

Với những thống kê cho thấy đây là một đồ thị lớn bao gồm rất nhiều đỉnh và cạnh.

3.4.2.4 Huấn luyện mô hình GNNs (Graph Neural Network):

Mô hình GNNs được xây dựng với sự hỗ trợ của thư viện `torch_geometric.nn` với `input_dim=10` và `hidden_layer=4` và 1 `output_layer` với activation là hàm sigmoid để phục vụ cho bài toán dự đoán nhị phân.

Trong quá trình huấn luyện mô hình, ta sẽ không dùng các node có nhãn là -1 (node test) để tính vào loss trong quá trình tối ưu hóa mô hình.

3.4.2.5 Kết quả dự đoán của mô hình GNNs.

Classification Report:					
	precision	recall	f1-score	support	
0	0.58	0.34	0.43	16383	
1	0.81	0.92	0.86	51316	
accuracy			0.78	67699	
macro avg	0.70	0.63	0.65	67699	
weighted avg	0.76	0.78	0.76	67699	

Kết quả dự đoán của mô hình ta thấy vẫn ở mức khá cao so với các mô hình Baseline

Chương 5. KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

5.1 Kết luận

5.1.1 Ưu điểm

- Thông qua quá trình thực hiện đồ án học hỏi tập được nhiều kiến thức mới về xử lý dữ liệu dạng bảng, khai thác dữ liệu và xây dựng khai thác dữ liệu mạng.
- Thực hiện được bài toán đề ra với hai hình thức là dự đoán qua các mô hình học máy từ các dữ liệu dạng bảng và dự đoán qua thuật toán GNN từ dữ liệu dnagj đồ thị.

5.1.2 Nhược điểm

- Đa số các thành viên chưa có kinh nghiệm, nên tằng trong việc thực hiện các dự án liên quan đến xử lý, khai phá dữ liệu nên gặp nhiều khó khăn trong quá trình tìm hiểu và phát triển đồ án.
- Chưa thực hiện được việc ứng dụng kết quả của việc dự đoán vào một sản phẩm thực tế để thấy được tính ứng dụng của bài toán đã đưa ra.

5.2 Hướng phát triển

- Thực hiện thêm nhiều thuật toán và các phương pháp khác để tăng khả năng dự đoán của bài toán bỏ học của học viên và cho ra kết quả dự đoán chính xác hơn
- Phát triển thành một sản phẩm thực tế để ứng dụng được việc dự đoán khả năng bỏ học của học viên vào các trường hợp thực tế.

TÀI LIỆU THAM KHẢO

1. Nhóm tác giả, “**Hệ thống dự đoán khả năng bỏ học của người dùng đối với khóa học**, Đồ án môn học Khai phá dữ liệu và ứng dụng, Trường Đại học Công nghệ Thông tin - ĐHQG TP HCM, 2024.
2. Feng, W., Tang, J. and Liu, T.X (2019). *Understanding dropouts in moocs*, *Proceedings of the AAAI Conference on Artificial Intelligence*. [Trực tuyến] Địa chỉ: <https://ojs.aaai.org/index.php/AAAI/article/view/3825>. [Truy cập lần cuối: 10 December 2024].
3. S. Kumar, X. Zhang, and J. Leskovec. [Predicting Dynamic Embedding Trajectory in Temporal Interaction Networks](#). ACM SIGKDD, 2019.