

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



ĐỀ TÀI: HỆ THỐNG KHUYẾN NGHỊ KHÓA HỌC  
CHO NỀN TẢNG HỌC TẬP TRỰC TUYẾN

ĐỒ ÁN MÔN HỌC

MÔN HỌC: KHAI THÁC DỮ LIỆU VÀ ỨNG DỤNG (CS313)

Nhóm 4

GVHD

ThS. Nguyễn Anh Thư

TP. HO CHI MINH, 6/2024

## LỜI CẢM ƠN

Chúng tôi xin chân thành cảm ơn ThS. Nguyễn Thị Anh Thư, người đã định hướng, giúp đỡ, trực tiếp hướng dẫn và tận tình chỉ bảo chúng tôi trong suốt quá trình nghiên cứu, xây dựng và hoàn thành đồ án này.

Chúng tôi cũng xin được cảm ơn tới gia đình, những người thân và bạn bè thường xuyên quan tâm, động viên, chia sẻ kinh nghiệm, cung cấp các tài liệu hữu ích trong thời gian học tập, nghiên cứu cũng như trong suốt quá trình thực hiện đồ án.

TP. HCM, ngày 6 tháng 6 năm 2024

## DANH SÁCH THÀNH VIÊN

STT	Họ và tên	MSSV
1	Đoàn Nhật Sang	21522542
2	Trương Văn Khải	21520274
3	Lê Ngô Minh Đức	21520195
4	Phạm Minh Quốc	22540017
5	Lê Yên Nhi	21522427
6	Hoàng Thị Mỹ Hạnh	21522044
7	Hoàng Tiến Đạt	21520696
8	Lê Minh Quang	21522510

## ĐÁNH GIÁ CỦA GIẢNG VIÊN HƯỚNG DẪN

TP. HCM, ngày 6 tháng 6 năm 2024

GVHD

## Nguyễn Anh Thư

## MỤC LỤC

---

1	TỔNG QUAN .....	1
1.1	Giới thiệu .....	1
1.2	Định nghĩa bài toán .....	2
1.3	Ứng dụng .....	2
1.4	Khó khăn và thử thách .....	3
1.5	Mục tiêu và phạm vi thực hiện .....	3
2	CÁC CÔNG TRÌNH NGHIÊN CỨU LIÊN QUAN .....	4
2.1	Bộ dữ liệu sử dụng .....	4
2.1.1	Giới thiệu bộ dữ liệu sử dụng .....	4
2.1.2	Mô tả sơ bộ về bộ dữ liệu .....	6
2.2	Phân tích kết quả khảo sát .....	7
2.3	Hướng phát triển đề tài .....	8
3	CƠ SỞ LÝ THUYẾT .....	9
3.1	Phương pháp tiếp cận gần đây - KGAT: Knowledge Graph Attention Network for Recommendation .....	9
3.1.1	Các kiến thức cần nắm .....	9
3.1.2	Tổng quan về KGAT .....	11
3.1.3	Các kỹ thuật sử dụng của KGAT .....	11
3.2	Phương pháp tiếp cận cổ điển .....	19
3.2.1	Khái niệm khuyến nghị .....	19
3.2.2	Content-based filtering [12] .....	19
3.2.3	Bayesian Personalized Ranking (BPR) .....	21

3.2.4	Factorization Machines – FM .....	24
3.2.5	Neural Factorization Machine – NFM.....	27
4	PHƯƠNG PHÁP ĐỀ XUẤT .....	29
4.1	Mô hình .....	29
4.2	Kiến trúc dữ liệu lớn.....	30
4.3	Ứng dụng web .....	37
4.3.1	Công nghệ được sử dụng .....	38
4.3.2	Thiết kế cơ sở dữ liệu.....	41
4.3.3	Thiết kế giao diện.....	41
5	THỰC NGHIỆM.....	46
5.1	Dữ liệu thực nghiệm.....	46
5.2	Phương pháp tổ chức dữ liệu thực nghiệm.....	46
5.2.1	Data translation .....	46
5.2.2	EDA, làm sạch dữ liệu .....	47
5.2.3	Feature selection .....	62
5.2.4	Data preprocessing .....	62
5.2.5	Data splitting .....	64
5.3	Độ đo đánh giá .....	64
5.4	Kích bản thực nghiệm theo thời gian trên kiến trúc dữ liệu lớn .....	65
5.4.1	Thông số chi tiết của các phương pháp.....	66
5.4.2	Đánh giá kết quả thực nghiệm .....	68
5.5	Trực quan hóa cách biểu diễn tri thức cho từng mô hình thực nghiệm .....	71
5.5.1	Content-based Filtering.....	71

5.5.2	Matrix Factorization – MF.....	72
5.5.3	Factorization Machine và Neural Factorization Machine.....	72
5.5.4	KGAT.....	74
6	KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN .....	76
6.1	Đánh giá các phương pháp.....	76
6.2	Hướng phát triển tiềm năng.....	80
7	TÀI LIỆU THAM KHẢO.....	81

## DANH MỤC TỪ VIẾT TẮT

Từ viết tắt	Từ chuẩn	Diễn giải
AI	Artificial Intelligence	Trí tuệ nhân tạo
DL	Deep Learning	Học sâu
MF	Matrix Factorization	Nhân tố hóa ma trận
FM	Factorization Machine	Máy nhân tố
KGAT	Knowledge Graph Attention	Cơ chế chú ý trên đồ thị tri thức
BPR	Bayesian Personalized Ranking	Hàm loss xếp hạng cá nhân hóa Bayesian
KG	Knowledge Graph	Đồ thị tri thức
CKG	Collaborative Knowledge Graph	Đồ thị tri thức hợp tác
BPRMF	BPR Matrix Factorization	Nhân tố hóa ma trận sử dụng hàm loss BPR
CSDL	Cơ sở dữ liệu	Cơ sở dữ liệu

## **DANH MỤC BẢNG**

---

Bảng 2.1 Bảng thống kê số lượng của từng loại tài nguyên .....	5
Bảng 5.1 Bảng thống kê số lượng của từng thực thể sau khi xử lý dữ liệu .....	63
Bảng 5.2 Bảng thống kê chi tiết từng loại liên kết sau khi thực hiện N-core filtering .....	63
Bảng 5.3 Bảng thống kê 10 kết quả nhóm đã thực nghiệm. Màu đỏ là kết quả tốt nhất. Màu lam là kết quả tốt thứ 2. ....	69

## DANH MỤC HÌNH ẢNH

---

Hình 1.1 Hình minh họa Input-Output của bài toán .....	2
Hình 3.1 Hình minh họa biểu đồ tri thức KG trong thực tế .....	10
Hình 3.2 Hình minh họa kiến trúc tổng quan mô hình khuyến nghị học sâu KGAT .....	11
Hình 3.3 Hình minh họa một CKG biểu diễn liên kết cho các loại thực thể .....	12
Hình 3.4 Hình minh họa cơ chế embeddings (bên trái) và TranR (bên phải) trong KGAT .....	13
Hình 3.5 Hình minh họa cơ chế Attention và cơ chế truyền thông tin giữa các lớp.	15
Hình 3.6 Hình minh họa giai đoạn dự đoán của mô hình KGAT .....	17
Hình 3.7 Hình minh họa ý tưởng chính của kỹ thuật Content-based filtering.....	20
Hình 3.8 Hình minh họa cách thức biểu diễn tri thức trên một utility matrix .....	20
Hình 3.9 Hình minh họa mã giả thuật toán BPR .....	23
Hình 3.10 Hình minh họa cách thức tổ chức bài toán theo BPRMF .....	23
Hình 3.11 Hình minh họa cách diễn giải lại mô hình Matrix Factorization .....	24
Hình 3.12 Hình minh họa cho cơ chế Factorization Machines.....	26
Hình 3.13 NFM (phản linear regression bậc một không hiển thị trên hình để rõ ràng hơn). .....	28
Hình 4.1 Hình minh họa quy trình thực nghiệm bài toán với cách tiếp cận sử dụng mô hình KGAT .....	29
Hình 4.2 Hình minh họa giai đoạn hậu xử lý để lọc ra được top-k khóa học được đề xuất .....	30
Hình 4.3 Hình minh họa kiến trúc lưu trữ và xử lý dữ liệu lớn cho đề tài của nhóm .....	31

Hình 4.4 Hình minh họa dữ liệu cho việc xử lý, phân tích được lưu trữ trên azure data lake gen 2 - 1 .....	32
Hình 4.5 Hình minh họa dữ liệu cho việc xử lý, phân tích được lưu trữ trên azure data lake gen 2 - 2 .....	32
Hình 4.6 Hình minh họa pipeline cho quá trình ingest dữ liệu cập nhật-1 .....	33
Hình 4.7 Hình minh họa pipeline cho quá trình ingest dữ liệu cập nhật-2 .....	33
Hình 4.8 Hình minh họa tạo databricks cluster trên Azure.....	34
Hình 4.9 Hình minh họa tạo các Script xử lý và khai phá dữ liệu trên Azure Databricks .....	34
Hình 4.10 Hình minh họa chạy Script translation dữ liệu trên Azure Databricks ....	34
Hình 4.11 Hình minh họa chạy Script EDA và Preprocessing dữ liệu trên Azure Databricks.....	35
Hình 4.12 Hình minh họa chạy Script Tramsform and Split dữ liệu trên Azure Databricks.....	35
Hình 4.13 Hình minh họa chạy Script Train-Eval Model trên Azure Databricks ...	36
Hình 4.14 Hình minh họa các model đã được đăng ký và lưu trữ trên Azure Machine Learning .....	37
Hình 4.15 Hình minh họa quy trình xây dựng website của nhóm .....	37
Hình 4.16 NextJS .....	38
Hình 4.17 FastAPI.....	39
Hình 4.18 MySQL.....	40
Hình 4.19 Lược đồ cơ sở dữ liệu .....	41
Hình 4.20 Hình minh họa sơ đồ use-case tổng quan của bài toán .....	42
Hình 4.21: Giao diện tìm kiếm ban đầu. ....	43

Hình 4.22 Giao diện tìm kiếm khi tìm kiếm với user id “U_1007443”.....	43
Hình 4.23 Giao diện tìm kiếm khi tìm kiếm với username “Huỳnh Oanh Vũ”.....	44
Hình 4.24 Giao diện thông tin chi tiết của người dùng.....	44
Hình 4.25 Giao diện thông tin chi tiết của khóa học (phần trên) .....	45
Hình 4.26 Giao diện thông tin chi tiết của khóa học (phần dưới).....	45
Hình 5.1 Hình minh họa quy trình xử lý dữ liệu của nhóm .....	46
Hình 5.2 Bảng thống kê mô tả của course.json.....	47
Hình 5.3 Thông kê cơ bản về số từ trong about, name sau khi segmented (len_about_segmented, len_name_segmented) .....	48
Hình 5.4 Biểu đồ cột thể hiện độ dài văn bản ở trường about, name.....	48
Hình 5.5 Bảng thống kê mô tả của course-field.json .....	49
Hình 5.6 Histogram thể hiện số lượng fields trong mỗi khóa học (bên trái) và bảng thống kê mô tả tương ứng (bên phải) .....	49
Hình 5.7 Biểu đồ cột thể hiện số lượng khóa học của các field (bên trái) và bảng thống kê mô tả tương ứng (bên phải). .....	50
Hình 5.8 Bảng thống kê mô tả của concept.json.....	50
Hình 5.9 Histogram thể hiện số lượng concept của mỗi khóa học (bên trái) và bảng thống kê mô tả tương ứng (bên phải) .....	51
Hình 5.10 Histogram thể hiện số lượng khóa học của mỗi concept (bên trái) và bảng thống kê mô tả tương ứng (bên phải). .....	51
Hình 5.11 Bảng thống kê mô tả của school.json.....	52
Hình 5.12 Histogram thể hiện số lượng khóa học của mỗi trường (bên trái) và bảng thống kê mô tả tương ứng (bên phải) .....	52
Hình 5.13 Bảng thống kê mô tả cho số lượng trường học của mỗi khóa học.....	53

Hình 5.14 Bảng thống kê mô tả của teacher.json.....	53
Hình 5.15 Histogram thể hiện số lượng teachers của mỗi khóa học (bên trái) và bảng thống kê mô tả tương ứng (bên phải). .....	54
Hình 5.16 Histogram thể hiện số lượng khóa học của mỗi teacher (bên trái) và bảng thống kê mô tả tương ứng (bên phải). .....	55
Hình 5.17 Bảng thống kê mô tả số lượng giáo viên của một tổ chức.....	55
Hình 5.18 Bảng thống kê mô tả của user.json .....	56
Hình 5.19 Value counts của trường gender của User. ....	57
Hình 5.20 Biểu đồ thể hiện số lượng User ứng với từng giới tính.....	57
Hình 5.21 Biểu đồ thể hiện tổng số lượng khóa học đăng ký của mỗi nhóm giới tính .....	58
Hình 5.22 Histogram thể hiện số lượng videos của mỗi khóa học(trái) và bảng thống kê mô tả tương ứng (phải). ....	59
Hình 5.23 Bảng thống kê mô tả của video.json .....	59
Hình 5.24 Thực hiện kiểm định phương sai ANOVA để xem số lượng khóa học đăng ký có bị phụ thuộc vào nhóm giới tính hay không.....	60
Hình 5.25 Một phần của tập luật kết hợp .....	61
Hình 5.26 Các khóa học thường xuất hiện cùng với khóa học 746997 .....	61
Hình 5.27 Sơ đồ phân rã của quy trình tiền xử lý dữ liệu .....	63
Hình 5.28 Reall@K của 10 thực nghiệm .....	69
Hình 5.29 NDCG@K của 10 thực nghiệm .....	70
Hình 5.30 Hình minh họa cách thức biểu diễn một ma trận Đặc trưng khóa học theo phương pháp Content-based filtering.....	72
Hình 5.31 Hình minh họa cách thức biểu diễn tri thức của MF theo dạng đồ thị (bên trái) và dạng ma trận (bên phải) .....	72

Hình 5.32 Hình minh họa cách thức tổ chức dữ liệu dưới dạng ma trận theo FM ...	73
Hình 5.33 Hình minh họa cách thức tổ chức dữ liệu dưới dạng đồ thị theo FM .....	74
Hình 5.34 Hình minh họa cách tổ chức dữ liệu thành một CKG.....	75

# 1 TỔNG QUAN

---

## 1.1 Giới thiệu

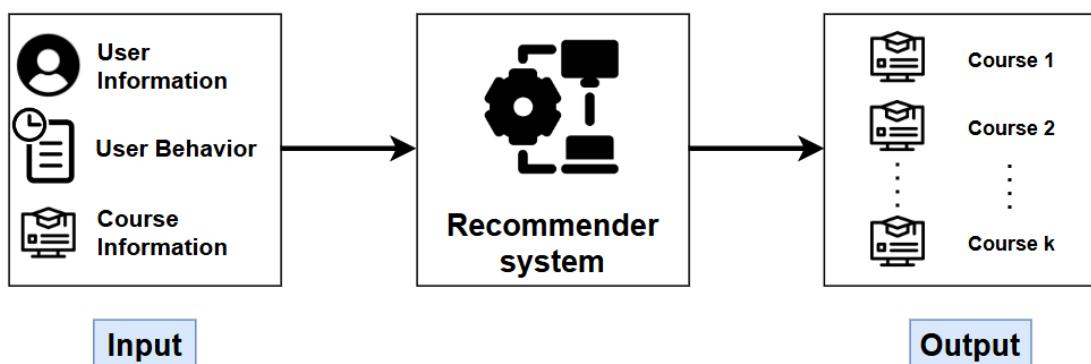
Khai phá dữ liệu và đặc biệt là dữ liệu lớn đang là lĩnh vực được các nhà khoa nghiên cứu quan tâm trong những năm gần đây. Ứng dụng của loại bài toán này khá đa dạng và phong phú và được thực hiện trong nhiều lĩnh vực khác nhau như: kinh doanh, giáo dục, y tế, tài chính, ngân hàng, ... Đặc biệt trong những năm gần đây, khai phá dữ liệu hay đặc biệt là khai phá dữ liệu lớn trong lĩnh vực giáo dục đang là đối tượng đang rất được quan tâm nghiên cứu vì tính thiết thực của chúng. Đối với việc giáo dục trực tuyến hiện tại, người dùng hay người học sẽ phải có sự chủ động và tự giác cao, vì có rất nhiều môn học thuộc rất nhiều các nhóm ngành học khác nhau trong danh sách đào tạo. Người dùng sẽ phải phân bổ các môn học các môn học theo các nhóm chuyên ngành cho từng thời điểm khác nhau để bổ sung hay tự cải thiện vốn kiến thức chuyên ngành cần có của mình. Trên thực tế, các nền tảng học tập trực tuyến không có bất kỳ sự ràng buộc trực tiếp nào về mặt thời gian, điểm số nên thường xảy ra các trường hợp như: các môn học không được hoàn thành đúng hạn theo thời gian dự tính hoặc không bao giờ được hoàn thành vì người dùng đã bỏ ngang vì chán nản trong quá trình học tập. Chính vì vậy, công tác cố vấn trên các nền tảng học tập trực tuyến được đặt ra là một công việc quan trọng trong hình thức học tập theo kiểu mới này. Đây cũng là bài toán được sinh ra trong lĩnh vực khai phá dữ liệu trong khi có số lượng dữ liệu lớn về người học cũng như hành vi học tập của người học trong quá trình tham gia các nền tảng học tập trực tuyến nhằm trợ giúp cho việc cải thiện hiệu suất học tập hay gợi ý các môn học thuộc đúng lĩnh vực được người dùng quan tâm.

Chính vì những lý do trên, nhóm chúng tôi chọn đề tài: “**Hệ thống khuyến nghị khóa học cho các nền tảng học tập trực tuyến**”. Nhóm chúng tôi hi vọng đồ án sẽ mang tính đóng góp thiết thực vào việc giải quyết các vấn đề mang tính cấp bách và cần thiết trong việc giáo dục trên các nền tảng học tập trực tuyến hiện tại.

## 1.2 Định nghĩa bài toán

Trong các nền tảng học tập trực tuyến, người học thường gặp khó khăn trong việc lựa chọn các khóa học. Từ đó một nhu cầu được đặt ra về một hệ thống khuyến nghị các khóa học phù hợp nhằm hỗ trợ cho việc cá nhân hóa quá trình học tập, bài toán sẽ được mô tả với đầu ra và đầu vào như sau:

- **Input:** Nguồn dữ liệu lớn trong các nền tảng học tập trực tuyến: Thông tin người học, thông tin khóa học, hoạt động học tập của người dùng.
- **Output:** Đề xuất top k ( $k \in N^*$ , trong đề án này chúng tôi chọn  $k = 10$ ) các khóa học phù hợp nhất với người dùng.



Hình 1.1 Hình minh họa Input-Output của bài toán

## 1.3 Ứng dụng

Bài toán được chúng tôi giải quyết nhằm tạo nên một hệ thống khuyến nghị các nguồn tài nguyên học tập. Chúng tôi thực hiện nhiệm vụ này trên dữ liệu lớn về nền tảng học tập trực tuyến, cụ thể là dữ liệu MOOCubeX [1] để đánh giá khả năng của tập dữ liệu này và thảo luận về cách nó có thể được sử dụng để tiến hành các nghiên cứu liên quan hoặc mở rộng trên các dữ liệu về nền tảng học tập trực tuyến khác.

Sau đó có thể sử dụng các dữ liệu học tập của các trường Đại học ở Việt Nam để xây dựng các ứng dụng có liên quan đến “Cố vấn học tập thông minh tại các trường đại học” hay “Cố vấn học tập thông minh cho các nền tảng học tập trực tuyến”.

## 1.4 Khó khăn và thử thách

Từ vấn đề khuyến nghị các khóa học cho người dùng trên các nền tảng học tập trực tuyến, một số vấn đề thách thức được đặt ra trước mắt như sau:

1. Dữ liệu: Các nền tảng học tập trực tuyến thường có số lượng người dùng, số lượng tài nguyên học tập rất lớn và được lưu trữ dưới dạng nhiều cấu trúc dữ liệu khác nhau như: có cấu trúc, bán cấu trúc hay phi cấu trúc nên việc xử lý loại dữ liệu này thường rất phức tạp cũng như bao gồm rất nhiều bước xử lý dữ liệu. Ngoài ra, chất lượng dữ liệu cũng là một vấn đề đáng được quan tâm vì chất lượng dữ liệu không được đảm bảo vì có thể đây là nguồn dữ liệu không đầy đủ, thiếu chính xác và có nhiều nhiễu.
2. Khả năng và hiệu suất của hệ thống: Vì đây là bài toán thực tế nên hệ thống phải được đảm bảo là có khả năng mở rộng để xử lý số lượng người dùng lớn và sử dụng dữ liệu một cách hiệu quả. Ngoài ra, còn phải đáp ứng yêu cầu về mặt hiệu suất thời gian thực để đảm bảo hệ thống vừa chính xác vừa xử lý với tốc độ cao nhằm phù hợp với yêu cầu sử dụng của người dùng.
3. Tính bảo mật: Vì yếu tố và các chính sách của các nền tảng học tập trực tuyến, nên phải đảm bảo về yêu cầu an toàn về thông tin của người dùng. Vì vậy, hệ thống phải đảm bảo rằng dữ liệu về thông tin cá nhân của người dùng phải được đảm bảo và không bị lạm dụng.

## 1.5 Mục tiêu và phạm vi thực hiện

**Mục tiêu:** Xây dựng hệ thống khuyến nghị khóa học dựa trên dữ liệu MOOCCubeX. Nhóm sẽ sử dụng một số phương pháp khuyến nghị đã có nhằm tìm ra phương pháp tốt nhất có thể trên dữ liệu này. Sau đó xây dựng một ứng dụng web để demo tính năng khuyến nghị mà nhóm phát triển.

**Phạm vi nghiên cứu:** Người dùng, khóa học và thông tin về hành vi học tập của người dùng trong MOOCCubeX.

## 2 CÁC CÔNG TRÌNH NGHIÊN CỨU LIÊN QUAN

### 2.1 Bộ dữ liệu sử dụng

#### 2.1.1 Giới thiệu bộ dữ liệu sử dụng

Sau quá trình nghiên cứu và tìm hiểu cũng như được sự đề xuất của giảng viên, nhóm chúng tôi quyết định chọn bộ dữ liệu MOOCCubeX.

Bộ dữ liệu được thu thập từ nền tảng XuetangX [2] - Đây là một trong những đối tác của edX. Tuy hệ thống ra mắt vào tháng 10 năm 2013 nhưng đến ngày 31 tháng 5 năm 2021 đã cung cấp hơn 6.000 khóa học, bao gồm các khóa từ Đại học Thanh Hoa, Đại học Bắc Kinh và các khóa học của edX từ MIT, Stanford, UC Berkeley, ... thu hút 4.500.000 người dùng đăng ký. XuetangX cung cấp đa dạng tài nguyên học tập, cho phép người dùng tự do ghi danh vào các khóa học và tham gia vào quá trình học đầy đủ bao gồm học qua video, làm bài tập và tham gia thảo luận. Các dữ liệu này có mối liên hệ chặt chẽ và được quản lý tốt, nên thường được sử dụng làm cơ sở lý tưởng cho MOOCCubeX. Sau đây là bảng thống kê số lượng chi tiết của từng loại tài nguyên trong MOOCCubeX:

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

Tài nguyên Khóa học – Giảng viên	<b>97,192</b> quan hệ
Tài nguyên phản hồi bình luận	<b>331,011</b> phản hồi
Tài nguyên User – Xiaomu	<b>108,351</b> quan hệ
Tài nguyên Course – Comment	<b>10,181,950</b> quan hệ
Tài nguyên User – Comment	<b>8,422,134</b> quan hệ
Tài nguyên User – Reply	<b>331011</b> quan hệ
Tài nguyên Comment - Reply	<b>370,493</b> quan hệ
Tài nguyên Concepts	<b>637,572</b> concepts
Tài nguyên Other	<b>210,349</b> mẫu
Tài nguyên Concept - Other	<b>379,926</b> quan hệ
Tài nguyên Concept - Paper	<b>5,410,752</b> quan hệ
Tài nguyên Concept-Problem	<b>33,180</b> quan hệ
Tài nguyên Concept-Video	<b>624,683</b> quan hệ
Tài nguyên Concept-Comment	<b>31,074</b> quan hệ
Tài nguyên CS	<b>492,102</b> mẫu
Tài nguyên Math	<b>331,202</b> mẫu
Tài nguyên Psy	<b>757,771</b> mẫu

Bảng 2.1 Bảng thống kê số lượng của từng loại tài nguyên

## **2.1.2 Mô tả sơ bộ về bộ dữ liệu**

Bộ dữ liệu gồm hai phần chính: Tài nguyên khóa học (Course Resource) và Hành vi học sinh (Student Behavior)

### **2.1.2.1 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 3.781 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 (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.

### **2.1.2.2 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 [2], 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.

## 2.2 Phân tích kết quả khảo sát

Vì đây là bài toán mang thiên hướng khuyến nghị, nhóm chúng tôi tập trung vào nghiên cứu tìm hiểu và nghiên cứu các bài toán khuyến nghị đã được công bố và đưa vào thực tiễn trước đó, bao gồm:

1. **KGAT: Knowledge Graph Attention Network for Recommendation** [3]: một phương pháp dựa trên GNN (Mạng Nơ-ron Đồ thị) sử dụng đồ thị tri thức nền (Knowledge Base) để cải thiện khuyến nghị. Phương pháp này được tái tạo dựa trên mạng co-occurrence network.
2. **Amazon.com Recommendations: Item-to-Item Collaborative Filtering** [4]: của Amazon năm 2003 giới thiệu một phương pháp cải tiến cho hệ thống gợi ý sản phẩm, được gọi là lọc cộng tác dựa trên mặt hàng (Item-to-Item Collaborative Filtering).
3. **BPR: Bayesian Personalized Ranking from Implicit Feedback** [5]: một kỹ thuật học máy mới để giải quyết vấn đề xếp hạng cá nhân hóa từ phản hồi ngầm của người dùng (implicit feedback). Phản hồi ngầm không bao gồm xếp hạng rõ ràng mà là các hành vi của người dùng như xem, mua hoặc nhập vào các sản phẩm.
4. **Fast Context-aware Recommendations with Factorization Machines** [6]: trình bày một phương pháp tiên tiến để cung cấp các khuyến nghị dựa trên ngữ cảnh nhanh chóng, sử dụng Máy phân tích nhân tố (Factorization Machines - FMs).
5. **Neural Factorization Machines for Sparse Predictive Analytics** [7]: giới thiệu một mô hình học máy tiên tiến kết hợp giữa Máy phân tích nhân tố (Factorization Machines - FMs) và Mạng nơ-ron (Neural Networks) để xử lý các bài toán phân tích dự đoán trên dữ liệu thưa

Song, sau quá trình đọc và tìm hiểu từng bài toán, chúng tôi nhận thấy được điểm nổi bật của bài báo **KGAT: Knowledge Graph Attention Network for**

**Recommendation** [3] về cả mặt hiệu suất và cách thức tổ chức dữ liệu, nên đây cũng chính là hướng nghiên cứu chính trong đồ án của chúng tôi.

### 2.3 Hướng phát triển đề tài

Do những hạn chế về mặt thời gian thực hiện cũng như tài nguyên của nhóm sử dụng, vì vậy trong tương lai, chúng tôi mong muốn sẽ được phát triển đề tài theo các hướng:

- Nghiên cứu thêm các phương pháp khai phá dữ liệu và tri thức: tận dụng được nhiều hơn nữa các công cụ, các nền tảng Điện Toán Đám Mây để hoàn thiện một quy trình xử lý dữ liệu thống nhất.
- Phân tích các phương pháp sâu hơn về các khai phá dữ liệu và xử lý dữ liệu để chọn được phương án tối ưu nhất cho các bài toán khuyến nghị đang được đề xuất.
- Thu thập và sử dụng các bộ dữ liệu lớn khác để tăng hiệu suất và độ chính xác dự đoán.
- Xây dựng hoàn thiện hơn một hệ thống đầy đủ các chức năng và kiến trúc như một hệ thống mang tính thực tiễn, giúp ích và nâng cao hiệu suất cá nhân hóa quá trình học tập của người dùng.

### 3 CƠ SỞ LÝ THUYẾT

#### 3.1 Phương pháp tiếp cận gần đây - KGAT: Knowledge Graph Attention Network for Recommendation

Bài báo đề xuất một phương pháp để cải thiện độ chính xác, đa dạng và khả năng giải thích của hệ thống gợi ý, cần phải xem xét thêm thông tin bổ sung thay vì chỉ dựa vào tương tác người dùng-sản phẩm. Thường thì các phương pháp truyền thống như học máy không đủ để nắm bắt các mối quan hệ phức tạp giữa các mục, từ đó cho ra kết quả dự đoán kết quả không được cao. Trong nghiên cứu này, nhóm tác giả sử dụng đồ thị tri thức (Knowledge Graph) để liên kết các mục với thuộc tính của chúng, cho rằng các quan hệ bậc cao là chìa khóa cho việc gợi ý hiệu quả. Tổng quan phương pháp này được gọi là Knowledge Graph Attention Network for Recommendation (KGAT) [3], mô hình hóa rõ ràng các kết nối bậc cao và sử dụng cơ chế chú ý để đánh giá tầm quan trọng của các liên kết. Kết quả thực nghiệm cho thấy KGAT vượt trội hơn các phương pháp tiên tiến hiện có và cung cấp khả năng giải thích tốt hơn.

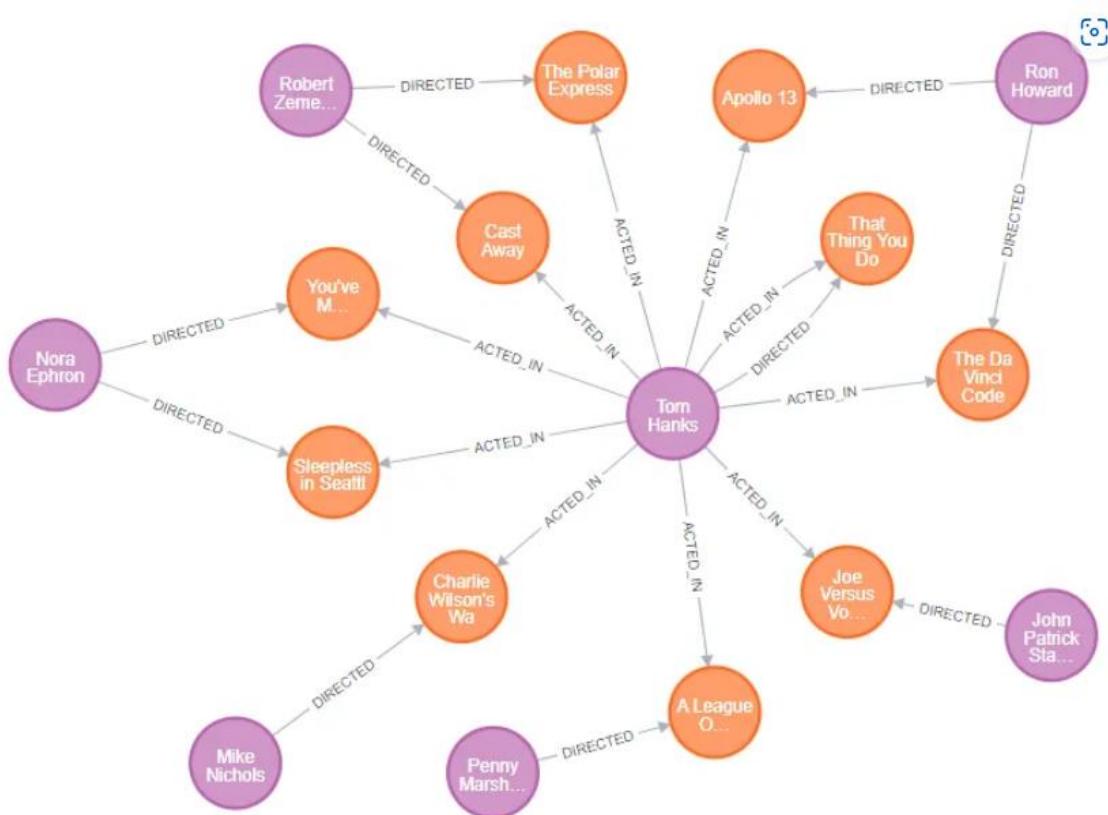
KGAT là một trong những hệ thống đề xuất tập trung nhiều hơn về những thông tin ẩn về người dùng. Bằng cách kết hợp nhúng nhiều trường thông tin ẩn vào mô hình, nhóm tác giả đã tạo được một mô hình không chỉ hiểu về những thông tin rõ ràng đã có của người dùng mà còn các tương tác ẩn giữa người dùng với các đối tượng đó.

##### 3.1.1 Các kiến thức cần nắm

1. Bài báo KGAT đề xuất một phương pháp xử lý dữ liệu tốt hơn bằng cách tận dụng trường thông tin bổ sung (Item side information) để tăng hiệu suất của mô hình.
2. Giải quyết bài toán đề xuất với cách tiếp cận là tạo ra một mô hình có mối quan hệ bậc cao rõ ràng giữa các trường thông tin và toàn diện với cách xử lý của một mạng thần kinh đồ thị (Graph Neural Network [8] [9] – GNN).
3. Tiến hành mở rộng và so sánh với các phương pháp kinh điển trước đó để chứng minh được hiệu suất rõ ràng của mô hình.

Về tổng quát, có thể thấy KGAT tập trung phát triển và cải thiện các kĩ thuật xử lý dữ liệu dựa trên GNN. Vậy GNN là gì? Cốt lõi của GNN là gì?

Đáp án của những câu hỏi trên nằm ở một câu trả lời duy nhất, đó là đồ thị tri thức (Knowledge Graph - KG). KG có thể được hiểu là một mạng phức tạp liên kết các thực thể - chẳng hạn như giữa người dùng (users), vật phẩm (items) và các đặc trưng (features) của chúng thông qua mối quan hệ phức tạp của chúng. Có thể dễ dàng hình dung thông qua đồ thị sau:



Hình 3.1 Hình minh họa biểu đồ tri thức KG trong thực tế

Thường thì, các thành phần cấu tạo nên một KG sẽ bao gồm:

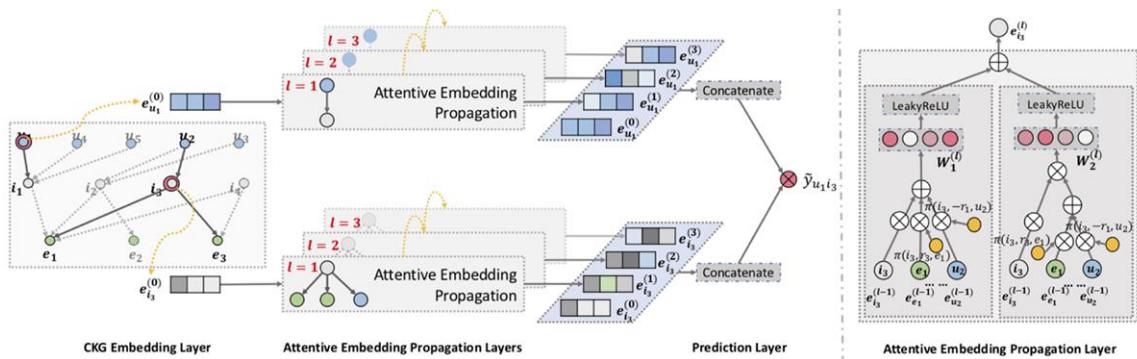
- Nút: đại diện cho các thực thể (ví dụ: người dùng, sản phẩm, danh mục, ...)
- Cạnh: mô tả mối quan hệ của các nút (ví dụ: đã mua, đã xem, .... Và ngược lại)

Có thể thấy đây là một kiến trúc mạng linh hoạt có nhiều lớp thông tin có thể được bổ sung và tích hợp với nhau, điều này cho phép một mạng KG có thể hiểu được toàn diện về ngữ cảnh và môi trường của người dùng nếu được tổ chức tốt.

### 3.1.2 Tổng quan về KGAT

Mô hình KGAT sử dụng thông tin bổ sung để xây dựng một biểu đồ tri thức nhằm nắm bắt mối quan hệ giữa người dùng các mục (items) và các tương tác người dùng (user interaction) với nhau. Mô hình KGAT sử dụng cơ chế học chú ý (Attention Mechanism) với đồ thị nhằm cho phép mô hình có thể hiểu được tầm quan trọng về sự khác nhau giữa các vật phẩm và những đặc trưng của chúng. Điểm khác biệt giữa mô hình KGAT so với các mô hình trước đây là nó được thiết kế với sự thay đổi nhằm làm phong phú thêm về sự hiểu biết của mô hình về hành vi và sở thích của người dùng. Điều này dẫn đến những dự đoán không chỉ dựa trên những thuộc tính của các mục (items) mà còn được định hình dựa trên đặc điểm của từng người dùng, giúp cải thiện đáng kể độ phù hợp và liên quan giữa các mục được mô hình đề xuất.

Đây là kiến trúc của mô hình KGAT:



Hình 3.2 Hình minh họa kiến trúc tổng quan mô hình khuyến nghị học sâu KGAT

### 3.1.3 Các kĩ thuật sử dụng của KGAT

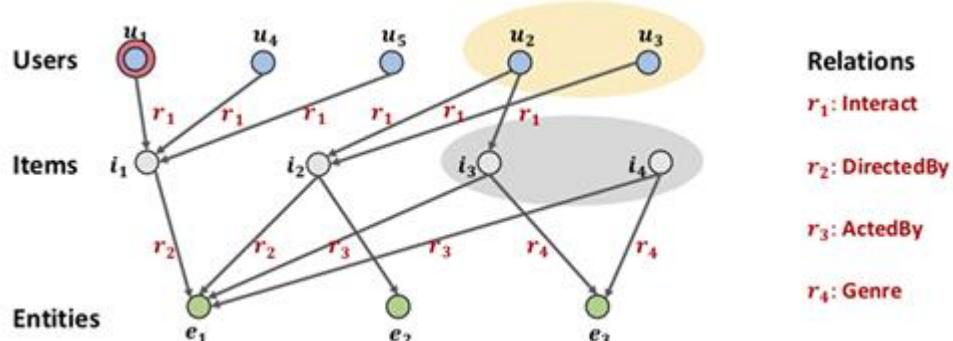
#### 3.1.3.1 Embedding layer.

**Về mặt lý thuyết**, tác giả tạo một collaborative knowledge graph – CKG. CKG là sự kết hợp giữa một KG và một user-item bipartile graph. CKG là có thể được hiểu là một dạng mở rộng của một KG, trong đó thông tin không chỉ đến từ thông tin của

User và Item mà còn đến từ các thông tin bổ trợ khác (ở hình sau là các Entities).

Mục tiêu của một CKG là sử dụng được thông tin phong phú từ nhiều nguồn để cải thiện độ chính xác và tăng khả năng dự đoán của mô hình. Để hiểu hơn một CKG là gì, chúng tôi có làm rõ hơn một số khái niệm như sau:

- KG là một biểu đồ tri thức nó bao gồm các nút (đại diện cho các thực thể như con người, địa điểm, vật phẩm) và các cạnh (đại diện cho mối quan hệ giữa các thực thể).
- User-item bipartite graph là đại diện cho toàn bộ những users, items, tương tác giữa users - items và mối tương tác giữa chúng. Biểu đồ này được xây dựng với các nút đại diện cho cả users và items, cạnh có thể được hiểu như là tương tác lẩn nhau giữa cái nút đó (ví dụ như đánh giá, đã xem, ... ).



Hình 3.3 Hình minh họa một CKG biểu diễn liên kết cho các loại thực thể

Điểm hay của KGAT là khai thác được các quan hệ bậc cao trong CKG, ví dụ như các kết nối dài hạn sau:

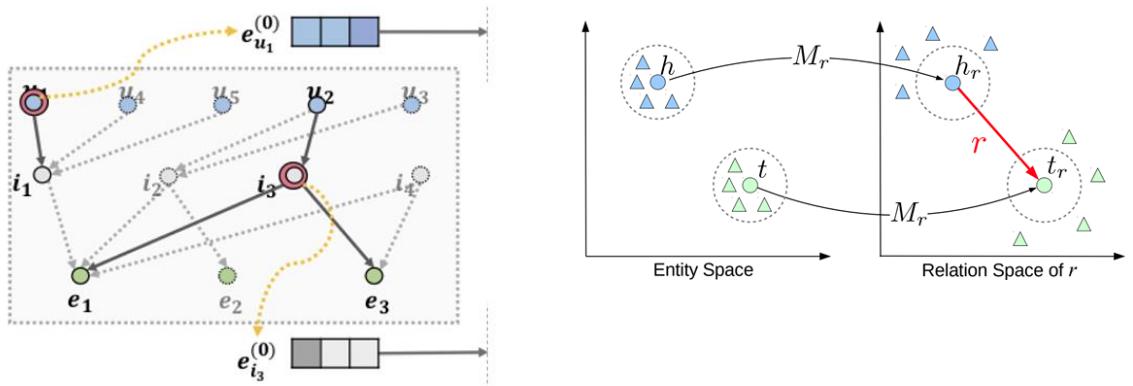
- $u_1 \xrightarrow{r1} i_1 \xrightarrow{-r2} e_1 \xrightarrow{r2} i_2 \xrightarrow{-r1} \{u_2, u_3\}$
- $u_1 \xrightarrow{r1} i_1 \xrightarrow{-r2} e_1 \xrightarrow{r3} \{i_3, i_4\}$

Trong đó,  $r$  là thuận theo chiều của cạnh,  $-r$  là ngược chiều của cạnh. Có thể nhìn thấy trên biểu đồ, những mối quan hệ này lần lượt đại diện cho những con đường dẫn đến các vòng tròn màu vàng và màu xám. Tuy nhiên ở đây họ sẽ gặp các thách thức không nhỏ về mặt tính toán do sự tăng cao về số lượng bậc của một nút và liệu mỗi

quan hệ bậc cao có đóng góp nhiều cho một điểm dự đoán, điều này khiến mô hình phải cân nhắc để chọn lọc chúng.

**Về cách thức thực hiện**, embedding layer tham số hóa các thực thể và quan hệ dưới dạng các vectors trong khi vẫn bảo toàn cấu trúc đồ thị.

Trong bài báo, nhóm tác giả sử dụng TransR [10] để tham số hóa các thực thể và mối quan hệ trong CKG thành các biểu diễn vector, xem xét sự biểu diễn của chúng với kết nối trực tiếp của mỗi bộ ba phần tử (h, r, t):



Hình 3.4 Hình minh họa cơ chế embeddings (bên trái) và TranR (bên phải) trong KGAT

Trong hình trên, hình bên phải thể hiện cách thức nhúng các nút dưới dạng vector và hình bên trái thể hiện được cơ chế bảo tồn cấu trúc của các cặp bộ ba (h, r, t). Toàn bộ quá trình trên được gọi là một quá trình Knowledge Graph Embedding. Đây là một phương pháp hiệu quả để biểu diễn các thực thể và các quan hệ dưới dạng vector, trong khi vẫn giữ được cấu trúc của một KG thông thường. Theo TranR, các embeddings của thực thể và quan hệ sẽ được học bằng cách tối ưu nguyên tắc chuyển đổi  $e_h^r + e_r \approx e_t^r$  nếu bộ ba (h, r, t) tồn tại trong đồ thị. Trong đó,  $e_h, e_t \in R^d, e_r \in R^k$  lần lượt là các embedding của h, r, t; và  $e_h^r, e_t^r \in R^k$  lần lượt là các embedding được chiếu vào không gian quan hệ r của  $e_h, e_t$ . Vì vậy, với một bộ ba (h, r, t), điểm số (điểm khả năng) của chúng được xác định bằng công thức:

$$g(h, r, t) = \|W_r e_h + e_r - W_r e_t\|_2^2$$

Trong đó:  $W_r \in R^{dxk}$  là ma trận biến đổi của mối quan hệ r, chiều các thực thể từ không gian thực thể d chiều vào không gian mối quan hệ k chiều. Điểm số càng thấp cho thấy bộ ba có khả năng tồn tại trong đồ thị, và ngược lại.

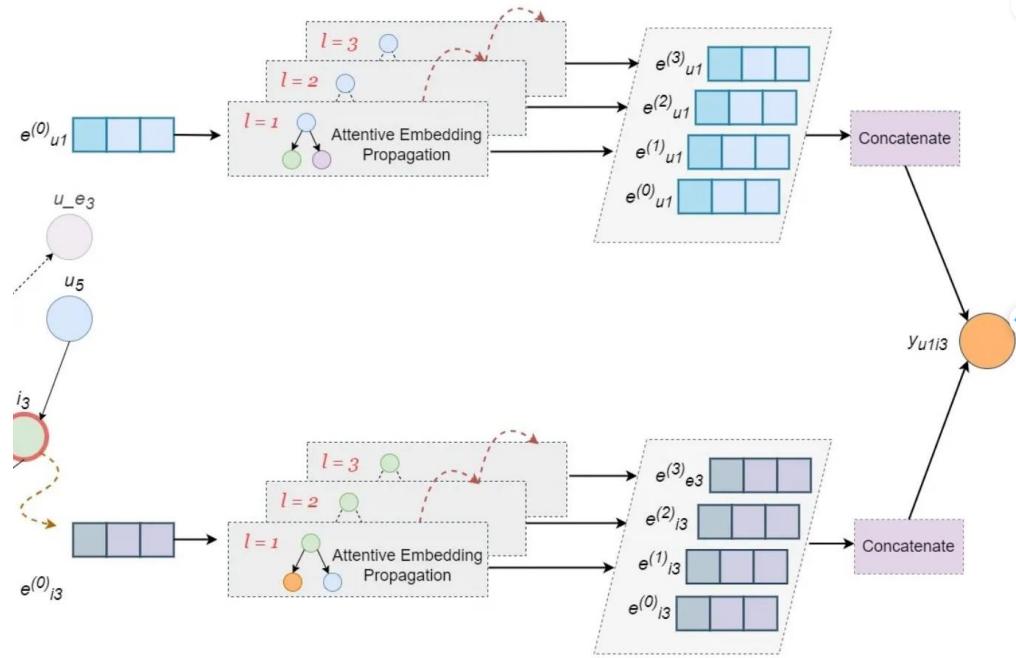
Quá trình huấn luyện của TransR cân nhắc thứ bậc tương đối giữa các bộ ba tồn tại và không tồn tại trong CKG, và khuyến khích sự phân biệt chúng thông qua hàm mất mát xếp hạng giữa các cặp:

$$L_{KG} = \sum_{(h,r,t,t') \in T} -\ln \sigma(g(h,r,t') - g(h,r,t))$$

Trong đó,  $T$  là tập hợp của các  $(h, r, t, t')$  sao cho  $(h, r, t)$  tồn tại trong CKG, còn  $(h, r, t')$  không tồn tại trong CKG.  $\sigma(\cdot)$  là sigmoid function. Lớp này mô hình các thực thể và quan hệ của các bộ ba, thêm các kết nối trực tiếp vào các vector đại diện, từ đó tăng khả năng đại diện mô hình.

### 3.1.3.2 Attentive Embedding propagation layers

**Về lý thuyết**, đây được xem như giai đoạn biểu diễn mỗi nút dưới dạng một vector bằng cấu trúc bảo toàn cấu trúc của CKG, để quy lan truyền để truyền các lớp nhúng học được từ các nút lân cận để cập nhập biểu diễn cho nó, sử dụng cơ chế học có chú ý (Attention) để học chi tiết từng trọng số cho các nút lân cận trong quá trình lan truyền. Giai đoạn này có thể được hiểu là giai đoạn huấn luyện các kết nối giữa các nút và thông tin liên kết.



*Hình 3.5 Hình minh họa cơ chế Attention và cơ chế truyền thông tin giữa các lớp*

Cách tiếp cận có hệ thống này biến mô hình KGAT thành một hệ thống kết hợp nắm bắt cả các thông tin items trực tiếp của người dùng và bối cảnh rộng hơn của các tùy chọn đó, tạo ra một hệ thống để xuất rõ ràng hơn có khả năng mang được nhiều thông tin để dự đoán hơn.

**Về cách thức thực hiện**, giai đoạn này sẽ được tạo thành từ các lớp đơn lẻ và bao gồm ba thành phần chính:

- **Information Propagation:** Lan truyền thông tin, một thực thể có thể tham gia vào nhiều bộ ba, đóng vai trò là cầu nối giữa hai bộ ba và lan truyền thông tin. Xét  $e_1 \xrightarrow{r^2} i_2 \xrightarrow{-r^1} u_2$  và  $e_2 \xrightarrow{r^3} i_2 \xrightarrow{-r^1} u_2$ , mục  $i_2$  nhận các thuộc tính  $e_1$  và  $e_2$  làm đầu vào để làm phong phú thêm các đặc điểm của chính nó, sau đó đóng góp vào sở thích của người dùng  $u_2$ , điều này có thể được mô phỏng bằng cách lan truyền thông tin từ  $e_1$  đến  $u_2$ .

$$e_{N_h} = \sum_{(h,r,t) \in N_h} \pi(h, r, t) e_t$$

Trong đó:

- $\pi(h, r, t)$  điều khiển hệ số suy giảm trên mỗi lần lan truyền tải trên cạnh  $(h, r, t)$ . Chỉ ra được mức độ thông tin được lan truyền từ  $t$  đến  $h$  phụ thuộc vào quan hệ  $r$ .
- $N_h = \{(h, r, t) | (h, r, t) \in G\}$  là ego network của  $h$ , một đồ thị con mà trong đó  $h$  là thực thể đầu.
- **Knowledge-aware Attention:**  $\pi(h, r, t)$  được cài đặt thông qua cơ chế học chú ý và được tính theo công thức:

$$\pi(h, r, t) = (W_r e_t)^T \tanh((W_r e_h + e_r))$$

Tác giả chọn sử dụng hàm *Tanh* như một hàm kích hoạt phi tuyến. Điều này nhằm để làm cho điểm số học chú ý (attention score) phụ thuộc dựa trên khoảng cách của  $e_h$  và  $e_t$  trong không gian quan hệ  $r$ .

Sau đó, các hệ số sẽ được chuẩn hóa trên tất cả bộ ba kết nối với  $h$  bằng hàm softmax:

$$\pi(h, r, t) = \frac{\exp(\pi(h, r, t))}{\sum_{(h, r', t') \in N_h} \exp(\pi(h, r', t'))}$$

- **Information Aggregation:** Giai đoạn cuối cùng là tổng hợp đại diện thực thể  $e_h$  và đại diện ego network của nó  $e_{N_h}$  để tạo ra đại diện mới của thực thể  $h$ . Trong bài báo có ba cách tổng hợp thông tin, lần lượt là: GCN aggregator [9], GraphSage aggregator [11] và Bi-interaction aggregator [3]. Do Bi-interaction cho kết quả tốt nhất trong paper, nên nhóm sẽ chỉ đề cập đến phương pháp này. Công thức của Bi-interaction aggregator như sau:

$$e_h = \text{LeakyReLU}\left(W_1(e_h + e_{N_h})\right) + \text{LeakyReLU}\left(W_2(e_h \odot e_{N_h})\right)$$

Trong đó:

- $W_1, W_2 \in R^{d' \times d}$ : Các ma trận trọng số có thể huấn luyện.
- $\odot$ : Phép nhân element-wise.

Nhìn chung, tác dụng của embedding propagation layer nằm ở việc khai thác rõ ràng các thông tin kết nối bậc một để liên kết các đại diện người dùng, sản phẩm, và các thực thể tri thức. Để khai thác các kết nối bậc cao, ta có thể chất chòng nhiều

propagation layers (higher-order propagation), tập hợp các thông tin được lan truyền từ các node ở xa. Nói theo toán học, tại bước thứ  $l$ , đại diện của một thực thể sẽ được tính đê quy như sau:

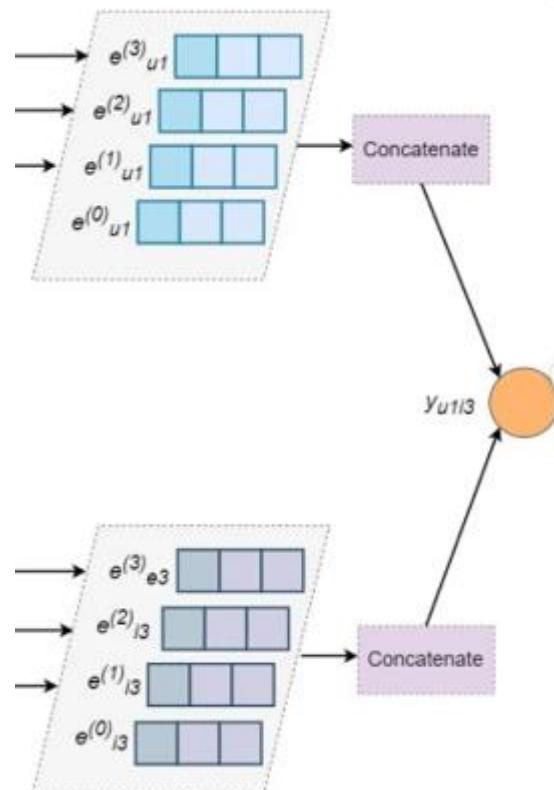
$$e_h^{(l)} = f(e_h^{(l-1)}, e_{N_h}^{(l-1)})$$

Với thông tin lan truyền trong  $l$ -ego network của thực thể  $h$  sẽ được định nghĩa như sau:

$$e_{N_h}^{(l-1)} = \sum_{(h,r,t) \in N_h} \pi(h, r, t) e_t^{(l-1)}$$

Trong đó,  $e_t^{(l-1)}$  là đại diện của thực thể  $t$  được tạo từ bước lan truyền thông tin trước đó.

### 3.1.3.3 Model prediction.



Hình 3.6 Hình minh họa giai đoạn dự đoán của mô hình KGAT

Sau khi thực hiện lan truyền thông tin qua L lớp, ta thu được nhiều đại diện của người dùng node  $u$ , bao gồm:  $\{e_u^{(1)}, \dots, e_u^{(L)}\}$ ; tương tự với sản phẩm node  $i$ , ta cũng thu được  $\{e_i^{(1)}, \dots, e_i^{(L)}\}$ . Tiếp đến, ta sẽ ghép nối embedding ban đầu với các embeddings mới thu được theo công thức sau:

$$e_u^* = e_u^{(0)} || \dots || e_u^{(L)}, e_i^* = e_i^{(0)} || \dots || e_i^{(L)}$$

Trong đó,  $e_u^*, e_i^*$  lần lượt là embedding tổng hợp của người dùng và sản phẩm. Cuối cùng, điểm số phù hợp (matching score) giữa người dùng và sản phẩm sẽ được tính bằng inner product:

$$\hat{y}(u, i) = {e_u^*}^T e_i^*$$

Trong giai đoạn này, để tối ưu, KGAT sử dụng BPR loss [5]. Cụ thể, phương pháp này giả định các sản phẩm đã được sử dụng, thứ chỉ ra sở thích của người dùng, nên được gán giá trị dự đoán cao hơn so với những sản phẩm chưa được tiêu dùng:

$$L_{CF} = \sum_{(u, i, j) \in O} -\ln (\hat{y}(u, i) - \hat{y}(u, j))$$

Trong đó,  $O$  là tập hợp các  $(u, i, j)$  sao cho người dùng  $u$  đã sử dụng sản phẩm  $i$  nhưng chưa sử dụng sản phẩm  $j$ .

### 3.1.3.4 Optimization

Toàn bộ quá trình huấn luyện của mô hình được tối ưu với một hàm loss được tổng hợp từ hai giai đoạn như sau:

$$L_{KGAT} = L_{KG} + L_{CF} + \lambda \|\Theta\|_2^2$$

Trong đó,  $\Theta$  là tập hợp các trọng số của mô hình,  $\lambda \|\Theta\|_2^2$  là L2 regularization term giúp tránh overfitting.

## 3.2 Phương pháp tiếp cận cổ điển

### 3.2.1 Khái niệm khuyến nghị

Có nhiều cách thức khai thác, xử lý, đưa ra kết quả khuyến nghị nhưng về cơ bản chúng được chia thành 3 nhánh chính:

- Khuyến nghị dựa trên lọc cộng tác (Collaborative Filtering)
- Khuyến nghị dựa trên lọc nội dung (Content based filtering)
- Khuyến nghị dựa trên phép lai (Hybrid)

Nhưng dù phân loại theo nhánh nào, chúng cũng đều khai thác 2 loại thực thể chính của dữ liệu và mối quan hệ giữa chúng:

- Item (vật phẩm): đối tượng chính mà người dùng tương tác, ví dụ: phim, video, sản phẩm.
- User (người dùng): đối tượng sử dụng trên hệ thống, đưa ra những mối tương tác đối với vật phẩm.
- Hành vi: tương tác giữa người dùng với vật phẩm. Bao gồm 2 loại là tương tác rõ ràng (đăng ký học, lượt thích, lượt chia sẻ, ...) và tương tác ngầm định (lịch sử thanh toán, lịch sử xem video, ...)

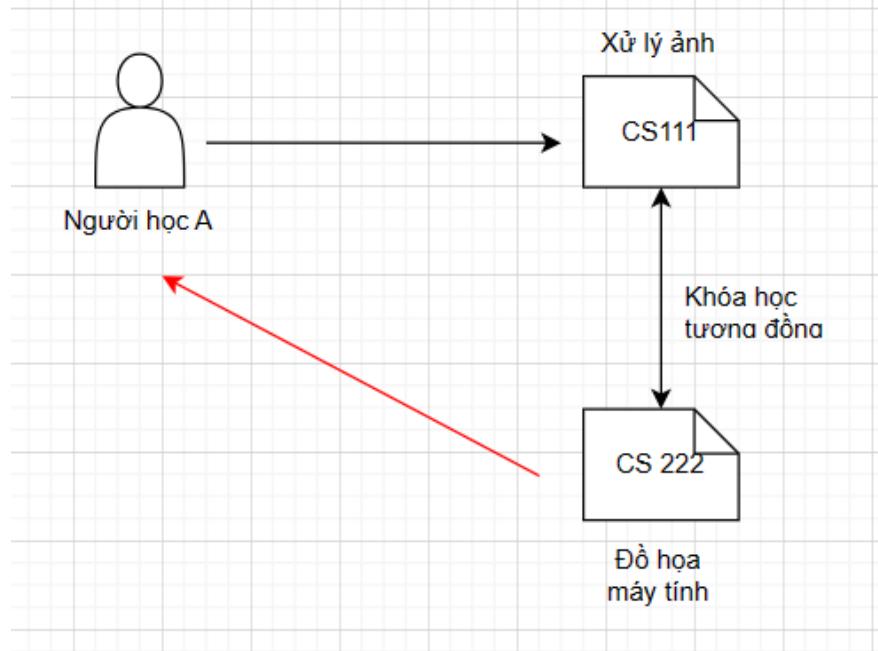
Để hiểu rõ hơn về các khuyến nghị này, chúng ta cùng đi vào cụ thể phương pháp lọc nội dung.

### 3.2.2 Content-based filtering [12]

**Khái niệm:** Dựa vào đặc tính của người dùng để đề xuất cho người dùng các khóa học tương tự.

**Ý tưởng:** Dựa vào các vật phẩm của dựa trên hành vi trước đó của người dùng và nội dung, thuộc tính của những vật phẩm để dự đoán các vật phẩm tương tự với những vật phẩm mà người dùng đã chọn trong quá khứ.

Dưới đây là một ví dụ minh họa:



*Hình 3.7 Hình minh họa ý tưởng chính của kỹ thuật Content-based filtering*

Content-based filtering dựa trên cơ sở chủ yếu là Utility Matrix. Có thể hình dung như sau, mỗi user sẽ có mức độ quan tâm tới từng item khác nhau. Mức độ quan tâm này, nếu đã biết trước, được gán cho một giá trị ứng với mỗi cặp user-item. Giả sử rằng mức độ quan tâm được đo bằng giá trị user rate cho item, ta tạm gọi giá trị này là rating. Tập hợp tất cả các ratings, bao gồm cả những giá trị chưa biết cần được dự đoán, tạo nên một ma trận gọi là utility matrix. Xét ví dụ sau:

	A	B	C	D	E	F
Mưa nửa đêm	5	5	0	0	1	?
Cỏ úa	5	?	?	0	?	?
Vùng lá me bay	?	4	1	?	?	1
Con cò bé bé	1	1	4	4	4	?
Em yêu trường em	1	0	5	?	?	?

*Hình 3.8 Hình minh họa cách thức biểu diễn tri thức trên một utility matrix*

Trong ví dụ này, có 6 users A, B, C, D, E, F và 5 bài hát. Các ô màu xanh thể hiện việc một user đã đánh giá một bài hát với ratings từ 0 (không thích) đến 5 (rất thích). Các ô có dấu ‘?’ màu xám tương ứng với các ô chưa có dữ liệu. Công việc của một Recommendation Systems là dự đoán giá trị tại các ô màu xám này, từ đó đưa ra gợi ý cho người dùng. Recommendation Systems, vì vậy, đôi khi cũng được coi là bài toán Matrix Completion (Hoàn thiện ma trận).

Rõ ràng rằng càng nhiều ô được điền thì độ chính xác của hệ thống sẽ càng được cải thiện. Vì vậy, các hệ thống luôn luôn hỏi người dùng về sự quan tâm của họ tới sản phẩm, và muốn người dùng đánh giá càng nhiều sản phẩm càng tốt. Việc đánh giá các sản phẩm, vì thế, không những giúp các người dùng khác biết được chất lượng sản phẩm mà còn giúp hệ thống biết được sở thích của người dùng, qua đó có chính sách quảng cáo hợp lý.

### 3.2.3 Bayesian Personalized Ranking (BPR)

BPR [5] là một hướng tiếp cận để tối ưu tham số của một mô hình khuyến nghị thông dụng từ thông tin phản hồi tiềm ẩn nhằm giải quyết những khó khăn gặp phải bởi dữ liệu của bài toán này.

Ví dụ: Hiện có một tập người dùng và một danh sách các sản phẩm mà người dùng muốn mua.

Gọi:

- U: tập người dùng:  $U = \{u_1, u_2, u_3, \dots, u_n\}$
- I: tập sản phẩm:  $I = \{i_1, i_2, i_3, \dots, i_n\}$

Trong giả định trên, ta gọi tập  $S \subseteq U \times I$  là tập các tương tác của người dùng. Nhiệm vụ của ta là phải đi tìm những sản phẩm  $i$  mà người dùng  $u$  có thể yêu thích, tức là việc đi tìm một danh sách xếp hạng theo thứ tự các sản phẩm  $i$  để gợi ý cho người dùng.

Mục tiêu của ta là đi tìm xếp hạng các sản phẩm để đề xuất cho người dùng, hình dung đơn giản có thể hiểu thì đây là một bài toán quan hệ, tức là ta đi tìm câu trả lời cho câu hỏi có hay không quan hệ giữa hai đối tượng đang xét. Quan hệ vừa đề cập là  $s >_u \subset I^2$  của tất cả sản phẩm. Lúc này, kí hiệu  $i >_u j$  được hiểu là người dùng  $u$  thích sản phẩm  $i$  hơn  $j$ .

Quan hệ này phải thỏa:

- $\forall i, j \in I: i \neq j \Rightarrow i >_u j \vee j >_u i$ : Toàn phần: Sản phẩm  $i$  và  $j$  luôn nằm ở một trong hai trường hợp: Người dùng thích  $i$  hơn  $j$  hoặc ngược lại.
- $\forall i, j \in I: i >_u j \wedge j >_u i \Rightarrow i = j$ : Phi đối xứng: Nếu người dùng thích  $i$  hơn  $j$  thì không thể xảy ra việc người dùng thích  $j$  hơn  $i$ . Do đó nếu điều trên xảy ra thì ta quy định  $i$  và  $j$  là cùng một sản phẩm  $i = j$ .
- $\forall i, j, k \in I: i >_u j \wedge j >_u k \Rightarrow i >_u k$ : Quan hệ bắc cầu: Nếu người dùng thích  $i$  hơn  $j$  và thích  $j$  hơn  $k$  thì có thể suy ra người dùng thích  $i$  hơn  $k$ .

Lúc này, một định nghĩa của bài toán được phát biểu:

$$I_u^+ := \{i \in I: (u, i) \in S\}$$

$$U_i^+ := \{u \in U: (u, i) \in S\}$$

Ở đây  $I_u^+$  là tập những sản phẩm mà người dùng đã tương tác (mua, thích, ...), tương tự  $U_i^+$  là tập người dùng có tương tác với sản phẩm.

Thay vì tiếp cận theo phương pháp truyền thống, BPR đặt ra một bài toán tiếp cận hoàn toàn mới.

Gọi  $D_s$  là dữ liệu ban đầu:

$$D_s = \{(u, i, j) | i \in I_u^+ \wedge j \in I \setminus I_u^+\}$$

Tập  $D_s$  là tập gồm  $(u, i, j)$ , ngữ nghĩa của cặp bộ ba ở đây là người dùng  $u$  thì giả định là thích  $i$  hơn  $j$  vì người dùng đã tương tác với  $i$  mà không tương tác với  $j$ .

### 3.2.3.1 Thuật toán BPR

Hướng tiếp cận của BPR là sử dụng thuật toán tối ưu SGD (stochastic gradient descent), tổng quát như sau:

```

FUNCTION LearnBPR( $D_S, \Theta$ ):
    Khởi tạo  $\Theta$ 
    While chưa hội tụ:
         $(u, i, j) \leftarrow$  chọn ngẫu nhiên từ  $D_S$ 
         $\Theta \leftarrow \Theta + \alpha \left( \frac{e^{-\hat{x}_{uij}}}{1+e^{-\hat{x}_{uij}}} \cdot \frac{\partial}{\partial \Theta} \hat{x}_{uij} + \lambda_\Theta \cdot \Theta \right)$ 
    Endwhile
    RETURN  $\Theta$ 

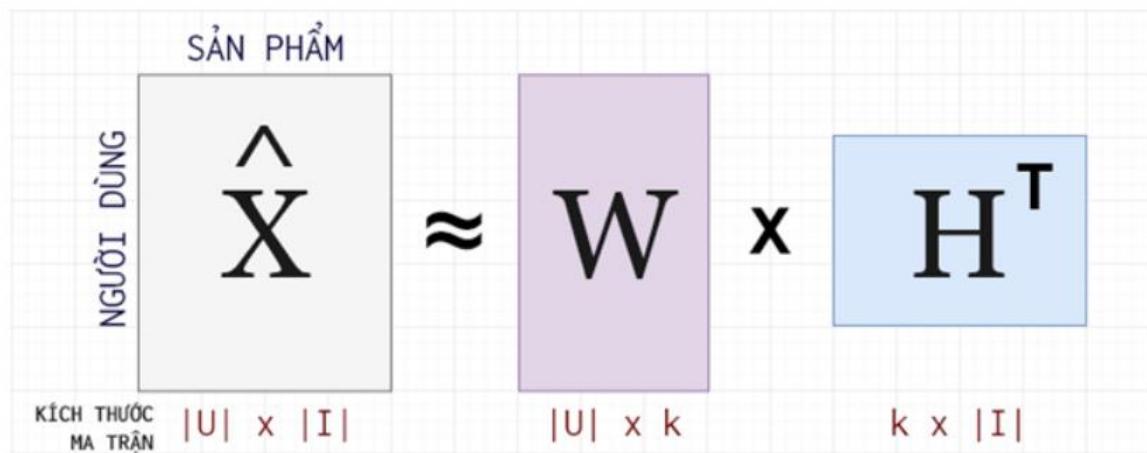
```

Hình 3.9 Hình minh họa mã giả thuật toán BPR

### 3.2.3.2 BPR cho Matrix Factorization – BPRMF

Với Matrix Factorization [5], bài toán dự đoán  $\hat{x}_{ui}$  có thể được xem như là ước lượng ma trận  $\hat{X}$  kích thước  $|U| \times |I|$  được phân tích thành hai ma trận W (kích thước  $|U| \times k$ ) và H (kích thước  $|I| \times k$ ) thỏa:

$$\hat{X} = WH^T$$



Hình 3.10 Hình minh họa cách tổ chức bài toán theo BPRMF

Từ công thức trên, có thể hiểu BPRMF [5] như sau:

- Mỗi người dùng và sản phẩm được mô tả bởi  $k$  đặc trưng.
- $k$  đặc trưng ở đây có thể hiểu như là những nhân tố ở bên dưới thể hiện mối liên hệ giữa người dùng và sản phẩm. Ví dụ: hệ khuyến nghị về phim,  $k$  đặc trưng có thể là: “hài hước”, “viễn tưởng”, “chiến tranh”, “tâm lý”,...

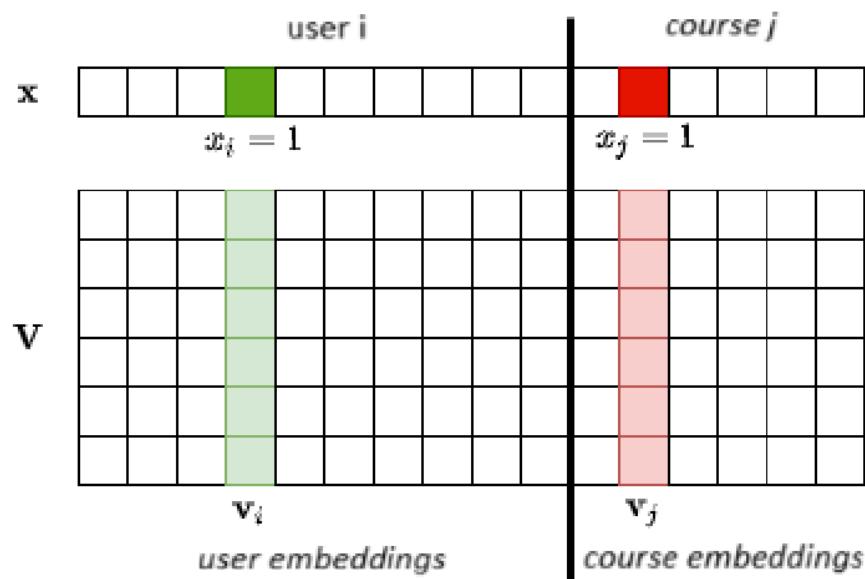
Và mức độ yêu thích của sản phẩm  $i$  đối với người dùng  $u$  được tính bởi:

$$\hat{x}_{ui} = \sum_{f=1}^k w_{uf} h_{if}$$

### 3.2.4 Factorization Machines – FM

Như phần 3.2.3.2, nhược điểm của mô hình BPRMF là không có khả năng mô hình hóa những thông tin bổ trợ giữa người dùng và sản phẩm. Do đó, một phương pháp mở rộng FM [6] đã ra đời. Đây cũng là phương pháp nền móng cho các kĩ thuật DL được ra đời cho bài toán khuyến nghị.

Ta có thể diễn giải lại MF như sau: dữ liệu đầu vào là cặp (người dùng, sản phẩm) được biểu diễn bằng vector  $\mathbf{x} \in R^{1 \times d}$  chỉ có hai phần tử khác 0 bằng tương ứng với chỉ số người dùng  $i$  và sản phẩm  $j$ :



Hình 3.11 Hình minh họa cách diễn giải lại mô hình Matrix Factorization

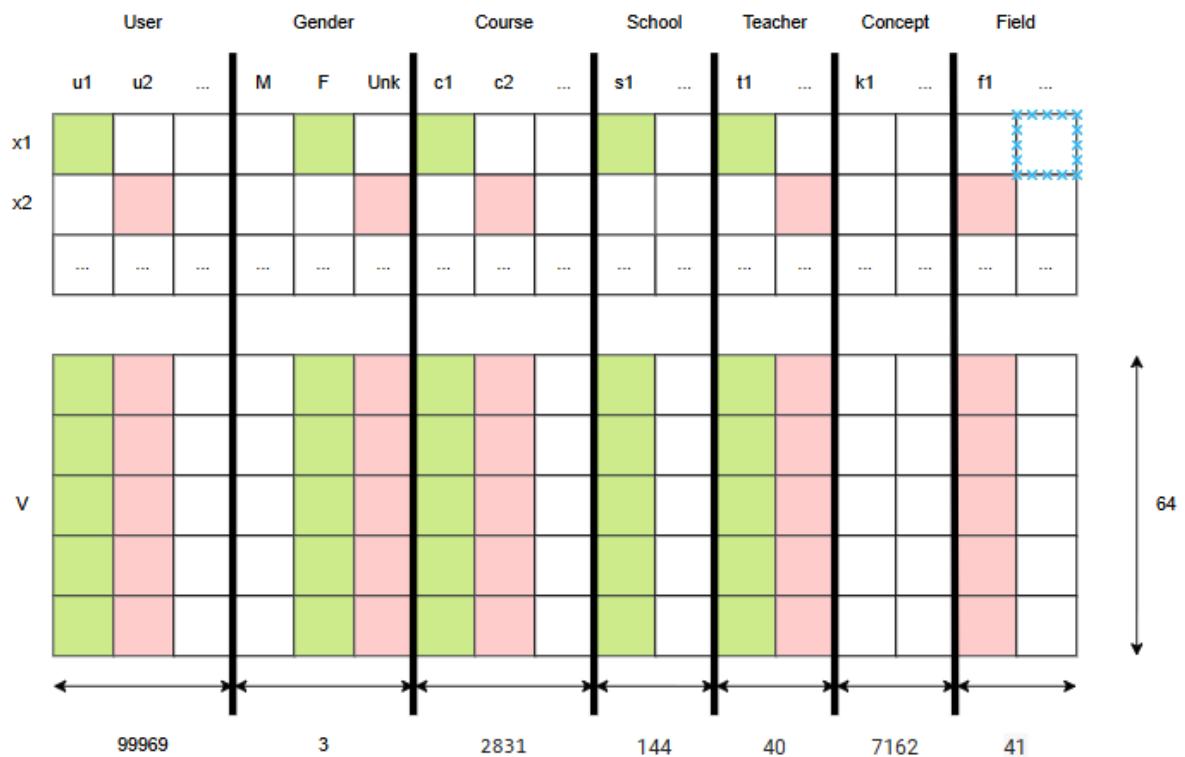
Khi đó, hai ma trận embeddings của người dùng và sản phẩm được nối lại với nhau thành một ma trận  $\mathbf{V} \in R^{k \times d}$ . Đồng thời tạo một ma trận hệ số tự do của người dùng và sản phẩm thành là  $\mathbf{w} \in R^{d \times 1}$ . Khi đó, độ quan tâm của người dùng  $i$  tới sản phẩm  $j$  được viết lại:

$$\hat{y}_{ij} = \mathbf{v}_i^T \mathbf{v}_j + w_i + w_j + w_0$$

Trong đó:

- $w_i$ : hệ số thiên hướng của người  $i$  thể hiện “độ khó” của người dùng.
- $w_j$ : hệ số thiên hướng ứng với độ yêu thích của sản phẩm  $j$ .
- $w_0$ : hệ số thiên hướng chung của các đánh giá trong bộ dữ liệu.

Như đã nói, FM có thể mở rộng ra với các thông tin bổ trợ của người dùng và sản phẩm. Ví dụ như về khuyến nghị cho người dùng một bộ phim, ta có thể xét mức độ ảnh hưởng của các thông tin bổ trợ như: giới tính, tuổi, nghề nghiệp, ... Những thành phần này sẽ được mã hóa thành các vector one-hot hoặc multi-hot vector. Nếu có thêm các dữ liệu dạng số khác, ta có thể thêm vào  $\mathbf{x}$  các thành phần tương ứng. Với mỗi thành phần được thêm vào  $\mathbf{x}$ , ta thêm một cột vector embedding vào  $\mathbf{V}$  như hình bên dưới đây.



Hình 3.12 Hình minh họa cho cơ chế Factorization Machines

Khi đó, độ quan tâm của người dùng có thể được dựng lên như sau:

$$\hat{y}_{ij} = w_0 + \mathbf{x}\mathbf{w} + \sum_{i=1}^d \sum_{j=i+1}^d \mathbf{v}_i^T \mathbf{v}_j x_i x_j$$

Trong đó:

- $w_0$  đóng vai trò như một hệ số bias trong mô hình hồi quy tuyến tính, nó có thể được xem như là một hệ số vô hướng cố định được thêm vào kết quả dự đoán cuối cùng để điều chỉnh sự lệch trung bình.
- $\mathbf{x}\mathbf{w}$ : đây là tích vô hướng vector đặc trưng đầu vào (input feature vector) và một vector trọng số  $\mathbf{w}$  tương ứng với các đặc trưng của  $\mathbf{x}$ .
- $\sum_{i=1}^d \sum_{j=i+1}^d \mathbf{v}_i^T \mathbf{v}_j x_i x_j$ : Đây là thành phần tương tác bậc hai giữa các đặc trưng.
  - $x_i, x_j$  lần lượt là phần tử thứ i, thứ j trong feature vector x.
  - $\mathbf{v}_i^T \mathbf{v}_j$  là tích vô hướng giữa giữa các vector embeddings tương ứng với từng đặc trưng đầu vào  $x_i$  và  $x_j$ .

- $\sum_{i=1}^d \sum_{j=i+1}^d \mathbf{v}_i^T \mathbf{v}_j x_i x_j$  là biểu diễn tổng tất cả các cặp tương tác giữa các đặc trưng có trong tập dữ liệu.

Đây chính là ý tưởng chính của FM. Đồng thời, nhờ vào việc  $x$  thường là một vector rất thưa (rất ít thành phần khác 0), việc huấn luyện và dự đoán trở nên rất nhanh ngay cả khi số lượng người dùng và sản phẩm lớn.

### 3.2.5 Neural Factorization Machine – NFM

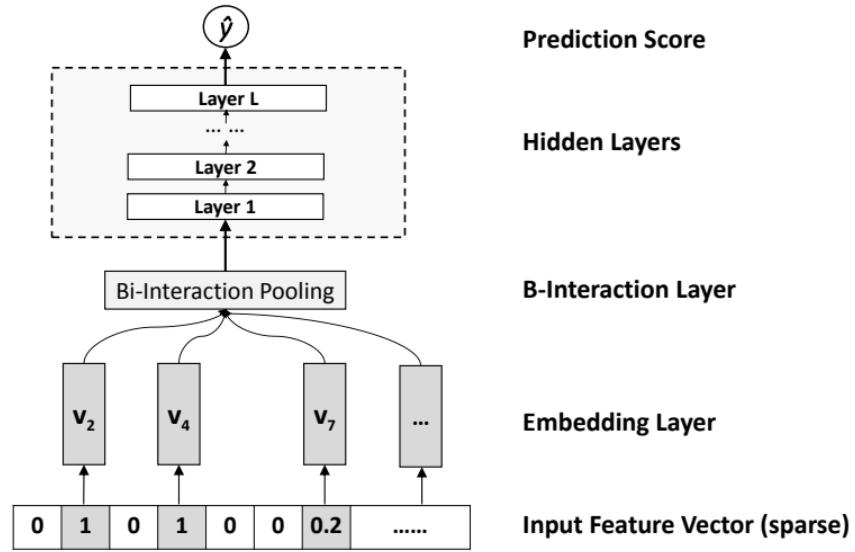
Mặc dù có khả năng mô hình hóa các thông tin bổ trợ của người dùng và sản phẩm, performance của FM vẫn bị hạn chế bởi tính tuyến tính của nó cũng như việc chỉ mô hình các tương tác đặc trưng (ví dụ như bậc 2) theo cặp. Với dữ liệu thực tế có cấu trúc cơ bản phức tạp và phi tuyến, FM sẽ không đủ khả năng biểu diễn. Tuy FM bậc cao hơn [13] đã được đề xuất, chúng vẫn thuộc họ mô hình tuyến tính và được cho là khó ước tính.

NFM [7] ra đời nhằm cải tiến FM bằng cách mô hình hóa các tương tác đặc trưng bậc cao và phi tuyến. Bằng cách sử dụng một phép toán mới trong mô hình neural network – Bilinear Interaction (Bi-Interaction) pooling, NFM được xem như là sự kết hợp của FM với neural network framework. Thông qua việc xếp chồng các lớp phi tuyến trên Bi-interaction pooling layer, NFM đã làm sâu hơn mô hình FM tuyến tính nồng, từ đó mô hình hóa các tương tác đặc trưng phi tuyến và bậc cao một cách hiệu quả, cải thiện performance của FM.

Với một feature vector thưa  $x \in R^n$  làm đầu vào, trong đó  $x_i = 0$  nghĩa là đặc trưng thứ  $i$  không tồn tại trong đối tượng, NFM dự đoán mục tiêu như sau:

$$\hat{y}(x) = w_0 + \mathbf{x}\mathbf{w} + f(x)$$

Trong đó, term đầu tiên và thứ 2 là phần linear regression giống với FM, thứ 3 mô hình bias toàn cục và trọng số của các đặc trưng. Còn term thứ 3  $f(x)$  là thành phần cốt lõi trong NFM để mô hình tương tác đặc trưng.  $f(x)$  chính là một multi-layer feed-forward neural network như Hình 3.13.



Hình 3.13 NFM (phản linear regression bậc một không hiển thị trên hình để rõ ràng hơn).

Do Embedding layer tương tự như FM nên ta sẽ tiếp tục tìm hiểu về Bi-Interaction pooling, Hidden layer, Prediction layer.

**Bi-interaction pooling:** chuyển đổi tập các embedding vectors  $V_x$  thành 1 vector như sau:

$$f_{BI}(V_x) = \sum_{i=1}^n \sum_{j=i+1}^n x_i \mathbf{v}_i \odot x_j \mathbf{v}_j$$

Trong đó,  $\odot$  là ký hiệu của element-wise product của 2 vectors;  $x_i, x_j$  lần lượt là phần tử thứ i và j trong feature vector  $\mathbf{x}$ .  $\mathbf{v}_i, \mathbf{v}_j$  lần lượt là embedding của feature thứ i và j.

**Hidden layer:** Trên Bi-interaction pooling là 1 chồng các lớp fully connected, thứ có khả năng mô hình các tương tác bậc cao giữa các đặc trưng. Mỗi hidden layer sẽ có một non-linear activation function như tanh, sigmoid, ReLU.

**Prediction layer:** Cuối cùng, vector đầu ra của hidden layer cuối sẽ được chuyển đổi thành prediction score:

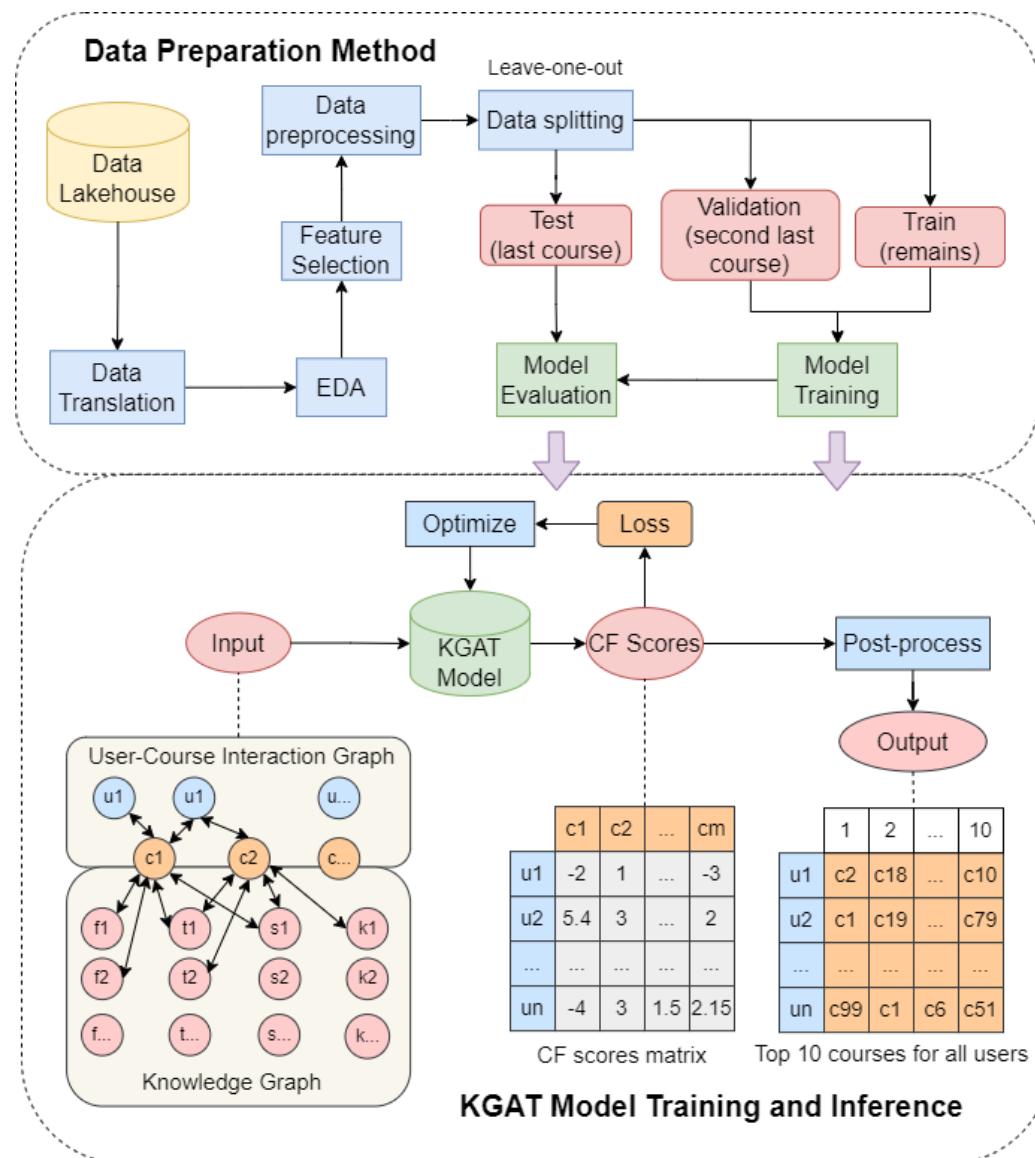
$$f(\mathbf{x}) = \mathbf{h}^T \mathbf{z}_L$$

Trong đó,  $\mathbf{h}$  là weight của prediction layer,  $\mathbf{z}_L$  là output vector của hidden layer cuối.

## 4 PHƯƠNG PHÁP ĐỀ XUẤT

### 4.1 Mô hình

Sau khi chạy thực nghiệm và đánh giá các phương pháp gợi ý trên dữ liệu MOOCCubeX [1], nhóm đã chọn ra được phương pháp tốt nhất là KGAT model (chi tiết kết quả trong mục 5). Vì vậy trong phần này, nhóm sẽ trình bày cách chuyển đổi từ input sang output của mô hình KGAT, còn phương pháp chuẩn bị dữ liệu sẽ được trình bày ở phần 5.

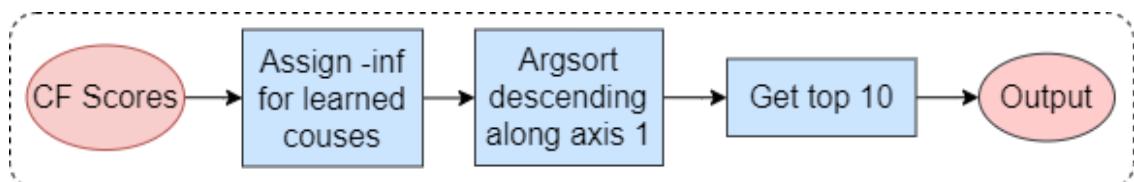


Hình 4.1 Hình minh họa quy trình thực nghiệm bài toán với cách tiếp cận sử dụng mô hình KGAT

Input của mô hình là collaborative knowledge graph chứa 2 đồ thị thành phần như Hình 4.1: user-item bipartite graph và knowledge graph. User-item bipartite graph biểu diễn mối quan hệ người dùng đã đăng ký khóa học nào và mối quan hệ ngược lại, nghĩa là khóa học được đăng ký bởi user nào. Còn knowlege graph biểu diễn mối quan hệ giữa khóa học và các thuộc tính của nó. Nếu xét node xuất phát là khóa học, đồ thị này sẽ có 4 loại quan hệ: khóa học có khái niệm (concept) nào; khóa học thuộc lĩnh vực (field) nào; khóa học được dạy bởi giáo viên (teacher) nào; khóa học được tổ chức bởi trường (school) nào. Và tính thêm chiều ngược lại, ta sẽ có tổng cộng  $4 \times 2 = 8$  loại quan hệ giữa khóa học và các thuộc tính.

Khi nhận được input, model sẽ trả về một ma trận collaborative scores  $cf\_scores \in R^{n_{users} \times n_{courses}}$ , với mỗi phần tử  $cf\_scores[i, j] \in R$  ( $i; j \in N; 0 \leq i < n_{users}; 0 \leq j < n_{courses}$ ) cho biết mức độ phù hợp giữa  $user_i$  và  $course_j$ . Nếu giá trị này càng lớn thì  $user_i$  càng phù hợp với  $course_j$  và ngược lại.

Sau đó, giai đoạn postprocess (Hình 4.2) sẽ gán giá trị âm vô cùng cho  $cf\_scores[i, j]$  nếu  $user_i$  đã đăng ký  $course_j$ ; tiếp đến argsort giảm dần theo chiều axis = 1 (để các khóa học mà  $user_i$  đã đăng ký không bị đe xuất lại) rồi lấy top 10 khóa học có score cao nhất ứng với mỗi user để làm output.

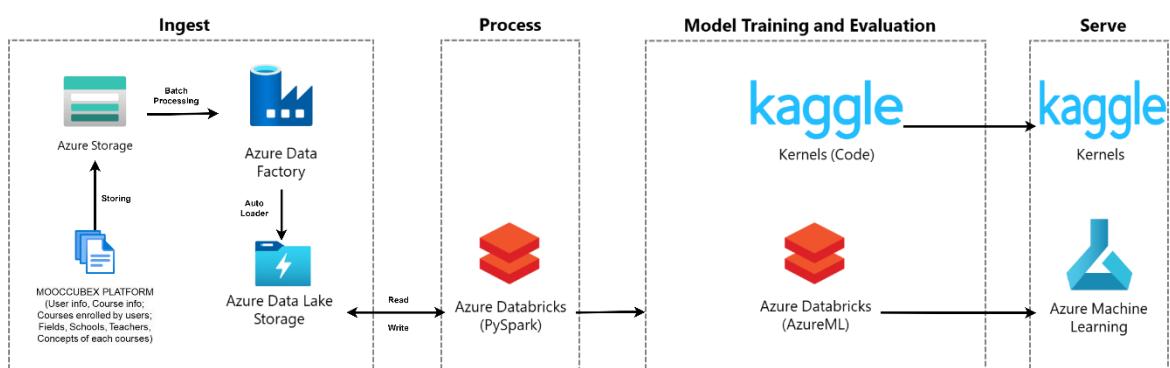


Hình 4.2 Hình minh họa giai đoạn hậu xử lý để lọc ra được top-k khóa học được đề xuất

## 4.2 Kiến trúc dữ liệu lớn

Microsoft Azure là một nền tảng điện toán đám mây (Cloud Computing) của Microsoft. Azure cung cấp một loạt các dịch vụ đám mây, bao gồm việc lưu trữ dữ liệu, xử lý dữ liệu, phân tích dữ liệu, học máy, lưu trữ ảo hóa, triển khai ứng dụng cũng như các công cụ và dịch vụ để giúp người dùng quản lý và giám sát tài nguyên của mình.

Tuy nhiên, đây là một nền tảng tính phí: Đối với gói sinh viên, nhiều dịch vụ bị giới hạn (trong đó có dịch vụ về lưu trữ và xử lý dữ liệu lớn); đối với gói xác thực cơ bản với thẻ tín dụng thì người dùng chỉ được miễn phí 200USD trong vòng 1 tháng/áp dụng cho mọi dịch vụ mà người dùng lựa chọn sử dụng nhưng cũng bị giới hạn về nhiều chức năng trên các dịch vụ đó. Do đó, để phù hợp về kinh phí cũng như đáp ứng được nhu cầu của đề tài thì nhóm chỉ sử dụng Microsoft Azure cho quá trình lưu trữ và xử lý dữ liệu lớn; đối với quá trình xây dựng và huấn luyện mô hình, nhóm sẽ thực hiện trên nền tảng Kaggle (một nền tảng trực tuyến được thiết kế cho cộng đồng những người dùng chuyên về khoa học dữ liệu và machine learning).



*Hình 4.3 Hình minh họa kiến trúc lưu trữ và xử lý dữ liệu lớn cho đề tài của nhóm*

Ingest:

- Azure Blob Storage: là giải pháp lưu trữ đối tượng trên Cloud. Blob Storage cho phép Microsoft Azure lưu trữ lượng dữ liệu phi cấu trúc lớn tùy ý và phục vụ chúng cho người dùng qua HTTP và HTTPS. Đây là nơi lưu trữ các tệp dữ liệu thô cần sử dụng cho đề tài từ nguồn MOOCubeX.
- Azure Data Lake Gen2: cho phép lưu trữ dữ liệu ở bất kỳ quy mô nào, từ dữ liệu thô chưa cấu trúc đến dữ liệu đã qua xử lý, giúp dễ dàng phân tích và khai thác giá trị từ dữ liệu, được thiết kế chủ yếu để hoạt động với Hadoop và tất cả các framework sử dụng Hệ thống tệp phân tán Apache Hadoop (HDFS) làm lớp truy cập dữ liệu - Azure Databricks. Đây là nơi lưu trữ các tệp dữ liệu thô mang tính cập nhật theo thời gian, phục vụ cho quá trình khai thác và xây dựng, huấn luyện cũng như cải tiến mô hình học máy.

Name	Modified	Access tier	Archive status	Blob type	Size	Lease state
ingestionDemo					-	...
ModelData					-	...
ModelStorage					-	...
PreprocessedData					-	...
RawData					-	...
translatedData					-	...

Hình 4.4 Hình minh họa dữ liệu cho việc xử lý, phân tích được lưu trữ trên azure data lake gen 2 - 1

Name	Modified	Access tier	Archive status	Blob type	Size	Lease state
L1					-	...
azure					-	...
entity_list_24052024102435.txt					-	...
kg_final_24052024102435.txt					-	...
relation_list_24052024102435.txt					-	...
user_list_24052024102435.txt					-	...
entity_list	5/23/2024, 11:33:54 AM	Hot (Inferred)		Block blob	297.15 kB	Available
interactions_n_core.txt	5/23/2024, 11:34:02 AM	Hot (Inferred)		Block blob	76.01 MiB	Available

Hình 4.5 Hình minh họa dữ liệu cho việc xử lý, phân tích được lưu trữ trên azure data lake gen 2 - 2

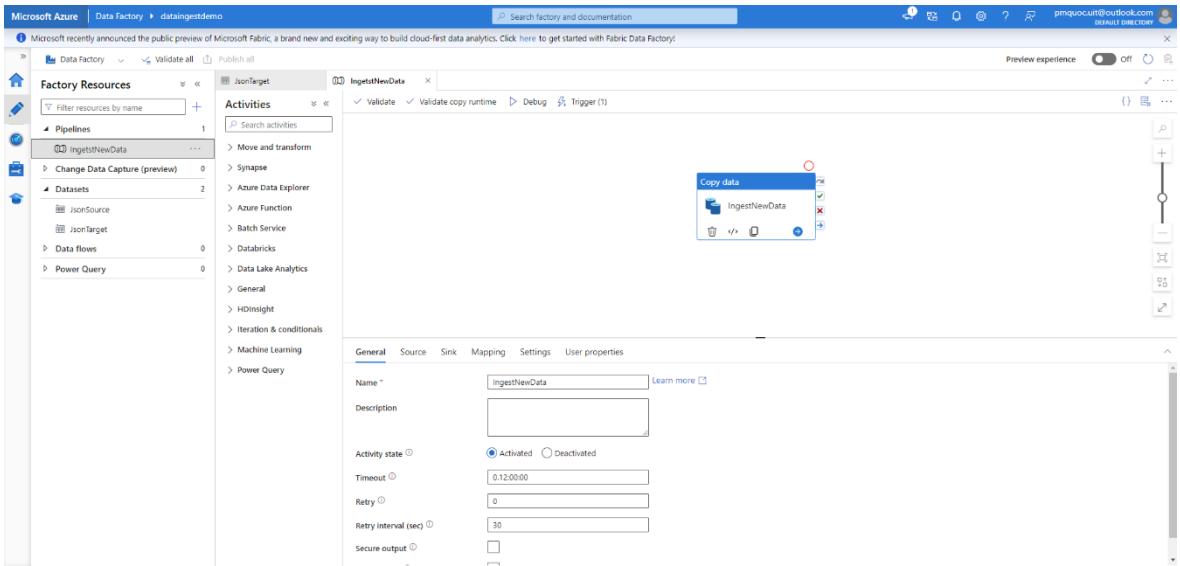
- Azure Data Factory: cho phép người dùng tạo, quản lý, và giám sát các luồng công việc tích hợp dữ liệu (data integration workflows), hỗ trợ việc di chuyển và chuyển đổi dữ liệu từ nhiều nguồn khác nhau đến các đích khác nhau. Một Pipeline được nhóm thiết kế từ dịch vụ này sẽ làm nhiệm vụ kiểm tra và cập nhật dữ liệu mới từ Azure Blob Storage vào Azure Data Lake Gen2 nhằm đảm bảo dữ liệu tại datalake luôn có tính cập nhật và sẵn sàng cho quá trình khai phá, xây dựng và cải tiến mô hình máy học.

Resource group (move)	Status	Location	Subscription (move)	Subscription ID
cs313-demo	Succeeded	Southeast Asia	Azure subscription 1	310e3256-6480-9915-9bdce5800e9

Azure Data Factory Studio

Launch studio

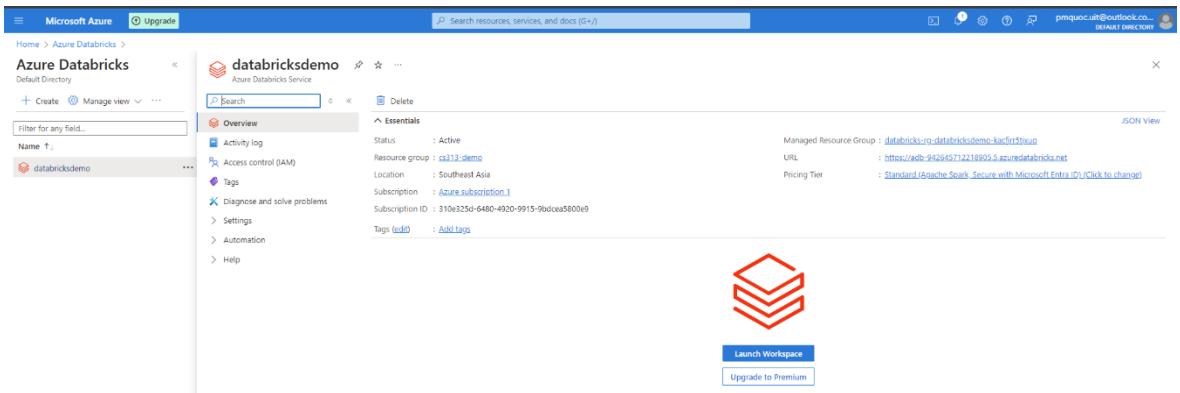
Hình 4.6 Hình minh họa pipeline cho quá trình ingest dữ liệu cập nhật-1



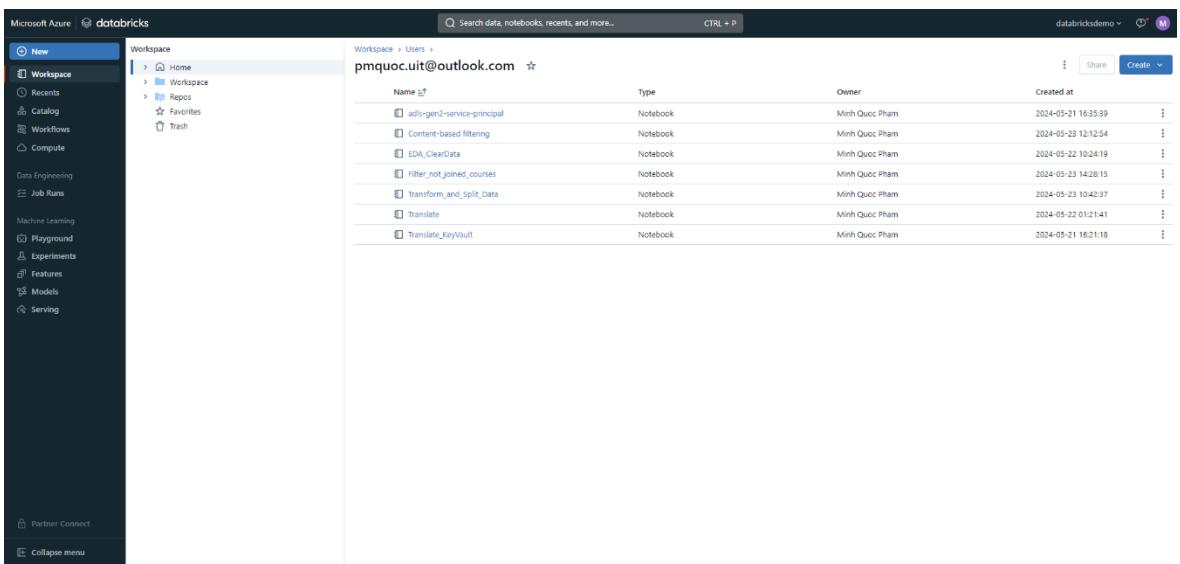
Hình 4.7 Hình minh họa pipeline cho quá trình ingest dữ liệu cập nhật-2

Process:

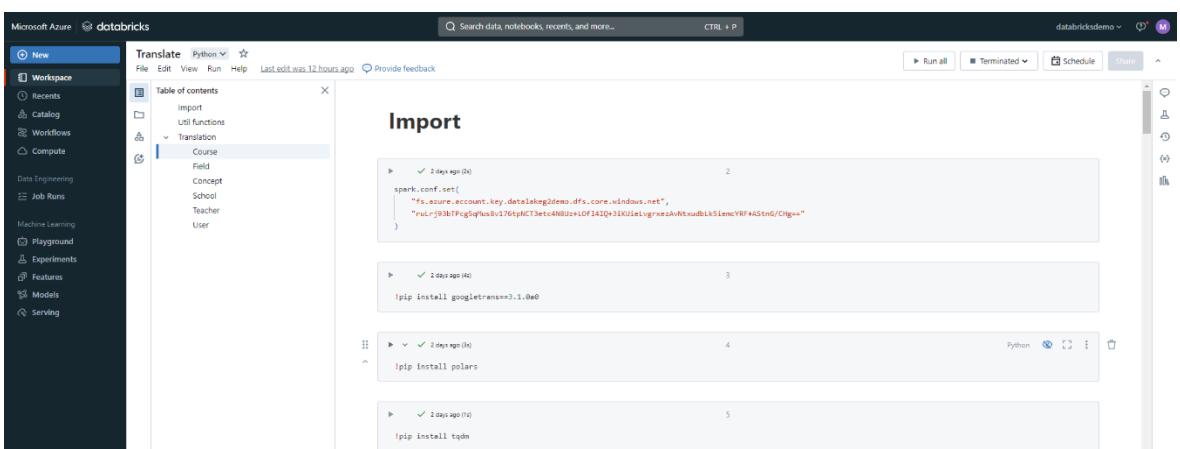
- Azure Databricks: là một dịch vụ phân tích dữ liệu lớn và học máy mạnh mẽ, linh hoạt và dễ sử dụng, được xây dựng trên nền tảng Apache Spark. Với khả năng tích hợp mạnh mẽ với các dịch vụ Azure khác, quản lý cụm tự động, và các tính năng bảo mật cao, Azure Databricks giúp các tổ chức dễ dàng xây dựng và triển khai các giải pháp phân tích và học máy quy mô lớn, từ đó khai thác tối đa giá trị từ dữ liệu của mình. Để giảm thiểu chi phí thử nghiệm trên Azure, nhóm sẽ thử nghiệm và xây dựng các mã nguồn python trên Google Colab cho quá trình Xử lý dữ liệu lớn (Data Translation, EDA and Data Preprocessing, Data Transform and Splitting); sau đó các mã nguồn này sẽ được Refactor lại với các thư viện cần thiết khác (PySpark, AzureMLCore) để phù hợp với môi trường xử lý phân tán trên Azure Databricks.



Hình 4.8 Hình minh họa tạo databricks cluster trên Azure



Hình 4.9 Hình minh họa tạo các Script xử lý và khai phá dữ liệu trên Azure Databricks



Hình 4.10 Hình minh họa chạy Script translation dữ liệu trên Azure Databricks

The screenshot shows the Microsoft Azure Databricks interface. On the left, the sidebar includes options like Workspace, Recents, Catalog, Workflows, Compute, Data Engineering, Job Runs, Machine Learning, Experiments, Features, Models, and Serving. The main area displays two notebooks: 'EDA\_ClearData' and 'Preprocess'. The 'EDA\_ClearData' notebook has a table of contents with sections such as Import, EDA, Thống kê mô tả và trực quan hóa dữ liệu, Xử lý dữ liệu, Phân tích thống kê, Khai phá dữ liệu, and ClearData-Làm sạch dữ liệu. The code editor shows Python scripts for data cleaning and preprocessing. The 'Preprocess' notebook also has a table of contents with sections like Import, Create new variables, N-core filtering, Knowledge Graph filtering, Mapping, Train-val-test split, Temp, and Statistics of knowledge graph. Its code editor shows Python scripts for data transformation and splitting.

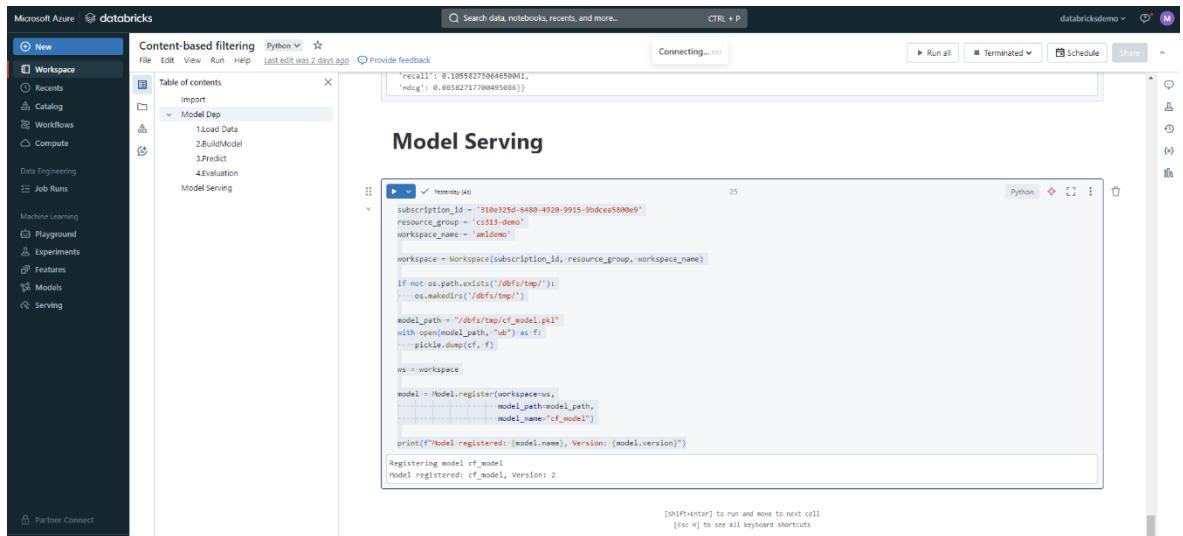
Hình 4.11 Hình minh họa chạy Script EDA và Preprocessing dữ liệu trên Azure Databricks

This screenshot shows the same Azure Databricks workspace as the previous one. It highlights the 'Transform\_and\_Split\_Data' notebook. The table of contents includes sections like Import, Create new variables, N-core filtering, Knowledge Graph filtering, Mapping, Train-val-test split, Temp, and Statistics of knowledge graph. The code editor contains Python scripts for transforming and splitting data, similar to the EDA and Preprocessing notebooks.

Hình 4.12 Hình minh họa chạy Script Tramsform and Split dữ liệu trên Azure Databricks

### Model Traning and Evaluation:

- Nhóm thực hiện tương tự quy trình Process cho việc xây dựng và huấn luyện mô hình Content-based filtering trên Azure Databricks.



Hình 4.13 Hình minh họa chạy Script Train-Eval Model trên Azure Databricks

- Đối với 4 mô hình còn lại, bao gồm mô hình KGAT, nhóm sẽ dùng dữ liệu đã xử lý để thực nghiệm trên nền tảng Kaggle.

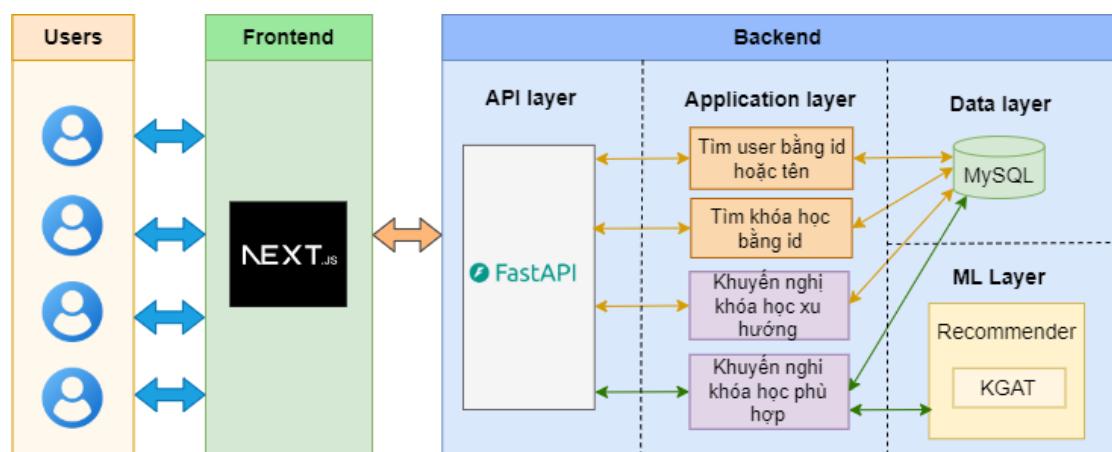
#### Model Serve:

- Azure Machine Learning: là một nền tảng giúp quản lý và theo dõi các thí nghiệm học máy, bao gồm việc lưu trữ và so sánh các kết quả huấn luyện, từ đó dễ dàng chọn lựa mô hình tốt nhất; bên cạnh đó AML còn cung cấp các công cụ để dễ dàng để dễ dàng triển khai và quản lý các mô hình học máy dưới dạng dịch vụ web (Web service). Mô hình Content-based filtering sau khi được xây dựng và đánh giá từ Azure Databricks sẽ được đăng ký và lưu trữ trên AML, phục vụ cho việc theo dõi, tối ưu hóa mô hình cũng như xây dựng và triển khai mô hình dưới dạng web service.

Hình 4.14 Hình minh họa các model đã được đăng ký và lưu trữ trên Azure Machine Learning

### 4.3 Ứng dụng web

Để thuận tiện cho việc demo cũng như thể hiện khả năng tích hợp của hệ thống gọi ý vào các nền tảng học tập trực tuyến, nhóm đã xây dựng một ứng dụng web với Nextjs làm frontend, fastapi làm giao diện lập trình ứng dụng giữa frontend và backend, MySQL làm database và sử dụng KGAT model làm hệ thống gợi ý. Framework của ứng dụng như hình sau:



Hình 4.15 Hình minh họa quy trình xây dựng website của nhóm

### 4.3.1 Công nghệ được sử dụng

Trong phần này, nhóm sẽ giới thiệu sơ lược về các công nghệ được sử dụng trong framework ứng dụng.

#### 4.3.1.1 NextJS

NextJS là framework mã nguồn mở được xây dựng trên nền tảng của React, cho phép chúng ta xây dựng các trang web tĩnh có tốc độ siêu nhanh và thân thiện với người dùng, cũng như xây dựng các ứng dụng web React. NextJS được ra đời vào năm 2016, thuộc sở hữu của Vercel. NextJS bắt đầu trở nên phổ biến vào năm 2018 và tiếp tục tăng trưởng mạnh mẽ trong cộng đồng phát triển web vào những năm sau đó. Sự kết hợp của các tính năng như Server-side Rendering (SSR) với Static Site Generation (SSG) đã giúp NextJS trở thành sự lựa chọn hấp dẫn cho nhiều dự án phát triển ứng dụng web.



Hình 4.16 NextJS

Ưu điểm:

- Mang lại khả năng SEO (Search Engine Optimization) tốt.
- Trải nghiệm người dùng tốt hơn.
- Làm việc với cơ chế SSG (Static Site Generation), SSR (Server Side Rendering) và cả CSR (Client Side Rendering).
- Khởi tạo nhanh chóng.
- Hỗ trợ nền React cực kì tốt. Hỗ trợ cấu trúc và tổ hợp một cách tối ưu.
- Hỗ trợ phát triển tính năng nhanh chóng cho việc cấu hình như: Webpack, Babel,... Bảo mật về dữ liệu.
- Khả năng thích ứng và đáp ứng thay đổi.

Nhược điểm:

- Ít plug-in thích ứng.
- Nextjs bị giới hạn bởi việc chỉ sử dụng bộ định tuyến trên tệp của nó, ta không thể nào sửa đổi cách nó giao dịch với các tuyến. Vì vậy, để sử dụng tuyến động, ta cần làm việc thêm với Node.js máy chủ.
- Nextjs không cung cấp nhiều trang nhất tích hợp, để làm việc ta cần phải tạo toàn bộ front-end từ đầu lên.

#### 4.3.1.2 FastAPI

FastAPI là một web framework Python hiện đại, rất hiệu quả trong việc xây dựng API, code đơn giản nhưng hỗ trợ tốt cho việc làm ra sản phẩm.



*Hình 4.17 FastAPI*

Các tính năng của FastAPI:

- Nhanh: Hiệu năng rất cao khi so sánh với NodeJS và Go. Một trong những Python framework nhanh nhất, giúp hệ thống truy vấn phản hồi nhanh hơn.
- Code nhanh: Tăng tốc độ phát triển tính năng từ 200% tới 300%
- Ít lỗi hơn: Do đơn giản nên giảm khoảng 40% những lỗi được sinh ra bởi developer.
- Trực giác tốt hơn: Được các trình soạn thảo hỗ trợ tuyệt vời. Completion mọi nơi. Ít thời gian gỡ lỗi.
- Dễ dàng: Được thiết kế để dễ dàng học và sử dụng. Ít thời gian đọc tài liệu.
- Ngắn: Tối thiểu việc lặp code. Các tham số truyền vào có nhiều tính năng. Ít bugs.
- Mạnh mẽ: hiệu năng mạnh mẽ, có thể tương tác API qua docs.

### 4.3.1.3 MySQL

MySQL là một hệ thống quản lý cơ sở dữ liệu quan hệ mã nguồn mở (RDBMS) dựa trên ngôn ngữ truy vấn có cấu trúc (SQL) được phát triển, phân phối và hỗ trợ bởi tập đoàn Oracle. MySQL chạy trên hầu hết tất cả các nền tảng, bao gồm cả Linux , UNIX và Windows. MySQL thường được kết hợp với các ứng dụng web



Hình 4.18 MySQL

Ưu điểm:

- Dễ sử dụng: MySQL là cơ sở dữ liệu tốc độ cao, ổn định, dễ sử dụng và hoạt động trên nhiều hệ điều hành cung cấp một hệ thống lớn các hàm tiện ích rất mạnh.
- Độ bảo mật cao: MySQL rất thích hợp cho các ứng dụng có truy cập CSDL trên Internet khi sở hữu nhiều tính năng bảo mật thậm chí là ở cấp cao.
- Đa tính năng: MySQL hỗ trợ rất nhiều chức năng SQL được mong chờ từ một hệ quản trị cơ sở dữ liệu quan hệ cả trực tiếp lẫn gián tiếp.
- Khả năng mở rộng và mạnh mẽ: MySQL có thể xử lý rất nhiều dữ liệu và hơn thế nữa nó có thể được mở rộng nếu cần thiết.
- Nhanh chóng: Việc đưa ra một số tiêu chuẩn cho phép MySQL để làm việc rất hiệu quả và tiết kiệm chi phí, do đó nó làm tăng tốc độ thực thi.

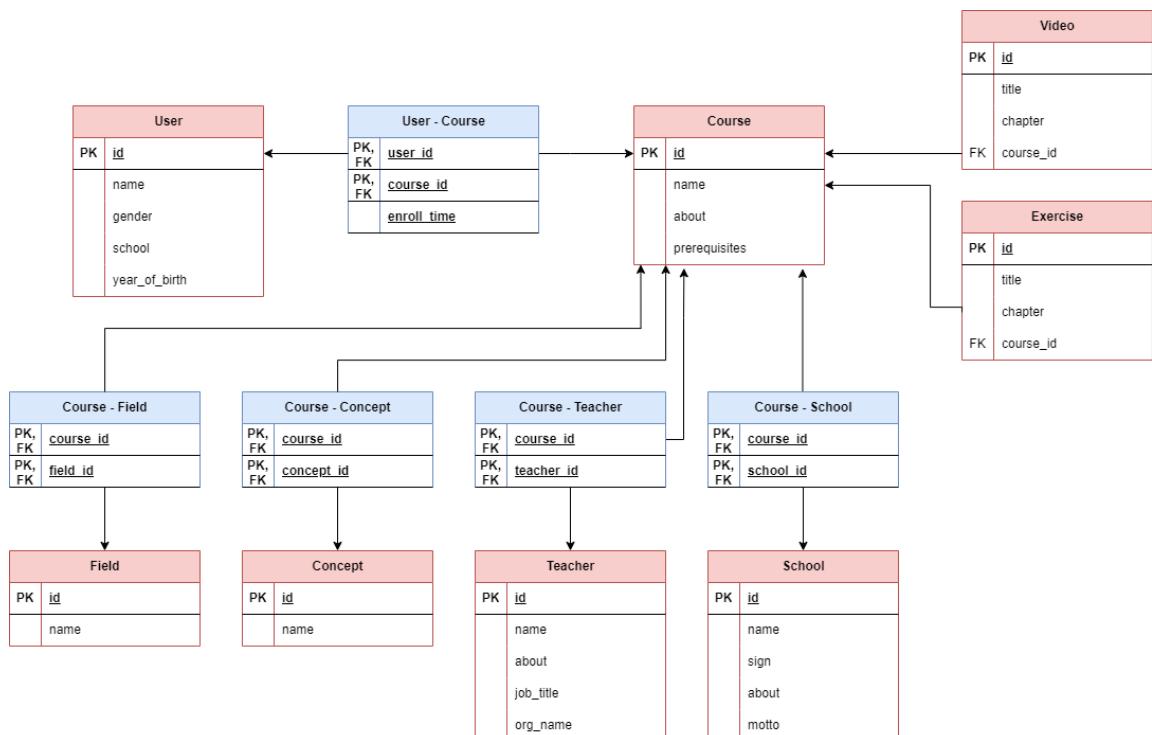
Nhược điểm:

- Giới hạn: Theo thiết kế, MySQL không có ý định làm tất cả và nó đi kèm với các hạn chế về chức năng mà một vào ứng dụng có thể cần.

- Độ tin cậy: Cách các chức năng cụ thể được xử lý với MySQL (ví dụ tài liệu tham khảo, các giao dịch, kiểm toán,...) làm cho nó kém tin cậy hơn so với một số hệ quản trị cơ sở dữ liệu quan hệ khác.
- Dung lượng hạn chế: Nếu số bản ghi lớn dần lên thì việc truy xuất dữ liệu trở nên khá khó khăn, khi đó ta sẽ phải áp dụng nhiều biện pháp để tăng tốc độ truy xuất dữ liệu như là chia tải database này ra nhiều server, hoặc tạo cache MySQL

### 4.3.2 Thiết kế cơ sở dữ liệu

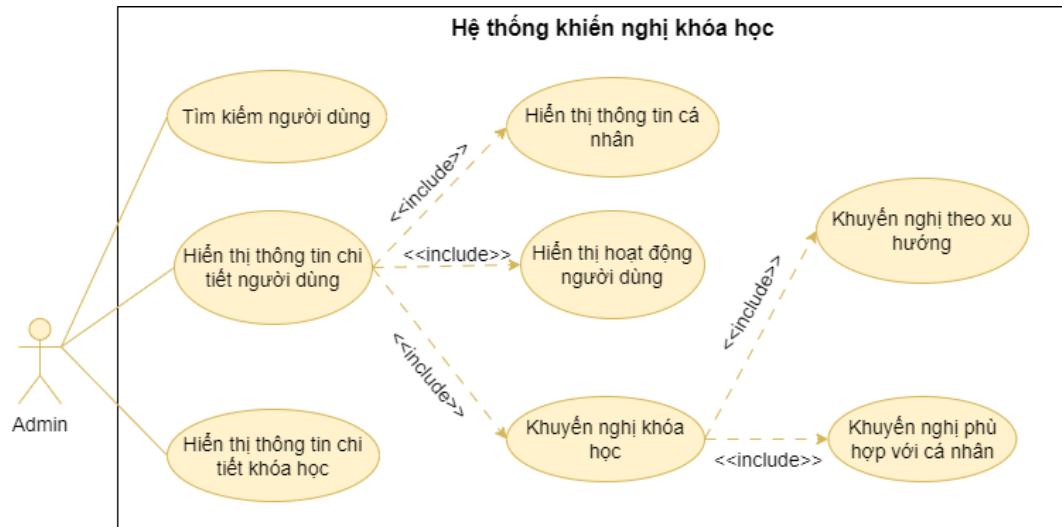
Dữ liệu của MOOCubeX được lưu trữ dưới các file json và txt, vì vậy nhóm cần phải chuyển các file này vào cơ sở dữ liệu của MySQL. Trước hết, nhóm cần phải thiết kế một lược đồ cơ sở dữ liệu để lưu trữ các thông tin cần thiết cho việc demo:



Hình 4.19 Lateral view of the database schema

### 4.3.3 Thiết kế giao diện

Sơ đồ use-case của ứng dụng như sau:



*Hình 4.20 Hình minh họa sơ đồ use-case tổng quan của bài toán*

Web được thiết kế để mô phỏng một số tính năng hỗ trợ các nền tảng học tập trực tuyến. Giả sử sau khi admin đăng nhập và vào tính năng “Search”, giao diện sẽ hiển thị như Hình 4.21. Giao diện chứa các thành phần chính như: input thông tin để tìm kiếm; nút tìm kiếm; bảng thông tin người dùng; nút back để quay trở lại trang chủ (nhưng ở đây nhóm chỉ demo tính năng tìm kiếm nên nút back chỉ tượng trưng). Trong đó, input thông tin để tìm kiếm gồm 2 ô chính: 1 input dùng để nhập user id, 1 input dùng để nhập user name. Khi nhập thông tin vào 1 trong 2 ô, sau đó click chuột sang ô còn lại thì thông tin ở ô được nhập trước đó sẽ bị xóa, điều này giúp hệ thống chỉ tìm kiếm dựa trên 1 trong 2 thông tin tại 1 thời điểm. Sau khi nhập thông tin và nhấn nút tìm kiếm, bảng thông tin người dùng sẽ chỉ giữ lại các người dùng có thông tin phù hợp như Hình 4.22 hay Hình 4.23.

Course Recommender System

Search Back

User ID:  Search

User ID	Username	Gender	School	Year of Birth
U_100066	Hồ Hải Duyên	2	华北电力大学	null
U_1000826	Dương Gia Ân	2	昆明理工大学	null
U_100096	Nguyễn Hồng Nhật	0	北京大学	null
U_1001553	Ngô Ngọc Huy	0	昆明理工大学	null
U_100156	Lý Ngọc Hoan	2	工大	null
U_1002253	Bùi Diễm Thư	1	亭湖高级中学	null
U_1002389	Lê Loan Châu	1	陕西科技大学	null
U_1002476	Hồ Thu Hằng	1	吉林信息工程大学	null
U_1002666	...	...	...	...

Hình 4.21: Giao diện tìm kiếm ban đầu.

Search Back

User ID:  Search

User ID	Username	Gender	School	Year of Birth
U_1007443	Huỳnh Oanh Vũ	2	暨南大学	null

Hình 4.22 Giao diện tìm kiếm khi tìm kiếm với user id “U\_1007443”

User ID	Username	Gender	School	Year of Birth
U_1007443	Huỳnh Oanh Vũ	2	聖南大学	null
U_21664797	Huỳnh Oanh Vũ	1	null	null

*Hình 4.23 Giao diện tìm kiếm khi tìm kiếm với username “Huỳnh Oanh Vũ”*

Khi click chuột vào 1 hàng trong bảng, ta sẽ chuyển sang giao diện thông tin chi tiết của người dùng như Hình 4.24. Giao diện này chứa 5 thành phần chính: bảng thông tin cá nhân (User Information); bảng các khóa học đã đăng ký (Enrolled Courses); bảng các khóa học xu hướng (Trending courses); bảng các khóa học phù hợp với người dùng (Courses you may like); Nút back dùng để quay lại giao diện tìm kiếm. Bảng các khóa học phù hợp với người dùng sử dụng mô hình khuyến nghị tốt nhất trong thực nghiệm của nhóm – KGAT.

Name	Schools	Number of users
nghe và nói tiếng anh sống động	Đại học Thanh Hoa	18378
giới thiệu về tám lý học	Đại học Thanh Hoa	15517

Name	Schools
nghe và nói tiếng anh sống động	Đại học Thanh Hoa
giới thiệu về tám lý học	Đại học Thanh Hoa
giới thiệu về tư tưởng mao trạch đồng và hệ thống lý luận về chủ nghĩa xã hội đặc	Đại học Thanh

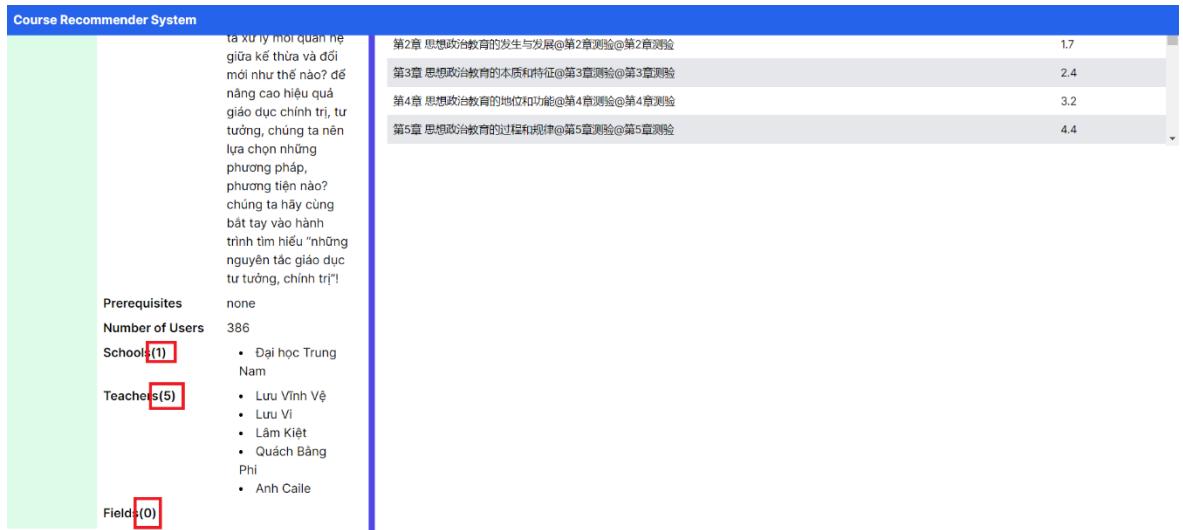
*Hình 4.24 Giao diện thông tin chi tiết của người dùng*

Khi click trái chuột vào 1 hàng (khóa học) trong bảng, ta sẽ chuyển sang giao diện thông tin chi tiết của khóa học như Hình 4.25 và Hình 4.26. Giao diện này gồm 4 thành phần chính: Bảng thông tin khóa học (Course information); Bảng Videos – cho biết thông tin cơ bản của các videos của khóa học; Bảng Exercises – cho biết thông tin cơ bản của các bài tập của khóa học. Trong giao diện, vị trí được khoanh vùng màu đỏ sẽ cho biết số lượng hàng hàng tương ứng trong bảng.



Course Information		
Name	nguyên tắc giáo dục tư tưởng, chính trị	Videos (89)
About	xin chào các bạn cùng lớp thân mến! bạn có muốn biết giáo dục tư tưởng, chính trị ra đời và phát triển như thế nào không? làm thế nào để suy nghĩ và hành vi chuyển hóa lẫn nhau? trong quá trình giáo dục chính trị, tư tưởng, chúng ta xử lý mối quan hệ giữa kế thừa và đổi mới như thế nào? để nâng cao hiệu quả giáo dục chính trị, tư tưởng, chúng ta nên lựa chọn những	
Exercises (14)		Chapter
Title		
第1章 思想政治教育与思想政治教育学述要@1.1 思想政治教育的概念@思想政治教育的概念	0	
第1章 思想政治教育与思想政治教育学述要@1.2 思想政治教育的内涵与外延@1.2思想政治教育的内涵与外延	0.1	
第1章 思想政治教育与思想政治教育学述要@1.3 思想政治教育学的研究对象@1.3 思想政治教育学的研究对象	0.2	
第1章 思想政治教育与思想政治教育学述要@1.4 思想政治教育学的基本范畴@思想政治教育学的基本范畴（一）	0.3	
第2章 思想政治教育与思想政治教育学述要@2.1 思想政治教育与思想政治教育学的关系@2.1思想政治教育与思想政治教育学的关系	0.21	
Exercise (14)		
Title		Chapter
第1章 思想政治教育与思想政治教育学述要@第1章测验	0.6	
第2章 思想政治教育的发生与发展@第2章测验@第2章测验	1.7	
第3章 思想政治教育的本质和特征@第3章测验@第3章测验	2.4	
第4章 思想政治教育的地位和功能@第4章测验@第4章测验	3.2	
第5章 思想政治教育的过程和规律@第5章测验@第5章测验	4.4	

Hình 4.25 Giao diện thông tin chi tiết của khóa học (phần trên)



Course Recommender System	
Prerequisites	ta xử lý mối quan hệ giữa kế thừa và đổi mới như thế nào? để nâng cao hiệu quả giáo dục chính trị, tư tưởng, chúng ta nên lựa chọn những phương pháp, phương tiện nào? chúng ta hãy cùng bắt tay vào hành trình tìm hiểu "những nguyên tắc giáo dục tư tưởng, chính trị"!
Prerequisites	none
Number of Users	386
Schools (1)	• Đại học Trung Nam
Teachers (5)	• Lưu Vĩnh Vệ • Lưu Vì • Lâm Kiệt • Quách Bằng Phi • Anh Calle
Fields (0)	

Hình 4.26 Giao diện thông tin chi tiết của khóa học (phần dưới)

## 5 THỰC NGHIỆM

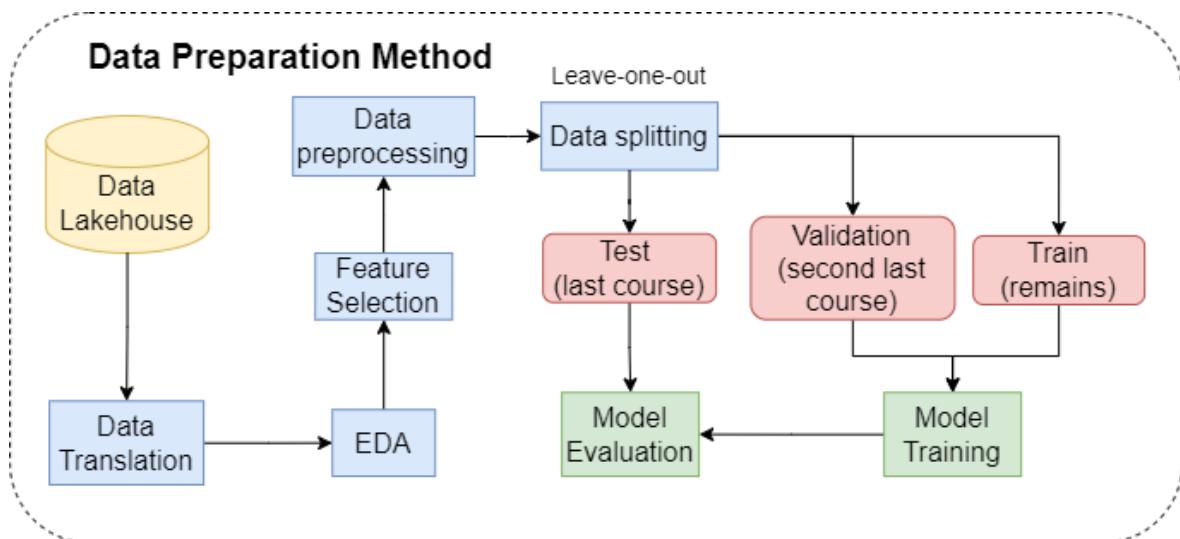
### 5.1 Dữ liệu thực nghiệm

Từ bộ dữ liệu MOOCCubeX, nhóm đã chọn lọc các files sau để xây dựng hệ thống gợi ý khóa học và ứng dụng web:

- Entities: course.json, teacher.json, school.json, concept.json, user.json, video.json.
- Relations: course-field.json, course-school.txt, course-teacher.txt, course-concept.txt, user-video.json, vid\_ccid.txt.

### 5.2 Phương pháp tổ chức dữ liệu thực nghiệm

Các thao tác chuẩn bị dữ liệu được thực hiện theo trình tự sau:



Hình 5.1 Hình minh họa quy trình xử lý dữ liệu của nhóm

#### 5.2.1 Data translation

Để dịch dữ liệu từ tiếng Trung sang tiếng Việt, nhóm sử dụng Googletrans [14], một thư viện python miễn phí và không giới hạn, triển khai API Google Translate. Thư viện này sử dụng API Google Translate Ajax để thực hiện lệnh gọi đến các phương thức như phát hiện và dịch. Nhưng do có một số trường có lượng dữ liệu cần dịch lớn nên thời gian dịch rất lâu, và trong quá trình dịch cũng xảy ra hiện tượng bị mất kết

nội. Bên cạnh đó, việc dịch một số trường có lượng dữ liệu lớn cũng không cần thiết để huấn luyện mô hình nên nhóm chỉ dịch một số trường có dữ liệu nhỏ để hiển thị thông tin trên ứng dụng web. Dưới đây là các trường mà nhóm đã dịch:

- course.json: name, prerequisites, fields
- user.json: không dịch, nhưng ở cột name, nhóm đã sử dụng bộ sinh tên tiếng Việt dựa trên giới tính được cung cấp bởi [15]. Việc này giúp thuận tiện cho việc mô phỏng khả năng tìm kiếm bằng username của ứng dụng web.
- school.json: name, about, motto
- teacher.json: name, job\_title, org\_name, about

## 5.2.2 EDA, làm sạch dữ liệu

Để hiểu rõ hơn về tập dữ liệu và xác định các mẫu, xu hướng tiềm ẩn, nhóm đã thực hiện phân tích dữ liệu khám phá

### 5.2.2.1 Thống kê mô tả, trực quan hóa dữ liệu, xử lý dữ liệu

#### 5.2.2.1.1 Course

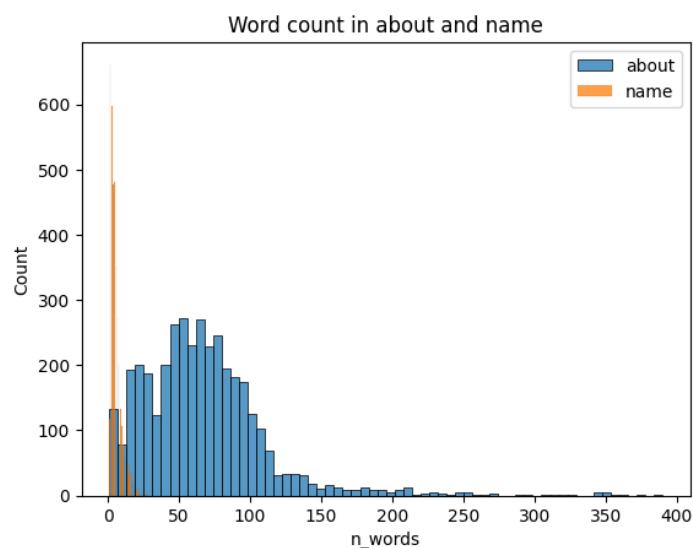
describe	id	name	field	prerequisites	about	resource	name_trans	about_trans
str	str	str	str	str	str	str	str	str
"count"	"3781"		"3781" "3781"	"3779"		"3779" "3781"	"3781"	"3781"
"null_count"	"0"		"0" "0"	"2"		"2" "0"	"0"	"0"

Hình 5.2 Bảng thống kê mô tả của course.json

course.json có 3781 hàng với các trường thông tin như sau: case\_id, name (đã được dịch), field (đã được dịch), prerequisites, about (đã được dịch), resource. Tất cả các trường đều có rất ít null. Nhận thấy bên cạnh sử dụng tfidf để tạo đặc trưng trên trường about, name, ta vẫn có thể sử dụng PhoBERT [16] để embed tạo đặc trưng. Do PhoBERT được huấn luyện trên dữ liệu ở mức độ từ, nên ta sẽ phải gôm các tiếng lại thành từ bằng rdrsegmenter của VNCoreNLP [17]. Sau đây là một số thống kê cơ bản:

describe	about_segmented	name_segmented	len_about_segmented	len_name_segmented
	str	str	f64	f64
"count"	"3781"	"3781"	3781.0	3781.0
"null_count"	"0"	"0"	0.0	0.0
"mean"	null	null	66.694261	5.525522
"std"	null	null	44.815107	3.791343
"min"	""	"" diễn_dàn gia...	1.0	1.0
"25%"	null	null	39.0	3.0
"50%"	null	null	62.0	5.0
"75%"	null	null	87.0	7.0
"max"	"🔥 " tâm_lý_học... "中医基础理论俄文版 (co b...		390.0	28.0

Hình 5.3 Thống kê cơ bản về số từ trong about, name sau khi segmented (len\_about\_segmented, len\_name\_segmented)



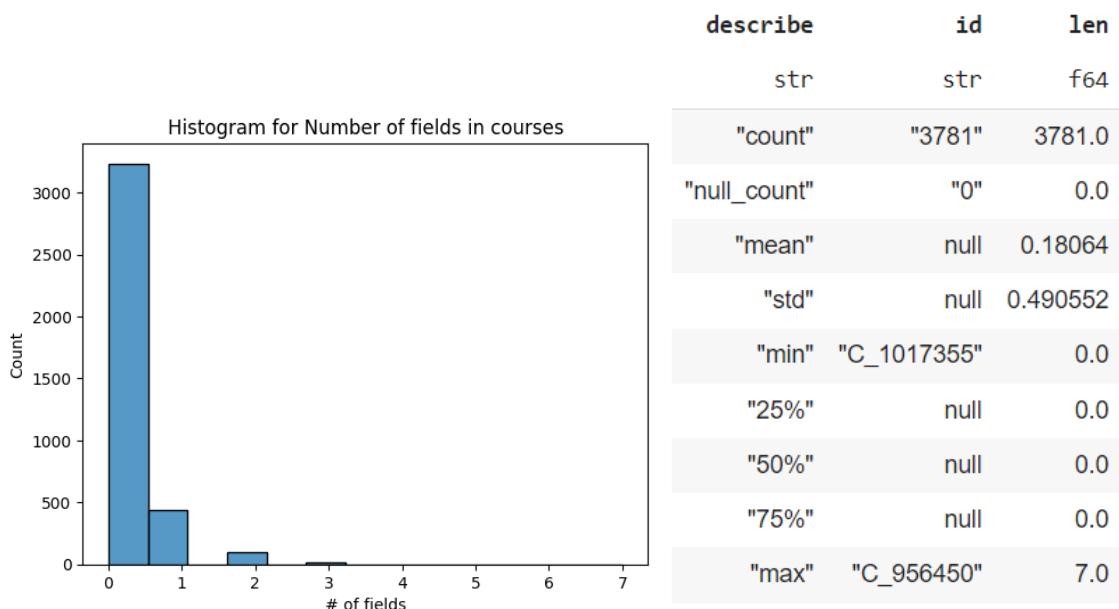
Hình 5.4 Biểu đồ cột thê hiện độ dài văn bản ở trường about, name.

Từ bảng trên, ta có độ dài ngắn nhất trong trường about, name là 1; độ dài lớn nhất trong about, name lần lượt là 390 và 28; độ dài trung bình trong about, name lần lượt là 67 và 6. Từ hình vẽ, ta thấy được độ dài văn bản trong trường about, name lần lượt có dạng phân phối chuẩn và long tail. Tiếp đến, ta sẽ xét trường field và file course-field.json.

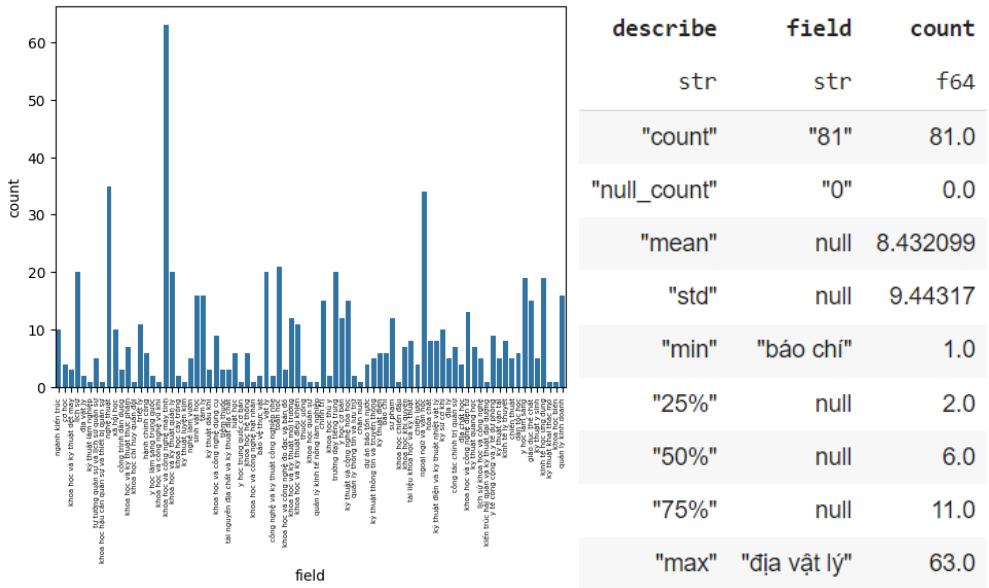
describe	course_id	course_name	field	course_name_trans
str	str	str	str	str
"count"	"632"	"632"	"632"	"632"
"null_count"	"0"	"0"	"0"	"0"

Hình 5.5 Bảng thống kê mô tả của course-field.json

Trường field của course có nội dung tương tự như file course-field.json. Ta sử dụng phép hợp để gồm 2 thông tin này lại với nhau. Ngoài ra, ta sẽ lọc bỏ những khóa học của file course-field.json mà không tồn tại trong course.json. Sau đó, ta trực quan hóa thông tin về field của khóa học như hình sau:



Hình 5.6 Histogram thể hiện số lượng fields trong mỗi khóa học (bên trái) và bảng thống kê mô tả tương ứng (bên phải)



*Hình 5.7 Biểu đồ cột thể hiện số lượng khóa học của các field (bên trái) và bảng thống kê mô tả tương ứng (bên phải).*

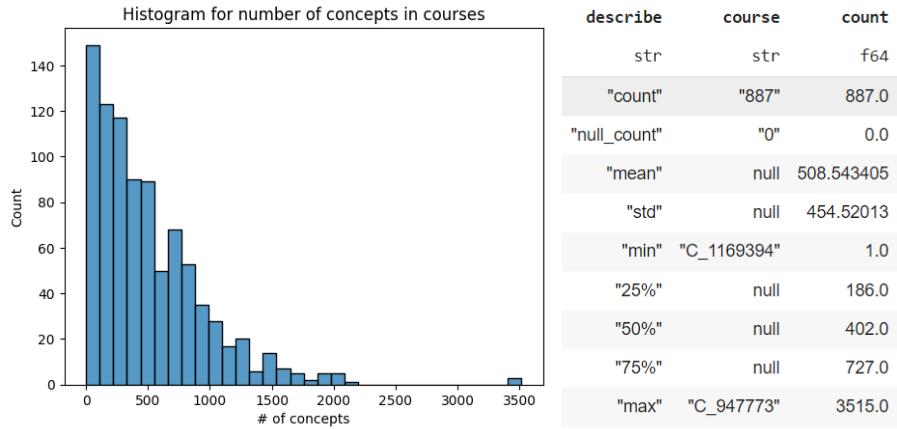
Biểu đồ cho thấy, có rất nhiều khóa học không có field nào, và có nhiều field có số lượng khóa học ít hơn 5. Điều này cho thấy, feature field của khóa học sẽ có rất ít đóng góp.

#### 5.2.2.1.2 Concept

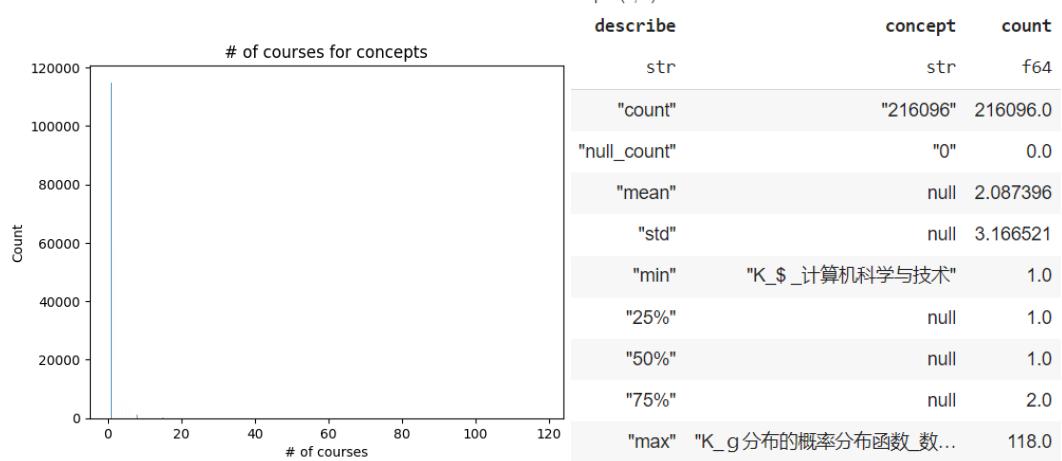
describe	id	name	context
str	str	str	str
"count"	"637572"	"637572"	"637572"
"null_count"	"0"	"0"	"0"

*Hình 5.8 Bảng thống kê mô tả của concept.json*

concept.json có 637572 hàng và các trường thông tin như id, name, context. Các trường thông tin này tuy không có null nhưng không có nhiều ý nghĩa, nên ta sẽ chỉ quan tâm đến liên kết giữa khóa học và concept (concept-course.txt với 451078 hàng). Do ta không quan tâm đến thông tin của concept nên sẽ không lọc bỏ những liên kết của các concept không hợp lệ. Sau đây là một số thông tin được trực quan hóa:



*Hình 5.9 Histogram thể hiện số lượng concept của mỗi khóa học (bên trái) và bảng thống kê mô tả tương ứng (bên phải).*



*Hình 5.10 Histogram thể hiện số lượng khóa học của mỗi concept (bên trái) và bảng thống kê mô tả tương ứng (bên phải).*

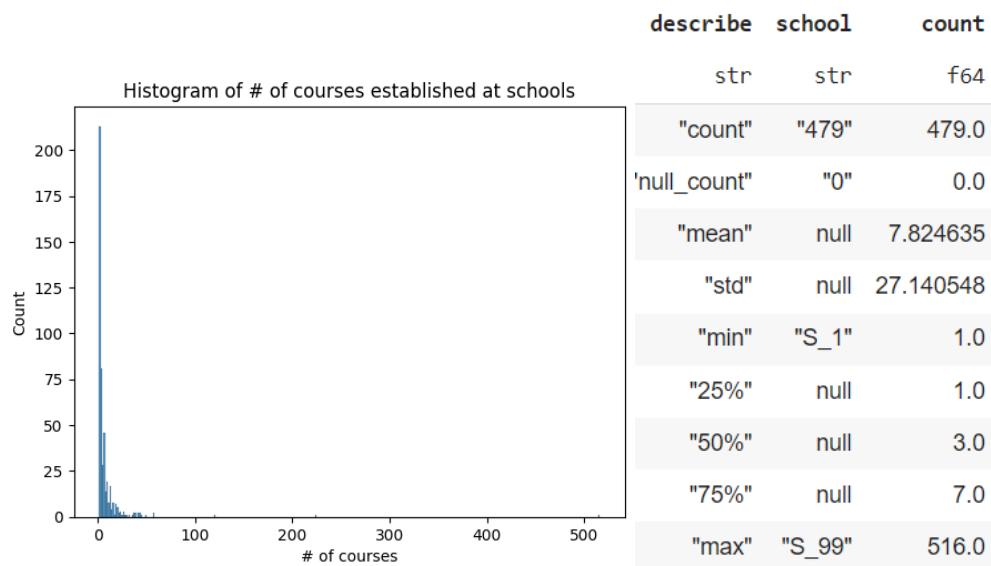
Số lượng concepts của khóa học có phân phối long tail. Số lượng concept nhiều nhất trong khóa học là 3515, ít nhất là 0. Phần lớn concepts có số lượng khóa học là 1. Số lượng khóa học ít nhất của 1 concept là 1, lớn nhất là 118. Số lượng concepts với số lượng khóa học  $\geq 5$  là 14016, điều này cho thấy, đây có thể là đặc trưng đóng góp nhiều vào dự đoán của mô hình.

### 5.2.2.1.3 School

describe	id	name	name_en	sign	about	motto	about_trans	motto_trans
str	str	str	str	str	str	str	str	str
"count"	"429"	"429"	"429"	"429"	"429"	"429"	"429"	"429"
"null_count"	"0"	"0"	"0"	"0"	"0"	"0"	"0"	"0"

Hình 5.11 Bảng thống kê mô tả của school.json

school.json gồm 429 hàng với các trường thông tin như: id, name (đã được dịch), name\_en, sign, about (đã được dịch), motto (đã được dịch). Tuy các trường thông tin này không có null nhưng lại không có quá nhiều ý nghĩa để ta khai thác. Vì vậy, ta sẽ chỉ quan tâm đến liên kết giữa khóa học và school (course-school.txt). Trong file course-school.txt có 3748 hàng. Tương tự concept, ta sẽ không lọc bỏ những liên kết của các school không hợp lệ. Một số thông tin được trực quan hóa như sau:



Hình 5.12 Histogram thể hiện số lượng khóa học của mỗi trường (bên trái) và bảng thống kê mô tả tương ứng (bên phải).

describe	course	count
str	str	f64
"count"	"3717"	3717.0
"null_count"	"0"	0.0
"mean"	null	1.00834
"std"	null	0.090955
"min"	"C_1017355"	1.0
"25%"	null	1.0
"50%"	null	1.0
"75%"	null	1.0
"max"	"C_956450"	2.0

Hình 5.13 Bảng thống kê mô tả cho số lượng trường học của mỗi khóa học.

Số lượng khóa học của một trường theo phân phối long tail. Phần lớn school có số lượng khóa học  $\leq 7$  (Phân vị 75%). Có một số school đặc biệt tổ chức từ 100 khóa học trở lên. Số lượng khóa học ít nhất của một trường là 1, nhiều nhất là 516. Có 185 schools tổ chức ít nhất 5 khóa học.

Ngoài ra, nhóm cũng thống kê xem có bao nhiêu trường cùng tổ chức 1 khóa học. Kết quả cho thấy phần lớn khóa học chỉ được tổ chức bởi 1 school (Phân vị 75%). Số schools ít nhất của một khóa học là 1 (không xét trường hợp khóa học không có school), nhiều nhất là 2. Điều này là khác biệt so với trường field, concept vì 1 khóa học có nhiều field, concept. Tuy nhiên sự khác biệt này không ảnh hưởng nhiều.

#### 5.2.2.1.4 Teacher

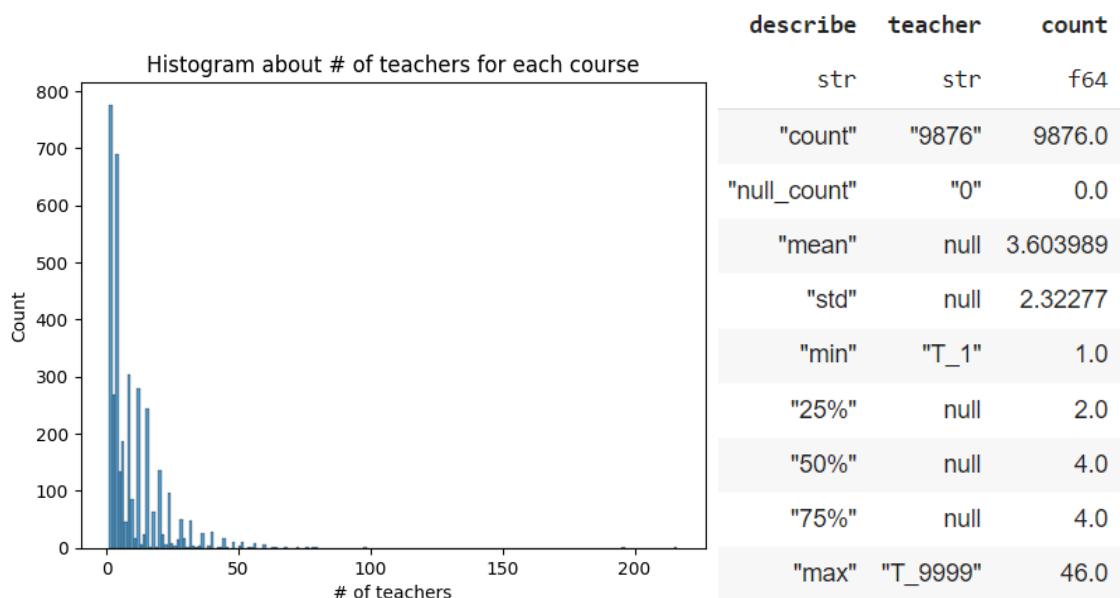
describe	id	name	name_en	about	job_title	org_name	about_trans
str	str	str	str	str	str	str	str
"count"	"17018"	"17018"	"9525"	"17018"	"14768"	"17018"	"13892"
"null_count"	"0"	"0"	"7493"	"0"	"2250"	"0"	"3126"

Hình 5.14 Bảng thống kê mô tả của teacher.json

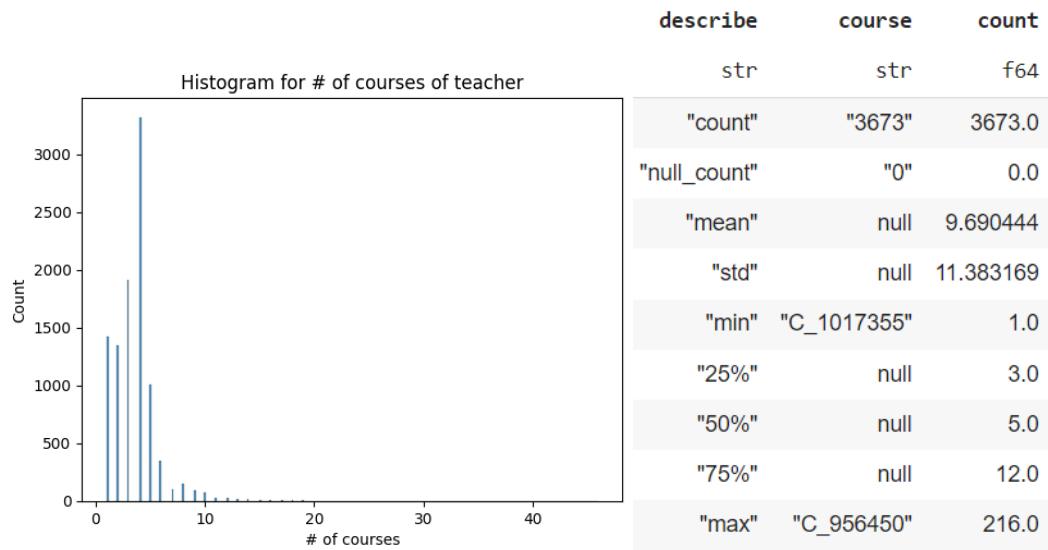
teacher.json gồm 17018 hàng và các trường thông tin như id, name, name\_en, about (đã được dịch), job\_title, org\_name. Trong đó, trường job\_title là chức danh của teacher, có thể có ích trong việc thiết kế feature. Thực chất, job\_title nên được xem như là văn bản thay vì categorical bởi vì job\_title không chỉ bao gồm chức vụ, học vị, mà còn có một số thông tin khác như tên công ty, phòng khoa... VD: "giáo sư, trường

kinh tế và quản lý, đại học thanh hóa", "phó bí thư, phó tổng bí thư đảng ủy viện kê toán công chứng trung quốc". Ta có thể sử dụng một mô hình bên ngoài (ChatGPT [18], Gemini [19]) để trích xuất các thông tin về chức vị, tổ chức, sau đó xem các biến này là categorical. Bên cạnh đó, trường org\_name, tổ chức của teacher, cũng có thể ảnh hưởng nhiều đến số lượng user tham gia vào khóa học.

Về liên kết giữa khóa học và teacher, file course-teacher.txt chứa 97192 hàng. Trong tương lai, nhóm sẽ dùng một số trường thông tin của teacher nếu có thể. Vì vậy, ta cần lọc bỏ các liên kết có khóa học hoặc teacher không tồn tại. Sau khi lọc bỏ, số hàng còn lại là 35593. Các thông tin được trực quan hóa như sau:



*Hình 5.15 Histogram thể hiện số lượng teachers của mỗi khóa học (bên trái) và bảng thống kê mô tả tương ứng (bên phải).*



*Hình 5.16 Histogram thể hiện số lượng khóa học của mỗi teacher (bên trái) và bảng thống kê mô tả tương ứng (bên phải).*

Phần lớn teachers có số lượng khóa học < 5. Có 1883 teachers với số lượng khóa học  $\geq 5$ . Phần lớn khóa học có số teachers  $\leq 12$  (Phân vị 75%).

Về org\_name của teacher, số lượng giáo viên trong một tổ chức như sau:

describe	count
str	f64
"count"	724.0
"null_count"	0.0
"mean"	13.640884
"std"	35.463706
"min"	1.0
"25%"	1.0
"50%"	4.0
"75%"	14.0
"max"	678.0

*Hình 5.17 Bảng thống kê mô tả số lượng giáo viên của một tổ chức.*

### 5.2.2.1.5 User

describe	id	name	gender	school	year_of_birth	course_order	enroll_time
str	str	str	f64	str	str	str	str
"count"	"3330294"	"3330240"	3.33024e6	"1128399"	"0"	"3330294"	"3330294"
"null_count"	"0"	"54"	54.0	"2201895"	"3330294"	"0"	"0"
"mean"	null	null	0.945575	null	null	null	null
"std"	null	null	0.83211	null	null	null	null
"min"	"U_10000"	""	0.0	"Queen's Univ..."	null	null	null
"25%"	null	null	0.0	null	null	null	null
"50%"	null	null	1.0	null	null	null	null
"75%"	null	null	2.0	null	null	null	null
"max"	"U_999999"	"□"	232.0	"🔧 工程大学"	null	null	null

Hình 5.18 Bảng thông kê mô tả của user.json

user.json gồm 3330294 hàng với các trường thông tin như id, name (được khởi tạo), gender, school, year\_of\_birth, course\_order, enroll\_time. Trường school có tới 66.12% là null, ngoài ra có một số ký tự đặc biệt như: "🔧 工程大学"; "Queen's Univ...", có lẽ trường này không được kiểm tra trước khi đưa vào CSDL của XuetangX. Trường year of birth gần như toàn bộ là null nên không đem lại ý nghĩa. Các khóa học trong course\_order sẽ tương ứng với enroll\_time và chúng được sắp xếp theo thời điểm tăng ký, nên sẽ thuận tiện cho việc chia tập train, val, test theo thời gian. Tiếp theo, nhóm sẽ lọc bỏ những khóa học không hợp lệ (không tồn tại trong course.json)

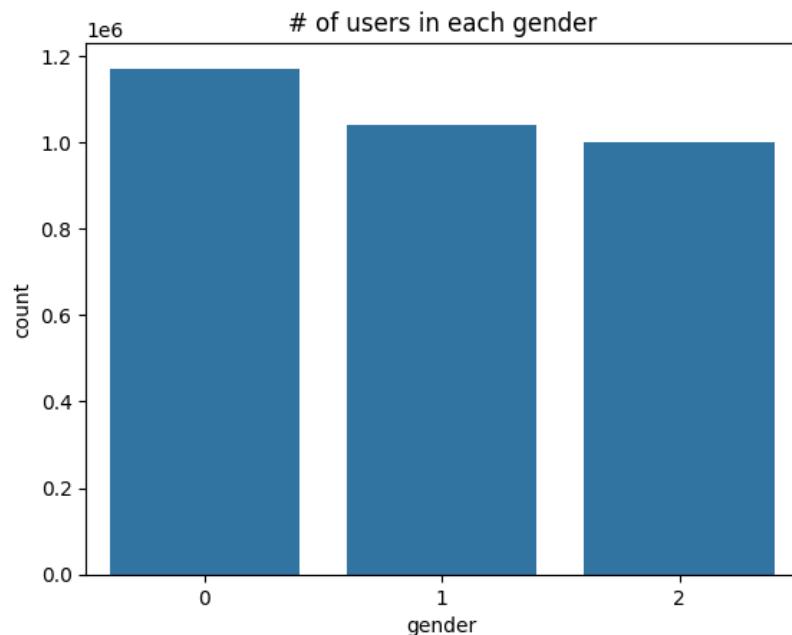
Sau đó, ta xét trường gender. Trường này chỉ có  $\frac{54}{3330294} * 100 = 0.0016\%$  giá trị null nên ta có thể sử dụng để làm feature.

gender	count
i64	u32
232	1
2	1040449
3	1
1	1067858
0	1221931
null	54

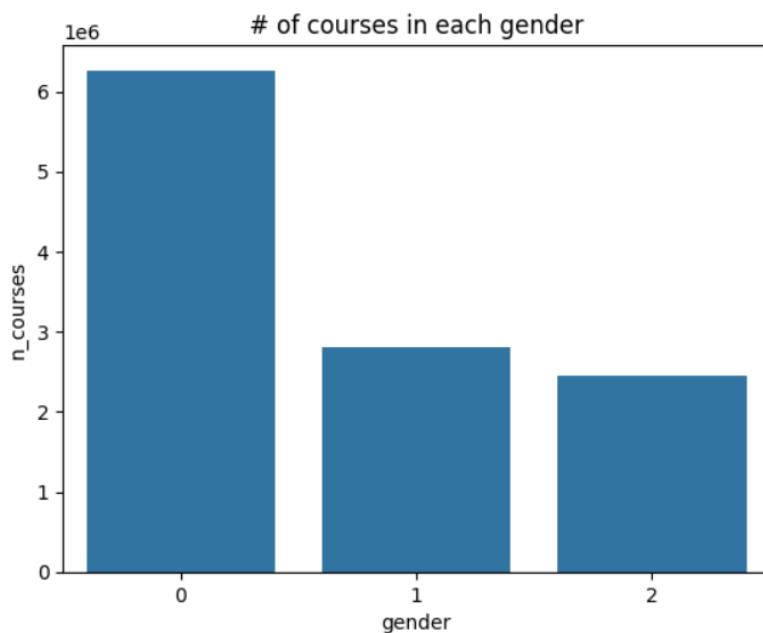
Hình 5.19 Value counts của trường gender của User.

Theo Hình 5.19, trường gender có 2 giá trị nhiều là 232, 3 và có rất ít hàng có gender là null, vì vậy ta sẽ bỏ đi những hàng chứa các giá trị này.

Số lượng users trong từng giới tính khá tương đồng (Hình 5.20). Số lượng khóa học của nhóm giới tính thứ 0 nhiều vượt trội so với 2 giới tính còn lại (Hình 5.21).



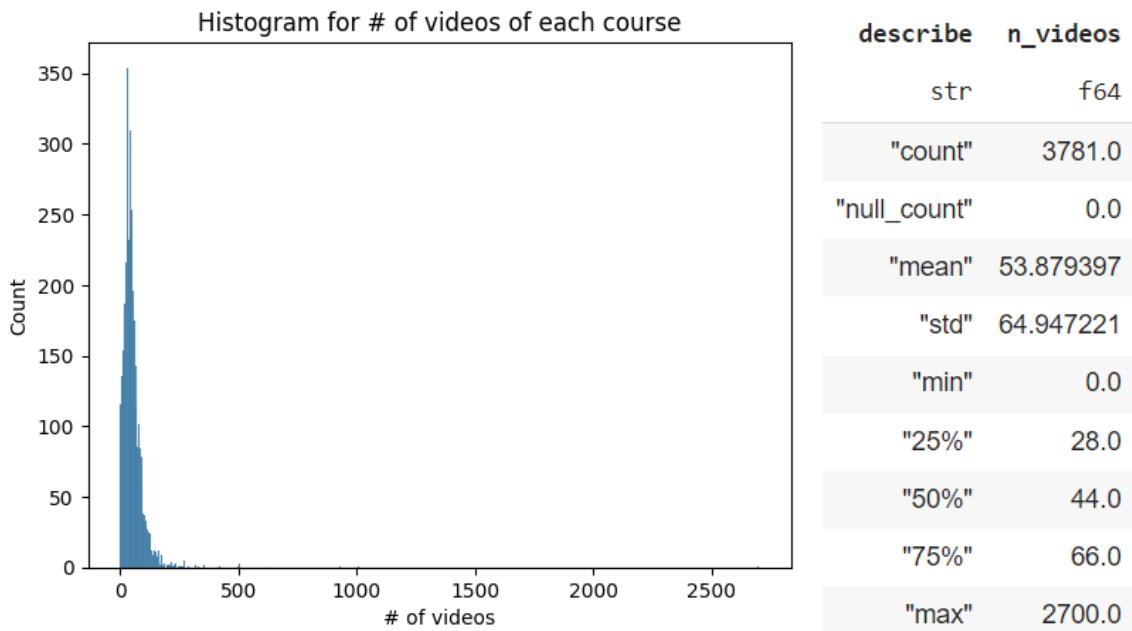
Hình 5.20 Biểu đồ thể hiện số lượng User ứng với từng giới tính.



Hình 5.21 Biểu đồ thể hiện tổng số lượng khóa học đăng ký của mỗi nhóm giới tính

#### 5.2.2.1.6 Video

Ở đây, ta sẽ tập trung vào danh sách các video của khóa học trong trường resource của course.json. Do trường resource chứa 2 loại tài nguyên là video và exercise, nên ta cần tách video ra để dễ dàng thống kê. Sau đây là một số thống kê:



*Hình 5.22 Histogram thể hiện số lượng videos của mỗi khóa học(trái) và bảng thống kê mô tả tương ứng (phải).*

Số videos nhiều nhất trong 1 khóa học là 2700. Số videos trung bình trong 1 khóa học là 53.88. Có một số khóa học không có video nào. Do videos có thể chỉ là thông tin bổ trợ cho bài toán, nên nhóm sẽ không xóa đi các khóa học không có videos nào. Tuy nhiên, các khóa học có thể chứa các video không tồn tại, nên chúng ta sẽ xét tiếp đến video.json và vid\_ccid.txt.

describe	ccid	name	start	end	text
str	str	str	str	str	str
"count"	"59581"	"59581"	"59581"	"59581"	"59581"
"null_count"	"0"	"0"	"0"	"0"	"0"

*Hình 5.23 Bảng thống kê mô tả của video.json*

video.json chứa 59581 hàng với các trường: ccid, name, start, end, text. Trong đó, trường start, end có thể được sử dụng để tính tổng thời lượng của từng video. Lưu ý, ccid khác với video\_id. Hiểu 1 cách đơn giản, khi ccid được trình chiếu tại một môn học nào đó thì nó mới là video\_id. Do đó, 1 video\_id sẽ tương ứng với 1 ccid, và 1 ccid sẽ tương ứng với nhiều video\_id, liên kết của chúng được lưu trữ trong vid\_ccid.txt.

vid\_ccid.txt chứa 2798892 hàng. Ta tiến hành lọc bỏ các liên kết không hợp lệ (video\_id không tồn tại trong resource của course.json). Kết quả thống kê cho thấy, trong vid\_ccid.txt, chỉ có 7% liên kết có ý nghĩa (video\_id tồn tại trong resource của course.json); có tới 63% ccid không tồn tại trong video.json (xét ccid unique).

### 5.2.2.2 Phân tích thống kê

Nhóm thực hiện kiểm định ANOVA để so sánh trung bình số lượng khóa học đăng ký giữa các nhóm giới tính.

Bộ dữ liệu bao gồm ba loại giới tính khác nhau được đánh theo số thứ tự 0, 1, 2. Sau đó nhóm gọi hàm scipy.stats.f\_oneway(g0, g1, g2) để thực hiện kiểm định ANOVA, kết quả trả về giá trị thống kê f và p-value.

```
[ ] g_nc = user_df.select('gender', 'course_order') \
    .with_columns(pl.col('course_order').list.len()) \
    .rename({'course_order': 'n_courses'}) \n\n
g0 = g_nc.filter(pl.col('gender') == 0).select('n_courses')
g1 = g_nc.filter(pl.col('gender') == 1).select('n_courses')
g2 = g_nc.filter(pl.col('gender') == 2).select('n_courses')\n\n
f, p = scipy.stats.f_oneway(g0, g1, g2)
print(f'Thống kê f : {f}')
print(f'p-value : {p}')
```

➡ Thống kê f : [26606.91252138]  
p-value : [0.]

Hình 5.24 Thực hiện kiểm định phương sai ANOVA để xem số lượng khóa học đăng kí có bị phụ thuộc vào nhóm giới tính hay không.

Với việc p-value < 0.05, ta có thể kết luận rằng: “Có sự khác nhau với số lượng khóa học đăng kí giữa 3 nhóm giới tính”.

### 5.2.2.3 Khai phá tri thức

Tại mục 5.2.2.1, nhóm đã thấy được bộ dữ liệu có số lượng user rất nhiều, tuy nhiên số lượng user đăng kí các khóa học lại khá ít (Hơn 75% user có số khóa học đăng kí

$\leq 2$ ). Vì vậy, nhóm sẽ lọc bỏ và chỉ lấy số lượng user đã đăng ký từ 5 khóa học trở lên.

Sau đó, nhóm thực hiện một số bước khai phá dữ liệu và tìm ra được các quy tắc kết hợp từ dữ liệu các khóa học đã được đăng ký bởi người dùng, sau đó dựa vào những quy tắc đã tìm được để dự đoán các khóa học mà một học viên có thể quan tâm. Các bước khai phá tri thức được thực hiện cụ thể như sau:

- Sử dụng thuật toán FP-Growth [20] để tìm được tập các khóa học phổ biến với min\_support là 0.01.
- Sau đó sử dụng hàm association\_rules từ thư viện mlxtend.frequent\_patterns để tạo ra được các quy tắc kết hợp từ tập các khóa học phổ biến.

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction	zhangs_metric
0	(948410)	(696700)	0.062076	0.103303	0.010022	0.161453	1.562902	0.003610	1.069346	0.384002
1	(696700)	(948410)	0.103303	0.062076	0.010022	0.097018	1.562902	0.003610	1.038697	0.401657
2	(948410)	(697791)	0.062076	0.139220	0.011482	0.184968	1.328602	0.002840	1.056130	0.263698
3	(697791)	(948410)	0.139220	0.062076	0.011482	0.082474	1.328602	0.002840	1.022232	0.287331
4	(948410)	(629559)	0.062076	0.119984	0.014332	0.230875	1.924216	0.006884	1.144179	0.512097
...	...	...	...	...	...	...	...	...	...	...
1771	(916828)	(735123)	0.077565	0.033061	0.010670	0.137569	4.161018	0.008106	1.121178	0.823553
1772	(758208)	(679390)	0.031045	0.078778	0.010767	0.346821	4.402512	0.008321	1.410367	0.797619
1773	(679390)	(758208)	0.078778	0.031045	0.010767	0.136674	4.402512	0.008321	1.122352	0.838948
1774	(758208)	(916828)	0.031045	0.077565	0.010183	0.328013	4.228900	0.007775	1.372698	0.787995
1775	(916828)	(758208)	0.077565	0.031045	0.010183	0.131285	4.228900	0.007775	1.115389	0.827735

1776 rows x 10 columns

Hình 5.25 Một phần của tập luật kết hợp

Từ tập quy tắc này, nhóm xây dựng một function nhỏ có input là khóa học, tập luật; output là các khóa học thường đi kèm. Ví dụ minh họa như sau:

```
# Giả định `new_student_courses` là một danh sách các khóa học mà một học viên mới muốn đăng ký
new_student_courses = [746997]

# Dự đoán course_order cho học viên mới
predicted_order = predict_course_order(new_student_courses, rules)

print("Recommended course order:", predicted_order)
```

⇒ Recommended course order: {697018, 782555, 696679}

Hình 5.26 Các khóa học thường xuất hiện cùng với khóa học 746997

#### 5.2.2.4 Làm sạch dữ liệu

Đầu tiên, chúng tôi tiến hành xử lý dữ liệu trùng lặp ở các bảng course, course-chool, teacher, course-teacher, course-concept, user. Kết quả phát hiện được số mẫu dữ liệu bị trùng lặp ở từng bảng như sau:

- course: Số dòng bị trùng lặp là 0, Số ID bị trùng lặp là 0.
- course-chool: Số dòng bị trùng lặp là 0.
- teacher: Số dòng bị trùng lặp là 0, Số ID bị trùng lặp là 0.
- course – teacher: Số dòng bị trùng lặp là 22281.
- course-concept: Số dòng bị trùng lặp là 0
- user: Số dòng bị trùng lặp là 0, Số ID bị trùng lặp là 0.

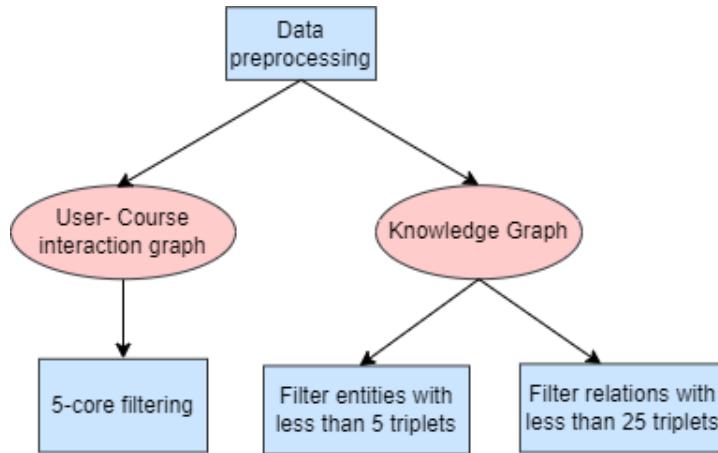
Đối với các mẫu dữ liệu bị trùng lặp, nhóm tiến hành xóa bỏ các mẫu giống nhau này và giữ lại một mẫu duy nhất.

#### 5.2.3 Feature selection

Trong phần này, nhóm sẽ tìm các thuộc tính có thể của khóa học để đưa vào Knowledge graph. Có 4 thuộc tính của khóa học được chọn: school, teacher, concept, field. Bên cạnh đó, nhóm đã thử kiểm tra xem liệu có thể sử dụng mức độ hoàn thành khóa học dựa trên thời gian xem video của từng user hay không. Mức độ này được tính theo công thức  $\frac{\text{tổng thời gian xem các video của 1 khóa học của 1 người dùng}}{\text{tổng thời gian video của khóa học đó}}$ . Tuy nhiên, lượng thông tin này rất ít. Để có được thông tin về tổng thời gian video của 1 khóa học, ta cần ánh xạ video\_id sang ccid (nhiều video\_id sẽ tương ứng với 1 ccid), sau đó lấy thông tin về thời gian của từng video. Nhưng đánh giá cho thấy, trong video\_id-ccid.txt có đến 63% ccid không tồn tại trong video.json (file chứa thông tin thời gian video). Vì vậy, nhóm sẽ không sử dụng thông tin thời gian này để tạo đặc trưng.

#### 5.2.4 Data preprocessing

Nhóm sẽ lọc bỏ đối tượng có liên kết ít để đảm bảo đồ thị collaborative knowledge graph không bị thừa thớt, đảm bảo chất lượng của bộ dữ liệu huấn luyện, đánh giá.



Hình 5.27 Sơ đồ phân rã của quy trình tiền xử lý dữ liệu

Đối với user-course bipartite (interaction) graph, nhóm sử dụng 5-core filtering, nghĩa là sẽ lọc bỏ những user đăng ký ít hơn 5 khóa học và những khóa học được đăng ký bởi ít hơn 5 user. Sau khi lọc, số lượng liên kết trong đồ thị đã giảm từ 11523022 xuống 7470942; số lượng user, khóa học còn lại lần lượt là 373351, 3118. Nhưng do không đủ tài nguyên tính toán, nhóm sẽ chỉ giữ lại 100000 user ngẫu nhiên rồi lọc lại như trên. Kết quả cuối cùng như bảng sau:

	Số lượng
User-course interactions	1992150
Users	99969
Courses	2831

Bảng 5.1 Bảng thống kê số lượng của từng thực thể sau khi xử lý dữ liệu

Đối với Knowledge graph, nhóm lọc bỏ những entity có số lượng triplet (course, relation, entity) ít hơn 5 và những relation có số lượng triplet ít hơn 25. Kết quả cuối như bảng sau:

Relation	# of triplets	# of unique entities
course.school	2296	144
course.concept	63680	7162
course.teacher	262	40
course.field	471	41
<b>Tổng</b>	<b>66709</b>	<b>7387</b>

Bảng 5.2 Bảng thống kê chi tiết từng loại liên kết sau khi thực hiện N-core filtering

Thực chất, trong paper KGAT [3], để đảm bảo chất lượng dữ liệu huấn luyện, nhóm nghiên cứu đã sử dụng 10-core filtering, lọc bỏ entity ít hơn 10 triplets, lọc bỏ relation ít hơn 50 triplets. Nhưng do paper KGAT [3] thực hiện gợi ý các sản phẩm như sách, bài hát,... có thời gian hoàn thành ngắn hơn nhiều so với việc hoàn thành một khóa học, nên việc giảm tiêu chí để lọc là điều cần thiết.

### 5.2.5 Data splitting

Nhóm chia dữ liệu theo chiến lược leave-one-out. Với mỗi user, nhóm giữ khóa học cuối cùng làm test, khóa học kế cuối làm val, các khóa học còn lại làm train.

## 5.3 Độ đo đánh giá

Để đánh giá performance của mô hình, nhóm sử dụng cách đánh giá leave-one-out (như đã đề cập trên phần data splitting), được sử dụng rộng rãi trong [21] [22] [23]. Các mô hình được đánh giá theo chiến lược “randomly sampling negative item”, nghĩa là bắt cặp khóa học ground truth trong tập test với 100 khóa học mà người dùng chưa đăng ký được lấy mẫu ngẫu nhiên.

Về độ đo đánh giá, chúng tôi sử dụng 2 độ đo đánh giá phổ biến: Recall và NDCG.

**Recall** là số liệu đo lường phần trăm các item (ở đây là khóa học) có liên quan đã được đề xuất, trong số tất cả các item liên quan có sẵn trong hệ thống. Recall cao hơn cho thấy hệ thống có thể đề xuất tỷ lệ các mặt hàng có liên quan cao hơn. Nói cách khác, nó đo lường mức độ hoàn thiện của hệ thống trong việc đề xuất tất cả các mục liên quan cho người dùng.

$$\text{Recall}@K = \frac{\text{Số items được khuyến nghị và có liên quan trong top} K}{\text{Số items liên quan trong top} K}$$

Trong đó,  $K$  là số items trong top đầu danh sách,  $@K$  cho thấy ta đang xét thang đo trên top- $K$  items

**NDCG** (Normalized Discounted Cumulative Gain) là một thang đo chất lượng xếp hạng. Nó so sánh xếp hạng được đề xuất với xếp hạng lý tưởng mà trong đó tất cả các items phù hợp đều ở top của danh sách. NDCG được tính theo công thức:

$$NDCG@K = \frac{DCG@K}{IDCG@K}$$

Tương tự như Recall, @K cho biết ta đang xét trên top-K items. Còn DCG là viết tắt của Discounted Cumulative Gain và được tính theo công thức:

$$DCG@K = \sum_{k=1}^K \frac{rel_i}{\log_2(i+1)}$$

$rel_i$  là điểm số liên quan (relavance score) của một item tại vị trí  $i$ .  $rel_i = 0$  khi item đó không liên quan và  $rel_i = 1$  nếu item đó liên quan. Công thức DCG@K sử dụng hàm log ở mẫu để đảm bảo sự khác biệt giữa vị trí 1 và 2 với vị trí 2 và 3,... Điều này phản ánh những items có xếp hạng cao sẽ ảnh hưởng đáng kể đến trải nghiệm của người dùng và có sự sụt giảm đáng kể khi item này nằm phía dưới trong danh sách. Tiếp đến, IDCG là viết tắt của Ideal Discounted Cumulative Gain, đại diện cho DCG tối đa có thể đạt được với cùng một bộ điểm số liên quan nhưng theo thứ tự xếp hạng hoàn hảo. Tóm lại, NDCG có thể nhận giá trị từ 0 đến 1. NDCG bằng 1 khi danh sách được sắp xếp hoàn hảo theo mức độ liên quan. NDCG bằng 0 khi trong top-K không chứa các items liên quan. NDCG nằm giữa 0 và 1 trong các trường hợp còn lại, giá trị càng cao thì hệ thống khuyến nghị càng tốt.

Để đánh giá hệ thống, nhóm dùng Recall, NDCG với  $K = 1, 5, 10$  tương tự như [1], [24]. Trong trường hợp  $K = 1$ , thì Recall và NDCG sẽ như nhau nên chỉ có Recall@1 bảng thống kê kết quả.

#### 5.4 Kích bản thực nghiệm theo thời gian trên kiến trúc dữ liệu lớn

Như đã nói ở mục 5.2.5, nhóm sẽ chia bộ dữ liệu thành 3 phần train, val, test. Với test chứa khóa học gần nhất mà người dùng đăng ký, val chứa khóa học kế cuối, train chứa các khóa học còn lại. Đây cũng chính là kích bản thực nghiệm theo thời gian trên kiến trúc dữ liệu lớn. Sau khi huấn luyện các mô hình, nhóm sẽ đánh giá trên tập test.

## 5.4.1 Thông số chi tiết của các phương pháp

### 5.4.1.1 Content-based filtering

#### 5.4.1.1.1 Feature 1

- Sử dụng trường name, about, field của khóa học. Dùng rdrsegmenter để gôm các tiếng lại thành từ tiếng Việt cho các trường name, about.
- TFIDF để chuyển thông tin thành vector
- Tính độ tương đồng giữa khóa học và người dùng bằng cosine

#### 5.4.1.1.2 Feature 2:

- Sử dụng school, concept của khóa học
- TFIDF để chuyển thông tin thành vector
- Tính độ tương đồng giữa khóa học và người dùng bằng cosine

#### 5.4.1.1.3 Feature 3:

- Sử dụng school, concept của khóa học
- Multi-hot encoding để chuyển thông tin thành vector
- Tính độ tương đồng giữa khóa học và người dùng bằng cosine

### 5.4.1.2 Matrix factorization (MF)

- Course embedding dimension = User embedding dimension = 64
- Lambda của l2 regularization =  $10^{-5}$
- Train batch size = 1024
- Test batch size = 10000
- Optimizer: Adam
- Learning rate = 0.0001
- Số epoch = 300
- Evaluate every 5 epochs
- Early stopping steps = 5 với Recall@10 → Thực nghiệm dừng tại epoch 140

#### 5.4.1.3 Factorization machine

- Course embedding dimension = User embedding dimension = 64
- Lambda của l2 regularization =  $10^{-5}$
- Train batch size = Test batch size = 1024
- Optimizer: Adam
- Learning rate = 0.0001
- Số epoch = 1000
- Evaluate every 5 epochs
- Early stopping steps = 10 với Recall@10

Thực hiện 2 thí nghiệm:

- Sử dụng giới tính người dùng → Dừng tại epoch 130.
- Không sử dụng giới tính người dùng → Dừng tại epoch 190.

#### 5.4.1.4 Neural Factorization Machine

- Course embedding dimension = User embedding dimension = 64
- Gồm 3 hidden layers nối tiếp nhau, mỗi hidden layer là một linear layer với activation là ReLU, được sau bởi 1 lớp dropout với dropout rate là 0.1. Ba hidden layers có output dimension lần lượt là 64, 32, 16.
- Lambda của l2 regularization =  $10^{-5}$
- Train batch size = Test batch size = 1024
- Optimizer: Adam
- Learning rate = 0.0001
- Số epoch = 1000
- Evaluate every 5 epochs
- Early stopping steps = 5 với Recall@10

Thực hiện 2 thí nghiệm:

- Sử dụng giới tính người dùng → Dừng tại epoch 90
- Không sử dụng giới tính người dùng → Dừng tại epoch 60

#### 5.4.1.5 Knowledge Graph Attention Network (KGAT)

- Course embedding dimension = User embedding dimension = Relation embedding dimension = 64
- Train batch size:
  - o Collaborative batch size = 1024
  - o Knowledge graph batch size = 2048
- Test batch size = 256
- Laplacian type: random walk
- Aggregation type: bi-interaction
- Độ sâu là 3 với hidden dimension là 64, 32, 16.
- Lambda của knowledge graph's l2 regularization =  $10^{-5}$
- Lambda của collaborative's l2 regularization =  $10^{-5}$
- Optimizer: Adam
- Learning rate = 0.0001
- Số epoch = 1000
- Evaluate every 5 epochs
- Early stopping steps = 10 với Recall@10

Thực hiện 2 thí nghiệm:

- Train từ đầu: Trong lần thực nghiệm đầu tiên, do thời gian huấn luyện vượt quá mức cho phép của Kaggle nên chỉ lưu được epoch 98. Sau đó, nhóm chạy thực nghiệm tiếp với parameters đã có. Trong lần thực nghiệm này, early stopping tại epoch 55 . Như vậy, tổng số epoch là 153 epoch.
- Train từ pretrained của MF: early stopping tại epoch 65

#### 5.4.2 Đánh giá kết quả thực nghiệm

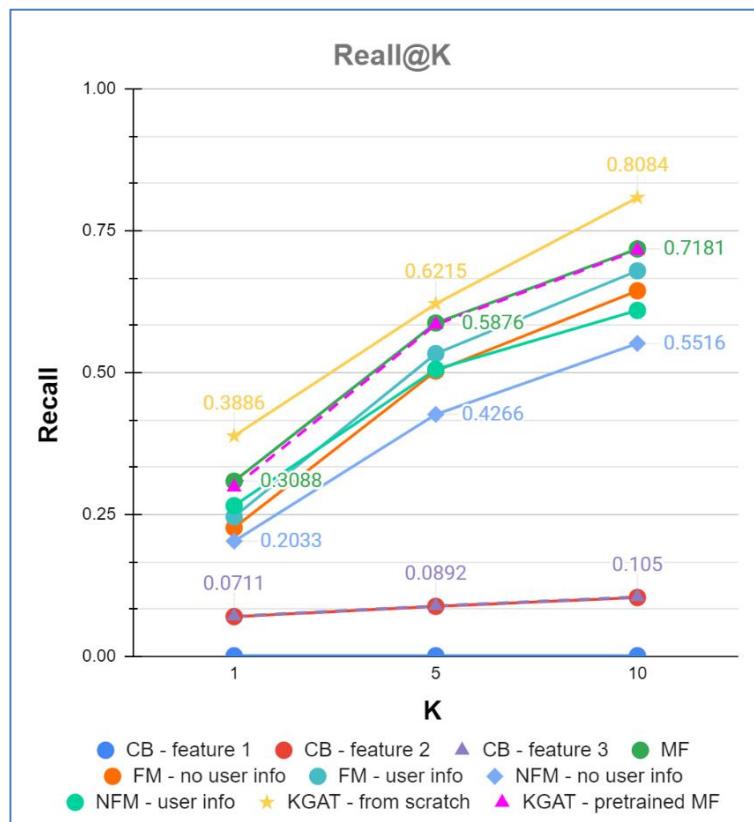
Sau đây là kết quả của 10 thí nghiệm:

Method	Metric	Recall			NDCG	
		@1	@5	@10	@5	@10

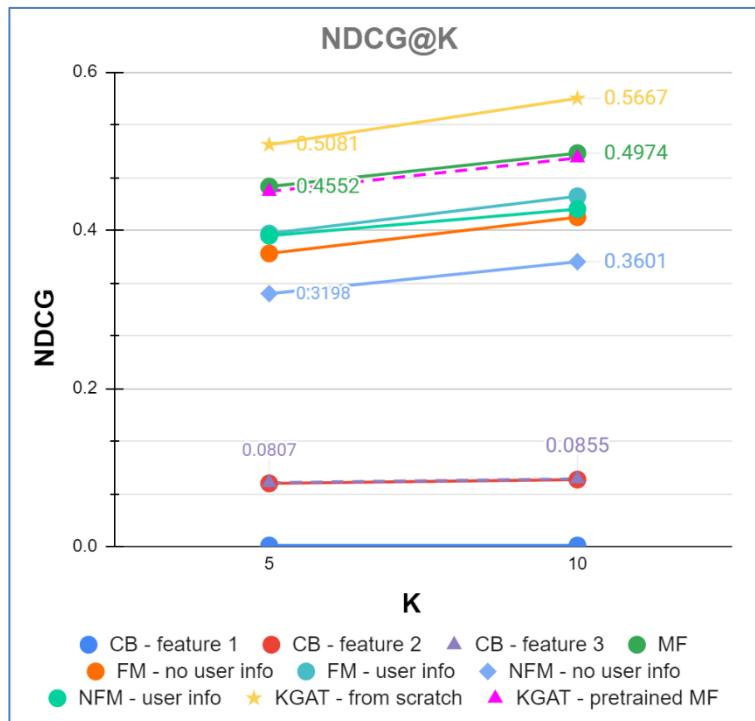
Content-based	Feature 1	0.0012	0.0013	0.0013	0.0013	0.0013
	Feature 2	0.0703	0.0884	0.104	0.0798	0.0846
	Feature 3	0.0711	0.0892	0.105	0.0807	0.0855
MF		<u>0.3088</u>	<u>0.5876</u>	<u>0.7181</u>	<u>0.4552</u>	<u>0.4974</u>
FM	Without user info	0.2268	0.503	0.6444	0.3706	0.4164
	With user info	0.2462	0.5336	0.6795	0.3958	0.4429
NFM	Without user info	0.2033	0.4266	0.5516	0.3198	0.3601
	With user info	0.2657	0.5059	0.6098	0.393	0.4266
KGAT	From scratch	<b>0.3886</b>	<b>0.6215</b>	<b>0.8084</b>	<b>0.5081</b>	<b>0.5667</b>
	With pretrained MF embeddings	0.2985	0.5849	0.7157	0.449	0.4914

Bảng 5.3 Bảng thống kê 10 kết quả nhóm đã thực nghiệm. Mầu đỏ là kết quả tốt nhất. Mầu lam là kết quả tốt thứ 2.

Để dễ quan sát, nhóm đã vẽ 2 đồ thị sau (Phóng to để dễ dàng quan sát hơn):



Hình 5.28 Reall@K của 10 thực nghiệm



Hình 5.29 NDCG@K của 10 thực nghiệm

Thực nghiệm cho thấy, phương pháp cho kết quả tốt nhất là KGAT from scratch với Recall@K, NDCG@K cao vượt trội. Điều này thể hiện sự hiệu quả của khả năng mô hình hóa các liên kết bậc cao trong đồ thị tri thức với Graph attention network. Phương pháp cho kết quả top 2 là MF, gần xấp xỉ với KGAT được huấn luyện tiếp từ pretrained của MF. Thấp nhất là content-based filtering với các feature khác nhau.

Việc KGAT from scratch cho kết quả cao hơn so với KGAT from pretrained cho ta thấy, việc dùng pretrained của MF để khởi tạo tham số ban đầu đã khiến KGAT rơi vào điểm tối ưu cục bộ không tốt.

Phương pháp content-based filtering cho kết quả thấp nhất bởi vì cách thiết kế feature vẫn còn đơn giản, chưa áp dụng được việc tối ưu hàm mất mát để học tham số mà chủ yếu chỉ dựa trên việc tính cosine similarity giữa các hand-made feature. Đối với feature 1, việc sử dụng các trường thông tin như name, about, field cho kết quả rất thấp, có thể bởi vì các trường about, name của khóa học mà user đăng ký có rất ít các từ trùng với khóa học mà ta thực sự mong muốn khuyến nghị cho họ. Ngoài ra, có rất ít khóa học chứa thông tin field. Trong thống kê ở Bảng 5.2, chỉ có 471 triplets của

quan hệ course.field, trong khi tổng số khóa học là 2831. Thấy được hạn chế của feature 1, nhóm đã thực nghiệm trên feature 2, feature 3 với thông tin concept và school. Theo thống kê thì quan hệ course.school có 2296 triplets với 144 trường duy nhất, còn quan hệ course.concept có 63680 triplets với 7162 concept duy nhất. Thực nghiệm cho thấy feature 2, 3 cho kết quả tốt hơn feature 1. Tuy nhiên, nhìn chung thì kết quả này vẫn khá thấp so với các phương pháp khác. Nguyên nhân là vì khi khuyến nghị khóa học cho 1 user cụ thể, phương pháp vẫn chưa tận dụng được thông tin các khóa học mà user khác đăng ký mà chỉ đơn thuần dựa trên các khóa học mà user đó đăng ký.

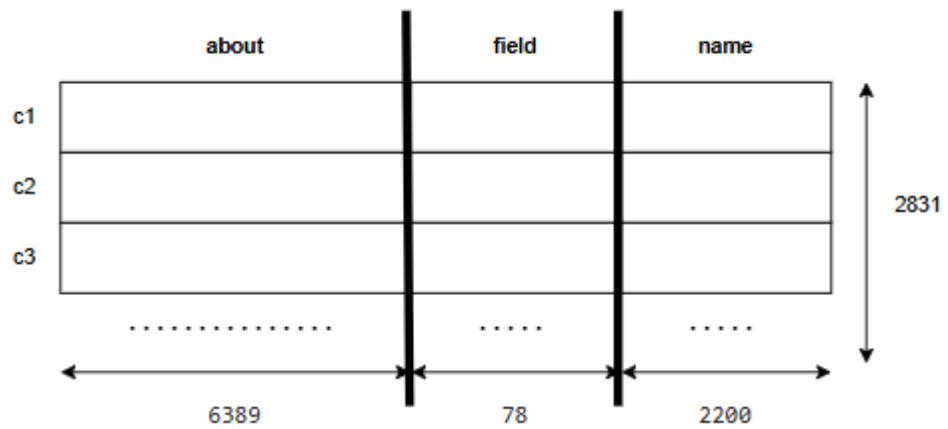
Mặc dù các phương pháp FM, NFM là các phương pháp cải tiến của MF, cho phép sử dụng thêm các thông tin phụ của khóa học để khuyến nghị, nhưng trong thực nghiệm, kết quả của chúng lại thấp hơn so với MF. Điều này cho thấy phương pháp tổng hợp các thông tin bổ trợ của chúng không hiệu quả trên MOOCubeX. Ngoài ra, các thí nghiệm của FM, NFM cũng cho thấy thông tin về giới tính người dùng cũng góp phần đáng kể vào hiệu suất mô hình khi giúp cải thiện Recall@10 từ 0.6444 lên 0.6795, NDCG@10 từ 0.4164 lên 0.4429 cho FM; cải thiện Recall@10 từ 0.5516 lên 0.6098, NDCG@10 từ 0.3601 lên 0.4266 cho NFM.

## 5.5 Trực quan hóa cách biểu diễn tri thức cho từng mô hình thực nghiệm

Đối với cách tổ chức dữ liệu của từng mô hình, chắc chắn có sự khác nhau giữa các phương pháp vì mỗi mô hình đều thực hiện theo một cách thức khác nhau để rút trích ra được từng đặc trưng riêng biệt của bộ dữ liệu.

### 5.5.1 Content-based Filtering

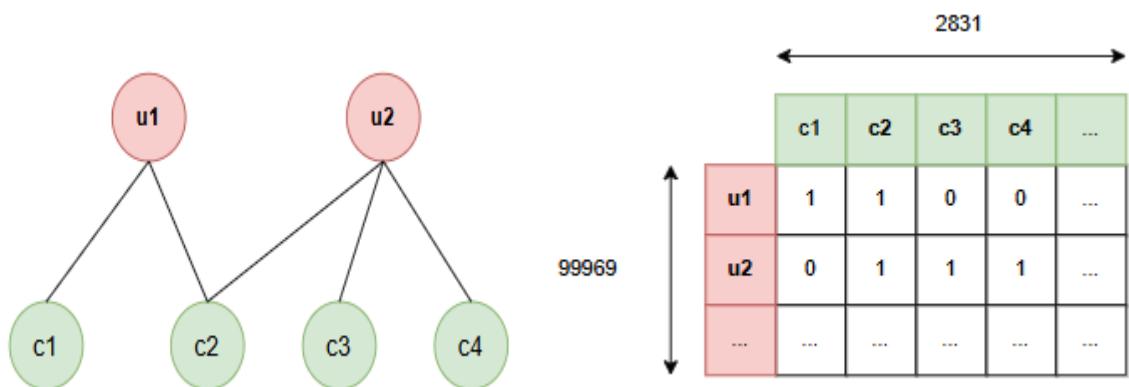
Đầu tiên là cách biểu diễn tri thức của mô hình Content-based filtering được chúng tôi tổ chức lại như sau:



Hình 5.30 Hình minh họa cách thức biểu diễn một ma trận Đặc trưng khóa học theo phương pháp Content-based filtering

### 5.5.2 Matrix Factorization – MF

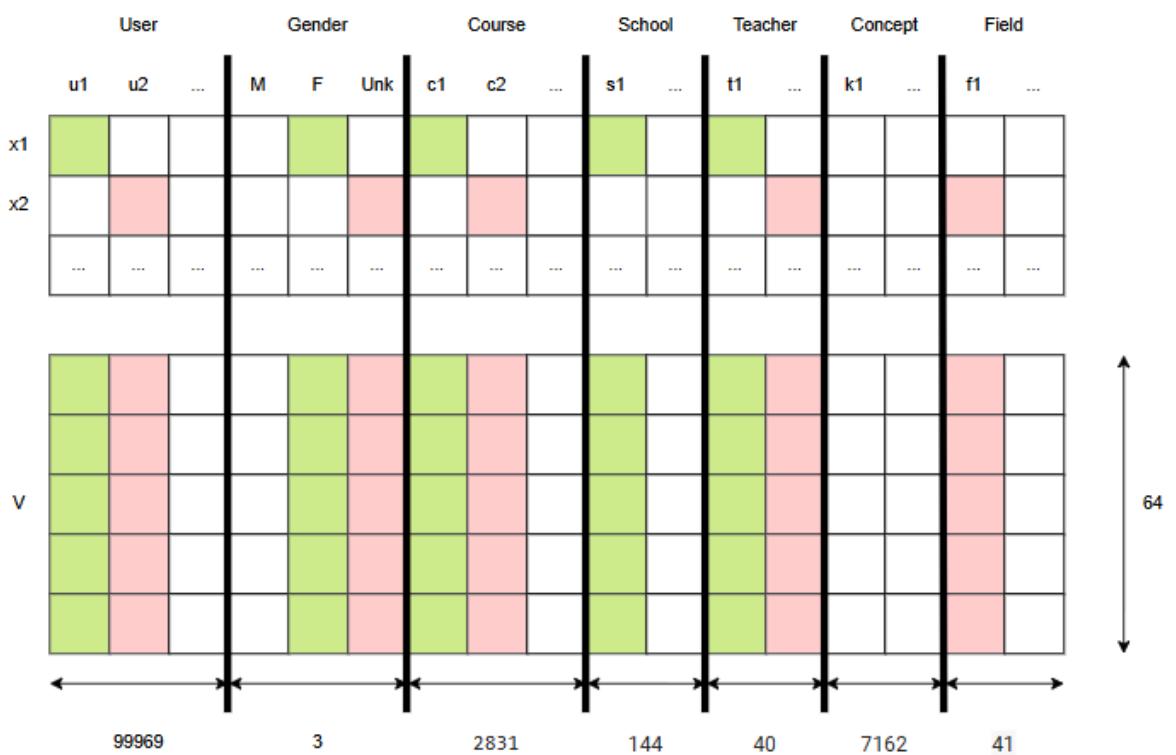
Mô hình MF là mô hình cơ sở của sự hình thành và phát triển của mô hình FM, nên nếu so về các thức biểu diễn tri thức của mô hình MF, chắc chắn sẽ còn nhiều hạn chế hơn so với FM. Sau đây là các biểu diễn tri thức của mô hình MF:



Hình 5.31 Hình minh họa cách thức biểu diễn tri thức của MF theo dạng đồ thị (bên trái) và dạng ma trận (bên phải)

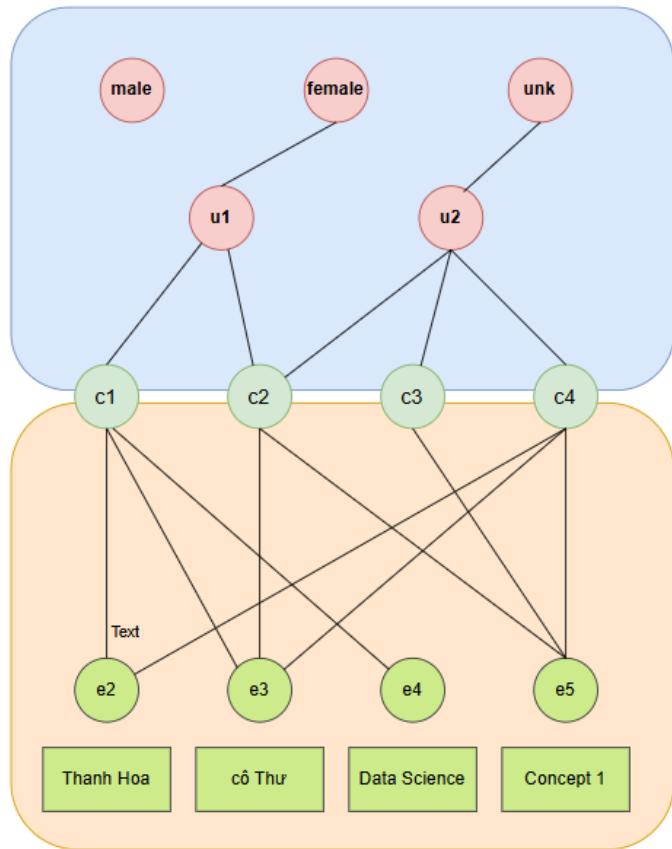
### 5.5.3 Factorization Machine và Neural Factorization Machine

Đối với mô hình (Neural) FM, họ lại xây dựng và tổ chức dữ liệu theo một ma trận có chứa các vector embeddings của cả những thông tin rõ ràng và cả những thông tin ngầm định, cách thức tổ chức dữ liệu để phục vụ mô hình FM được chúng tôi thực hiện lại như sau:



*Hình 5.32 Hình minh họa cách thức tổ chức dữ liệu dưới dạng ma trận theo FM*

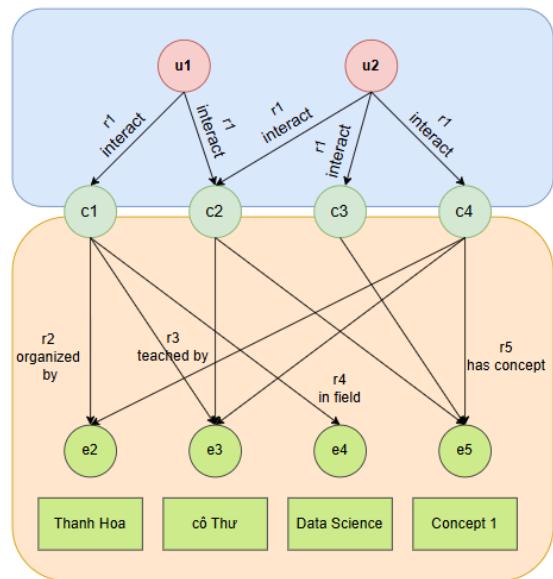
Tuy nhiên, có thể hình dung một cách khái quát hơn các thức tổ chức này có khá nhiều điểm tương đồng với cách thức tổ chức dữ liệu dưới dạng đồ thị, có thể được tái hiện như sau:



Hình 5.33 Hình minh họa cách thức tổ chức dữ liệu dưới dạng đồ thị theo FM

#### 5.5.4 KGAT

Đối với KGAT, họ chủ yếu xây dựng và tổ chức bộ dữ liệu theo một CKG – một đồ thị kiến thức toàn diện để có thể nắm bắt được nhiều hơn nữa các thông tin ngầm định của bộ dữ liệu. Cách thức nhóm chúng tôi theo phương pháp đề xuất của họ như sau:



Hình 5.34 Hình minh họa cách tổ chức dữ liệu thành một CKG

## 6 KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

---

### 6.1 Đánh giá các phương pháp

Mỗi phương pháp đều có ưu và khuyết điểm:

- Content-based filtering:
  - o Ưu điểm:
    - Không cần huấn luyện, suy luận nhanh.
    - Không cần dữ liệu từ người dùng khác. Khi có một xuất hiện khóa học mới chưa được đăng ký, ta hoàn toàn có thể gọi dựa nội dung, thông tin của nó để gợi ý.
    - Khả năng cá nhân hóa cao: Do phân tích sở thích cá nhân của người dùng, bộ lọc dựa trên nội dung có thể đưa ra những đề xuất phù hợp và chính xác hơn.
  - o Khuyết điểm:
    - Performance thấp nếu hand-made feature không hiệu quả
    - Không cần dữ liệu từ người dùng khác cũng chính là khuyết điểm của phương pháp này, vì nó sẽ chỉ gợi ý những khóa học có nội dung tương tự với các khóa user đã đăng ký, thiếu đi tính đa dạng.
- Matrix factorization:
  - o Ưu điểm:
    - Đầu tiên, nó có thể xử lý dữ liệu thưa thớt và không đầy đủ, điều này thường xảy ra đối với xếp hạng mục cũng như việc đăng ký khóa học của người dùng. MF có thể điền vào các giá trị còn thiếu và dự đoán xếp hạng cho các mục hoặc người dùng chưa nhìn thấy.
    - Thứ hai, nó có thể làm giảm chiều và độ phức tạp của dữ liệu, điều này có thể cải thiện hiệu quả và khả năng mở rộng của hệ

thông gợi ý. MF có thể nén một ma trận lớn thành các ma trận nhỏ hơn để nắm bắt thông tin cần thiết và giảm nhiễu.

- Thứ ba, nó có thể khám phá các đặc trưng và mẫu tiềm ẩn không rõ ràng hoặc rõ ràng trong dữ liệu. MF có thể tiết lộ những điểm tương đồng và sở thích tiềm ẩn giữa người dùng và mặt hàng, điều này có thể nâng cao chất lượng và tính đa dạng của các đề xuất

- Nhược điểm:

- Đầu tiên, nó có thể bị overfitting và underfitting, điều này có thể ảnh hưởng đến tính chính xác và tính khái quát của các đề xuất. Overfitting xảy ra khi các vectơ đặc trưng fit quá tốt với dữ liệu và thu được nhiễu hoặc các ngoại lệ, trong khi underfitting xảy ra khi các vectơ đặc trưng fit quá tệ với dữ liệu và bỏ lỡ thông tin quan trọng. Để tránh những vấn đề này, việc phân tích FM cần phải cân bằng sự đánh đổi giữa việc overfitting và việc regularize các vectơ đặc trưng.
- Thứ hai, nó có thể nhạy cảm với việc lựa chọn các tham số, chẳng hạn như số lượng đặc trưng, tốc độ học và regularization term. Các tham số này có thể ảnh hưởng đến hiệu suất và độ hội tụ của thuật toán nhân tử hóa ma trận.
- Thứ ba, nó có thể bị giới hạn bởi các giả định về tính tuyến tính và tính độc lập, những giả định này có thể không đúng đắn với một số dữ liệu hoặc kịch bản. MF giả định rằng xếp hạng là sự kết hợp tuyến tính của các đặc điểm và các đặc điểm này độc lập với nhau. Tuy nhiên, trong một số trường hợp, xếp hạng có thể phụ thuộc vào các đặc điểm phi tuyến tính hoặc tương tác hoặc vào các yếu tố bên ngoài như bối cảnh, thời gian hoặc ảnh hưởng xã hội.

- Thứ 4, không có khả năng mô hình hóa các thông tin hỗ trợ về người dùng và sản phẩm.

- Factorization machine:

o Ưu điểm:

- FM là một phương pháp mở rộng của MF ở đó thông tin về sự tương tác giữa nhiều thành phần thông tin khác nhau được mô hình hóa dưới dạng một biểu thức bậc hai hoặc cao hơn. Thông thường, chỉ các tương tác bậc hai được sử dụng để giảm độ phức tạp tính toán.
- Có khả năng mô hình hóa trên cả đặc trưng thừa thót và dày đặc.
- Có khả năng mô hình hóa các thông tin hỗ trợ về người dùng và sản phẩm.
- FM cũng giải quyết được vấn đề “khởi đầu lạnh” khi một người dùng hoặc sản phẩm chưa hề có tương tác nhưng đã có thông tin riêng về người dùng/sản phẩm đó.

o Khuyết điểm:

- Nó có thể bị overfitting nếu số lượng hệ số quá lớn hoặc dữ liệu quá .
- Nó có thể không nắm bắt được các tương tác đặc trưng phi tuyến hoặc phức tạp, thứ không thể xấp xỉ bởi inner products
- Nó có thể nhạy cảm với việc lựa chọn các siêu tham số, chẳng hạn như regularization term, learning rate.

- Neural Factorization Machine:

o Ưu điểm:

- Neural network trong NFM có thể nắm bắt các mối quan hệ phi tuyến tính phức tạp giữa các đặc trưng, dẫn đến độ chính xác dự đoán tốt hơn so với FM, đặc biệt là đối với dữ liệu phức tạp.

- Tương tự như FM, NFM vượt trội trong việc xử lý dữ liệu thưa thớt, điều này thường gặp trong các hệ thống đề xuất nơi người dùng chỉ có thể tương tác với một phần rất nhỏ các mục.
- NFM có thể được mở rộng quy mô để xử lý các tập dữ liệu lớn một cách hiệu quả nhờ các kỹ thuật tham số hóa hiệu quả của nó.
- NFM cho phép kết hợp nhiều loại đặc trưng, bao gồm dữ liệu phân loại và dữ liệu số mà không cần feature engineering thủ công cần thiết trong các mô hình truyền thống.

○ Khuyết điểm:

- Neural network có độ phức tạp cao hơn khi so với FM. Điều này có thể khiến việc đào tạo và diễn giải kết quả của NFM trở nên khó khăn hơn.
- Đào tạo NFM đòi hỏi nhiều tài nguyên tính toán hơn so với các mô hình đơn giản hơn như FM do kiến trúc mạng thần kinh phức tạp.
- Giống như FM, NFM phụ thuộc rất nhiều vào chất lượng và số lượng dữ liệu để có hiệu suất tối ưu. Dữ liệu không đầy đủ có thể dẫn đến việc overfitting hoặc khai thác kém.
- Có thể không sử dụng hiệu quả các thông tin bổ trợ của người dùng và sản phẩm, dẫn đến cho kết quả thấp hơn so với MF

- Knowlede Graph Attention Network:

○ Ưu điểm:

- Mô hình hóa các kết nối bậc cao trong đồ thị tri thức theo kiểu từ đầu đến cuối
- Sử dụng embedding của các node lân cận để điều chỉnh embedding của 1 node và sử dụng kỹ thuật attention để phân biệt mức độ quan trọng của các lân cận. Điều này giúp tạo các

embedding hiệu quả hơn, từ đó giúp hệ thống khuyến nghị có performance tốt hơn.

- Khuyết điểm: KGAT có độ phức tạp cao nhất trong các phương pháp được sử dụng. Điều này có thể khiến việc đào tạo và diễn giải kết quả của KGAT trở nên khó khăn hơn.

## 6.2 Hướng phát triển tiềm năng

- Dữ liệu:
  - Thu thập thêm các thông tin của người dùng, khóa học để có thể tạo thêm nhiều đặc trưng cho mô hình.
- Mô hình:
  - Sử dụng thêm thông tin giới tính người dùng cho KGAT
  - Thực nghiệm trên các phương pháp khác như BERT4Rec, GRU4Rec [25], TrueLearn [26],...
- Ứng dụng web:
  - Bổ sung các tính năng cần thiết để tạo nên một ứng dụng học tập trực tuyến hoàn chỉnh: gợi ý tên khi nhập tên tìm kiếm người dùng, hỗ trợ phân quyền,...
  - Xây dựng web responsive với các thiết bị di động khác như điện thoại, iPad,...
  - Sử dụng các cơ sở dữ liệu, công cụ tìm kiếm hiệu quả hơn giúp tăng tốc độ truy vấn.
  - Đưa ứng dụng lên cloud computing, tự động hóa toàn bộ quá trình từ ingest data, store data, đến huấn luyện mô hình máy học, triển khai ứng dụng web.
  - Tích hợp tính năng gợi ý vào một ứng dụng học tập trực tuyến hiện có, ví dụ như: XueTangX, Coursera,...

## TÀI LIỆU THAM KHẢO

- [1] Yu, J.; Wang, Y.; Zhong, Q.; Luo, G.; Mao, Y.; Sun, K.; Feng, W.; Xu, W.; Cao, S.; Zeng, K.; et al., “MOOCubeX: A Large Knowledge-centered Repository for Adaptive Learning in MOOCs,” trong *In Proceedings of the 30th ACM International Conference on Information & Knowledge Management*, 2021.
- [2] “XuetangX: Online Courses from Top Universities,” Tsinghua University, [Trực tuyến]. Available: <https://www.xuetangx.com/global>. [Đã truy cập 28th May 2024].
- [3] Xiangnan He, Yixin Cao, Meng Liu, and Tat-Seng Chua, “KGAT: Knowledge graph attention network for recommendation,” trong *KDD 2019*, 2019.
- [4] G.Linden, B. Smith, and J. York, “Amazon.com recommendations: item-to-item collaborative filtering,” *IEEE Internet Computing*, tập 7, số 1, pp. 76 - 80, 22 January 2003.
- [5] Steffen Rendle, Christoph Freudenthaler, Zeno Gantner, and Lars Schmidt-Thieme, “BPR: Bayesian Personalized Ranking from Implicit Feedback,” trong *UAI*, 2009.
- [6] Steffen Rendle, Zeno Gantner, Christoph Freudenthaler, and Lars Schmidt-Thieme, “Fast context-aware recommendations with factorization machines,” trong *SIGIR*, 2011.

- [7] Xiangnan He, Tat-Seng Chua, “Neural Factorization Machines for Sparse Predictive Analytics,” trong *SIGIR*, 2017.
- [8] Petar Velickovic, Guillem Cucurull, Arantxa Casanova, Adriana Romero, Pietro, “Graph Attention Networks,” trong *ICLR*, 2018.
- [9] Welling, Thomas N. Kipf and Max, “Semi-Supervised Classification with Graph Convolutional Networks,” trong *ICLR*, 2017.
- [10] Lin, Yankai, et al., “Learning entity and relation embeddings for knowledge graph completion.,” trong *Proceedings of the AAAI conference on artificial intelligence*, 2015.
- [11] William L. Hamilton, Zhitao Ying, and Jure Leskovec, “Inductive Representation Learning on Large Graphs,” trong *NeurIPS*, 2017.
- [12] Someren, R. Van Meteren and M. Van, “Using Content-Based Filtering for Recommendation,” trong *MLnet/ECML 2000 Workshop*, 2000.
- [13] Rendle, S., “Factorization machines,” trong *ICDM*, 2010.
- [14] S. H. Han, “PyPI,” Python Software Foundation, 14th June 2020. [Trực tuyến]. Available: <https://pypi.org/project/googletrans/>. [Đã truy cập 28th May 2024].
- [15] L. Long, “vietnamese-fullname-generator,” 19th March 2021. [Trực tuyến]. Available: <https://github.com/lhlong/vietnamese-fullname-generator/commits/main/>. [Đã truy cập 28th May 2024].

- [16] Dat Quoc Nguyen, Anh Tuan Nguyen, “PhoBERT: Pre-trained language models for Vietnamese,” trong *Findings of the Association for Computational Linguistics: EMNLP 2020*, 2020.
- [17] Thanh Vu, Dat Quoc Nguyen, Dai Quoc Nguyen, Mark Dras, and Mark Johnson, “VnCoreNLP: A Vietnamese Natural Language Processing Toolkit,” trong *Proceedings of NAACL: Demonstrations*, 2018.
- [18] “ChatGPT,” OpenAI, [Trực tuyến]. Available: <https://chatgpt.com/>. [Đã truy cập 28th May 2024].
- [19] “Gemini,” Google, [Trực tuyến]. Available: <https://gemini.google.com/app>. [Đã truy cập 28th May 2024].
- [20] Han, J., Pei, J., Yin, Y. et al, “Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach,” trong *Data Mining and Knowledge Discovery 8*, 2004.
- [21] Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua, “Neural collaborative filtering,” trong *Proceedings of the 26th international*, 2017.
- [22] McAuley, ] Wang-Cheng Kang and Julian, “Self-attentive sequential recommendation,” trong *Proceedings of 2018 IEEE International Conference on Data Mining (ICDM)*, 2018.
- [23] Wang, Jiaxi Tang and Ke, “Personalized top-n sequential recommendation via convolutional sequence embedding,” trong

*Proceedings of the Eleventh ACM International Conference on Web Search and Data Mining*, 2018.

- [24] Fei Sun, Jun Liu, Jian Wu, Changhua Pei, Xiao Lin, Wenwu Ou, Peng Jiang, “BERT4Rec: Sequential Recommendation with Bidirectional Encoder Representations from Transformer,” trong *C/CM*, 2019.
- [25] Balázs Hidasi, Alexandros Karatzoglou, Linas Baltrunas, and Domonkos Tikk, “Session-based recommendations with recurrent neural networks,” trong *arXiv preprint arXiv:1511.06939*, 2015.
- [26] Sahan Bulathwela, Maria Perez-Ortiz, Emine Yilmaz, and John Shawe-Taylor, “Truelearn: A family of bayesian algorithms to match lifelong learners to open educational resources,” trong *Proceedings of the AAAI Conference on Artificial Intelligence*, 2020.