

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

Hồ Hữu Ngọc - Nguyễn Phát Minh

Hệ Thống Đề Xuất Phim Dựa Trên
Mô Hình Lai Sử Dụng Đồ Thị Và Bộ Mã
Hóa Tự Động

KHÓA LUẬN TỐT NGHIỆP CỬ NHÂN
CHƯƠNG TRÌNH CHÍNH QUY

Tp. Hồ Chí Minh, tháng 06/2023

**TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN**

Hồ Hữu Ngọc - 19120602

Nguyễn Phát Minh - 19120586

**Hệ Thống Đề Xuất Phim Dựa Trên
Mô Hình Lai Sử Dụng Đồ Thị Và Bộ Mã
Hóa Tự Động**

**KHÓA LUẬN TỐT NGHIỆP CỬ NHÂN
CHƯƠNG TRÌNH CHÍNH QUY**

GIÁO VIÊN HƯỚNG DẪN

ThS. Phạm Trọng Nghĩa

Tp. Hồ Chí Minh, tháng 06/2023

Lời cảm ơn

Khoảng thời gian làm khóa luận đối với mỗi sinh viên là một trong những khoảng thời gian quan trọng nhất đối với bất kỳ một sinh viên nào, nó không những đánh dấu những bước trưởng thành mà còn là một dấu mốc quan trọng trước ngưỡng cửa nghề nghiệp của mỗi cá nhân chúng em.

Lời đầu tiên chúng em xin dành lời cảm ơn đặc biệt đến với thầy ThS. Phạm Trọng Nghĩa đã hướng dẫn và dìu dắt chúng em trong suốt quá trình thực hiện đề tài khóa luận tốt nghiệp. Từ những gợi ý và lời khuyên của thầy mà nhóm có thể tư duy sáng tạo để giải quyết những bài toán, những vấn đề, cũng như có những cải tiến, thay đổi phù hợp để có thể nâng cao tri thức đối với đề tài mà nhóm chọn giải quyết. Nhóm em sẽ không thể hoàn thành khóa luận này nếu không có sự động viên, góp ý từ thầy.

Chúng em xin phép cảm ơn các thầy cô đã và đang giảng dạy tại Đại học Khoa học Tự nhiên, ĐHQG Thành phố Hồ Chí Minh, đặc biệt là các thầy cô giảng viên tại khoa Công nghệ thông tin vì đã truyền thụ những kiến thức nền tảng quan trọng và những kinh nghiệm quý báu của quý thầy công trong suốt thời gian theo học tại khoa.

Lời cuối cùng, chúng em xin cảm ơn gia đình đã luôn là chỗ dựa tinh thần, luôn ủng hộ chúng em trên con đường đã chọn. Chúng em cũng rất hạnh phúc vì đã có thể cùng kề vai sát cánh từ thời điểm đầu tiên đến cuối cùng của khóa luận này.



fit@hcmus

TRƯỜNG ĐẠI HỌC KHOA HỌC TỰ NHIÊN
KHOA CÔNG NGHỆ THÔNG TIN

ĐỀ CƯƠNG KHOÁ LUẬN TỐT NGHIỆP

HỆ THỐNG ĐỀ XUẤT PHIM DỰA TRÊN MÔ HÌNH LAI SỬ DỤNG PHƯƠNG PHÁP ĐỒ THỊ VÀ BỘ MÃ HÓA TỰ ĐỘNG

(Hybrid approach for Movie Recommendation Based on Graph and Autoencoder)

1 THÔNG TIN CHUNG

Người hướng dẫn:

– Th.S Phạm Trọng Nghĩa (Khoa Công nghệ Thông tin)

Nhóm sinh viên thực hiện:

1. Hồ Hữu Ngọc (MSSV: 19120602)
2. Nguyễn Phát Minh (MSSV: 19120586)

Loại đề tài: Nghiên cứu

Thời gian thực hiện: Từ 01/2023 đến 07/2023

2 NỘI DUNG THỰC HIỆN

2.1 Giới thiệu về đề tài

Cùng với sự phát triển và bùng nổ của internet, các kênh tìm hiểu về thông tin, giải trí, thương mại điện tử...cũng phát triển nhanh chóng và mạnh mẽ. Giờ đây, có thể tìm kiếm được mọi thứ trên internet, từ tài liệu, sách, truyện, phim, video đến mặt hàng, sản phẩm ...Hệ thống đề xuất (Recommendation system) trong các lĩnh vực thương mại điện tử nói chung và trong ngành giải trí nói riêng đã ra đời và trở thành một trong những lĩnh vực nghiên cứu quan trọng. Với hệ thống này, người dùng có thể dễ dàng xử lý tình trạng quá tải thông tin và đưa ra những nội dung hoặc lời khuyên phù hợp cho từng khách hàng sử dụng dịch vụ. Nói riêng về lĩnh vực giải trí, nhu cầu của con người đang được đề cao và chú trọng, một trong số đó là lĩnh vực phim điện ảnh. Hiện nay, với sự đa dạng của các thể loại, số lượng phim thì việc đề xuất cho người dùng những bộ phim phù hợp với sở thích của họ là vô cùng quan trọng. Với mục đích như trên, nhóm em thực hiện nghiên cứu để tìm ra hệ thống đề xuất các bộ phim có hiệu quả tốt đáp ứng được nhu cầu của người xem và doanh nghiệp trong thực tế.

2.2 Mục tiêu đề tài

Hiện nay, có khá nhiều công trình nghiên cứu về các mô hình đề xuất phim cho người dùng. Nhiều mô hình mới được áp dụng vào thực tế và chất lượng của các mô hình này cũng ngày càng được cải tiến theo thời gian. Tuy nhiên, những mô hình khác nhau đưa lại những hiệu quả và có những ưu nhược điểm khác nhau. Trong khóa luận này, nhóm đưa ra ba mục tiêu quan trọng :

1. Nghiên cứu các thuật toán, mô hình cũ và mới đã được xây dựng trước đây để phục vụ cho bài toán đề xuất các bộ phim khác nhau cho người xem.
2. Chọn lựa thuật toán cũng như mô hình phù hợp sao cho đạt được hiệu quả

tốt khi chạy thử trên bộ dữ liệu Movielens[1], sau đó cố gắng thực hiện một vài cải tiến nhằm tăng hiệu suất cho mô hình.

3. Đưa ra những hướng nghiên cứu kế tiếp, làm nền tảng cho những nghiên cứu tiếp theo để mô hình có thể đưa vào thực tế, mang lại hiệu quả như mong đợi.

2.3 Phạm vi của đề tài

Trong khóa luận này, ngoài việc trình bày cơ sở lý thuyết về hệ thống đề xuất và các phương pháp lọc thông tin như lọc dựa trên nội dung, lọc cộng tác, lọc kết hợp, máy học và cả học sâu trong hệ thống đề xuất phim. Khóa luận sẽ tập trung nghiên cứu về phương pháp Hybrid Graph Recommendation System(GHRS)[2] kết hợp giữa mô hình Graph-Based[3] và Autoencoder[4] trong lĩnh vực học sâu sau đó tiến hành thực nghiệm trên tập dữ liệu thử nghiệm MovieLens để so sánh và đánh giá mức độ hiệu quả của phương pháp này so với một số phương pháp khác đã được trình bày.

2.4 Cách tiếp cận dự kiến

Nhiều nghiên cứu về các hệ thống đề xuất phim đã xuất hiện trong hai thập kỷ qua và bao gồm các dịch vụ có giá trị để tăng doanh thu của các công ty khác nhau. Hầu hết các hệ thống đề xuất hiện tại đều dựa vào cách tiếp cận dựa trên phương pháp lọc nội dung(Content-Based) hoặc cách tiếp cận dựa trên lọc cộng tác(Collaborative-Filtering), máy học, học sâu,... Trong khóa luận này, nhóm chúng em tìm hiểu phương pháp mới đang được nghiên cứu trong những năm gần đây: Sử dụng mô hình dựa trên đồ thị (Graph-Based) để tính toán sự tương đồng trong xếp hạng của người dùng, kết hợp với những thông tin khác của như độ tuổi, giới tính, nghề nghiệp... Sau đó, bằng việc sử dụng ưu điểm trích xuất đặc trưng của Autoencoder, mô hình sẽ trích xuất được những đặc trưng mới dựa trên tất cả các thuộc tính kết hợp. Từ một số nghiên cứu nhóm đã tìm hiểu, phương pháp này sử dụng với bộ dữ liệu MovieLens cho kết quả đề xuất vượt trội hơn nhiều so

với một số phương pháp đề xuất hiện có.

Xây dựng mô hình đề xuất theo phương pháp Hybrid Graph Recommendation System bao gồm 7 bước:

- **Bước 1:** Trong bước đầu tiên, mô hình cần xây dựng một đồ thị với những người dùng như các nút. Hai người dùng sẽ được kết nối dựa trên những đặc điểm giống nhau của họ. Cạnh kết nối một cặp người dùng là những người có sự tương đồng về xếp hạng các bộ phim.
- **Bước 2:** Trong bước thứ hai, một tập hợp thông tin của người dùng sẽ được trích xuất từ đồ thị.
- **Bước 3:** Trong bước thứ ba, mô hình tiến hành kết hợp thông tin phụ như giới tính và độ tuổi với các đặc trưng dựa trên đồ thị ở bước 2 làm đầu vào cho giai đoạn Autoencoder.
- **Bước 4:** Trong bước này, mô hình áp dụng kỹ thuật Autoencoder để trích xuất các đặc trưng mới và giảm kích thước của dữ liệu.
- **Bước 5:** Trong bước này, mô hình sử dụng các đặc trưng mới được mã hóa bởi Autoencoder để phân cụm người dùng, sử dụng thuật toán K-mean[5] để tạo ra một số lượng nhỏ các nhóm người dùng có sự tương đồng.
- **Bước 6:** Trong bước thứ sáu, mô hình sẽ phân người dùng mới vào cụm thích hợp dựa trên các tính năng được mã hóa và dự đoán xếp hạng các mục mới mà người dùng đó chưa xếp hạng.
- **Bước 7:** Trong bước cuối cùng, mô hình dự đoán xếp hạng của người dùng cho tất cả các mục theo xếp hạng trung bình của cụm và tiến hành đề xuất các bộ phim cho người xem.

2.5 Kết quả dự kiến của đề tài

- Đầu tiên, nhóm có thể hiểu cách sử dụng các phương pháp khác nhau trong mô hình đề xuất phim hiện có và cài đặt thành công một hệ thống để có thể sử dụng trong thực tế.
- Kiểm tra và so sánh được tốc độ và độ chính xác của một số mô hình hiện có trong lĩnh vực đề xuất phim.
- Cuối cùng, nhóm hiểu được điểm mạnh, điểm yếu của mô hình Hybrid Graph Recommendation System và ứng dụng để xây dựng hệ thống đề xuất phim tự động với bộ dữ liệu Movilens.

2.6 Kế hoạch thực hiện

Thời gian	Nội dung	Phân công
01/01 - 15/01	<p>Giai đoạn 1:</p> <ul style="list-style-type: none">• Tìm hiểu tổng quan bài đề xuất phim cho người dùng.• Tìm hiểu về những công trình, nghiên cứu được sử dụng trong lĩnh vực đề xuất phim.• Tìm hiểu các công trình, bài báo nghiên cứu đã thực hiện về đề tài này.• Tìm hiểu các bộ dữ liệu sử dụng cho bài toán này.	Ngọc(1,2) Minh(3,4)


15/01 - 15/02	<p>Giai đoạn 2:</p> <ul style="list-style-type: none"> • Tìm hiểu về mô hình GHRs(Hybrid Graph Recommendation System). • Tìm hiểu về thuật toán Graph-Bases được sử dụng ở trong mô hình. • Tìm hiểu về các kĩ thuật của Autoencoder và các thông số của mô hình. 	Ngọc(1,2) Minh(1,3)
15/02 - 01/04	<p>Giai đoạn 3:</p> <ul style="list-style-type: none"> • Cài đặt và tiến hành chạy thử mô hình GHRs-Hybrid Graph Recommendation system • Đọc hiểu rõ mã nguồn của mô hình. • Hoàn chỉnh cài đặt mô hình. 	Tất cả thành viên của nhóm
01/04 - 01/05	<p>Giai đoạn 4:</p> <ul style="list-style-type: none"> • Phân tích điểm mạnh, điểm yếu của mô hình. • Đề xuất các cải tiến để nâng cao hiệu suất và giảm thiểu chi phí cho giải thuật trên. 	Ngọc(1) Minh(2)
01/05 - 20/05	<p>Giai đoạn 5:</p> <ul style="list-style-type: none"> • Tiến hành cài đặt thử các cải tiến đã được đề xuất. • Phân tích, bàn luận về kết quả thực nghiệm. • Đề xuất hướng nghiên cứu cải tiến cho đề tài. 	Tất cả thành viên của nhóm

20/05 - 20/06	Giai đoạn 6: <ul style="list-style-type: none"> • Viết báo cáo cho khóa luận. • Hoàn thiện mã nguồn của khóa luận. 	Tất cả thành viên của nhóm
20/06 - 01/07	Giai đoạn 7: <ul style="list-style-type: none"> • Chuẩn bị slide, tài liệu cho phần phản biện và thuyết trình trước hội đồng. 	Tất cả thành viên của nhóm


3 Tài liệu

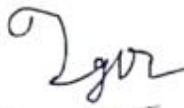
- [1] University of Minnesota “Movielens Dataset 100k, 1M, 10M, 25M Rating”,
<https://grouplens.org/datasets/movielens>.
- [2] Zahra Zamanzadeh Darban, Mohammad Hadi Valipour “Graph-based Hybrid Recommendation System with Application to Movie Recommendation”
<https://arxiv.org/abs/2111.11293>
- [3] Bitnine Global Inc “Special about a graph-based recommendation system”
<https://bitnine.net/blog-graph-database/graph-based-recommendation-system/?ekattemp=1>
- [4] Zhang et al “Autoencoder to a recommender system”, 2019
- [5] Rishabh Ahuja, Arun Solanki, Anand Nayyar “Movie Recommender System Using K-Means Clustering AND K-Nearest Neighbor” <https://icccexplore.icce.org/document/8776969>

XÁC NHẬN
 CỦA NGƯỜI HƯỚNG DẪN
 (Ký và ghi rõ họ tên)


 Phạm Trung Nghĩa

TP. Hồ Chí Minh, ngày... tháng... năm...
 NHÓM SINH VIÊN THỰC HIỆN
 (Ký và ghi rõ họ tên)


 Nguyễn Thái Minh


 Hồ Hữu Ngọc

Mục lục

Lời cảm ơn	i
Đề cương chi tiết	ii
Mục lục	ix
Thuật ngữ sử dụng trong báo cáo	xii
Tóm tắt	xiv
1 Giới thiệu đề tài	1
1.1 Đặt vấn đề	1
1.2 Mục tiêu và phạm vi của đề tài	2
1.2.1 Mục tiêu đề ra	2
1.2.2 Phạm vi đề tài	2
1.3 Đóng góp của đề tài	3
1.4 Khái quát về phương pháp đề xuất	3
1.4.1 Khái quát bài toán	3
1.4.2 Phương pháp sử dụng dữ liệu của hệ thống đề xuất	4
1.5 Cấu trúc khóa luận	5
2 Các công trình liên quan	6
2.1 Các mô hình trong hệ thống đề xuất không cá nhân hóa .	6
2.2 Các mô hình trong hệ thống đề xuất cá nhân hóa	6
2.2.1 Đề xuất dựa trên nội dung	7
2.2.2 Lọc cộng tác	8
2.2.3 Phương pháp lai	10
2.3 Hệ thống đề xuất sử dụng Autoencoder	13
2.4 Hệ thống đề xuất sử dụng mô hình Graph-based	13
2.5 Hệ thống đề xuất có sử dụng thuật toán k-Means	14

3	Phương pháp đề xuất	15
3.1	Cơ sở lí thuyết	15
3.1.1	Graph-based	15
3.1.2	Deep learning	17
3.1.3	Mạng nơ ron nhân tạo (ANN)	17
3.1.4	Autoencoder	20
3.1.5	Elastic net regularization	24
3.1.6	Kmeans	25
3.2	Cơ sở dữ liệu thực nghiệm	30
3.2.1	Bộ dữ liệu sử dụng	30
3.2.2	Phân tích và thống kê cơ bản	34
3.3	Tổng quan phương pháp tiếp cận	42
3.3.1	Hệ Thống Đề Xuất Phim Dựa Trên Mô Hình Lai Sử Dụng Đồ Thị Và Bộ Mã Hóa Tự Động	42
3.3.2	Thuật toán và mã giả để xây dựng hệ thống	43
4	Kết quả thực nghiệm	45
4.1	Môi trường thực nghiệm	45
4.1.1	Tập dữ liệu thực nghiệm	45
4.1.2	Môi trường thực nghiệm	45
4.1.3	Ngôn ngữ và thư viện lập trình	45
4.1.4	Phương pháp kiểm định	46
4.2	Quá trình xây dựng mô hình thực nghiệm	47
4.3	Kết quả mô hình và so sánh	55
4.3.1	Kết quả so sánh với các phương pháp cơ bản	55
4.3.2	Kịch bản cho vấn đề khởi động nguội	55
4.4	Thảo luận	56
5	Kết luận và hướng phát triển	57
5.1	Kết luận	57
5.2	Hướng phát triển	57
	Tài liệu tham khảo	59

Danh sách hình

2.1	Các mô hình trong hệ thống đề xuất	7
3.1	Kiến trúc của mạng nơ ron nhân tạo cơ bản [14]	19
3.2	Kiến trúc mạng của Autoencoder [11]	21
3.3	Đồ thị hàm biến dạng của thuật toán k-Means.	28
3.4	Thông tin người dùng trong USERS	31
3.5	Thông tin xếp hạng trong RATINGS	32
3.6	Thông tin các bộ phim trong MOVIES	33
3.7	Tổng các bộ phim theo xếp hạng	34
3.8	25 Bộ phim được xếp hạng nhiều nhất	35
3.9	Mức độ xem phim khác nhau giữa các độ tuổi	35
3.10	Số lượng người xem theo ngành nghề	36
3.11	Tổng số bộ phim của các thể loại	37
3.12	Số lượng người xem theo giới tính	37
3.13	Phân phối trung bình xếp hạng của mỗi bộ phim	38
3.14	Phân phối trung bình xếp hạng của mỗi người dùng	38
3.15	10 người dùng có số lượng xếp hạng nhiều và ít nhất	39
3.16	Số lượng xếp hạng của 943 người dùng cho các bộ phim	40
3.17	Số lượng 2 người xem trùng xếp hạng trên một số bộ phim	40
3.18	Kiến trúc tổng thể của mô hình	42
4.1	Biểu đồ 943 người dùng với $\alpha = 0.01$	48
4.2	Đặc trưng của 943 người dùng	50
4.3	Distort score on elbow	51
4.4	Distance of Average Silhouette	51
4.5	Số lượng người dùng trong mỗi cụm	52
4.6	Ma trận cụm người dùng - cụm	52
4.7	Ma trận cụm-bộ phim đã dự đoán đầy đủ xếp hạng	53
4.8	Ma trận người dùng - phim đã được dự đoán	54

Danh sách bảng

1.1	Xếp hạng của người dùng u cho bộ phim i	3
4.1	Các độ lỗi ở trong mô hình lai giữa đồ thị và bộ mã hóa tự động của 5 tập dữ liệu	55
4.2	So sánh độ lỗi với các phương pháp cơ bản	55
4.3	Độ lỗi của các phương pháp cho vấn đề khởi động nguội .	56

Thuật ngữ sử dụng trong báo cáo

- Autoencoder: bộ mã hóa tự động.
- Bias: thiên vị của người dùng hoặc sản phẩm .
- Collaborative filtering (CF): lọc cộng tác.
- Content-based filtering: lọc dựa trên nội dung.
- Graph-based: dựa trên dữ liệu đồ thị.
- K-means clustering: phương pháp phân loại nhóm một tập dữ liệu thành K cụm.
- Knowledge-based: lọc dựa trên kiến thức.
- Movielens: Bộ dữ liệu được sử dụng để kiểm nghiệm, bao gồm 100K, 1M, 10M, 20M, 25M chứa từ 100,000 đánh giá đến 25,000,000 đánh giá theo thứ tự.
- Rating (r): đánh giá .
- Sparsity: Dữ liệu thừa thớt hay thiếu dữ liệu.
- User (u): người dùng, trong báo cáo này có thể coi là người xem phim.
- Item (i): sản phẩm, trong báo cáo này có thể xem là bộ phim.

Tóm tắt

Trong thập kỉ vừa qua, việc lọc và đề xuất cho người dùng xem những bộ phim phù hợp với nhu cầu và sở thích của người xem là vấn đề được quan tâm và chú trọng. Với sự phát triển mạnh mẽ của lĩnh vực khoa học máy tính nói chung và Khoa học dữ liệu nói riêng thì chủ đề này càng thu hút nhiều sự chú ý và quan tâm của các chuyên gia, nhà nghiên cứu. So với các phương pháp trước đây là content-based (đề xuất dựa theo nội dung) và collaborative filtering (lọc công tác) thì phương pháp đề xuất phim dựa trên mô hình lai sử dụng đồ thị và bộ mã hóa tự động (Hybrid approach for Movie Recommendation based on Graph and Autoencoder) kết hợp cả hai phương pháp để cải thiện độ chính xác.

Những công trình nghiên cứu liên quan đến đề xuất thường gặp vấn đề về thiếu dữ liệu (Sparsity) và khởi động nguội (cold-start), và phương pháp lai được đề xuất để giải quyết hai vấn đề này. Phương pháp này đạt hiệu quả rất cao trong việc xử lí hai vấn đề trên và đồng thời đạt độ chính xác rất cao nên mục tiêu của nhóm là tái tạo lại phương pháp.

Phương pháp lai này sử dụng dữ liệu trực tiếp từ đánh giá của người dùng và thông tin riêng của người dùng (hoặc các bộ phim) để tạo ra một bảng các đặc điểm so sánh độ giống nhau giữa các người dùng. Sau đó sử dụng bộ mã hóa tự động để rút trích các dữ liệu quan trọng trước khi phân cụm cho người dùng và thực hiện dự đoán.

Trong khóa luận này, nhóm chúng tôi sẽ tìm hiểu khái niệm chung về hệ thống đề xuất cho các bộ phim, sau đó sẽ tập trung vào khảo sát các nhóm thuật toán phổ biến trong các hệ thống gợi ý hiện nay. Cuối cùng, nhóm sẽ thực hiện viết mã thực thi đầy đủ cho phương pháp lai (Hybrid approach for Movie Recommendation based on Graph and Autoencoder) nhóm đang tìm hiểu và thử nghiệm trên các bộ dữ liệu thực tế, qua đó hiểu rõ ưu điểm và nhược điểm của các phương pháp này khi được áp dụng thực tế.

Chương 1

Giới thiệu đề tài

1.1 Đặt vấn đề

Khi người xem truy cập vào một trang web để tìm kiếm và xem những bộ phim thì vấn đề lớn được đặt ra: "Làm sao để trang web có thể hiển thị cho người xem được bộ phim mà họ sẽ thích?" ... Vấn đề quyết định cho câu trả lời chính là trang web cần xây dựng một hệ thống đề xuất phim có hiệu quả cho người xem.

Hiện nay, mọi hệ thống đề xuất có hiển thị quảng cáo nói chung hay những bộ phim trong các trang web nói riêng đều sử dụng hệ thống đề xuất để đưa ra những quảng cáo hay đề xuất cho người sử dụng. Để làm được điều đó, hệ thống đề xuất sử dụng các thuật toán để phân tích, dự đoán dựa trên dữ liệu hành vi người dùng lưu lại. Nhờ đó, hệ thống sẽ cá nhân hóa tới người dùng và biết chính xác được từng người sử dụng có nhu cầu gì, muốn xem gì để đưa ra đề xuất thích hợp.

Hệ thống đề xuất góp phần không nhỏ đến thành công của các trang web lớn và với mỗi một hệ thống lại cần tinh chỉnh một bộ đề xuất phù hợp với dữ liệu mà web hay ứng dụng sở hữu và sử dụng bộ đề xuất nào để có thể đạt được hiệu quả cao với tài nguyên mà họ đang sở hữu.

Đa số các bộ đề xuất đều đạt hiệu quả cao nếu như có đủ dữ liệu nhưng lại kém hiệu quả khi gặp dữ liệu nhỏ, điều này đối với các web hay app có ít dữ liệu sẽ không thể tận dụng được lợi ích của bộ đề xuất. Phương pháp lai được đề xuất để giải quyết vấn đề nằm ở các bộ dữ liệu nhỏ : thiếu dữ liệu và khởi động nguội.

1.2 Mục tiêu và phạm vi của đề tài

1.2.1 Mục tiêu đề ra

Hiện nay, có khá nhiều công trình nghiên cứu về các mô hình đề xuất phim cho người dùng. Nhiều mô hình mới được áp dụng vào thực tế và chất lượng của các mô hình này cũng ngày càng được cải tiến theo thời gian. Tuy nhiên, những mô hình khác nhau đưa lại những hiệu quả và có những ưu nhược điểm khác nhau. Trong khóa luận này, nhóm đưa ra ba mục tiêu quan trọng :

1. Nghiên cứu các thuật toán, mô hình cũ và mới đã được xây dựng trước đây để phục vụ cho bài toán đề xuất các bộ phim khác nhau cho người xem.
2. Chọn lựa thuật toán cũng như mô hình phù hợp sao cho đạt được hiệu quả tốt khi chạy thử trên bộ dữ liệu Movielens[8], sau đó cố gắng thực hiện một vài cải tiến nhằm tăng hiệu suất cho mô hình.
3. Đưa ra những hướng nghiên cứu kế tiếp, làm nền tảng cho những nghiên cứu tiếp theo để mô hình có thể đưa vào thực tế, mang lại hiệu quả như mong đợi.

1.2.2 Phạm vi đề tài

Trong khóa luận này, ngoài việc trình bày cơ sở lý thuyết về hệ thống đề xuất và các phương pháp lọc thông tin như lọc dựa trên nội dung, lọc cộng tác, lọc kết hợp, máy học và cả học sâu trong hệ thống đề xuất phim. Khóa luận sẽ tập trung nghiên cứu về phương pháp lai sử dụng đồ thị và bộ mã hóa tự động(Hybrid approach for Movie Recommendation Based on Graph and Autoencoder) kết hợp giữa mô hình Graph-Based, thuật toán k-Mean và Autoencoder trong lĩnh vực học sâu sau đó tiến hành thực nghiệm trên tập dữ liệu thử nghiệm Movielens để so sánh và đánh giá mức độ hiệu quả của phương pháp này so với một số phương pháp khác đã được trình bày.

1.3 Đóng góp của đề tài

- Tìm hiểu các mô hình khác nhau trong hệ thống đề xuất phim hiện có và cài đặt các mô hình để có thể sử dụng kiểm nghiệm thực tế.
- Kiểm tra và so sánh được tốc độ và độ chính xác của một số mô hình hiện có trong lĩnh vực đề xuất phim.
- Tìm hiểu được điểm mạnh, điểm yếu của mô hình Hybrid approach for Movie Recommendation on Graph-based and Autoencoder và ứng dụng để xây dựng hệ thống đề xuất phim tự động với bộ dữ liệu Movielens-100k.
- Đánh giá phương pháp và đưa ra kết luận.

1.4 Khái quát về phương pháp đề xuất

1.4.1 Khái quát bài toán

Trước khi khóa luận trình bày về các phương pháp đề xuất, cần làm rõ 2 thuật ngữ được sử dụng: Người dùng(user) và sản phẩm(item). Thứ nhất, khái niệm người dùng ở đây là người sử dụng hệ thống để thực hiện các thao tác xem, đánh giá, bình luận, ... Thứ hai, khái niệm sản phẩm là mặt hàng như các video, bộ phim, bản nhạc, bài báo, ... riêng trong khóa luận này thì chủ yếu item là các bộ phim. Trong hầu hết các hệ thống đề xuất, dữ liệu được cung cấp dưới dạng đánh giá của người người dùng về sản phẩm.

	Movie 1	Movie 2	Movie 3	Movie 4
User 1	2	4	?	...	4
User 2	5	?	3	...	?
User 3	1	?	?	...	2
.....
User 4	3	?	5	...	?

Bảng 1.1: Xếp hạng của người dùng u cho bộ phim i

Bài toán được đặt ra như sau: Với các tập dữ liệu người dùng U , tập các bộ phim I và tập dữ liệu $D = u, i, r_{u,i}$ trong đó $u \in U, i \in I, r_{u,i}$ là

đánh giá của người dùng u cho bộ phim i . Cần dự đoán đánh giá(hay xếp hạng) chưa biết của một người dùng thứ n nào đó u_n cho bộ phim i_m như trong bảng 1.1 miêu tả.

1.4.2 Phương pháp sử dụng dữ liệu của hệ thống đề xuất

Để giải quyết bài toán đề xuất, có nhiều phương pháp có thể giải quyết được việc đó. Mỗi phương pháp sử dụng dữ liệu người dùng - sản phẩm theo những cách khác nhau. Nhìn chung, có thể phân loại cách sử dụng dữ liệu người dùng - sản phẩm thành 2 nhóm:

Sử dụng dữ liệu rõ ràng và sử dụng dữ liệu ẩn.

a. Dữ liệu rõ ràng

Những dữ liệu rõ ràng được sử dụng trong các thuật toán đề xuất phim như: Thông tin của các bộ phim và người xem, dữ liệu đánh giá của toàn bộ người xem cho các bộ phim.

b. Dữ liệu ẩn

Dựa trên dữ liệu đánh giá của người dùng và những dữ liệu rõ ràng, có thể phân tích ra những dữ liệu ẩn có ích cho việc dự đoán. Cụ thể trong hệ thống đề xuất phim, những dữ liệu ẩn có thể là: Movie biases, xu hướng người dùng, sở thích của người dùng.

1.5 Cấu trúc khóa luận

- Chương 1: “Giới thiệu đề tài” - Khóa luận giới thiệu tổng quan về đề tài đề xuất phim cho người xem, mục đích thực hiện đề tài, ứng dụng thực tế của đề tài, khái quát bài toán cần giải quyết và các đóng góp cụ thể của nhóm.
- Chương 2: “Các công trình liên quan” - Tìm hiểu tổng quan các công trình có liên quan đến đề tài mà nhóm đang nghiên cứu.
- Chương 3: “Phương pháp đề xuất” - Trình bày các phương pháp tiếp cận và bàn luận lý do đề xuất.
- Chương 4: “Kết quả thực nghiệm” - Trình bày chi tiết các tập dữ liệu được sử dụng trong đề tài, chi tiết cách cài đặt, quá trình huấn luyện, đánh giá mô hình và kết quả thu được.
- Chương 5: “Kết luận và hướng phát triển” - Khóa luận nhận xét và đánh giá cụ thể các kết quả thu được trong quá trình thực nghiệm, qua đó đề xuất một số hướng giải quyết nhằm nâng cao chất lượng mô hình trong tương lai.
- Tài liệu tham khảo - Mục trích dẫn các bài báo, tạp chí của các công trình nghiên cứu, bộ dữ liệu, các liên kết được tham khảo và dẫn xuất trong khóa luận.

Chương 2

Các công trình liên quan

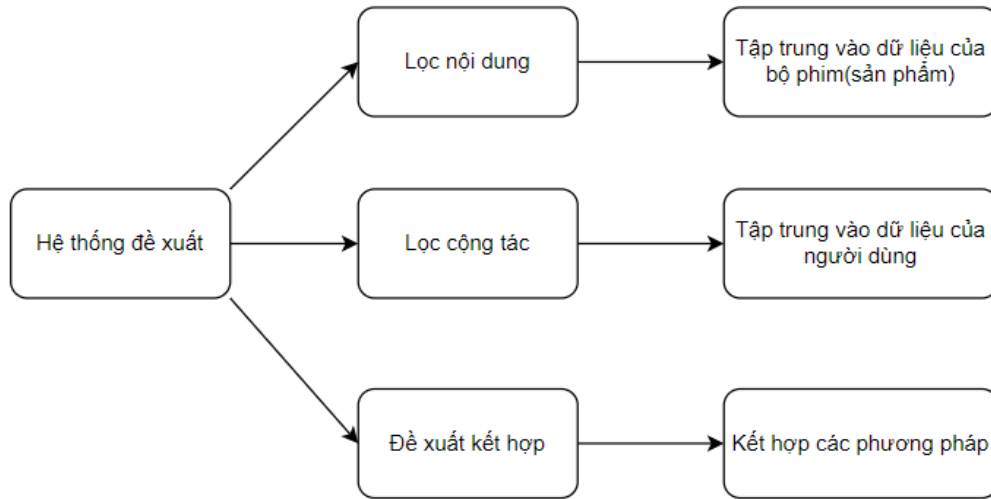
Chương 2 sẽ giới thiệu một số công trình nghiên cứu, hệ thống đề xuất có liên quan đến phương pháp mà nhóm đang tiếp cận.

2.1 Các mô hình trong hệ thống đề xuất không cá nhân hóa

Trong nhóm kỹ thuật này, do chúng khá đơn giản, dễ cài đặt nên thường được các website/hệ thống tích hợp vào, gồm cả các website thương mại, website tin tức, hay giải trí. Chẳng hạn như trong một số trang xem phim trực tuyến, người ta thường gợi ý các sản phẩm được xem, thích, bình luận,... nhiều nhất; gợi ý các bộ phim mới nhất; gợi ý các bộ phim cùng thể loại, đạo diễn ...; Tuy vậy, bất lợi của các phương pháp này là không cá nhân hóa cho từng người dùng, nghĩa là tất cả các người dùng đều được gợi ý giống nhau khi chọn cùng bộ phim. Do phương pháp này khá đơn giản nên nhóm không đề cập và tìm hiểu nhiều, thay vào đó nhóm tập trung nghiên cứu và tìm hiểu về các phương pháp cá nhân hóa cho người dùng ở trong phần 2.2.

2.2 Các mô hình trong hệ thống đề xuất cá nhân hóa

Có 3 hệ thống đề xuất sử dụng các phương pháp chính gồm: Lọc nội dung, lọc cộng tác và đề xuất kết hợp.



Hình 2.1: Các mô hình trong hệ thống đề xuất

2.2.1 Đề xuất dựa trên nội dung

Ý tưởng chính của phương pháp đề xuất này là: Đề xuất cho người dùng u những sản phẩm tương tự với những sản phẩm i đã được người dùng u đánh giá cao từ trước đó.

2.2.1.1 Phương pháp dự đoán

a. Thông tin các bộ phim

Cần xây dựng thông tin cho mỗi bộ phim i . Thông tin của bộ phim được xây dựng dưới dạng vector, là tập các đặc trưng của bộ phim đó.

b. Thông tin của người dùng

Có thể xây dựng được thông tin của người dùng là vector đánh giá trung bình của người dùng u đó cho từng thành phần trong vector đặc trưng của bộ phim.

c. Dự đoán

Để dự đoán độ "yêu thích" của người dùng u và bộ phim i :

$$l(u, i) = \cos(u, i) = \frac{u \cdot i}{||u|| ||i||} \quad (2.1)$$

2.2.2 Lọc cộng tác

Quá trình lọc cộng tác dựa trên một khái niệm đã được con người sử dụng trong nhiều thế kỷ: tìm kiếm ý kiến từ người khác để đưa ra các đề xuất sẽ hỗ trợ họ trong quá trình ra quyết định của chính họ. Nói chung, lọc cộng tác là quá trình lọc thông tin hoặc mẫu sử dụng các kỹ thuật liên quan đến sự hợp tác giữa nhiều nguồn dữ liệu. Có nhiều cách tiếp cận để giải quyết bài toán lọc cộng tác: Cách tiếp cận dựa trên bộ nhớ (memory-based), tiếp cận dựa trên mô hình (model-based), kết hợp nhiều cách tiếp cận: Kết hợp các thuật toán của các cách tiếp cận dựa trên bộ nhớ và thuật toán của cách tiếp cận dựa trên mô hình với nhau để đưa ra kết quả tốt hơn.

2.2.2.1 Cách tiếp cận dựa trên bộ nhớ

Các thuật toán dựa trên bộ nhớ, còn được gọi là dựa trên vùng lân cận, là các phỏng đoán tính toán dự đoán xếp hạng cho người dùng u dựa trên xếp hạng trước đó của những người hàng xóm gần nhất của anh ta, được xác định trước đó bằng sự tương đồng. Nói cách khác, giá trị xếp hạng không xác định $r_{u,i}$, đại diện cho giá trị mà người dùng u sẽ cung cấp cho một mục i , sẽ được tính như một tập hợp các xếp hạng tương tự nhất của người dùng cho mục i .

Phương pháp lọc cộng tác với cách tiếp cận dựa trên bộ nhớ có đặc trưng cơ bản là thường sử dụng toàn bộ dữ liệu đã có để dự đoán đánh giá của một người dùng nào đó về sản phẩm mới. Cách tiếp cận dựa trên bộ nhớ thường được chia làm 2 loại: Dựa trên người dùng và dựa trên sản phẩm.

a. Dựa trên người dùng:

Phương pháp được tóm tắt với 2 bước như sau:

- Bước 1: Tìm kiếm những người dùng có đánh giá tương tự với người dùng cần được dự đoán.
- Bước 2: Sử dụng đánh giá từ người dùng được tìm thấy ở bước 1 để tính toán dự đoán cho người cần được dự đoán.

b. Dựa trên sản phẩm:

Phương pháp được tóm tắt với 2 bước như sau:

- Bước 1: Xây dựng một ma trận để xác định mối quan hệ giữa các cặp sản phẩm với nhau
- Bước 2: Kiểm tra thị hiếu của người dùng cần dự đoán bằng cách kiểm tra ma trận và kết hợp dữ liệu của người dùng đó.

2.2.2.2 Cách tiếp cận dựa trên mô hình

Các thuật toán dựa trên mô hình tìm cách sử dụng tập xếp hạng để tìm hiểu một mô hình có thể được sử dụng để dự đoán xếp hạng mà người dùng u sẽ đưa ra cho một mục i . Các mô hình được phát triển bằng cách sử dụng các cách khai phá dữ liệu khác nhau, các thuật toán học máy để dự đoán đánh giá của người dùng về các bộ phim chưa được đánh giá.

2.2.2.3 Một số hạn chế của lọc cộng tác

Lọc cộng tác gặp phải một số khó khăn như sau:

- **Vấn đề thừa thớt dữ liệu (data sparsity)**

Vấn đề thừa thớt dẫn đến một vấn đề phổ biến khác gọi là khởi động nguội đề cập đến người dùng (hoặc mục) vẫn chưa đánh giá bất kỳ mục nào trong hệ thống. Vì không có thông tin liên quan đến sở thích của người dùng mới nên rõ ràng có khó khăn trong việc dự đoán xếp hạng cho những trường hợp này.

- **Khả năng mở rộng (scalability)**

Một vấn đề phổ biến khác là vấn đề về khả năng mở rộng của thuật toán đề xuất, vì chi phí tính toán tăng lên khi có nhiều người dùng và vật phẩm hơn trong hệ thống.

- **Khởi động nguội (cold start)** Vấn đề thừa thớt dẫn đến một vấn đề phổ biến khác gọi là khởi động nguội đề cập đến người dùng (hoặc mục) vẫn chưa đánh giá bất kỳ mục nào trong hệ thống. Vì không có thông tin liên quan đến sở thích của người dùng mới nên rõ ràng có khó khăn trong việc dự đoán xếp hạng cho những trường hợp này.

- **Vấn đề từ đồng nghĩa (synonymy)**

Từ đồng nghĩa đề cập đến xu hướng của một số mục giống nhau hoặc

rất giống nhau có tên hoặc mục khác nhau. Hầu hết các hệ thống gợi ý không thể phát hiện ra mối liên hệ tiềm ẩn này và do đó đối xử với các sản phẩm này một cách khác biệt.

- **Cừu xám (gray sheep) và cừu đen(black sheep)**

Cừu xám đề cập đến những người dùng có ý kiến không nhất quán đồng ý hoặc không đồng ý với bất kỳ nhóm người nào và do đó mô hình khó phân loại những người này vào các nhóm phù hợp. Cừu đen là một nhóm người có sở thích riêng khiến cho các đề xuất gần như không thể thực hiện được.

2.2.3 Phương pháp lai

Nghiên cứu gần đây đã chứng minh rằng một phương pháp lai, kết hợp lọc cộng tác và lọc dựa trên nội dung có thể hiệu quả hơn trong một số trường hợp. Các phương pháp lai có thể được thực hiện theo nhiều cách:

- Bằng cách đưa ra các dự đoán dựa trên nội dung và dựa trên lọc cộng tác riêng biệt và sau đó kết hợp chúng.
- Bằng cách thêm các khả năng dựa trên nội dung vào phương pháp cộng tác (và ngược lại).
- Bằng cách thống nhất các phương pháp tiếp cận thành một mô hình.

Một số nghiên cứu thực nghiệm so sánh hiệu suất của phương pháp lai với các phương pháp cộng tác thuần túy và chứng minh rằng các phương pháp lai có thể cung cấp các khuyến nghị chính xác hơn các phương pháp thuần túy. Những phương pháp này cũng có thể được sử dụng để khắc phục một số vấn đề thường gặp trong hệ thống gợi ý như khởi động nguội và vấn đề thừa thớt dữ liệu. Netflix là một ví dụ tốt về việc sử dụng các hệ thống hybrid recommender. Trang web đưa ra các đề xuất bằng cách so sánh thói quen xem và tìm kiếm của những người dùng tương tự (ví dụ: lọc cộng tác) cũng như bằng cách cung cấp những bộ phim có chung đặc điểm với những bộ phim mà người dùng đánh giá cao (lọc dựa trên nội dung). Một loạt các kỹ thuật khác đã được đề xuất làm cơ sở cho các hệ thống gợi ý: các kỹ thuật hợp tác (collaborative), dựa trên nội dung

(content-based), dựa trên kiến thức (knowledge-based) và nhân khẩu học (demographic techniques). Mỗi kỹ thuật này đều có những thiếu sót, như vấn đề khởi động nguội cho các hệ thống cộng tác và dựa trên nội dung (phải làm gì với người dùng mới với ít xếp hạng) và tắc nghẽn kỹ thuật tri thức (knowledge engineering bottleneck) trong các phương pháp dựa trên tri thức. Một hệ thống gợi ý lai là một hệ thống trong đó kết hợp nhiều kỹ thuật với nhau để đạt được một số sức mạnh tổng hợp giữa chúng.

- Cộng tác – Collaborative: Hệ thống tạo đề xuất chỉ sử dụng thông tin về hồ sơ xếp hạng cho những người dùng hoặc mục khác nhau. Các hệ thống cộng tác định vị “người dùng/mục” ngang hàng với lịch sử xếp hạng tương tự như người dùng hoặc mục hiện tại và tạo đề xuất sử dụng vùng lân cận này. Các thuật toán dựa trên người dùng và dựa trên hàng gần nhất có thể được kết hợp để giải quyết vấn đề khởi động nguội và cải thiện kết quả đề xuất.
- Dựa trên nội dung – Content-based: Hệ thống tạo đề xuất từ hai nguồn: các tính năng liên quan đến sản phẩm và xếp hạng mà người dùng đã cung cấp cho họ. Đề xuất dựa trên nội dung coi đề xuất là sự cố phân loại người dùng cụ thể và tìm hiểu trình phân loại cho lượt thích và không thích của người dùng dựa trên các tính năng của sản phẩm. Phương pháp này có thể kết hợp với một số phương pháp khác như lọc cộng tác để tạo ra sự đa dạng cho các bộ phim trong việc đề xuất phim cho người xem.
- Dựa trên tri thức – knowledge-based: Trình giới thiệu dựa trên kiến thức gợi ý các sản phẩm dựa trên các suy luận về nhu cầu và sở thích của người dùng. Kiến thức này đôi khi sẽ chứa kiến thức chức năng rõ ràng về cách các tính năng sản phẩm nhất định đáp ứng nhu cầu của người dùng.
- Nhân khẩu học – demographic techniques: Trình giới thiệu nhân khẩu học cung cấp các đề xuất dựa trên hồ sơ nhân khẩu học của người dùng. Sản phẩm được đề xuất có thể được sản xuất cho các mục nhân khẩu học khác nhau, bằng cách kết hợp xếp hạng của người dùng trong các mục đó.

Thuật ngữ Hybrid recommender systems được sử dụng ở đây để mô tả bất kỳ hệ thống recommender nào kết hợp nhiều kỹ thuật đề xuất với nhau để tạo dữ liệu đầu ra của nó. Các hệ thống đề xuất kết hợp thường kết hợp các kỹ thuật từ các kỹ thuật khác nhau để bù đắp các hạn chế của từng phương pháp trong khi tối đa hóa lợi thế của nó. Nói chung, một kỹ thuật mới được tạo ra với sự pha trộn các lớp thuật toán khác nhau hoặc thậm chí kết hợp các kỹ thuật dự đoán khác nhau bằng cách sử dụng các mô hình khác nhau nhằm đưa lại kết quả tốt nhất cho mô hình hoặc tận dụng được những điểm mạnh của các mô hình khác.

Có bảy kỹ thuật lai cơ bản (hybridization techniques):

- Có trọng số (Weighted): Điểm số của các thành phần đề xuất khác nhau được kết hợp theo số lượng.
- Chuyển đổi (Switching): Hệ thống chọn giữa các thành phần đề xuất và áp dụng hệ thống được chọn.
- Hỗn hợp (Mixed): Các khuyến nghị từ những người giới thiệu khác nhau được trình bày cùng nhau để đưa ra đề xuất.
- Kết hợp tính năng (Feature Combination): Các tính năng được lấy từ các nguồn tri thức khác nhau được kết hợp với nhau và được đưa ra cho một thuật toán gợi ý duy nhất.
- Tính năng tăng cường (Feature Augmentation): Một kỹ thuật gợi ý được sử dụng để tính toán một tính năng hoặc tập hợp các tính năng, sau đó là một phần của đầu vào cho kỹ thuật tiếp theo.
- Cascade: Các khuyến nghị được ưu tiên nghiêm ngặt, với những ưu tiên thấp hơn phá vỡ các mối quan hệ trong việc tính điểm của những người cao hơn.
- Cấp độ meta (Meta-level): Một kỹ thuật đề xuất được áp dụng và tạo ra một số loại mô hình, sau đó là đầu vào được sử dụng bởi kỹ thuật tiếp theo.

2.3 Hệ thống đề xuất sử dụng Autoencoder

Trong những năm gần đây, mạng nơ-ron học sâu hay tên gọi tiếng anh là Deep neural networks đã được sử dụng phổ biến nhất trong các hệ thống đề xuất. Với mức độ bùng nổ hàng ngày của khối lượng lớn dữ liệu đã dẫn đến sự xuất hiện của mô hình Dữ liệu lớn. Lượng thông tin ngày càng tăng có sẵn trên Internet khiến các cá nhân ngày càng khó tìm thấy những gì họ cần một cách nhanh chóng và dễ dàng. Các hệ thống khuyến nghị đã xuất hiện như một giải pháp để khắc phục vấn đề này. Lọc cộng tác được sử dụng rộng rãi trong loại hệ thống này, nhưng kích thước lớn và dữ liệu thừa thớt luôn là vấn đề chính. Với ý tưởng học sâu ngày càng trở nên quan trọng, một số công trình đã xuất hiện để cải thiện kiểu lọc này. Trong đó, một hệ thống đề xuất trong đó sử dụng Autoencoder dựa trên phương pháp lọc cộng tác được nghiên cứu như: Diana Ferreira và các cộng sự 2020 [6], AutoRec: An Automated Recommender System do Ting-Hsiang Wang và các cộng sự 2017 [17]. Nghiên cứu của họ đã cho thấy mức độ hiệu quả khi tận dụng những lợi ích của Autoencoder trong khi xây dựng một hệ thống đề xuất.

Từ những nghiên cứu đó cho thấy có 2 cách để áp dụng Autoencoder vào hệ thống đề xuất :

- Điền trực tiếp vào ma trận tương tác ở tầng tái xây dựng (reconstruction layer hay còn gọi là bước decoded)
- Giảm chiều dữ liệu, rút trích những đặc trưng quan trọng.

Tuy nhiên, cần phải kết hợp Autoencoder với những phương pháp lọc cộng tác khác nhau để đưa lại hiệu quả cho hệ thống đề xuất.

2.4 Hệ thống đề xuất sử dụng mô hình Graph-based

Để giải quyết vấn đề bùng nổ thông tin và nâng cao trải nghiệm người dùng trong các ứng dụng trực tuyến khác nhau, các hệ thống gợi ý đã được phát triển để mô hình hóa sở thích của người dùng. Mặc dù đã có nhiều nỗ lực hướng tới các đề xuất được cá nhân hóa hơn, nhưng các hệ thống

đề xuất vẫn gặp phải một số thách thức, chẳng hạn như dữ liệu thừa thớt và bắt đầu nguội. Trong những năm gần đây, việc tạo ra các đề xuất với biểu đồ tri thức như thông tin phụ đã thu hút được sự quan tâm đáng kể. Cách tiếp cận như vậy không chỉ có thể làm giảm bớt các vấn đề nêu trên để có khuyến nghị chính xác hơn mà còn cung cấp giải thích cho các mục được đề xuất chẳng hạn như: Knowledge Graph-based Recommendation Systems do SajishaPS và các cộng sự năm 2019 [15]. Và sau khi sử dụng phương pháp Graph-based để xây dựng và trích xuất được các thông tin của người dùng thì mô hình cũng cần kết hợp với các phương pháp lọc cộng tác khác để có thể xây dựng được một hệ thống hoàn chỉnh cho người dùng.

2.5 Hệ thống đề xuất có sử dụng thuật toán k-Means

K-Means[10] là một thuật toán khá phổ biến trong các hệ thống đề xuất lọc cộng tác. Thuật toán sẽ tiến hành gom những cụm người dùng có sở thích hay đặc điểm tương đồng nhau. Trong thuật toán k-Means mỗi cụm dữ liệu được đặc trưng bởi một tâm . Tâm là điểm đại diện nhất cho một cụm và có giá trị bằng trung bình của toàn bộ các quan sát nằm trong cụm ví dụ như: AI Movies Recommendation System Based on K-Means Clustering Algorithm của Syed Muhammad Asad năm 2020 [9]. Tương tự với một số thuật toán khác k-Means sẽ được nâng cao hiệu suất khi kết hợp với những thuật toán hay mô hình khác trong hệ thống đề xuất.

Chương 3

Phương pháp đề xuất

Chương này trình bày cơ sở lý thuyết về mạng thần kinh, học sâu, đồ thị, bộ mã hóa tự động và một số thuật toán nhằm cung cấp cơ sở để hiểu phương pháp được sử dụng cho các thử nghiệm được thực hiện.

3.1 Cơ sở lý thuyết

3.1.1 Graph-based

Trong toán học, đồ thị là cấu trúc toán học được sử dụng để mô hình hóa các mối quan hệ theo cặp giữa các đối tượng[4]. Một đồ thị trong này được tạo thành từ các đỉnh (còn được gọi là nút hoặc điểm) được kết nối bởi các cạnh (còn được gọi là liên kết hoặc đường). Sau khi xây dựng được một đồ thị. Đồ thị có thể tiến hành trích xuất các đặc trưng hoặc sự tương đồng giữa các thành phần trong đồ thị dựa vào các độ đo:

- **Page rank:[2]** là thuật toán đo lường ảnh hưởng chuyển tiếp hoặc khả năng kết nối của các nút. Chúng ta có thể tính **Page rank** bằng cách tính phân phối lặp lại xếp hạng của một nút (dựa trên mức độ) trên các nút lân cận hoặc duyệt ngẫu nhiên biểu đồ và đếm tần suất chạm vào từng nút trong các đường dẫn này. Trong khóa luận này, nhóm đã sử dụng cách đầu tiên.
- **Degree Centrality:[3]** Đo lường số lượng các mối quan hệ đến và đi từ một nút. Thuật toán Degree Centrality có thể được sử dụng để tìm mức độ phổ biến của các nút riêng lẻ. Các giá trị độ trung tâm được chuẩn hóa bằng cách chia cho tổng số lượng các nút khác có trong biểu đồ.
- **Closeness Centrality:[3]** là một cách để phát hiện các nút có thể

lan truyền thông tin hiệu quả. Closeness Centrality của một nút u đo khoảng cách trung bình của nó (tính theo khoảng cách) tới tất cả $n-1$ nút khác.

$$C_c(u) = \frac{n-1}{\sum_{v=1} d_{v,u}} \quad (3.1)$$

Với $d_{v,u}$ là khoảng cách giữa hai điểm u và v . Và n là số lượng nút của đồ thị. Các nút có Closeness Centrality cao là các nút có khoảng cách ngắn nhất đến tất cả các nút khác.

- **Betweenness Centrality:**[3] là một yếu tố mà để phát hiện mức độ ảnh hưởng của một nút có luồng thông tin trong đồ thị. Nó thường được sử dụng để tìm các nút đóng vai trò là cầu nối từ phần này sang phần khác của đồ thị. Thuật toán Betweenness Centrality tính toán đường đi ngắn nhất (có trọng số) giữa mọi cặp nút trong đồ thị liên thông, sử dụng thuật toán tìm kiếm theo chiều rộng. Mỗi nút nhận được một số điểm, dựa trên số lượng các con đường ngắn nhất đi qua thông qua nút.

$$C_B(u) = \frac{\sigma(s, t|u)}{\sum_{s,t \in V} \sigma(s, t)} \quad (3.2)$$

Trong đó V là tập hợp các nút, $\sigma(s, t)$ là đường đi ngắn nhất giữa hai nút (s, t) . $\sigma(s, t|u)$ là số của những con đi qua một số nút khác s và t . Nếu $s = t$ thì $\sigma(s, t) = 1$, còn nếu $v \in s, t$ thì $\sigma(s, t|u) = 0$

- **Load Centrality:**[13] là tỷ lệ của tất cả các đường đi ngắn nhất đi qua nút đó.
- **Average Neighbor Degree:**[5] là mức độ trung bình của vùng lân cận của mỗi nút. Trung bình bậc của nút i là:

$$AND(u) = \frac{1}{N(u)} \sum_{v \in N(u)} k(v) \quad (3.3)$$

Trong đó $N(u)$ là số lượng của những nút lân cận nút u . $k(v)$ là bậc của nút v . Với v thuộc tập hợp các nút lân cận của u .

Đối với đồ thị có trọng số ta có một cách tính khác tương tự:

$$AND^w(u) = \frac{1}{S(u)} \sum_{v \in N(u)} w_{uv} k(v) \quad (3.4)$$

3.1.2 Deep learning

Deep Learning (DL) là một chủ đề nóng trong cộng đồng học máy. Sự phổ biến của việc áp dụng học sâu vào hệ thống khuyến nghị trở nên phổ biến trong từ năm 2016 đến nay, với hội thảo Deep Learning for recommender Systems tại ACM RecSys 2016. Mạng nơ-ron hồi quy (RNN) có một số thuộc tính làm cho chúng trở nên phù hợp để mô hình hóa chuỗi các phiên truy cập của người dùng. Đặc biệt, chúng có khả năng kết hợp đầu vào từ các sự kiện xảy ra trong quá khứ, cho phép dự đoán tốt hơn ý định của người dùng. Tầm quan trọng của mạng thần kinh nhân tạo (ANN) đã được thảo luận thường xuyên trước đây khi chúng ta nói về phân loại hình ảnh, nhận dạng giọng nói và các vấn đề khác trong AI. Mạng lưới thần kinh rất phù hợp để giúp con người giải quyết các vấn đề và thách thức trong các tình huống thực tế bằng cách cải thiện quy trình ra quyết định trong các lĩnh vực khác nhau. Đề xuất phim là một trong số đó.

Trong các hệ thống đề xuất phim, ANN đặc biệt hữu ích và có thể được sử dụng làm bộ mã hóa tự động (Autoencoder) trong nhiều lĩnh vực. Mạng thần kinh sử dụng dữ liệu đào tạo để dự đoán các đề xuất phim với độ chính xác cao cho người dùng mục tiêu. Do đó, phần quan trọng nhất là có được bộ dữ liệu phim phù hợp để tạo mô hình mạng thần kinh cho hệ thống đề xuất phim.

3.1.3 Mạng nơ-ron nhân tạo (ANN)

Mạng nơ-ron nhân tạo là một phương thức trong lĩnh vực trí tuệ nhân tạo, nó gồm một chuỗi những thuật toán được đưa ra để tìm kiếm các mối quan hệ cơ bản trong tập hợp các dữ liệu thông và được lấy cảm hứng từ cách thức hoạt động của não bộ con người. Phương thức này tạo

ra một hệ thống thích ứng được máy tính sử dụng để học hỏi từ sai lầm của chúng và liên tục cải thiện. Vì vậy, mạng nơ-ron nhân tạo nhằm tới giải quyết các vấn đề phức tạp, chẳng hạn như tóm tắt tài liệu hoặc nhận diện khuôn mặt, với độ chính xác cao hơn.

Hiện nay, mạng nơ-ron nhân tạo đang đóng một vai trò khá quan trọng trong cuộc sống. Mạng nơ-ron nhân tạo sau khi được huấn luyện có thể giúp máy tính đưa ra những quyết định nhanh chóng mà không cần nhận hỗ trợ từ con người. Việc này giúp cho con người tiết kiệm được một lượng lớn thời gian nhưng vẫn mang lại hiệu quả tốt.

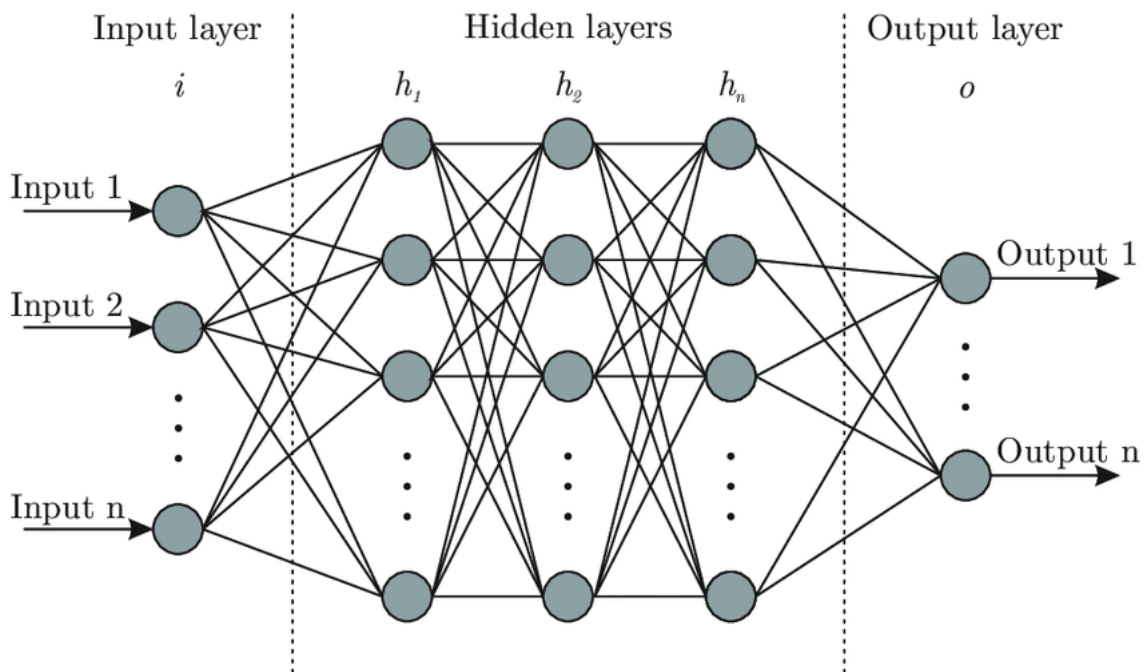
Mạng nơ-ron có rất nhiều công dụng trong cuộc sống, nó được sử dụng trong nhiều trường hợp thuộc các lĩnh vực khác nhau, chẳng hạn như:

1. Chẩn đoán bệnh trong y tế bằng cách phân loại ảnh y khoa.
2. Dự đoán tài chính bằng cách xử lý, phân tích dữ liệu tài chính trong quá khứ.
3. Tiếp thị, lọc dữ liệu bằng cách phân tích dữ liệu về hành vi người dùng trên mạng xã hội.
4. Hệ thống nhận diện hình ảnh trên ô tô tự lái để chúng có thể nhận ra các biển báo giao thông cũng như những người tham gia giao thông khác.
5. Hỗ trợ các nhân viên trực tổng đài và tự động phân loại cuộc gọi.

Bộ não con người chính là nguồn cảm hứng cho kiến trúc mạng nơ-ron. Các tế bào não của con người (các nơ-ron) liên kết với nhau tạo thành một mạng lưới phức tạp giúp não bộ xử lý thông tin. Tương tự như vậy, các nơ-ron nhân tạo cũng liên kết với nhau theo một kiến trúc nhất định để cùng xử lý một vấn đề. Kiến trúc của một mạng nơ-ron nhân tạo bao gồm 3 lớp:

1. Lớp đầu vào: Lớp này là lớp đầu tiên tiếp nhận dữ liệu từ thế giới bên ngoài. Các nút mạng ở lớp này xử lý dữ liệu, phân tích hoặc phân loại và sau đó chuyển dữ liệu sang lớp tiếp theo. Lớp này nằm ở bên trái cùng trong sơ đồ mạng trong hình 3.1.3 bên dưới.

2. Lớp ẩn: Dữ liệu đi vào lớp ẩn từ lớp đầu vào hoặc các lớp ẩn trước đó. Một mạng nơ ron nhân tạo có thể có một hoặc nhiều lớp ẩn. Mỗi lớp ẩn phân tích dữ liệu đầu ra từ lớp trước, xử lý dữ liệu và rồi chuyển dữ liệu sang lớp tiếp theo.
3. Lớp đầu ra: Lớp đầu ra cho ra kết quả cuối cùng của tất cả dữ liệu được xử lý bởi mạng nơ ron nhân tạo. Lớp này có thể có một hoặc nhiều nút tùy thuộc vào mục tiêu mà mạng hướng đến. Lớp này nằm ở bên phải cùng ở sơ đồ mạng trong hình 3.1.3 bên dưới.



Hình 3.1: Kiến trúc của mạng nơ ron nhân tạo cơ bản [14]

Việc học tập của mạng nơ ron nhân tạo là sự thích ứng của mạng để xử lý tốt hơn một nhiệm vụ bằng cách xem xét các quan sát mẫu. Việc học liên quan đến việc điều chỉnh trọng số (và ngưỡng tùy chọn) của mạng để cải thiện độ chính xác của kết quả. Điều này được thực hiện bằng cách giảm thiểu các lỗi quan sát được. Việc học hoàn tất khi kiểm tra các quan sát bổ sung không giúp giảm tỷ lệ lỗi một cách hữu ích. Ngay cả sau khi học, tỷ lệ lỗi thường không bằng 0. Nếu sau khi học, tỷ lệ lỗi quá cao, mạng thường phải được thiết kế lại. Trên thực tế, điều này được thực hiện bằng cách xác định hàm chi phí được đánh giá định kỳ trong quá trình học. Miễn là đầu ra của nó tiếp tục giảm, việc học vẫn tiếp tục. Chi phí thường được định nghĩa là một thống kê giá trị của nó chỉ có thể xấp xỉ.

Tỉ lệ học tập (Learning rate): Tỷ lệ học tập xác định quy mô của các bước khắc phục mà mô hình thực hiện để điều chỉnh các lỗi trong mỗi lần quan sát. Tốc độ học tập cao rút ngắn thời gian đào tạo nhưng với độ chính xác cuối cùng thấp hơn, trong khi tốc độ học tập thấp hơn mất nhiều thời gian hơn nhưng có khả năng cho độ chính xác cao hơn.

Hàm chi phí (Cost function): Mặc dù có thể xác định hàm chi phí một cách đặc biệt, nhưng thường thì sự lựa chọn được xác định bởi các thuộc tính mong muốn của hàm (chẳng hạn như độ lỗi) hoặc do nó phát sinh từ mô hình.

Lan truyền ngược (Backpropagation): Lan truyền ngược là một phương pháp được sử dụng để điều chỉnh trọng số kết nối để bù cho từng lỗi được tìm thấy trong quá trình học. Số lượng lỗi được phân chia hiệu quả giữa các kết nối. Về mặt kỹ thuật, backprop tính toán độ dốc (đạo hàm) của hàm chi phí được liên kết với một trạng thái nhất định đối với các trọng số. Việc cập nhật trọng số có thể được thực hiện thông qua giảm độ dốc ngẫu nhiên hoặc các phương pháp đặc biệt khác.

3.1.4 Autoencoder

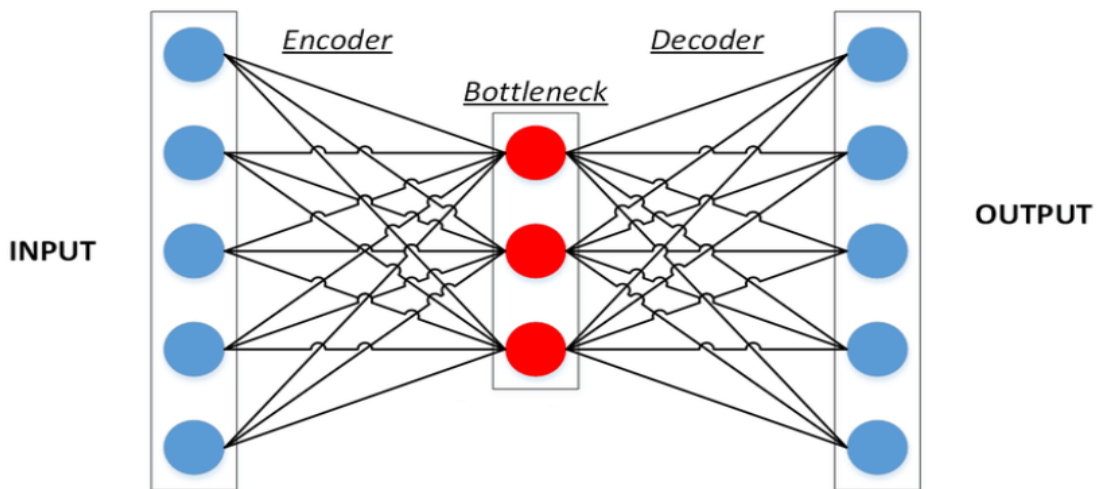
Autoencoder là một loại mạng thần kinh chuyển tiếp nguồn cấp dữ liệu cụ thể, trong đó đầu vào giống với đầu ra. Họ nén đầu vào thành mã có chiều thấp hơn và sau đó tái tạo lại đầu ra từ biểu diễn này. Mã này là một "tóm tắt" hoặc "nén" nhỏ gọn của đầu vào, còn được gọi là biểu diễn không gian tiềm ẩn. Autoencoder là mạng ANN có khả năng học hiệu quả các biểu diễn của dữ liệu đầu vào mà không cần nhãn, nói cách khác, giả sử từ một hình ảnh có thể tái tạo ra một bức ảnh có liên quan chặt chẽ với bức ảnh đầu vào đó. Đầu vào loại mạng neural này không có nhãn, tức là mạng có khả năng học không giám sát (Unsupervised Learning)

Đầu vào được mạng mã hóa để chỉ tập trung vào các đặc trưng quan trọng nhất, tùy vào bài toán cụ thể. Các biểu diễn (coding) thường có chiều nhỏ hơn so với input của Autoencoder, đó là lý do Autoencoder có thể dùng trong các bài toán giảm chiều dữ liệu hoặc trích xuất đặc trưng.

3.1.4.1 Kiến trúc của Autoencoder

Autoencoder bao gồm 3 phần chính(hình 3.2):

1. Encoder: Module có nhiệm vụ nén dữ liệu đầu vào thành một biểu diễn được mã hóa (coding), thường nhỏ hơn một vài bậc so với dữ liệu đầu vào.
2. Bottleneck: Module chứa các biểu diễn tri thức được nén (chính là đầu ra của Encoder), đây là phần quan trọng nhất của mạng bởi nó mang đặc trưng của đầu vào, có thể dùng để tái tạo lại dữ liệu, lấy đặc trưng của dữ liệu.
3. Decoder: Module giúp mạng giải nén các biểu diễn tri thức và tái cấu trúc lại dữ liệu từ dạng mã hóa của nó, mô hình học dựa trên việc so sánh đầu ra của Decoder với đầu vào ban đầu (Input của Encoder).



Hình 3.2: Kiến trúc mạng của Autoencoder [11]

Cả bộ mã hóa và bộ giải mã đều là các mạng thần kinh chuyển tiếp được kết nối đầy đủ, về cơ bản là các ANN. Code là một lớp duy nhất của ANN với kích thước do chúng ta lựa chọn. Số nút trong các lớp (kích thước mã) là một siêu tham số được đặt trước khi huấn luyện cho mô hình Autoencoder.

Cụ thể hơn, đầu tiên đầu vào đi qua bộ mã hóa, là một ANN được kết nối đầy đủ dùng để tạo mã. Bộ giải mã, có cấu trúc ANN tương tự, sau

đó tạo đầu ra chỉ bằng cách sử dụng mã. Mục tiêu là để có được một đầu ra giống với đầu vào. Nhưng tùy các trường hợp và bài toán cụ thể, ví dụ như làm nét ảnh, làm mờ ảnh, hay kéo sáng ảnh mà đầu ra và đầu vào sẽ giống về nội dung chứ giá trị thì không hẳn.

3.1.4.2 Cách mô hình hoạt động

1. **Encoder:** Bao gồm một tập hợp các convolutional blocks, theo sau là các polling modules giúp cho việc nén đầu vào của mô hình thành một phần nhỏ gọn hơn, được gọi là Bottleneck. Ở đây thì ngoài việc sử dụng các Convolutional + Pooling thì có thể chỉ cần sử dụng các khối fully connected tùy vào đầu vào và bài toán yêu cầu. Ví dụ bài toán làm mờ chữ số viết tay, thì chỉ cần một vài lớp Fully connected cũng đã làm rất tốt nhiệm vụ của mình rồi Tiếp đến, với đầu ra là Bottleneck, bộ Decoder sẽ giải mã bằng một loạt các module Upsampling (hoặc Fully Connected) để đưa đặc trưng nén về dạng hình ảnh. Trong các bài toán đơn giản thì đầu ra được mong đợi là giống với đầu vào (nhiều hơn hoặc nét hơn, ...) Tuy nhiên với các bài toán cao hơn thì ảnh đầu ra mong muốn là một ảnh hoàn toàn mới, mang một mối liên hệ chặt chẽ với ảnh đầu vào, được hình thành từ đặc trưng của ảnh đầu vào đã cung cấp
2. **Bottleneck:** Phần quan trọng nhất của Autoencoder, cũng là phần mang kích thước nhỏ nhất Bởi Bottleneck được thiết kế từ việc mã hóa tối đa thông tin của ảnh đầu vào, vậy nên có thể nói rằng bottleneck mang đặc trưng, chi thức của ảnh đầu vào Với cấu trúc mã hóa – giải mã, mô hình trích xuất được đặc trưng của ảnh dưới dạng dữ liệu và thiết lập được mối tương quan giữa input và output của mạng. Bottleneck có kích thước nhỏ hơn ảnh đầu vào cũng như mang nhiều thông tin đặc trưng giúp ngăn cản mạng ghi nhớ quá nhiều. Bottle neck càng nhỏ, rủi ro overfitting càng thấp, tuy nhiên nếu kích thước Bottleneck quá nhỏ sẽ hạn chế khả năng lưu trữ thông tin của ảnh đầu vào, gây khó khăn cho việc giải mã ở khối Decoder.
3. **Decoder:** Khối cuối cùng, mang nhiệm vụ giải mã từ Bottleneck để

tái tạo lại hình ảnh đầu vào dựa vào các đặc trưng “tiềm ẩn” bên trong Bottleneck

3.1.4.3 Một số Hyper-parameter quan trọng trong Autoencoder

Có 4 siêu tham số mà chúng ta cần đặt trước khi đào tạo bộ mã hóa tự động:

1. **Code size:** kích thước của Bottleneck là 1 hyperparameter rất quan trọng được sử dụng mà chúng ta cần lưu ý. Kích thước Bottleneck quyết định lượng thông tin được nén. “Nhiều quá không tốt mà ít quá cũng không ổn”
2. **Number of layers:** giống với hầu hết các mạng neural, một hyperparameter quan trọng để điều chỉnh độ sâu của encoder và decoder trong Autoencoder, càng sâu thì mô hình càng phức tạp, nhưng càng nông thì mô hình chạy càng nhanh, càng light weights
3. **Number of nodes per layer:** Số lượng nodes trên 1 layer quyết định số weights ra sẽ sử dụng trên từng layer. Thông thường, số lượng nút này giảm dần theo mỗi lớp tiếp theo bởi đầu vào của lớp này là đầu ra của lớp trước đó, và đầu vào này thì dần trở nên nhỏ hơn trên các lớp
4. **Loss function:** Là một phương thức không thể thiếu trong mạng neural. Hàm loss này sẽ phụ thuộc vào kiểu input và output của mô hình chúng ta muốn đáp ứng. Ví dụ với việc xử lý ảnh, các hàm loss thông thường được ưa chuộng là Mean Square Error (MSE) và L1 Loss. Nếu các giá trị đầu vào nằm trong phạm vi $[0, 1]$ thì thường sử dụng crossentropy. Trong một số trường hợp khác, mô hình sử dụng lỗi bình phương trung bình hoặc các hàm loss khác phù hợp với từng bài toán cụ thể.

3.1.4.4 Autoencoder denoising

Autoencoder denoising là một bộ mã hóa tự động, thay vì chỉ mã hóa trong giai đoạn mã hóa, mô hình sẽ làm hỏng ngẫu nhiên một số thành phần của nó để có được các tính năng mạnh mẽ hơn. Bộ mã hóa tự động nhận điểm dữ liệu bị hỏng làm đầu vào và được đào tạo để dự đoán điểm dữ liệu gốc, không bị hỏng như đầu ra của nó. Điều này ngăn bộ mã hóa tự động tìm hiểu chức năng nhận dạng một cách đơn giản.

3.1.5 Elastic net regularization

Huấn luyện mạng thần kinh có nghĩa cần đạt được sự cân bằng giữa tối ưu hóa và tối ưu hóa quá mức. Các mô hình được tối ưu hóa quá mức là mô hình hoạt động thực sự tốt trên tập huấn luyện. Tuy nhiên trong quá trình huấn luyện, mô hình sẽ tập trung cực tiểu hóa hàm mất mát và có thể gây ra hiện tượng overfitting cho cả mô hình làm cho khi thực hiện trên các tập thử nghiệm hay thực tế. Các kỹ thuật chính quy hóa có thể được sử dụng để giảm thiểu những vấn đề này. Về cơ bản, chính quy hóa là quá trình giới hạn (kiểm soát) quá trình học của một mô hình bằng cách thêm một hình phạt vào hàm lỗi (Loss function) mà mô hình đang cố gắng giảm thiểu.

$$\mathbf{L}_\phi = \mathbf{L}_\phi + \mathbf{\Omega}_\phi \quad (3.5)$$

Trong đó, \mathbf{L}_ϕ (Loss function) là hàm lỗi mà mô hình đang cố gắng giảm, $\mathbf{\Omega}_\phi$ (regularization term) là chính quy hóa được thêm vào hàm lỗi của mô hình. Trong thống kê và đặc biệt, trong việc điều chỉnh các mô hình hồi quy tuyến tính, logistic hay mô hình mạng thần kinh, Elastic net regularization [12] là một phương pháp hồi quy chính quy kết hợp tuyến tính các hình phạt L1(lasso) và L2(ridge).

- Lasso Còn được gọi là Chính quy hóa L1, phương pháp này nhanh hơn Forward Stepwise nếu có một số lượng lớn các yếu tố dự đoán. Phương pháp này ngăn chặn việc trang bị quá mức bằng cách thu hẹp (tức là bằng cách áp dụng hình phạt) đối với các tham số. Nó có thể thu nhỏ một số tham số về 0, thực hiện một phép chọn biến.
- Ridge Còn được gọi là Chính quy hóa L2, phương pháp này ngăn

chặn việc trang bị quá mức bằng cách thu hẹp (nghĩa là bằng cách áp dụng hình phạt) đối với các tham số. Nó thu nhỏ tất cả các tham số theo cùng một tỷ lệ nhưng không loại bỏ tham số nào và không phải là phương pháp chọn biến.

- Elastic net regularization được gọi là Chính quy hóa $L1 + L2$, phương pháp này ngăn việc khớp quá mức bằng cách thu hẹp (nghĩa là bằng cách áp dụng hình phạt) đối với các tham số. Nó có thể thu nhỏ một số tham số về 0, thực hiện lựa chọn biến.

3.1.6 Kmeans

K-means[16] là một thuật toán phân cụm đơn giản thuộc loại học không giám sát (tức là dữ liệu không có nhãn) và được sử dụng để giải quyết bài toán phân cụm. Ý tưởng của thuật toán phân cụm k-means là phân chia 1 bộ dữ liệu thành các cụm khác nhau. Trong đó số lượng cụm được cho trước là k . Công việc phân cụm được xác lập dựa trên nguyên lý: Các điểm dữ liệu trong cùng 1 cụm thì phải có cùng 1 số tính chất nhất định. Tức là giữa các điểm trong cùng 1 cụm phải có sự liên quan lẫn nhau. Đối với máy tính thì các điểm trong 1 cụm đó sẽ là các điểm dữ liệu gần nhau.

3.1.6.1 Ý tưởng của thuật toán K-means

1. Khởi tạo K điểm dữ liệu trong bộ dữ liệu và tạm thời coi nó là tâm của các cụm dữ liệu của chúng ta.
2. Với mỗi điểm dữ liệu trong bộ dữ liệu, tâm cụm của nó sẽ được xác định là 1 trong K tâm cụm gần nó nhất.
3. Sau khi tất cả các điểm dữ liệu đã có tâm, tính toán lại vị trí của tâm cụm để đảm bảo tâm của cụm nằm ở chính giữa cụm.
4. Bước 2 và bước 3 sẽ được lặp đi lặp lại cho tới khi vị trí của tâm cụm không thay đổi hoặc tâm của tất cả các điểm dữ liệu không thay đổi.

3.1.6.2 Chọn số lượng cụm K phù hợp

1. Lựa chọn số lượng cụm:

Chỉ việc lựa chọn số cụm k đã có thể tách thành 1 bài toán riêng. Không có 1 con số k nào là hợp lý cho tất cả các bài toán. Tùy theo tập dữ liệu của bài toán để xác định xem trong đó có thể có bao nhiêu cụm? Nhưng không phải lúc nào cũng có thể làm thế. Cách làm duy nhất là bạn hãy thử với từng giá trị $k=1,2,3,4,5,\dots$ để xem kết quả phân cụm thay đổi như thế nào. Một số nghiên cứu cho thấy việc thay đổi k sẽ có hiệu quả nhưng sẽ dừng lại ở 1 con số nào đó. Như vậy bạn hoàn toàn có thể thử xem dữ liệu của mình tốt với giá trị k nào đó.

2. Khởi tạo k vị trí ban đầu :

Đầu tiên, ta tiến hành khởi tạo k tâm cụm này phân bố đồng đều trên không gian của bộ dữ liệu. Điều đó có thể làm khi bạn có thể xác định được không gian và tính chất của dữ liệu. Nhưng ít nhất, các tâm cụm được khởi tạo cũng đừng quá gần nhau, cũng đừng trùng nhau. Hoặc là ta sẽ chạy thuật toán nhiều lần để lấy kết quả tốt nhất trong các lần chạy đó. Với điều kiện khởi tạo tâm của k cụm là ngẫu nhiên.

3. Về vấn đề tính dừng (hội tụ) :

Đối với những trường hợp dữ liệu phức tạp, thuật toán k -means sẽ rất lâu hoặc không bao giờ hội tụ. Tức là sẽ không bao giờ xác định được tâm cụm cố định để kết thúc bài toán. Hoặc là phải chạy qua rất nhiều bước lặp. Trong những trường hợp như vậy, thay vì phải tìm được k tâm cụm cố định thì ta sẽ dừng bài toán khi sự thay đổi ở một con số chấp nhận được. Tức là giữa hai lần cập nhật tâm cụm thì chênh lệch vị trí giữa tâm cũ và mới nhỏ hơn một số delta cho phép nào đó.

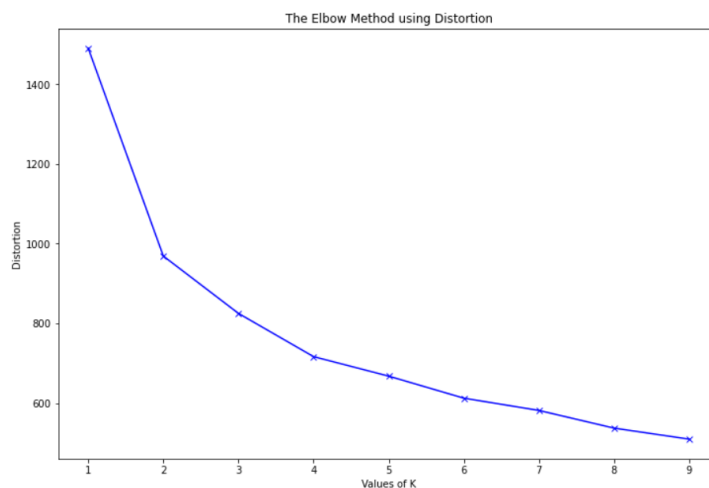
3.1.6.3 Hạn chế của thuật toán K-Means

Thuật toán K-Means có một số hạn chế đó là:

1. Chúng ta cần phải xác định trước số cụm cho thuật toán: Vì bộ dữ liệu của chúng ta chưa được gán nhãn nên dường như chúng ta không có thông tin nào về số lượng cụm hợp lý. Chúng ta chỉ có thể thực hiện phương pháp thử và sai (try and error) và xác định số cụm thông qua một phương pháp chẳng hạn như Elbow.
2. Vị trí tâm của cụm sẽ bị phụ thuộc vào điểm khởi tạo ban đầu của chúng: Những vị trí khởi tạo khác nhau có thể dẫn tới cách phân cụm khác nhau, mặc dù thuật toán có cùng thiết lập số cụm.
3. Đối với những bộ dữ liệu có hình dạng phức tạp hoặc mất cân bằng thì thuật toán không hội tụ về qui luật phân chia tổng quát. Chẳng hạn như dữ liệu có dạng đường viền hình tròn bao ngoài một hình tròn ở bên trong nó; dữ liệu hình tròn ốc; dữ liệu có phân phối dẹt; dữ liệu bị mất cân bằng phân phối giữa các cụm.
4. Thuật toán rất nhạy cảm với outliers: Khi xuất hiện outliers thì thường khiến cho tâm cụm bị chệch và do đó dự báo cụm không còn chuẩn xác. Chính vì thế chúng ta cần phải loại bỏ outliers trước khi huấn luyện thuật toán.
5. Thuật toán nhạy cảm với độ lớn đơn vị của biến: Khi áp dụng thuật toán trên các biến có sự khác biệt về mặt đơn vị thì khoảng cách chủ yếu bị ảnh hưởng bởi các biến có đơn vị lớn hơn và khiến cho kết quả phân cụm bị chệch. Chính vì thế chúng ta cần phải chuẩn hoá biến để loại bỏ sự khác biệt đơn vị trước khi đưa vào huấn luyện mô hình.
6. Thuật toán K-Means yêu cầu phải tính khoảng cách từ một điểm tới toàn bộ các tâm cụm để tìm ra tâm cụm gần nhất. Như vậy chúng ta cần phải load toàn bộ dữ liệu lên RAM, đối với những bộ dữ liệu kích thước lớn thì sẽ vượt quá khả năng lưu trữ của RAM. Khi đó chúng ta cần phải huấn luyện thuật toán theo phương pháp online learning. Kỹ thuật này sẽ được giới thiệu ở bên dưới. [1]

3.1.6.4 Phương pháp Elbow

Trong thuật toán K-Means thì chúng ta cần phải xác định trước số cụm. Câu hỏi đặt ra là đâu là số lượng cụm cần phân chia tốt nhất đối với một bộ dữ liệu cụ thể? Phương pháp Elbow là một cách giúp ta lựa chọn được số lượng các cụm phù hợp dựa vào đồ thị trực quan hoá bằng cách nhìn vào sự suy giảm của hàm biến dạng và lựa chọn ra điểm khuỷu tay (elbow point). Để tìm hiểu phương pháp Elbow, bên dưới hình 3.3 chúng ta cùng thử nghiệm vẽ biểu đồ hàm biến dạng bằng cách điều chỉnh số lượng cụm của thuật toán K-Means.



Hình 3.3: Đồ thị hàm biến dạng của thuật toán k-Means.

Điểm khuỷu tay là điểm mà ở đó tốc độ suy giảm của hàm biến dạng sẽ thay đổi nhiều nhất. Tức là kể từ sau vị trí này thì gia tăng thêm số lượng cụm cũng không giúp hàm biến dạng giảm đáng kể. Nếu thuật toán phân chia theo số lượng cụm tại vị trí này sẽ đạt được tính chất phân cụm một cách tổng quát nhất mà không gặp các hiện tượng vị khớp (overfitting). Trong hình trên thì ta thấy vị trí của điểm khuỷu tay chính là $k=2$ vì khi số lượng cụm lớn hơn 2 thì tốc độ suy giảm của hàm biến dạng dường như không đáng kể so với trước đó.

Phương pháp Elbow là một phương pháp thường được sử dụng để lựa chọn số lượng cụm phân chia hợp lý dựa trên biểu đồ, tuy nhiên có một số trường hợp chúng ta sẽ không dễ dàng phát hiện vị trí của Elbow, đặc biệt là đối với những bộ dữ liệu mà qui luật phân cụm không thực sự dễ

dường được phát hiện. Nhưng nhìn chung thì phương pháp Elbow vẫn là một phương pháp tốt nhất được ứng dụng trong việc tìm kiếm số lượng cụm cần phân chia.

3.1.6.5 Phương pháp Silhouette

Chỉ số Silhouette là chỉ số đánh giá các kết quả phân cụm phổ biến và được sử dụng nhiều nhất. Phân tích chỉ số Silhouette mục đích để đo lường mức độ tối ưu khi một quan sát, một điểm dữ liệu được phân vào các cluster bất kỳ. Cụ thể, phương pháp Silhouette, sẽ cho chúng ta biết những điểm dữ liệu hay những quan sát nào nằm gọn bên trong cụm (tốt) hay nằm gần ngoài rìa cụm (không tốt) để đánh giá hiệu quả phân cụm.

Silhouette đo lường khoảng cách của một điểm dữ liệu trong cụm đến Centroid, điểm trung tâm của cụm, và khoảng cách của chính điểm đó đến điểm trung tâm của cụm gần nhất (hoặc đến các điểm trung tâm của các cụm còn lại, và chọn ra khoảng cách ngắn nhất). Đó là trường hợp đo lường cho K-means clustering.

Nếu các cluster được tìm không phải dựa trên clustering, thì Silhouette cũng sẽ đo lường theo cách tương tự nhưng thay vì tính khoảng cách giữa điểm đó với điểm trung tâm, thì chúng ta sẽ tính khoảng cách trung bình với tất cả các điểm còn lại trong cluster của điểm đó, và khoảng cách trung bình với tất cả các điểm còn lại của các cluster khác (lấy khoảng cách trung bình ngắn nhất)

Giả sử mạng lưới được chia thành k cụm.

Với mỗi node i , đặt:

- $a(i)$ là khoảng cách trung bình từ i tới tất cả các node trong cùng cụm với i .
- $b(i)$ là khoảng cách trung bình ngắn nhất từ i tới bất kỳ cụm nào không chứa i .

Cụm tương ứng với $b(i)$ này được gọi là cụm hàng xóm của i .

Khi đó:

$$S(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} \quad (3.6)$$

$s(i)$ nằm trong đoạn $[-1,1]$. $s(i)$ càng gần 1 thì node i càng phù hợp với cụm mà nó được phân vào. $s(i) = 0$ thì không thể xác định được i nên thuộc về cụm nào giữa cụm hiện tại và cụm hàng xóm của nó. $s(i)$ càng gần -1 thì chứng tỏ i bị phân sai cụm, nó nên thuộc về cụm hàng xóm chứ không phải cụm hiện tại.

3.2 Cơ sở dữ liệu thực nghiệm

3.2.1 Bộ dữ liệu sử dụng

Dữ liệu được sử dụng trong thực hành là MovieLens-100k. Đây là một bộ dữ liệu khá đầy đủ để thực hành các thuật toán recommendation bởi vì nó chứa đầy đủ cả 3 thông tin: Ratings, Users và Movies. Trong đó file ratings chính là ma trận tiện ích (utility) chứa 100.000 giá trị rating của các cặp (user, movie). File users là thông tin của 943 người dùng của trang phim MovieLen và file movies là thông tin về gần 1682 bộ phim được rating.

Bộ dữ liệu Movielens 100K đã làm sạch - loại bỏ những người dùng có ít hơn 20 xếp hạng hoặc không có đầy đủ thông tin nhân khẩu học. Vì vậy, mỗi users sẽ rating ít nhất 20 bộ phim và cụ thể các trường thông tin trong từng file như sau:

- **USERS:** Thông tin người dùng bao gồm các trường UserID|Gender|Age|Occupation|Zip-code. Trong đó:
 - Gender: giới tính nhận 2 giá trị F là nữ, M là nam.
 - Age: khoảng tuổi của người xem.
 - Occupation: Nghề nghiệp của người dùng bao gồm các nghề: 'technician', 'other', 'writer', 'executive', 'administrator', 'student', 'lawyer', 'educator', 'scientist', 'entertainment', 'programmer', 'librarian', 'homemaker', 'artist', 'engineer', 'marketing', 'none', 'healthcare', 'retired', 'salesman', 'doctor'.

	UID	age	gender	job	zip
0	1	24	M	technician	85711
1	2	53	F	other	94043
2	3	23	M	writer	32067
3	4	24	M	technician	43537
4	5	33	F	other	15213
...
938	939	26	F	student	33319
939	940	32	M	administrator	02215
940	941	20	M	student	97229
941	942	48	F	librarian	78209
942	943	22	M	student	77841

943 rows × 5 columns

Hình 3.4: Thông tin người dùng trong USERS

- **RATINGS:** Bao gồm các trường UserID|MovieID|Rating|Timestamp.
Trong đó:
 - UserIDs: có giá trị nằm trong khoảng 1 đến 943 là id xác định người dùng.
 - MovieIDs: có giá trị nằm trong khoảng 1 đến 1682 là các id của các bộ phim.
 - Ratings: là giá trị đánh giá của người xem cho các bộ phim theo thang đo từ 5 sao.
 - Timestamp biểu diễn thời điểm tiến hành rating.

	UID	MID	rate	time
0	196	242	3	881250949
1	186	302	3	891717742
2	22	377	1	878887116
3	244	51	2	880606923
4	166	346	1	886397596
...
99995	880	476	3	880175444
99996	716	204	5	879795543
99997	276	1090	1	874795795
99998	13	225	2	882399156
99999	12	203	3	879959583

100000 rows × 4 columns

Hình 3.5: Thông tin xếp hạng trong RATINGS

- **MOVIES:** Thông tin của các bộ phim gồm một số trường quan trọng như: MovieID|Title|Release Date|Genres. Trong đó:
 - MID: có giá trị nằm trong khoảng 1 đến 1682 là các ID của các bộ phim.
 - Titles: Tiêu đề của bộ phim.
 - Release Date: Ngày phát hành bộ phim.
 - Genres: Thể loại phim gồm các loại như: ["unknow", "Action", "Adventure", "Animation", "Children", "Comedy", "Crime", "Documentary", "Drama", "Fantasy", "Film-Noir", "Horror", "Musical", "Mystery", "Romance", "Sci-Fi", "Thriller", "War", "Western"]

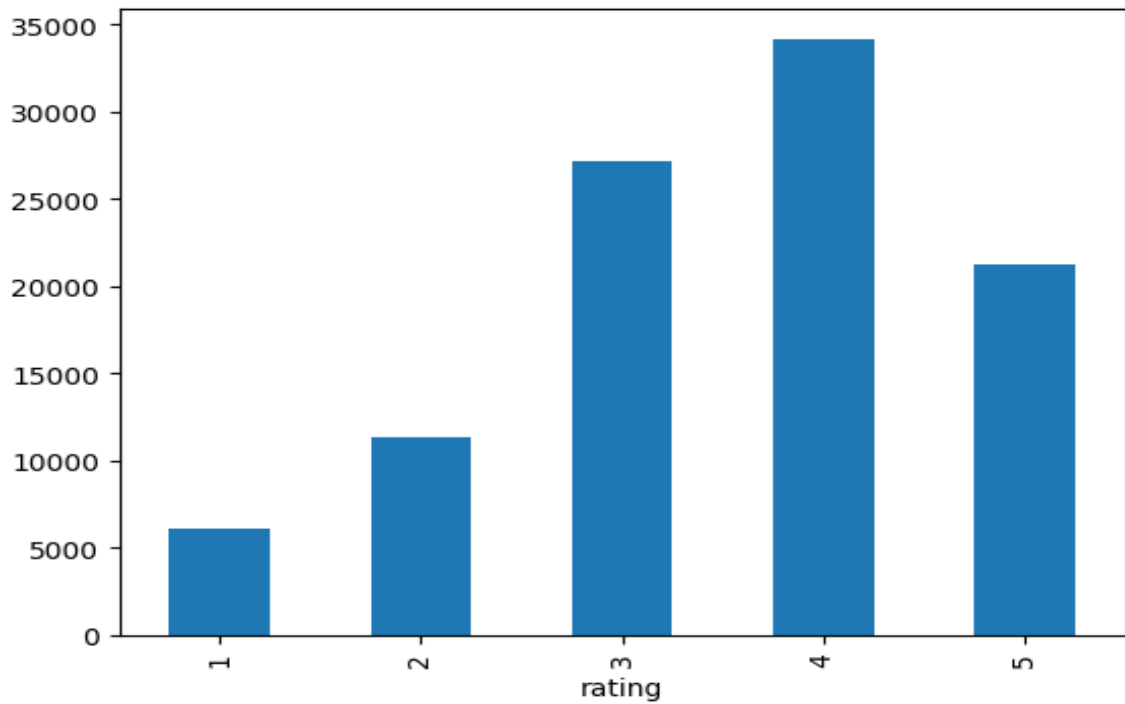
MID	Title	Release Date	Video release date	URL	unknown	Action	Adventure	Animation	Children	...	
1	Toy Story (1995)	01-Jan-1995	NaN	http://us.imdb.com/M/title-exact?Toy%20Story%2...	0	0	0	1	1	...	
2	GoldenEye (1995)	01-Jan-1995	NaN	http://us.imdb.com/M/title-exact?GoldenEye%20(...	0	1	1	0	0	...	
3	Four Rooms (1995)	01-Jan-1995	NaN	http://us.imdb.com/M/title-exact?Four%20Rooms%...	0	0	0	0	0	...	
4	Get Shorty (1995)	01-Jan-1995	NaN	http://us.imdb.com/M/title-exact?Get%20Shorty%...	0	1	0	0	0	...	
5	Copycat (1995)	01-Jan-1995	NaN	http://us.imdb.com/M/title-exact?Copycat%20(1995)	0	0	0	0	0	...	
...	

Hình 3.6: Thông tin các bộ phim trong MOVIES

Mục tiêu của chúng ta là xây dựng một mô hình kết hợp để dự báo xác suất để một khách hàng sẽ tương tác với một bộ phim nào đó. Đầu vào của mô hình sẽ là những thông tin về khách hàng và lịch sử xếp các bộ phim của họ và các trường thông tin liên quan đến bộ phim. Đầu ra là giá trị xếp hạng của bộ phim mà khách hàng chưa đánh giá, giá trị này sẽ được giải thích rõ ở mục 4 thực nghiệm.

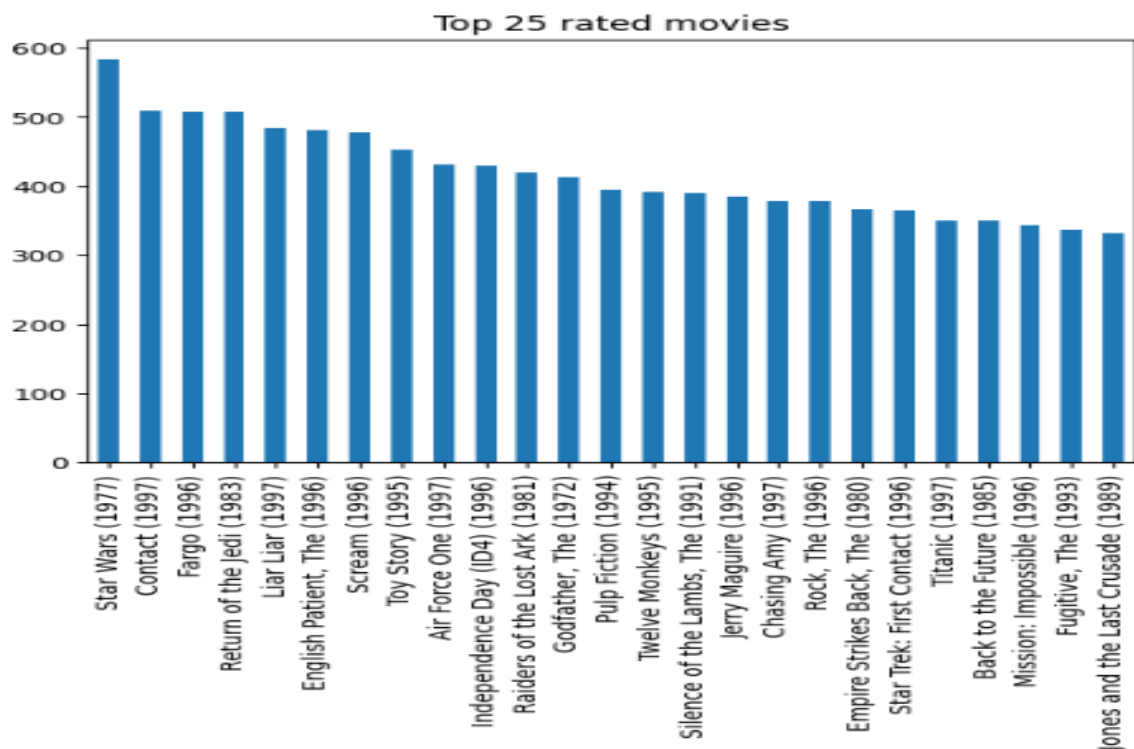
3.2.2 Phân tích và thống kê cơ bản

3.2.2.1 Thống kê cơ bản



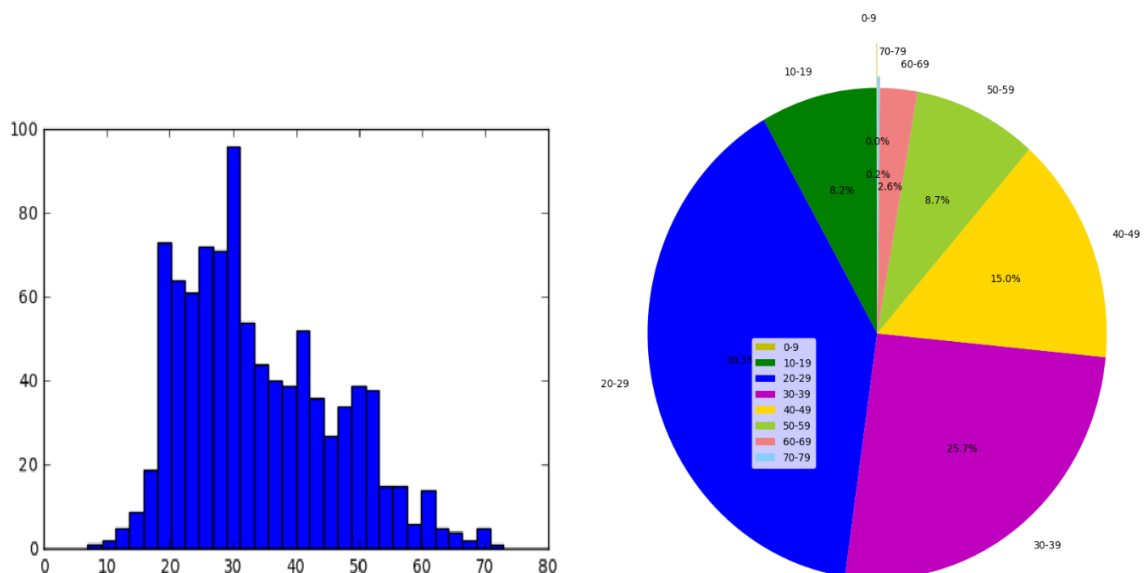
Hình 3.7: Tổng các bộ phim theo xếp hạng

Từ hình 3.7 ta thấy: các bộ phim được xếp hạng từ 1 - 5 sao. Số lượng bộ phim được xếp hạng 4 sao là nhiều nhất, số lượng phim được xếp hạng 1 sao là ít nhất. Các bộ phim được xếp hạng chủ yếu từ 3 đến 5 sao.



Hình 3.8: 25 Bộ phim được xếp hạng nhiều nhất

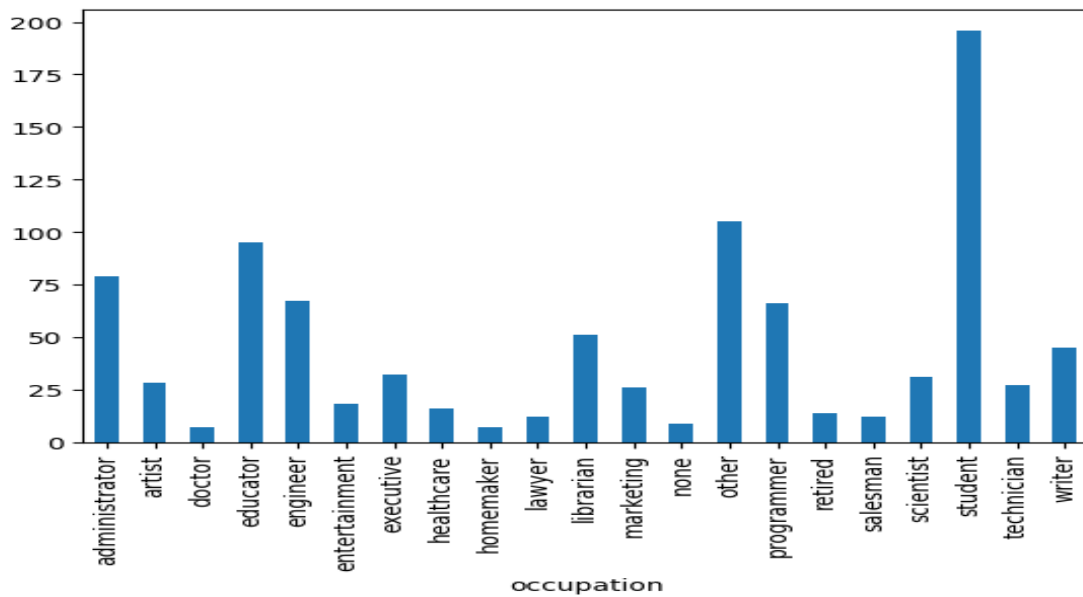
Trong 3.8 là 25 bộ phim được nhiều người xếp hạng nhất. Trong đó bộ phim được nhiều người xếp hạng nhất là bộ phim Star Wars với gần 600 xếp hạng.



Hình 3.9: Mức độ xem phim khác nhau giữa các độ tuổi

Theo dữ liệu quan sát được từ hình 3.9 Độ tuổi cho xếp hạng phim

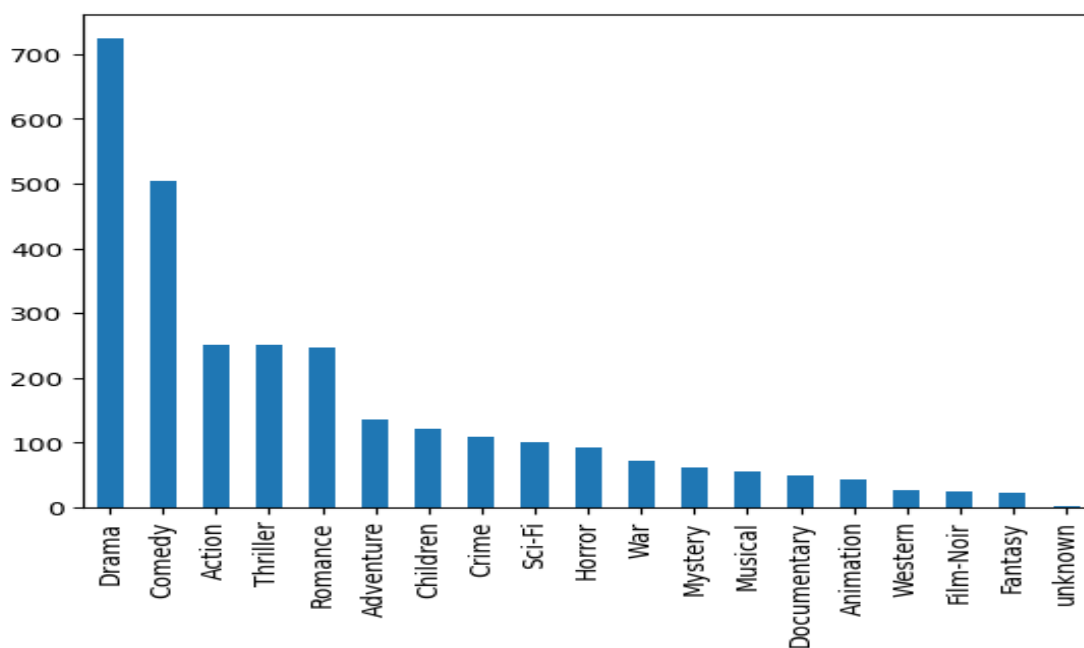
nhiều nhất là từ 20 đến 39 tuổi. Và theo bộ dữ liệu này những người 30 tuổi thường xuyên xem phim và cho đánh giá nhiều nhất.



Hình 3.10: Số lượng người xem theo ngành nghề

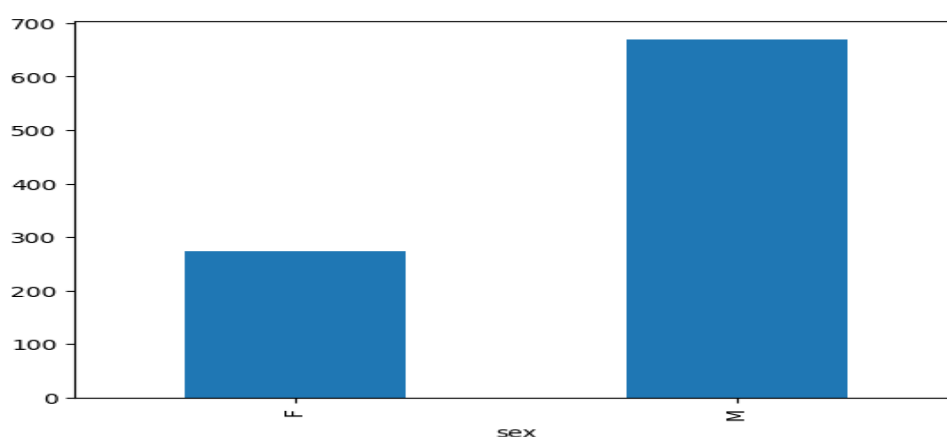
Từ hình 3.10 rút ra được một số nhận xét như sau:

- Ngành nghề student, educator, administrator có tỷ lệ người xem đông nhất. Điều này cho thấy không hẳn cứ là quản lý thì sẽ không có thời gian xem phim. Trái lại họ sắp xếp công việc rất hợp lý nên vẫn có những thời gian giải trí, phim ảnh.



Hình 3.11: Tổng số bộ phim của các thể loại

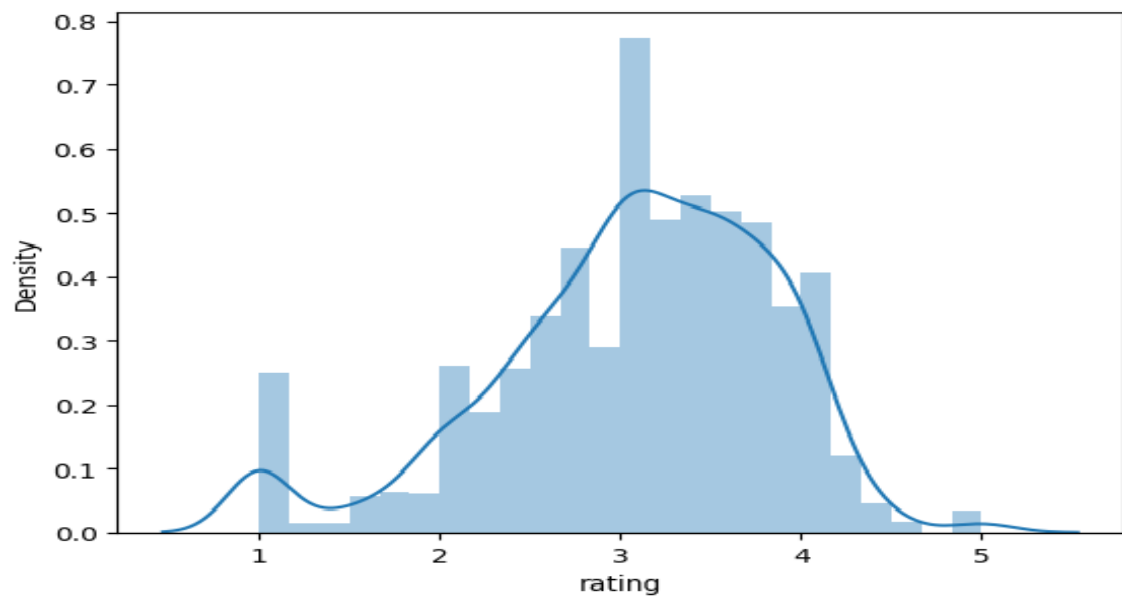
Theo hình 3.11 thì các thể loại phim bao gồm: 'unknown', 'Action', 'Adventure', 'Animation', 'Children', 'Comedy', 'Crime', 'Documentary', 'Drama', 'Fantasy', 'Film-Noir', 'Horror', 'Musical', 'Mystery', 'Romance', 'Sci-Fi', 'Thriller', 'War', 'Western'. Trong đó thể loại Drama có nhiều bộ phim nhất với 725 bộ phim. Và những bộ phim không biết thể loại chiếm 2 bộ phim.



Hình 3.12: Số lượng người xem theo giới tính

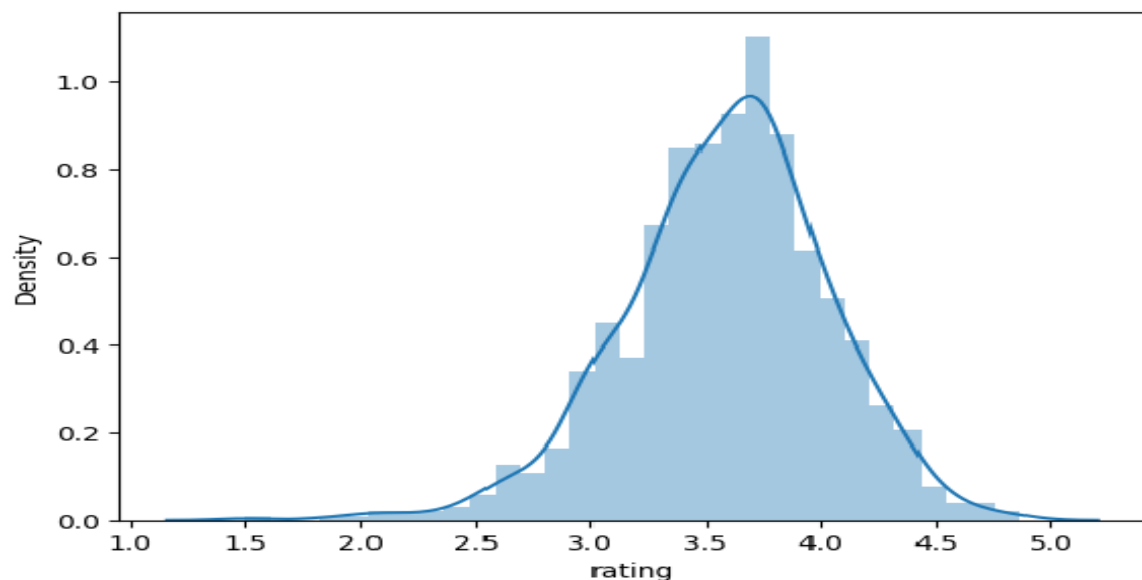
Theo biểu đồ trực quan của bộ dữ liệu ở hình 3.12, số người xem có giới tính nam vào khoảng 670 người nhiều hơn số lượng người xem thuộc

giới tính nữ vào khoảng 273 người.



Hình 3.13: Phân phối trung bình xếp hạng của mỗi bộ phim

Theo biểu đồ trực quan 3.13 ta thấy được, các bộ phim được đánh giá trung bình từ 1 đến 5 sao và số lượng bộ phim được đánh giá trung bình từ 3 đến 4 sao là nhiều nhất. Các bộ phim được đánh giá ở mức 5 sao là tương đối ít.



Hình 3.14: Phân phối trung bình xếp hạng của mỗi người dùng

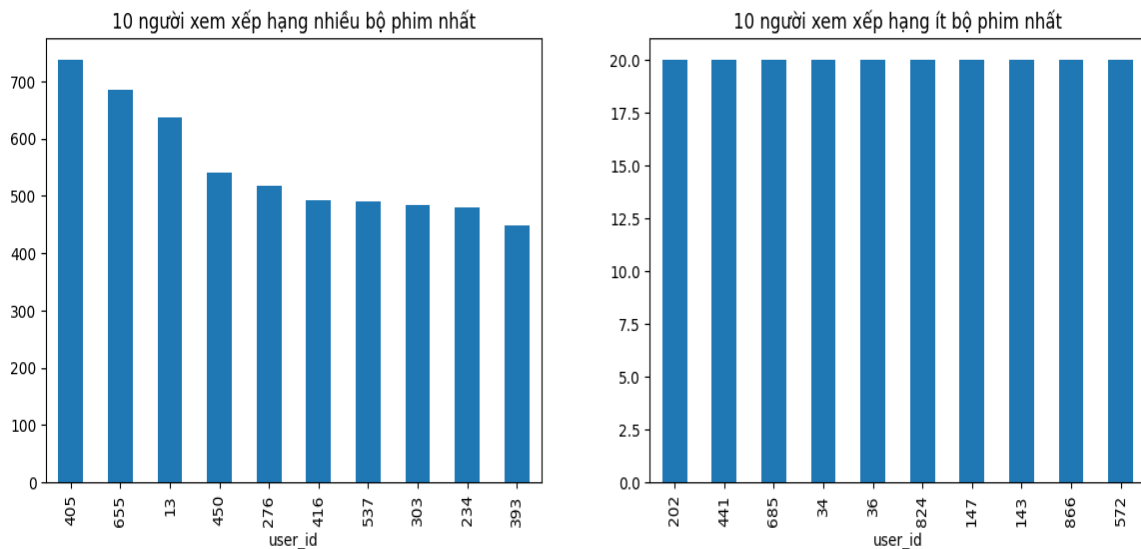
Theo hình 3.14 ta có 2 nhận xét:

- Hầu hết các bộ phim có xếp hạng trung bình trong khoảng từ 3-4.

Cũng có những bộ phim trung bình xếp hạng rất thấp (chỉ 1-2) và một số được xếp hạng khá cao (4-5).

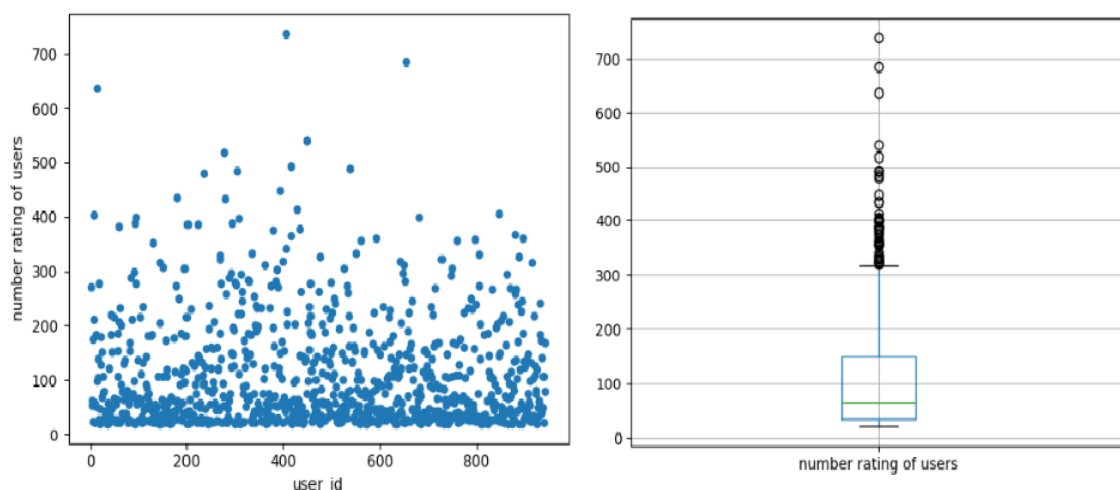
- Các khách hàng cũng phân biệt thành khách hàng khó tính và dễ tính. Đối với khách hàng khó tính, điểm trung bình xếp hạng chỉ nằm trong khoảng từ 2-3 và khách hàng dễ tính là 4-5.

3.2.2.2 Phân tích cơ bản



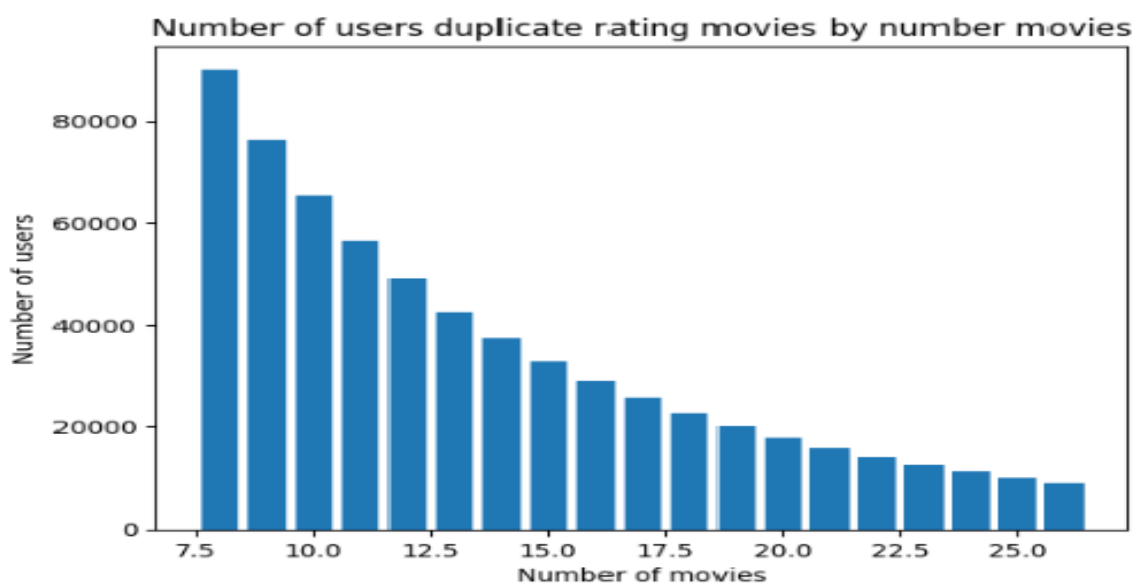
Hình 3.15: 10 người dùng có số lượng xếp hạng nhiều và ít nhất

Dựa vào hình 3.15 ta thấy, người xem xếp hạng nhiều nhất với số lượng trên 700 bộ phim. Những người dùng có số lượng xếp hạng các bộ phim ít hơn 20 đã được làm sạch và loại bỏ từ đầu, nên những người dùng có số lượng xếp hạng thấp nhất bắt đầu từ 20 xếp hạng. Và có khá nhiều người xem có số lượng xếp hạng này.



Hình 3.16: Số lượng xếp hạng của 943 người dùng cho các bộ phim

Từ hình 3.16, số lượng xếp hạng của người dùng bắt đầu từ 200 đến hơn 700 xếp hạng. Trong đó số lượng xếp hạng của người xem nhiều nhất từ khoảng 20 đến 150 bộ phim.



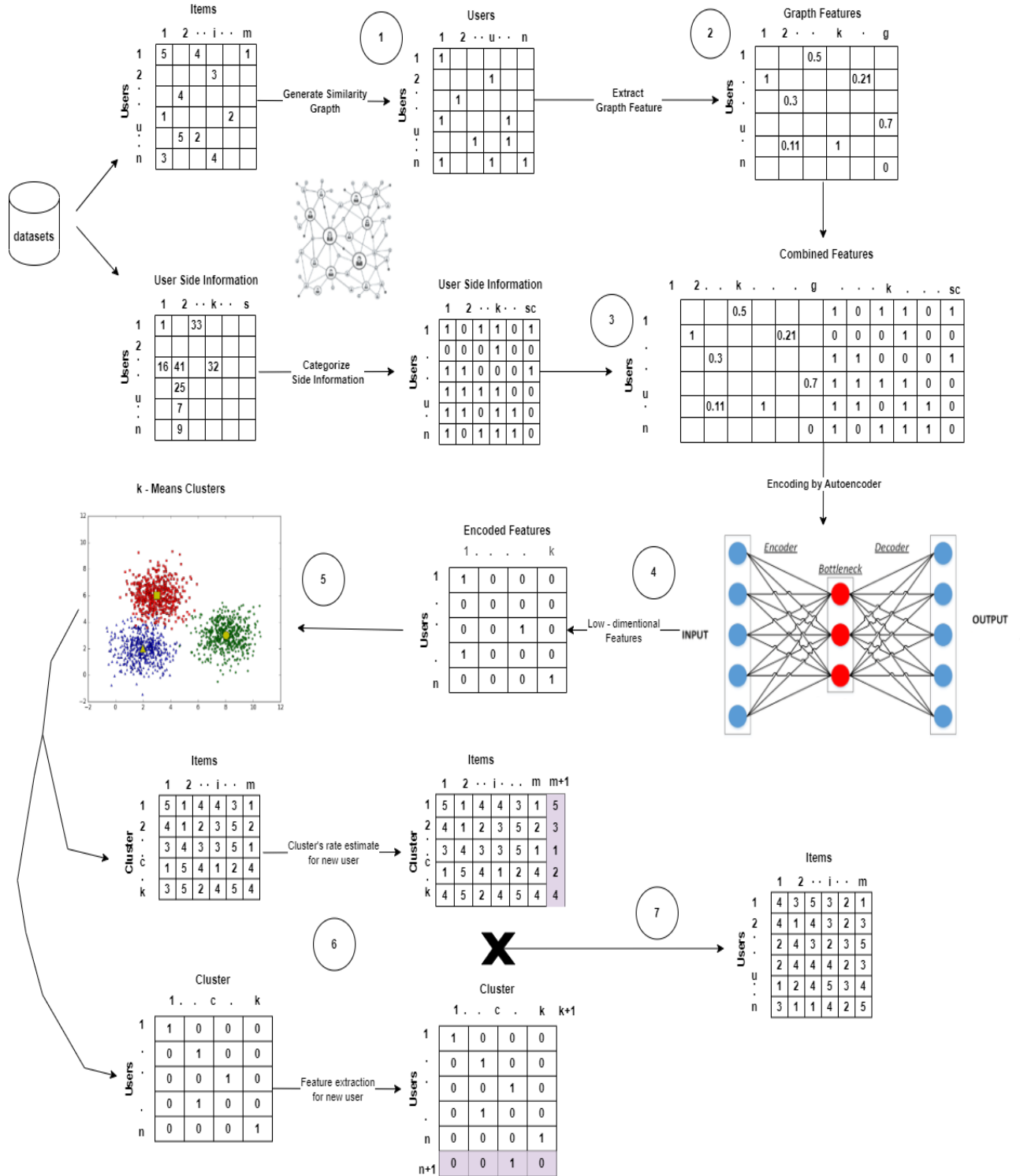
Hình 3.17: Số lượng 2 người xem trùng xếp hạng trên một số bộ phim

Từ biểu đồ 3.17, ta thấy khi chọn số bộ phim từ 8 - 26 bộ phim thì số lượng 2 người xem trong tập dữ liệu trùng xếp hạng khá lớn. Mục đích của việc xác định giá trị này nhằm xác định được thuật toán để tìm 2 người xem có giống nhau không và khả năng xây dựng một đồ thị có sự giống

nhau giữa những người xem với nhau. Điều này sẽ được giải thích kĩ hơn trong phần thực nghiệm mô hình ở phần 4.2 của khóa luận.

3.3 Tổng quan phương pháp tiếp cận

3.3.1 Hệ Thống Đề Xuất Phim Dựa Trên Mô Hình Lai Sử Dụng Đồ Thị Và Bộ Mã Hóa Tự Động



Hình 3.18: Kiến trúc tổng thể của mô hình

Mô hình đề xuất phim của nhóm được minh họa như hình 3.18. Trong đó: đầu vào là các thông tin của người dùng và ma trận các xếp hạng còn thiếu của họ. Đầu ra là một ma trận đầy đủ dự đoán các xếp hạng của họ cho tất cả các bộ phim.

Kiến trúc của mô hình bao gồm 7 bước:

- **Bước 1:** Trong bước đầu tiên, mô hình cần xây dựng một đồ thị với những người dùng như các nút. Hai người dùng sẽ được kết nối dựa trên những đặc điểm giống nhau của họ. Cạnh kết nối một cặp người dùng là những người có sự tương đồng về xếp hạng các bộ phim.
- **Bước 2:** Trong bước thứ hai, một tập hợp thông tin của người dùng sẽ được trích xuất từ đồ thị.
- **Bước 3:** Trong bước thứ ba, mô hình tiến hành kết hợp thông tin phụ như giới tính và độ tuổi với các đặc trưng dựa trên đồ thị ở bước 2 làm đầu vào cho giai đoạn Autoencoder.
- **Bước 4:** Trong bước này, mô hình áp dụng kỹ thuật Autoencoder để trích xuất các đặc trưng mới và giảm kích thước của dữ liệu.
- **Bước 5:** Trong bước này, mô hình sử dụng các đặc trưng mới được mã hóa bởi Autoencoder để phân cụm người dùng, sử dụng thuật toán K-mean để tạo ra một số lượng nhỏ các nhóm người dùng có sự tương đồng.
- **Bước 6:** Trong bước thứ sáu, mô hình sẽ phân người dùng mới vào cụm thích hợp dựa trên các tính năng được mã hóa và dự đoán xếp hạng các mục mới mà người dùng đó chưa xếp hạng.
- **Bước 7:** Trong bước cuối cùng, mô hình dự đoán xếp hạng của người dùng cho tất cả các mục theo xếp hạng trung bình của cụm và tiến hành đề xuất các bộ phim cho người xem.

3.3.2 Thuật toán và mã giả để xây dựng hệ thống

Dưới đây là các bước thực hiện để xây dựng thuật toán theo mô hình 7 bước trong mục 3.3.1 ở trên:

Algorithm 1: Chi tiết thuật toán lai giữa đồ thị và bộ mã hóa tự động

- 1: **Đầu vào:** U, I, R
 - 2: **Đầu ra:** Dự đoán xếp hạng của người dùng cho tất cả các bộ phim.
 - 3: Đặt α = phần trăm của các bộ phim có xếp hạng tương tự giữa hai người xem.
 - 4: Xây dựng đồ thị tương tự và xem xét các người dùng là các nút. Hai người dùng tương tự nhau sẽ được kết nối bằng các cạnh.
 - 5: Trích xuất các đặc trưng của người dùng(nút) dựa trên dữ liệu đồ thị được xây dựng vào bảng F_g .
 - 6: Thông tin phía người dùng được xử lý trước và phân loại vào bảng F_s
 - 7: Kết hợp 2 bảng F_g và F_s thành một bảng F_t chứa thông tin đặc trưng của người dùng. Sử dụng Autoencoder để huấn luyện mô hình với thông số tốt nhất.
 - 8: Trích xuất thông tin đặc trưng của người dùng và giảm chiều của dữ liệu từ Autoencoder vào bảng F_e
 - 9: Tìm hệ số cụm tối ưu để phân loại người dùng và khởi tạo các cụm
 - 10: Phân loại người dùng vào các cụm C thích hợp từ dữ liệu của bảng F_e bằng cách sử dụng thuật toán K-Means.
 - 11: Tạo ma trận người dùng - cụm (UC)
 - 12: Dự đoán xếp hạng của cụm cho tất cả các bộ phim vào ma trận CI
 - 13: **Nếu:** Những người dùng khác trong cụm đã đánh giá cho bộ phim rồi **thì:**
 - 14: CI_{ci} = Trung bình tất cả các xếp hạng của những người dùng đã xếp hạng cho bộ phim ở trong cụm đó
 - 15: **Nếu:** Bộ phim chưa được người xem nào trong cụm đánh giá **thì:**
 - 16: CI_{ci} = Trung bình tất cả các xếp hạng của những bộ phim tương tự bộ phim đó.
 - 17: **Nếu:** Bộ phim chưa được người xem nào trong cụm đánh giá và không có bộ phim nào tương tự như bộ đó **thì:**
 - 18: CI_{ci} = Trung bình tất cả xếp hạng của tất cả người dùng trong cụm đó.
 - 19: Dự đoán xếp hạng của người dùng cho các bộ phim trong ma trận $R' = UC \times CI$
 - 20: Tính toán số lượng phim đề xuất cho người dùng u
-

Chương 4

Kết quả thực nghiệm

4.1 Môi trường thực nghiệm

4.1.1 Tập dữ liệu thực nghiệm

Tập dữ liệu thực nghiệm được sử dụng trong khóa luận là Movielens 100K đã được giới thiệu ở mục 3.2. Đầu vào là các thông tin của người dùng, thông tin của các bộ phim, và xếp hạng của người dùng cho các bộ phim mà người dùng đánh giá. Tập dữ liệu 100.000 xếp hạng này đã được chia sẵn thành 5 tập dữ liệu với cách chia khác nhau để phù hợp cho việc kiểm định mô hình. Trong đó các tập được chia đều tương tự như u1.base và tập u1.test trong đó:

- u1.base: 80000 xếp hạng sử dụng làm tập huấn luyện.
- u1.test: 20000 xếp hạng tiếp theo dùng để kiểm tra.

Tương tự với tập dữ liệu u1.base và u1.test thì các tập dữ liệu còn lại từ u2.base đến u5.base và u2.test đến u5.test cũng được chia theo tỉ lệ 80/20 được lấy từ tập dữ liệu 100000 xếp hạng gốc để làm tập dữ liệu huấn luyện và kiểm tra.

4.1.2 Môi trường thực nghiệm

Toàn bộ quá trình cài đặt huấn luyện và kiểm thử được cài đặt trên Google Colab với CPU là Intel(R) Xeon(R) 2.20GHz, ổ cứng có dung lượng là 108GB, bộ nhớ RAM là 13GB.

4.1.3 Ngôn ngữ và thư viện lập trình

Xuyên suốt quá trình thực hiện đề tài này, nhóm sử dụng Python 3.7 làm ngôn ngữ lập trình chính. Ngoài ra, nhóm sử dụng một số thư viện

như pandas, numpy, sklearn, matplotlib và một số thư viện khác để đọc, trực quan và tiền xử lý dữ liệu cho mô hình. Nhóm xây dựng đồ thị(Graph-based) bằng thư viện Networkx, xây dựng mô hình autoencoder bằng thư viện Pytorch và cuối cùng là sử dụng thuật toán k-Means có sẵn trong thư viện sklearn để phân loại người dùng vào các cụm thích hợp.

4.1.4 Phương pháp kiểm định

Nhóm chúng tôi kiểm định 5 lần trên 5 bộ dữ liệu MovieLens 100K được chia để làm thành tập huấn luyện và thử nghiệm để đo lường hiệu suất của hệ thống đề xuất phim dựa trên mô hình lai sử dụng đồ thị và bộ mã hóa tự động một cách chính xác nhất. Các số liệu dự đoán cuối cùng là trung bình của các lần lặp huấn luyện và thử nghiệm dựa trên 5 bộ dữ liệu được nói ở trên. Tập huấn luyện bao gồm danh sách người dùng với các xếp hạng nhất định, thông tin của người dùng và thông tin phụ của các bộ phim. Chúng tôi sử dụng Root Mean Square Error (RMSE) để tính độ lỗi của mô hình dự đoán xếp hạng.

$$RMSE = \sqrt{\frac{\sum_{u,i}^{u \in U, i \in I} (R_{ui} - R'_{ui})^2}{Number of Ratings}} \quad (4.1)$$

Trong đó U số lượng người dùng, I là số bộ phim, R_{ui} và R'_{ui} là những đánh giá thật của người dùng và đánh giá được dự đoán từ mô hình cho một bộ phim.

Bên cạnh độ đo ở trên, chúng tôi cũng sử dụng Precision và Recall (số liệu phổ biến nhất để đánh giá các hệ thống truy xuất thông tin) là số liệu đánh giá để đo lường độ chính xác của mô hình đề xuất. Precision đo tỷ lệ các đề xuất chính xác cho tổng trên tổng số đề xuất. Recall hiển thị tỷ lệ các đề xuất đúng so với toàn bộ thông tin chính xác. Để làm được điều này, chúng tôi phải tách xếp hạng các bộ phim thành hai loại có ngưỡng trong khi xem xét xếp hạng thực tế của chúng, tức là không thích và có thích để tính Precision và Recall. Về vấn đề này, các mục xếp hạng [1 – 3] được coi là không liên quan và được xếp hạng [4 – 5] là có liên quan. Ngoài ra, các bộ phim trong tập dữ liệu đã được chia thành được chọn và không được chọn dựa trên xếp hạng dự đoán của họ. Do đó, Precision và Recall

của mô hình có thể được định nghĩa là:

$$Precision = \frac{TP}{TP + FP} \quad (4.2)$$

$$Recall = \frac{TP}{TP + FN} \quad (4.3)$$

Trong đó **TP** là True Positive(số bộ phim có liên quan và được chọn), FP là False Positive(số bộ phim đúng được chọn nhưng không liên quan) và FN là False Negative (số bộ phim có liên quan nhưng không được chọn).

4.2 Quá trình xây dựng mô hình thực nghiệm

Phần này sẽ trình bày về quá trình cài đặt và xây dựng mô hình dựa theo thuật toán 3.3.2 và mô hình 3.18 trong mục 3.3 mà nhóm giới thiệu ở trên.

Bước 1: Xây dựng mô hình Graph-based

Ta tiến hành sử dụng u1.base(nhóm lấy tập u1.base để trình bày và tương để có thể xây dựng một biểu đồ tương tự với mỗi người dùng là các nút, các cạnh được kết nối với nhau dựa trên sự tương tự giữa hai người xem. Tập dữ liệu này nhóm sử dụng 3 trường thông tin và loại bỏ trường thông tin về Timestamp vì nó không có ý nghĩa trong việc xây dựng biểu đồ tương tự mà nhóm hướng đến. Nhóm sử dụng các trường thông tin:

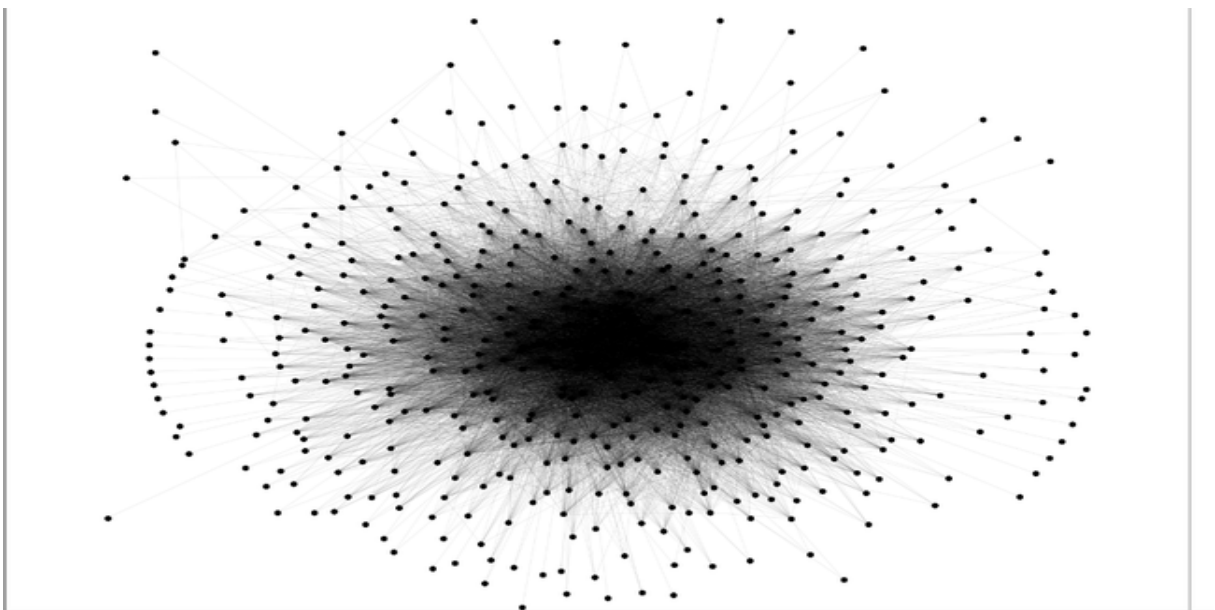
1. UserIDs: là các id của người dùng bắt đầu từ 1 đến 943.
2. MovieIDs: là các id của bộ phim bắt đầu từ 1 đến 1682.
3. Ratings: là giá trị đánh giá của người dùng cho các bộ phim bắt đầu từ 1 đến 5 sao.

Cách tính độ tương tự giữa 2 người xem:

Để xây dựng được đồ thị thì nhóm cần cài đặt một phương thức để tính độ tương tự giữa 2 người xem. Nhóm xác định hai người xem tương tự nhau khi hai người xem đó có số lượng các bộ phim có cùng xếp hạng vượt qua một ngưỡng α mà nhóm xác định.

$\alpha = [0.005, 0.01, 0.015]$ là các ngưỡng mà nhóm tìm hiểu. α này là tỉ lệ các bộ phim được cho là ở mức trùng lặp xếp hạng phù hợp so với toàn bộ các bộ phim có ở trong tập dữ liệu. Tập dữ liệu movielens có tổng cộng 1682 bộ phim tương ứng với α ta sẽ có ngưỡng của các bộ phim lần lượt là 8,16,25 bộ phim. Dựa vào sự phân tích ở mục 3.3, từ 2 biểu đồ 3.17 và 3.16 thì ta thấy được số lượng trùng hợp xếp hạng bắt đầu từ ngưỡng 8 bộ phim là tương đối lớn. Và nó dần được giảm đi khi ngưỡng các bộ phim được tăng lên. Mặc dù có khá nhiều bộ phim và dữ liệu cũng khá thừa thớt thì tỉ lệ trùng hợp xếp hạng giữa 2 người xem trong thực tế có thể là ít xảy ra, nhưng trong bộ dữ liệu này những người dùng ít hơn 20 xếp hạng đã được loại bỏ khỏi tập dữ liệu 100K. Và bộ dữ liệu này cũng thường xuyên được sử dụng để nghiên cứu khác nhau nên có thể có một số sự tương đồng nào đó ở trong bộ dữ liệu này mà tác giả đã xử lí. Vậy nên, số liệu mà nhóm thấy được trong bộ dữ liệu là tương đối hợp lí.

Sau nhiều lần thử nghiệm và trực quan, để tránh sự thừa thớt của biểu đồ và tránh sự overfitting underfitting của mô hình thì nhóm quyết định chọn mức $\alpha = 0.01$ tương ứng là 16 bộ phim làm ngưỡng xét sự tương đồng giữa hai người xem. Tức là, hai người dùng được xem là tương tự nhau khi có từ 16 bộ phim trở lên có cùng xếp hạng. Sau khi xác định được sự tương đồng giữa những người dùng với nhau nhóm tiến hành xây dựng biểu đồ với sự tương tự giữa người dùng với nhau.



Hình 4.1: Biểu đồ 943 người dùng với $\alpha = 0.01$

Bước 2: Trích xuất đặc trưng của người dùng từ đồ thị

Mô hình trích xuất đặc trưng của người dùng từ biểu đồ bằng các độ đo Page rank, Degree Centrality, Closeness Centrality, Betweenness Centrality, Load Centrality, Average Neighbor Degree đã được giới thiệu ở mục 3.1.1.

Bước 3: Trích xuất đặc trưng và giảm chiều dữ liệu của người dùng bằng Autoencoder

Ở bước này mô hình sẽ sử dụng tập u.user. Tập dữ liệu này nhóm sử dụng 4 trường thông tin và loại bỏ trường thông tin về Zip-code vì nó không có ý nghĩa về đặc trưng của người dùng. Trong tập tin gồm:

1. Gender: giới tính nhận 2 giá trị F là nữ, M là nam.
2. Age: khoảng tuổi của khách hàng.
3. Occupation: Nghề nghiệp của người dùng

Tiền xử lí dữ liệu trước khi đưa vào mô hình Autoencoder

1. Chia độ tuổi của thành các khoảng độ tuổi [0, 10, 20, 30, 40, 50, 100] tương ứng với [1,2,3,4,5,6]. Sau đó tiến hành OneHotEncoder các trường dữ liệu giới tính, nghề nghiệp, độ tuổi. Onehotencoder có chức năng đưa các trường dữ liệu về một vector. Với mỗi trường dữ liệu người đó thuộc về sẽ có chỉ số là 1 còn các chỉ số khác là 0.
2. Nhóm tiến hành ghép những đặc trưng được trích xuất từ biểu đồ và các trường dữ liệu của người dùng đã được OneHotEncoder thành một bảng chứa đặc trưng của 943 người dùng trong tập dữ liệu movie-lens 100K. Từ đây, đặc trưng của người dùng sẽ được biểu diễn dưới 35 chiều từ 29 chiều của OneHotEncoder(2 chiều của giới tính, 21 chiều của nghề nghiệp, 5 chiều của độ tuổi) và 6 chiều là độ đo được trích xuất từ đồ thị tương tự của người dùng 4.2. Cuối cùng là điền những chỗ khuyết cho bảng bằng giá trị '0', vì autoencoder chỉ chạy được khi dữ liệu không bị khuyết.

	age1	age2	age3	age4	age5	age6	gender1	gender2	job1	job2	...	job18	job19	job20	job21	PR	CD	CC	CB	LC	AND
0	0.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	...	0.0	0.0	1.0	0.0	0.137604	0.163380	0.652632	0.000088	0.000089	0.868979
1	0.0	0.0	0.0	0.0	0.0	1.0	1.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
2	0.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	...	0.0	0.0	0.0	1.0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
3	0.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	...	0.0	0.0	1.0	0.0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
4	0.0	0.0	0.0	1.0	0.0	0.0	1.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.047768	0.028169	0.570377	0.000000	0.000000	0.801968
...
938	0.0	0.0	1.0	0.0	0.0	0.0	1.0	0.0	0.0	0.0	...	0.0	1.0	0.0	0.0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
939	0.0	0.0	0.0	1.0	0.0	0.0	0.0	1.0	1.0	0.0	...	0.0	0.0	0.0	0.0	0.098709	0.104225	0.639835	0.000041	0.000041	0.868993
940	0.0	1.0	0.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	...	0.0	1.0	0.0	0.0	0.000000	0.000000	0.000000	0.000000	0.000000	0.000000
941	0.0	0.0	0.0	0.0	1.0	0.0	1.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.057652	0.042254	0.595010	0.000000	0.000000	0.868531
942	0.0	0.0	1.0	0.0	0.0	0.0	0.0	1.0	0.0	0.0	...	0.0	1.0	0.0	0.0	0.260785	0.343662	0.715110	0.004526	0.004515	0.709226

943 rows × 35 columns

Hình 4.2: Đặc trưng của 943 người dùng

Bước 4: Trích xuất đặc trưng với mô hình Autoencoder

Sau khi có bộ dữ liệu đặc trưng 35 chiều từ bước tiền xử lí 4.2. Các đặc trưng sẽ được đưa vào mô hình Autoencoder để trích xuất những đặc trưng nhất của người dùng và giảm chiều của dữ liệu thuận lợi cho việc tính toán phía sau. Để tránh việc overfitting cho mô hình nhóm đã sử dụng Autoencoder Denoising 3.1.4.4, và sử dụng Elastic regularization (mục 3.1.5 đã được nhóm trình vào mô hình để đào tạo tập huấn luyện.

Để kiểm tra mô hình autoencoder thì sẽ lấy 20% users từ dữ liệu đang có làm tập kiểm thử cho autoencoder, còn lại 80% đem đi huấn luyện.

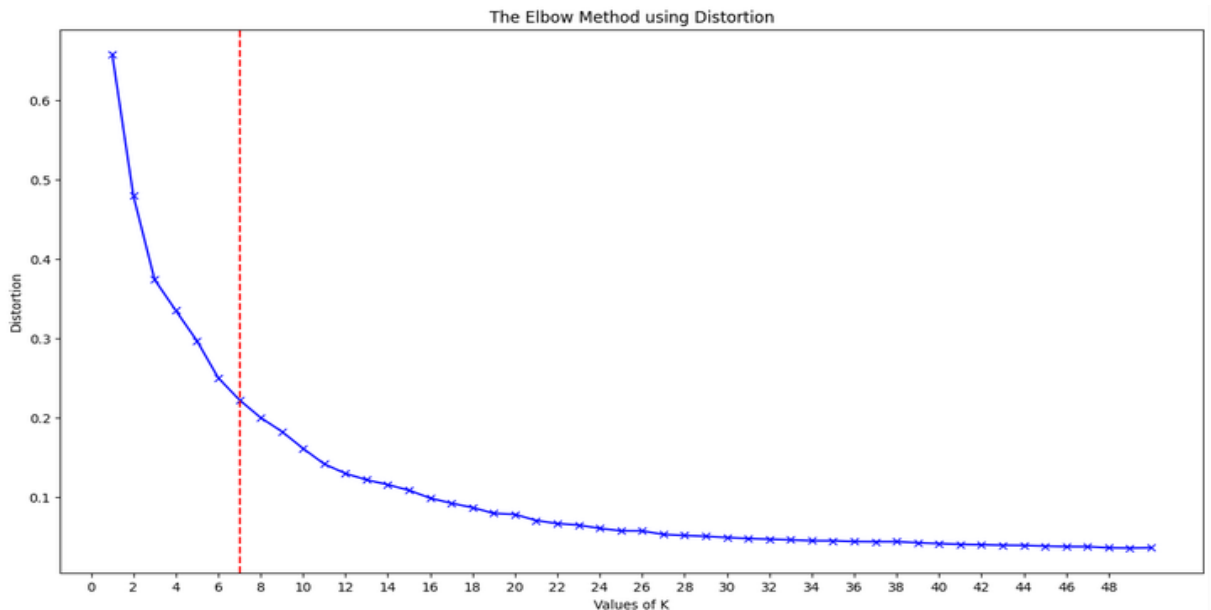
Ở mô hình này nhóm cài đặt như sau

- Số tầng : 5 tầng gồm 1 input, 2 hidden layer (pha encode và pha decode) , 1 bottleneck, 1 output
- Hàm lỗi : Mean squared error
- Hàm kích hoạt : toàn bộ đều là ReLu
- Kết quả bottleneck : 4 chiều

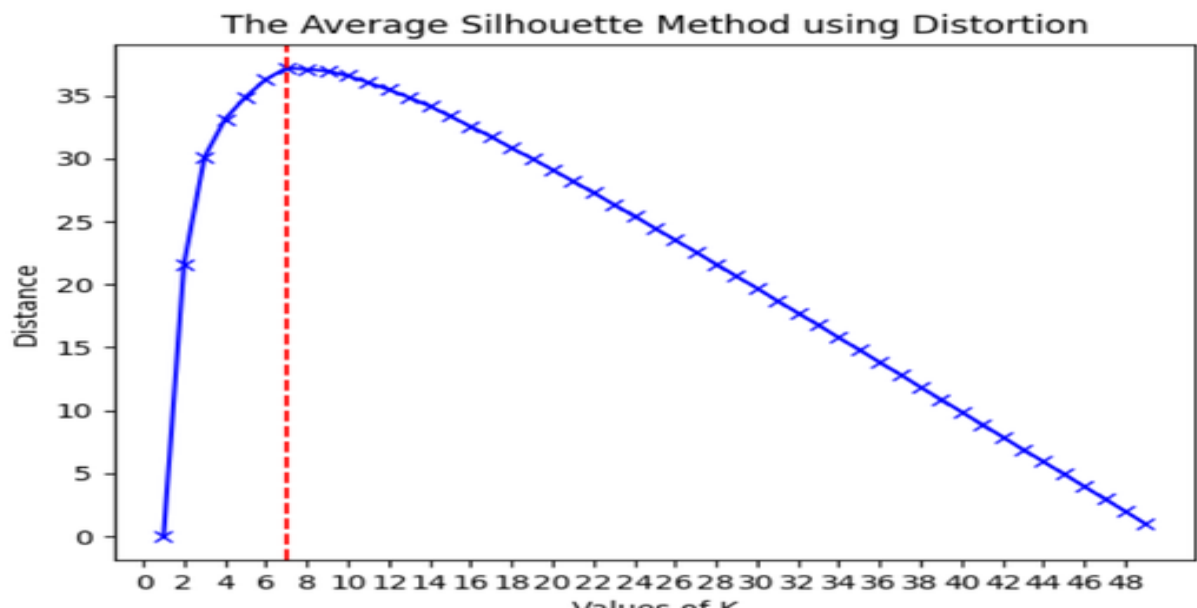
Bước 5: Sử dụng K-Means để phân loại người dùng theo cụm

Sau khi rút trích được những đặc trưng nhất của người dùng từ mô hình Autoencoder. Mô hình sẽ sử dụng thuật toán K-Means để phân loại

người dùng thành các cụm. Bằng cách sử dụng phương pháp Elbow(mục 3.1.6.4) và phương pháp Silhouette(mục 3.1.6.5) đã giới thiệu, ta sẽ chọn ra được điểm k tối ưu nhất cho thuật toán. Ta trực quan hóa dữ liệu khi được phân vào các cụm để có thể xem được Distort score on Elbow và Distance of Average Silhouette như trong 2 hình 4.3 và 4.4. Dựa vào sự trực quan này, chúng ta sẽ chọn được số cụm cho thuật toán k-Means của mô hình. Trong trường hợp này thì 7 sẽ là số cụm được chọn.



Hình 4.3: Distort score on elbow



Hình 4.4: Distance of Average Silhouette

Sau khi xác định được hệ số k tối ưu mô hình sẽ gom những người xem

vào cụm thích hợp dựa vào thuật toán k-means.4.5

```
Cluster 0: 99 users
Cluster 1: 156 users
Cluster 2: 122 users
Cluster 3: 131 users
Cluster 4: 61 users
Cluster 5: 219 users
Cluster 6: 155 users
```

Hình 4.5: Số lượng người dùng trong mỗi cụm

Bước 6: Dự đoán xếp hạng của người dùng Mô hình tiếp tục tách từ kết quả của thuật toán k-Means thành hai ma trận khác nhau, mục đích hỗ trợ cho việc dự đoán xếp hạng của người dùng.

	cluster0	cluster1	cluster2	cluster3	cluster4	cluster5	cluster6
1	1.0	0.0	0.0	0.0	0.0	0.0	0.0
2	0.0	0.0	0.0	0.0	1.0	0.0	0.0
3	0.0	0.0	0.0	0.0	1.0	0.0	0.0
4	1.0	0.0	0.0	0.0	0.0	0.0	0.0
5	0.0	0.0	0.0	0.0	0.0	0.0	1.0
...
939	0.0	0.0	0.0	0.0	1.0	0.0	0.0
940	0.0	1.0	0.0	0.0	0.0	0.0	0.0
941	0.0	0.0	0.0	1.0	0.0	0.0	0.0
942	0.0	0.0	0.0	0.0	0.0	1.0	0.0
943	1.0	0.0	0.0	0.0	0.0	0.0	0.0

943 rows × 7 columns

Hình 4.6: Ma trận cụm người dùng - cụm

- Ma trận thứ nhất (hình 4.6) chứa thông tin của user-cluster với user thuộc cluster nào thì sẽ được đánh dấu là '1' các cột cluster còn lại là '0'.

	1	2	3	4	5	6	7	8	9	10	...	1673	1674	1675	1676	1677	1678	1679	1680	1681	1682
cluster0	4.0	2.947368	2.533333	3.433333	3.0	2.666667	3.939394	3.807692	3.625	4.166667	...	3.413534	3.745846	3.745846	3.745846	3.0	1.0	3.0	2.0	3.225352	3.745846
cluster1	3.86747	3.230769	2.285714	3.527778	3.285714	3.857143	3.75	4.081081	3.862069	4.2	...	3.321429	4.0	3.685406	3.685406	3.685406	3.685406	3.6	3.625	3.0	3.685406
cluster2	3.631579	3.0	5.0	3.605014	3.75	5.0	3.5625	3.333333	3.666667	4.25	...	3.438776	3.935622	3.935622	3.935622	3.935622	3.935622	4.0	3.847222	3.359606	3.935622
cluster3	3.840426	3.366667	3.045455	3.631579	3.157895	3.666667	3.657895	4.261905	3.953125	3.681818	...	3.0	3.730891	3.0	2.0	3.730891	3.730891	3.730769	3.64117	3.294462	3.730891
cluster4	4.0	3.057143	3.25	3.608696	3.4	2.6	3.984375	3.783784	3.782609	3.857143	...	3.381679	3.557272	3.557272	3.557272	3.557272	3.557272	3.4	3.551331	3.192287	3.0
cluster5	3.88	3.75	4.0	3.083333	2.833333	4.0	4.05	4.0	3.619048	3.5	...	3.482234	3.600457	3.600457	3.600457	3.600457	3.600457	4.0	3.57947	3.22366	3.600457
cluster6	3.864865	3.333333	3.25	3.555556	4.0	3.809886	3.666667	4.0	3.931034	3.0	...	3.71875	3.809886	3.809886	3.809886	3.809886	3.809886	3.666667	3.783784	3.404959	3.809886

7 rows × 1682 columns

Hình 4.7: Ma trận cụm-bộ phim đã dự đoán đầy đủ xếp hạng

- Ma trận thứ hai (hình 4.7) chứa dự đoán cho các bộ phim của mỗi cụm cho các bộ phim.

Trong bước này hệ thống chia làm 3 trường hợp để dự đoán xếp hạng của các cụm cho các bộ phim.

1. Nếu bộ phim chưa có xếp hạng và đã được những người dùng tương tự khác ở trong cụm xếp hạng rồi. Mô hình sẽ tiến hành lấy trung bình tất cả xếp hạng của những người trong cụm đó làm xếp hạng của cụm cho bộ phim này.
2. Nếu bộ phim đó chưa có người nào ở trong cụm xếp hạng. Hệ thống sẽ tiến hành tìm những bộ phim tương tự trong tập dữ liệu u.item dựa vào thể loại. Hai bộ phim giống nhau khi có cùng thể loại với nhau. Sau khi tìm được những bộ phim có thể loại tương tự nhau. Xếp hạng chưa biết của bộ phim này sẽ bằng trung bình tất cả xếp hạng các bộ phim tương tự với bộ phim này.
3. Nếu bộ phim đó không có người nào trong cụm từng xem, không có các bộ phim tương tự hoặc những bộ phim tương tự cũng không có xếp hạng, thì xếp hạng của bộ phim này sẽ bằng trung bình tất cả xếp hạng của những người dùng ở trong cụm.

Sau các bước điền những giá trị xếp hạng của các cụm cho các bộ phim thì ta có một ma trận dự đoán giữa các cụm cho các bộ phim với đầy đủ tất cả các xếp hạng 4.7

Bước 7: Tính toán xếp hạng của tất cả người xem và hoàn thành hệ thống

Sau khi đã có 2 tập dữ liệu thì thực hiện tính dự đoán cho đánh giá của người xem bằng cách nhân 2 tập dữ liệu đã có ở bước 6 lại với nhau theo cách nhân 2 ma trận. Ở bước này sau khi hoàn thành thì tất cả các xếp hạng của các bộ phim chưa được đánh giá của người dùng cũ và mới đều được dự đoán 4.8.

	1	2	3	4	5	6	7	8	9	10	...	1673	1674	1675	1676	1677	1678	1679	1680	1681	1682
1	3.821429	2.888889	4.4	3.0	3.5	5.0	3.916667	3.75	3.789474	2.6	...	3.342246	3.726755	3.726755	3.726755	3.726755	3.726755	3.833333	3.716418	3.276688	3.726755
2	3.884058	3.24	2.75	3.5	3.294118	3.571429	3.833333	4.113636	4.078431	3.96	...	3.431118	4.0	3.741872	3.741872	3.741872	3.741872	3.869565	3.630836	3.327549	3.741872
3	3.761905	3.0	3.0	3.428571	3.5	3.5	3.47619	4.111111	3.125	3.25	...	3.33	3.269907	3.0	2.0	3.269907	3.269907	3.2	3.286207	2.962193	3.269907
4	3.521739	3.0	3.666667	5.0	3.333333	3.626667	3.55	4.166667	3.909091	3.0	...	3.512397	3.626667	3.626667	3.626667	3.626667	3.626667	3.8	3.78453	3.415441	3.626667
5	3.860759	3.269231	3.24	3.382353	3.4	3.8	3.802469	3.902439	3.840909	4.125	...	3.462848	3.677536	3.677536	3.677536	3.0	1.0	3.0	2.0	3.219738	3.677536
...
939	3.521739	3.0	3.666667	5.0	3.333333	3.626667	3.55	4.166667	3.909091	3.0	...	3.512397	3.626667	3.626667	3.626667	3.626667	3.626667	3.8	3.78453	3.415441	3.626667
940	3.884058	3.24	2.75	3.5	3.294118	3.571429	3.833333	4.113636	4.078431	3.96	...	3.431118	4.0	3.741872	3.741872	3.741872	3.741872	3.869565	3.630836	3.327549	3.741872
941	3.521739	3.0	3.666667	5.0	3.333333	3.626667	3.55	4.166667	3.909091	3.0	...	3.512397	3.626667	3.626667	3.626667	3.626667	3.626667	3.8	3.78453	3.415441	3.626667
942	3.894737	3.5	2.75	3.615385	3.25	3.5	4.0	4.0625	4.136364	3.8	...	3.401274	3.830527	3.830527	3.830527	3.830527	3.830527	3.909091	3.681957	3.525	3.830527
943	3.961538	3.142857	2.72	3.784314	3.052632	2.333333	3.88	4.0	3.880952	3.583333	...	3.382234	3.72171	3.72171	3.72171	3.72171	3.72171	3.380952	3.561644	3.0	3.72171

943 rows × 1682 columns

Hình 4.8: Ma trận người dùng - phim đã được dự đoán

Sau khi có hoàn thành được bảng dự đoán. Hệ thống sẽ sử dụng tập dữ liệu `u1.test` để thực hiện đo độ chính xác của hệ thống và đưa ra được một số so sánh với các hệ thống đề xuất khác.

4.3 Kết quả mô hình và so sánh

4.3.1 Kết quả so sánh với các phương pháp cơ bản

Tập dữ liệu	u1	u2	u3	u4	u5	Mean
RMSE	1.0868	1.069	1.0729	1.06355	1.071	1.0745
PRECISION	0.671	0.65	0.686	0.713	0.60	0.6744
RECALL	0.716	0.799	0.726	0.775	0.77	0.756
MSE OF AUTOENCODER	0.074	0.0754	0.0742	0.0735	0.07418	0.0738

Bảng 4.1: Các độ lỗi ở trong mô hình lai giữa đồ thị và bộ mã hóa tự động của 5 tập dữ liệu

Nhận xét: Từ bảng 4.1 ta thấy mô hình lai giữa đồ thị và bộ mã hóa tự động cho độ lỗi tương đối ổn định, không có sự chênh lệch giữa các tập dữ liệu khác nhau.

METHOD	RMSE
<i>Graph-based with Autoencoder</i>	1.0745
<i>User-Based</i>	1.56
<i>Item-Based</i>	2.84

Bảng 4.2: So sánh độ lỗi với các phương pháp cơ bản

Nhận xét: Từ bảng 4.2 ta thấy sự vượt trội hơn hẳn của phương pháp kết hợp so với những mô hình truyền thống như User-based hay Item-based.

4.3.2 Kích bản cho vấn đề khởi động nguội

Ngoài phương pháp chia dữ liệu theo bộ dữ liệu pháp xếp hạng của người dùng cho các bộ phim. Nhóm thực hiện chia dữ liệu theo người dùng để kiểm tra thông số xử lý của hệ thống cho người dùng mới. Tức là, nhóm sẽ xóa dữ liệu xếp hạng của 20% người xem, chỉ để lại thông tin về nhân khẩu học của người dùng đó. Sau đó đưa những người dùng này vào để hệ thống dự đoán tất cả xếp hạng của những người xem này. Do không có nhiều tài liệu liên quan đến xử lý vấn đề khởi động nguội, nên nhóm sử dụng độ đo MAE để so sánh với hai vấn đề xử lý khởi động nguội của hai phương pháp cơ bản là User-based collaborative Filtering và

Item-Based Collaborative Filtering trong tài liệu tham khảo mà nhóm tìm hiểu được.[7]

METHOD	MAE
<i>Graph-based with Autoencoder</i>	0.845
<i>User-Based Collaborative Filtering</i>	3.36
<i>Item-Based Collaborative Filtering</i>	3.56

Bảng 4.3: Độ lỗi của các phương pháp cho vấn đề khởi động nguội

Nhận xét: So sánh về vấn đề khởi động nguội cho những người xem mới, mô hình lai kết hợp giữa đồ thị và bộ mã hóa tự động cho độ lỗi khá tốt và vượt trội so với những phương pháp cơ bản hiện có.

4.4 Thảo luận

- So với các mô hình truyền thống thì mô hình Graph-based kết Autoencoder đạt hiệu quả cao hơn đối với tập dữ liệu gặp vấn đề thiếu dữ liệu. Đồng thời cũng đạt kết quả tương đối tốt đối với tình trạng khởi động nguội.
- Mô hình áp dụng được dữ liệu ngoài lề để tăng thêm hiệu quả và giải quyết vấn đề đưa ra.
- Tuy nhiên vẫn chưa tái tạo lại thành công về độ lỗi RMSE của bài báo gốc.
- Có thể thấy mô hình tích hợp nhiều kĩ thuật khác nhau và các thông tin khác nhau nhưng ở phần dự đoán đánh giá thì lại khá đơn giản
- Nhóm cũng đã đọc các bài báo khác liên quan đến phân cụm dữ liệu cho tập movielen 100K thì thấy được việc phân cụm đạt hiệu quả không cao bằng các mô hình model-based collaborative. Các mô hình đạt hiệu quả cao nhất ở tập movielen 100K thì thường dễ thấy sẽ có nhiều biến số hơn như bias của người dùng, bias của sản phẩm và liên kết thẳng tới việc đưa ra dự đoán.
- Theo các thông tin trên thì nhóm kết luận với cách tính dự đoán thì không thể đạt hiệu quả cao nhất mà bài báo gốc đưa ra.

Chương 5

Kết luận và hướng phát triển

5.1 Kết luận

Ý tưởng chính của phương pháp được đề xuất là tìm mối quan hệ giữa những người dùng dựa trên sự tương đồng của họ và biểu diễn dưới dạng các nút trong biểu đồ, đồng thời kết hợp chúng với thông tin phía người dùng để giải quyết vấn đề khởi động nguội. Thêm vào đó, chúng tôi đã áp dụng Autoencoder để trích xuất các đặc trưng của người xem với số chiều thấp hơn và nhiều thông tin hơn. Điều này làm bước phân cụm cuối cùng chính xác hơn và hiệu suất cao hơn.

Khóa luận đã trình bày tổng quan về bài toán đề xuất cũng như vai trò của bài toán trong xã hội hiện nay. Bên cạnh đó, nhóm đã kiểm nghiệm phương pháp sử dụng mô hình lai kết hợp giữa 3 mô hình Graph-based, Autoencoder và k-Means. Đồng thời nhóm đã tự cài đặt lại mô hình để thử nghiệm với bộ dữ liệu Movielens 100K. Kết quả của hệ thống đề xuất cho kết quả tương đối tốt so với các hệ thống đề xuất truyền thống. Kiến trúc này có thể tận dụng được đồng thời rất nhiều các thông tin cả về phía user, sản phẩm và lịch sử tương tác giữa user và sản phẩm. Do đó thông tin chúng học được là các véc tơ user và items có tính cá nhân hóa cao. Không chỉ riêng trong đề xuất các bộ phim, có thể sử dụng mô hình cho mọi hệ thống đề xuất có sử dụng được thông tin phụ của người dùng và vật phẩm.

5.2 Hướng phát triển

Trong tương lai, khóa luận cần cải thiện tốc độ huấn luyện của thuật toán và cải tiến các phương pháp tìm các hệ số tối ưu cho hệ thống. Xây dựng hệ thống đề xuất cho nhiều loại dữ liệu khác ở những thống khác nhau

ngoài lĩnh vực phim điện ảnh. Ngoài ra quá trình xây dựng và triển khai thuật toán sẽ gặp phải những khó khăn nhất định về dữ liệu, huấn luyện, dự báo mà nhóm cần phải khắc phục, đặc biệt là đối với những hệ thống lớn từ một triệu cho tới vài triệu người dùng. Bên cạnh đó, Autoencoder có thể là một lĩnh vực quan trọng cho nghiên cứu trong tương lai của nhóm nhằm chọn lựa được cấu trúc cấu trúc Autoencoder tốt nhất nhằm tăng khả năng trích xuất đặc trưng, giảm thời lượng đào tạo và cuối cùng là tăng hiệu suất và kết quả cho cả hệ thống đề xuất.

Tài liệu tham khảo

Tiếng Anh

- [1] Alejandro Bellogín, Pablo Castells. “A Performance Prediction Approach to Enhance Collaborative Filtering Performance”. In: *Advances in Information Retrieval, 32nd European Conference on IR Research, ECIR 2010, Milton Keynes, UK, March 28-31, 2010* (2010). URL: https://www.researchgate.net/publication/221397285_A_Performance_Prediction_Approach_to_Enhance_Collaborative_Filtering_Performance.
- [2] Amine, Amrani. *PageRank algorithm, fully explained*. URL: <https://towardsdatascience.com/pagerank-algorithm-fully-explained-dc794184b4af> (visited on 06/21/2020).
- [3] Bhasin, Jatin. *Graph Analytics — Introduction and Concepts of Centrality*. URL: <https://towardsdatascience.com/graph-analytics-introduction-and-concepts-of-centrality-8f5543b55de3> (visited on 06/21/2023).
- [4] Diamantaras, Konstantinos et al. “A Graph-based Method for Session-based Recommendations”. In: (2021), pp. 185–207. URL: <https://arxiv.org/abs/2106.12085>.
- [5] Donovan, Elizabeth A. and Barrus, Michael D. “Neighborhood degree lists of graphs”. In: (2018).
- [6] Ferreira, Diana, Sil, Sofia, and Machado, António Abelha and José. “Recommendation System Using Autoencoders”. In: (2020). URL: <https://www.mdpi.com/2076-3417/10/16/5510#>.
- [7] Mahdavi, Mehregan. “A New Collaborative Filtering Algorithm Using K-means Clustering and Neighbors’ Voting”. In: *11th International Conference on Hybrid Intelligent Systems, HIS 2011, Melacca,*

- Malaysia* (2011). URL: <https://www.researchgate.net/publication/220980957>.
- [8] Minnesota, University of. *Movielens Dataset 100k, 1M, 10M, 25M Rating*. URL: <https://grouplens.org/datasets/movielens>. (visited on 06/21/2020).
 - [9] Muhammad, Asad Syed. “AI Movies Recommendation System Based on K-Means Clustering Algorithm”. In: (2020), pp. 15–167. URL: <https://asdkazmi.medium.com/ai-movies-recommendation-system-with-clustering-based-k-means-algorithm-f04467e02fcd>.
 - [10] Na, Shi, ., Liu Xumin, and Yong, Guan. “Research on k-means Clustering Algorithm: An Improved k-means Clustering Algorithm”. In: (2010), pp. 15–167. URL: <https://ieeexplore.ieee.org/document/5453745>.
 - [12] Online. *Elastic net regularization*. URL: https://en.wikipedia.org/wiki/Elastic_net_regularization (visited on 06/22/2015).
 - [13] Online. *Load Centrality*. URL: https://www.centiserver.org/centrality/Load_Centrality/ (visited on 06/21/2023).
 - [14] Online. *Phân loại neural network*. URL: <https://tnquangblog.wordpress.com/2021/02/18/phan-loai-neural-network/> (visited on 06/21/2023).
 - [15] P.S, Sajisha, V.S, bAnoop, and K.A, cAnsal. “Knowledge Graph-based Recommendation Systems:” in: (2019).
 - [16] Rishabh Ahuja Arun Solanki, Anand Nayyar. “Movie Recommender System Using K-Means Clustering AND K-Nearest Neighbor”. In: *2019 9th International Conference on Cloud Computing, Data Science Engineeringa* (2019). URL: https://www.researchgate.net/publication/334763301_Movie_Recommender_System_Using_K-Means_Clustering_AND_K-Nearest_Neighbor.
 - [11] Sublime, Jeremie. *Automatic Post-Disaster Damage Mapping Using Deep-Learning Techniques for Change Detection: Case Study of the Tohoku Tsunami*. URL: <https://www.researchgate.net/figure/>

Basic-architecture-of-a-single-layer-autoencoder-made-of-an-encoder-going-from-the-input_fig3_333038461 (visited on 06/21/2023).

- [17] Wang, Ting-Hsiang et al. “AutoRec: An Automated Recommender System”. In: (2020). URL: <https://arxiv.org/abs/2007.07224>.