

BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI

NGUYỄN NGỌC LONG

ĐA, KLTN ĐẠI HỌC/ CAO ĐẲNG KHOA HỌC MÁY TÍNH

NGHIÊN CỨU VÀ ỨNG DỤNG PHƯƠNG PHÁP
COLLABORATIVE FILTERING VÀ CONTENT-BASED
FILTERING TRONG XÂY DỰNG MÔ HÌNH KHUYẾN NGHỊ
PHIM

KHOA HỌC MÁY TÍNH

CBHD: ThS. Đặng Quỳnh Nga
Sinh viên: Nguyễn Ngọc Long
Mã số sinh viên: 2020601627

Hà Nội – Năm 2024

BỘ CÔNG THƯƠNG
TRƯỜNG ĐẠI HỌC CÔNG NGHIỆP HÀ NỘI

ĐA, KLTN ĐẠI HỌC/ CAO ĐẲNG KHOA HỌC MÁY TÍNH
NGHIÊN CỨU VÀ ỨNG DỤNG PHƯƠNG PHÁP COLLABORATIVE
FILTERING VÀ CONTENT-BASED FILTERING TRONG XÂY DỰNG
MÔ HÌNH KHUYẾN NGHỊ PHIM

CBHD: ThS. Đặng Quỳnh Nga
Sinh viên: Nguyễn Ngọc Long
Mã số sinh viên: 2020601627

Hà Nội – Năm 2024

LỜI NÓI ĐẦU

Cùng với sự phát triển mạnh mẽ của mạng internet đời sống con người ngày càng được nâng cao cả về vật chất lẫn tinh thần. Về nhu cầu vật chất, hiện nay, chúng ta đang trải qua thời kỳ khi nhu cầu về đồ ăn, trang phục không chỉ đơn thuần là đầy đủ mà còn là sự thưởng thức. Đồ ăn không chỉ đáp ứng nhu cầu dinh dưỡng của con người mà còn trở thành một trải nghiệm mới về hương vị và nền văn hóa của các nước khác nhau. Trang phục không chỉ mang lại sự ấm áp mà còn thể hiện phong cách của từng cá nhân.

Còn về mặt tinh thần, nhu cầu giải trí của con người được nâng cao rõ rệt, từ các lễ hội truyền thống, đến các khu vui chơi mua sắm hay đơn giản là xem phim, nghe nhạc, chơi game... Chính vì thế hiện nay người ta không cần phải đi tới trực tiếp các rạp chiếu phim mà vẫn có thể tận hưởng trọn vẹn bộ phim đó ngay tại căn nhà của mình chỉ bằng cách tìm kiếm trên các trang mạng.

Tuy nhiên, điều này đặt ra một câu hỏi quan trọng: “Làm sao để tìm kiếm những bộ phim thú vị như vậy?” hay là “Tìm kiếm thể loại phim đó ở đâu”. Chính lẽ đó hệ thống gợi ý phim theo sở thích đã ra đời áp dụng các công nghệ tiên tiến, hiện đại của các ngành học như học máy, trí tuệ nhân tạo, khai phá dữ liệu... để giúp người xem có thể dễ dàng tìm kiếm bộ phim mà họ muốn xem dựa vào sở thích.

Hiện nay, không chỉ ở Việt Nam mà còn trên thế giới, các ứng dụng của lĩnh vực trí tuệ nhân tạo, học máy, xử lý ảnh hay khai phá dữ liệu được áp dụng rộng rãi trong các lĩnh vực có thể kể đến như: trong giáo dục, ứng dụng trong công nghệ điểm danh học sinh, sinh viên, cán bộ công – nhân viên bằng khuôn mặt, vân tay, hay trong lĩnh vực y tế, nghiên cứu và dự đoán bệnh truyền nhiễm, hay trong lĩnh vực sinh học, chuẩn đoán bệnh trên cây trồng, hay trong kinh tế, dự đoán giá nhà ...

Bằng cách ứng dụng các kiến thức của lĩnh vực khai phá dữ liệu nói riêng và lĩnh vực của ngành công nghệ thông tin nói chung, trong bài báo cáo này giới thiệu và tìm hiểu về đề tài: **“Nghiên cứu và ứng dụng phương pháp Collaborative Filtering và Content-Based Filtering trong xây dựng mô hình khuyến nghị phim”**.

Bài báo cáo bao gồm 4 phần:

Chương 1: Khảo sát và phát biểu bài toán

Trong chương một, báo cáo sẽ tiến hành tìm hiểu về phim ảnh là gì, sau đó tiếp tục tìm hiểu hệ thống khuyến nghị, nhu cầu của mọi người về phim ảnh, từ đó xác định các yêu cầu cần thiết cho hệ thống khuyến nghị phim theo sở thích. Sau khi đã tìm hiểu kỹ các yếu tố cần thiết, sẽ tiến hành phân tích đầu vào, đầu ra của bài toán hệ thống gợi ý phim theo sở thích bằng cách ứng dụng các kỹ thuật của khai thác dữ liệu.

Chương 2: Các kỹ thuật giải quyết bài toán

Sau khi đã tìm hiểu và xác định rõ ràng được yêu cầu của bài toán, bài báo cáo tiếp tục trình bày các kỹ thuật hiện có cùng các ưu và nhược điểm của chúng. Sau đó bài báo cáo sẽ sử dụng một trong các phương pháp đó để giải quyết bài toán.

Chương 3: Chương trình thực nghiệm

Tại chương 3, bài báo cáo tập trung trình bày quá trình thực nghiệm bằng cách áp dụng giải pháp đã đưa ra ở chương 2. Quá trình thực nghiệm được đánh giá và sử dụng một số cung cụ hỗ trợ và tiến hành so sánh kết quả, đồng thời vẽ đồ thị và lập bảng thống kê. Từ đó đưa ra kết luận và nhận xét về các kết quả thu được.

Chương 4: Xây dựng sản phẩm demo

Ở chương 4, bài báo cáo sẽ đưa ra phương pháp xây dựng một sản phẩm thực tế áp dụng các kỹ thuật giải quyết bài toán ở chương 3 kết hợp với giao diện cụ thể bằng cách phân tích thiết kế hệ thống.

Báo cáo này có ý nghĩa quan trọng trong việc tạo ra các hệ thống khuyến nghị giải trí cá nhân, giúp cải thiện trải nghiệm người dùng và tối ưu hóa quá trình tìm kiếm phim. Đồng thời, nó cũng mở ra những cơ hội cho việc phát triển và cải thiện các thuật toán khuyến nghị sử dụng trí tuệ nhân tạo trong nhiều lĩnh vực khác nhau.

Em xin gửi lời cảm ơn chân thành đến cô Đặng Quỳnh Nga - người đã tận tình hướng dẫn, giúp đỡ em hoàn thành bài báo cáo đồ án tốt nghiệp này.

Sự hướng dẫn tận tình, chu đáo của cô đã giúp em hiểu rõ hơn về nội dung bài học, từ đó có thể hoàn thành bài báo cáo một cách đầy đủ và chính xác. Bên cạnh đó, cô cũng đã giúp em rèn luyện kỹ năng viết báo cáo khoa học, giúp em có thêm kinh nghiệm trong việc nghiên cứu và thực hiện các bài tập về sau.

Trong quá trình làm báo cáo khó tránh khỏi những thiếu sót. Em rất mong nhận được ý kiến đóng góp từ thầy cô để học thêm được nhiều kinh nghiệm và sẽ hoàn thành tốt hơn bài báo cáo tới.

Em xin chúc thầy cô luôn dồi dào sức khỏe, luôn vui vẻ và thành công trong cuộc sống.

Em xin chân thành cảm ơn!

Sinh viên thực hiện

Nguyễn Ngọc Long

DANH MỤC CÁC TỪ VIẾT TẮT

KNN	K-Nearest Neighbors
SVM	Support Vector Machine
RS	Recommend System

DANH MỤC CÁC HÌNH ẢNH

Hình 1-1 Hình ảnh rạp chiếu phim	1
Hình 1-2 Hình ảnh bộ phim đầu tiên	2
Hình 1-3 Hình ảnh các bộ phim nổi tiếng.....	3
Hình 1-4 Hệ thống khuyến nghị phim được sử dụng trong Netflix.....	5
Hình 1-5 Hệ thống gợi ý video trên Youtube	5
Hình 2-1 Minh họa thuật toán SVM	9
Hình 2-2 Thuật toán SVM trong không gian 2 chiều (R^2) và không gian 3 chiều (R^3)	10
Hình 2-3 Minh họa mô hình hồi quy tuyến tính	13
Hình 2-4 Minh họa thuật toán KNN	15
Hình 2-5 Thống kê mức lương trung bình theo vị trí công việc IT	17
Hình 2-6 Minh họa phương pháp Collaborative Filtering	19
Hình 2-7 Các bước xây dựng thuật toán Neighborhood-based Collaborative Filtering	20
Hình 2-8 Minh họa phương pháp Content-based Filtering.....	24
Hình 2-9 Minh họa sản phẩm của website Sixdo.vn	25
Hình 2-10 Ứng dụng của Content-based Filtering trên website Sixdo.vn..	25
Hình 2-11 Thông tin các khóa học trên Coursera.org	26
Hình 2-12 Ứng dụng của Content-based Filtering trên Coursera.org	26
Hình 3-1 Thống kê tần suất sử dụng các ngôn ngữ lập trình tính từ 2014-2022	27
Hình 3-2 IDE Pycharm	29
Hình 3-3 File dữ liệu gốc khi mới tải về.....	30
Hình 3-4 Chọn file dữ liệu cần chuyển định dạng.....	31
Hình 3-5 Cửa sổ Text Import Wizard Step 1 of 3	31
Hình 3-6 Cửa sổ Text Import Wizard Step 2 of 3	32

Hình 3-7 File user.dat sau khi đã chuyển định dạng về user.csv.....	33
Hình 3-8 File Movie.dat sau khi đã chuyển định dạng về movie.csv.....	34
Hình 3-9 File Rating.dat sau khi đã chuyển định dạng thành Rating.csv...	34
Hình 4-1 Flask.....	47
Hình 4-2 HTML	49
Hình 4-3 Một đoạn code bằng HTML	50
Hình 4-4 Bootstrap với giao diện lúc đầu	50
Hình 4-5 Bootstrap với giao diện mới	51
Hình 4-6 Use-case hệ thống	52
Hình 4-7 Màn hình chính của sản phẩm	58
Hình 4-8 Màn hình chọn người dùng muốn dự đoán phim	59
Hình 4-9 Danh sách các phim được hệ thống gợi ý bằng Collaborative Filtering	59
Hình 4-10 Màn hình lựa chọn tên phim muốn gợi ý	59
Hình 4-11 Danh sách các phim được gợi ý bằng Content-based Filtering.	60
Hình 4-12 Màn hình lựa chọn ID người dùng và tên phim muốn gợi ý.....	60
Hình 4-13 Danh sách các phim được gợi ý bằng Hybrid Filtering	60

MỤC LỤC

LỜI NÓI ĐẦU.....	i
DANH MỤC CÁC TỪ VIẾT TẮT.....	iv
DANH MỤC CÁC HÌNH ẢNH	v
MỤC LỤC.....	vii
CHƯƠNG 1. KHẢO SÁT VÀ PHÁT BIỂU BÀI TOÁN	1
1.1 Điện ảnh và phim ảnh.....	1
1.1.1 Điện ảnh là gì?.....	1
1.1.2 Sự ra đời của ngành điện ảnh	2
1.1.3 Phim là gì?.....	3
1.1.4 Phân loại phim.....	3
1.2 Hệ thống khuyến nghị	4
1.3 Hệ thống khuyến nghị phim theo sở thích	6
CHƯƠNG 2. CÁC KỸ THUẬT GIẢI QUYẾT BÀI TOÁN	9
2.1 Thuật toán SVM(Support Vector Machine).....	9
2.2 Thuật toán Naïve Bayes	11
2.3 Mô hình hồi quy tuyến tính.....	13
2.4 Thuật toán K-Nearest Neighbors (KNN)	14
2.5 Phương pháp Collaborative Filtering và Content-Based Filtering trong khai phá dữ liệu.....	16
2.5.1 Khai phá dữ liệu	16
2.5.2 Collaborative Filtering.....	18
2.5.3 Content-Based Filtering.....	23
CHƯƠNG 3. Thực nghiệm	27
3.1 Bộ dữ liệu	27
3.2 Ngôn ngữ sử dụng	27
3.3 Công cụ sử dụng.....	29
3.4 Quá trình thực nghiệm.....	30

3.4.1 Tiền xử lý dữ liệu	30
3.4.2 Quy trình thực nghiệm.....	35
3.5 Kết quả thực nghiệm	45
CHƯƠNG 4. XÂY DỰNG SẢN PHẨM DEMO	47
4.1 Công cụ sử dụng.....	47
4.1.1 Flask.....	47
4.1.2 HTML	48
4.1.3 Bootstrap.....	50
4.2 Phân tích thiết kế hệ thống	52
4.2.1 Use case Xem kết quả dự của phương pháp Collaborative Filtering.....	52
4.2.2 Use case Xem kết quả dự đoán của phương pháp Content-based Filtering.....	53
4.2.3 Use case Xem kết quả dự đoán của phương pháp Hybrid	54
4.3 Quy trình xây dựng giao diện.....	54
4.4 Giao diện sản phẩm	58
KẾT LUẬN	61
TÀI LIỆU THAM KHẢO.....	63

CHƯƠNG 1. KHẢO SÁT VÀ PHÁT BIỂU BÀI TOÁN

1.1 Điện ảnh và phim ảnh

1.1.1 Điện ảnh là gì?

Điện ảnh tiếng Anh là cinema bao gồm các bộ phim được tạo nên từ khung hình chuyển động, kỹ thuật ghi lại hình ảnh, âm thanh, ánh sáng... Bên cạnh đó, điện ảnh còn là sự kết hợp giữa hai yếu tố nghệ thuật và kỹ thuật. Vì thế, điện ảnh còn liên quan đến các công đoạn làm, quảng bá và phân phối phim ảnh gọi chung là công nghiệp điện ảnh.[1]



Hình 1-1 Hình ảnh rạp chiếu phim

Kể từ khi xuất hiện, điện ảnh luôn đóng vai trò quan trọng đối với đời sống tinh thần của con người. Những điều mà điện ảnh truyền tải đến con người luôn mang giá trị nhân văn sâu sắc qua các tác phẩm kinh điển, ăn sâu trong tiềm thức của con người.

Điện ảnh là loại hình nghệ thuật thị giác dùng để mô phỏng những trải nghiệm truyền đạt ý tưởng, kể chuyện và khơi gợi cảm xúc, được thực hiện bằng cách ghi lại hình ảnh bằng máy ảnh hoặc bằng cách tạo hình ảnh bằng kỹ thuật hoạt hình hoặc hiệu ứng hình ảnh.[1]

1.1.2 Sự ra đời của ngành điện ảnh

Vào nửa cuối thế kỷ 19, điện ảnh bắt đầu được khai sinh và chỉ tập trung vào việc ghi lại hình ảnh của chuyển động. Điển hình nhất là những phát minh của Louis Le Prince, Eadweard James Muybridge, Étienne-Jules Marey hay Thomas Edison. Các nhà sử học đã chọn ngày 28/12/1895 là ngày khai sinh ra nghệ thuật điện ảnh. Bởi đây là ngày công chiếu phim chuyển động đầu tiên được tổ chức tại Salon Indien (Phòng Ấn Độ) nằm dưới tầng hầm của quán cafe Grand Café tại Paris, Pháp.[1]



Hình 1-2 Hình ảnh bộ phim đầu tiên

Sự ra đời của điện ảnh đã nhanh chóng nhận được nhiều sự đón nhận của công chúng. Sau đó, điện ảnh được thương mại hóa và công nghiệp điện ảnh đã xuất hiện.

1.1.3 Phim là gì?

Phim là tác phẩm điện ảnh bao gồm phim truyện, phim tài liệu, phim khoa học, phim hoạt hình [2]



Hình 1-3 Hình ảnh các bộ phim nổi tiếng

1.1.4 Phân loại phim

Hiện nay, phim điện ảnh được biết đến với 4 loại hình phổ biến sau:

- **Phim truyện:** Là loại hình phim có cốt truyện hư cấu, diễn viên đóng và tạo bối cảnh giả, tạo ảo giác giống như cuộc đời thực.
- **Phim khoa học viễn tưởng:** Chủ đề phim sẽ tập trung vào các khía cạnh khoa học, công nghệ và viễn tưởng. Mục đích là đưa ra những ý tưởng về tương lai, vũ trụ và nhằm nâng cao nhận thức khoa học và sự phát triển của nhân loại.
- **Phim hoạt hình:** Phim được tạo ra với các diễn viên là các hình ảnh chuyển động. Chúng được tạo ra bằng cách kết hợp các hình vẽ hoặc nhờ bằng máy tính. Sau đó sử dụng phương pháp quay từng hình, chiếu lên màn chiếu liên tục để tạo ảo giác.
- **Phim tài liệu:** Loại hình phim này đi thẳng vào những vấn đề trong cuộc sống, ghi lại hình ảnh, hành động thực của con người.

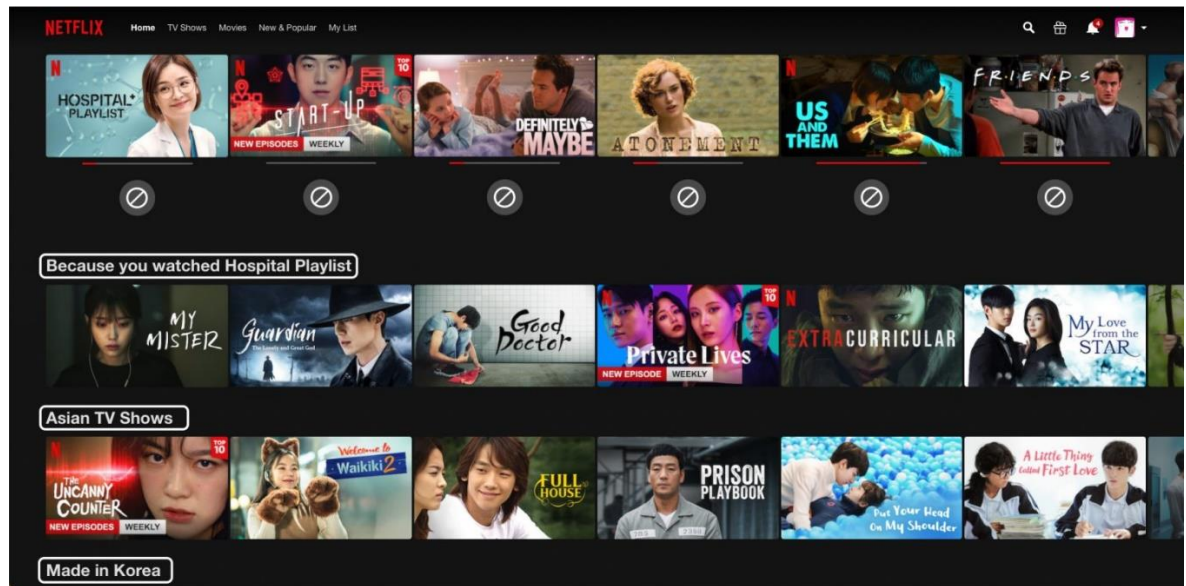
1.2 Hệ thống khuyến nghị

Nhắc đến Trí tuệ nhân tạo (AI), nhiều người sẽ nghĩ đến các ứng dụng như nhận dạng ảnh, chatbot, hay các hệ thống xe tự lái. Nhưng trên thực tế, ứng dụng của AI đã được sử dụng phổ biến từ lâu và hiện có mặt hầu hết trên các nền tảng trực tuyến, đặc biệt là trong các hệ thống khuyến nghị(recommender systems).[3]

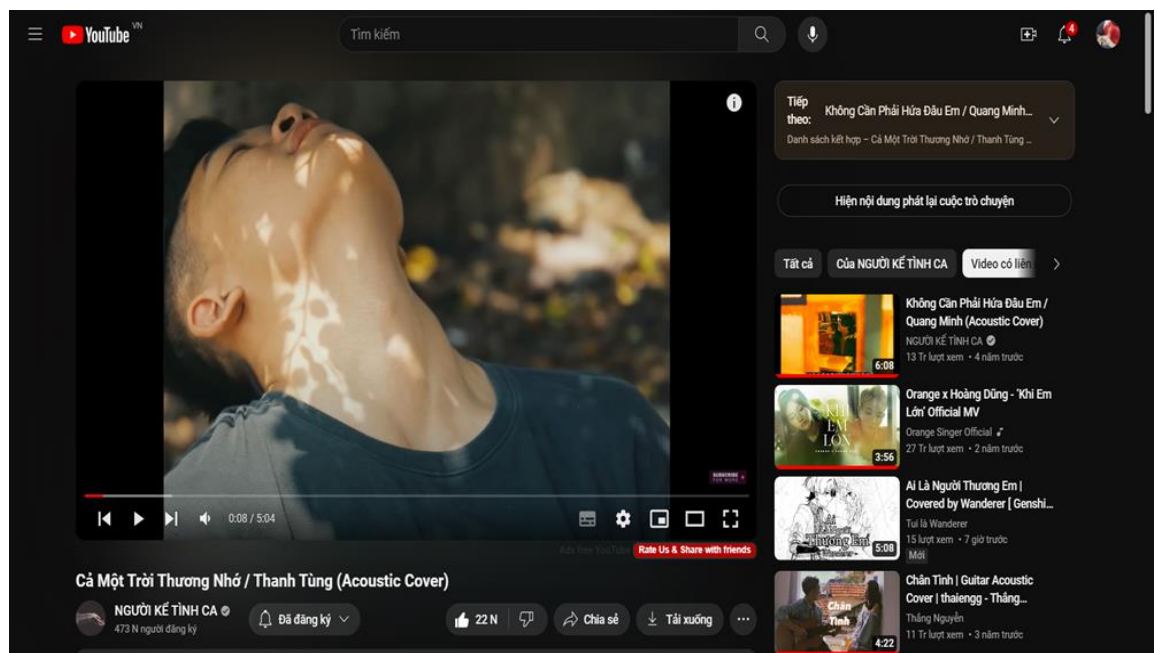
Hệ thống khuyến nghị (Recommender System) sử dụng công nghệ AI để thực hiện phân tích và hiểu khối dữ liệu cá nhân, từ đó, đưa ra các dự đoán, gợi ý đề xuất phù hợp với sở thích của người dùng tại thời điểm bất kỳ trên các ứng dụng và nền tảng trực tuyến giúp tiết kiệm thời gian tìm kiếm, truy cập nội dung dễ dàng, đồng thời, giúp doanh nghiệp nâng cao trải nghiệm khách hàng.

Hệ thống khuyến nghị là các công cụ và kỹ thuật phần mềm cung cấp các đề xuất về các hạng mục/nội dung đề xuất cho người dùng. Các đề xuất liên quan đến các quá trình ra quyết định khác nhau tại những thời điểm bất kỳ cho người dùng, ví dụ như mua hàng hóa nào, nghe nhạc gì hoặc đọc tin tức gì.

Hệ thống khuyến nghị có thể sử dụng các kỹ thuật AI như học máy để hiểu được sở thích của người dùng, nhờ vậy đưa ra những dự đoán và khuyến nghị những sản phẩm/dịch vụ/nội dung mà người dùng có thể quan tâm (hàng hoá, phim, sách, video, tin tức, bài hát, khách sạn, khoá học, v.v.). Một số hệ thống khuyến nghị tiêu biểu bao gồm hệ thống của Amazon, Netflix và Youtube.[3]



Hình 1-4 Hệ thống khuyến nghị phim được sử dụng trong Netflix



Hình 1-5 Hệ thống gợi ý video trên Youtube

Lý do cần có hệ thống khuyến nghị là bởi số lượng sản phẩm/dịch vụ/nội dung được cung cấp trực tuyến quá nhiều và người dùng khó tìm được thứ mình cần. Khi người dùng vào website cung cấp sản phẩm, hệ thống khuyến nghị sẽ trả về một danh sách ngắn các sản phẩm mà người dùng nhiều khả năng sẽ chọn, có thể bao gồm cả những thứ mà người đó không biết từ trước.

Như vậy, hệ thống khuyến nghị giúp tiết kiệm thời gian, tăng tốc độ tìm kiếm và giúp người dùng truy cập tới nội dung họ quan tâm một cách dễ dàng hơn, đồng thời, gợi ý tới người dùng những đề xuất mới mà trước đây họ chưa từng biết đến.

Với khả năng của hệ thống khuyến nghị, các doanh nghiệp sử dụng chúng để giới thiệu sản phẩm tới người tiêu dùng, giúp gia tăng doanh số nhờ các ưu đãi, sản phẩm, dịch vụ được khuyến nghị một cách cá nhân hóa, làm nâng cao trải nghiệm khách hàng. Điều này cải thiện lợi thế cạnh tranh của doanh nghiệp và giảm thiểu tỉ lệ khách hàng rời bỏ và đến với đối thủ cạnh tranh khi họ nhận thấy doanh nghiệp hiểu nhu cầu của họ và cung cấp cho họ những thứ họ muốn.

Hệ thống khuyến nghị là thành phần không thể thiếu của các nền tảng trực tuyến cung cấp đa dạng các loại hình dịch vụ, từ các website thương mại điện tử tới nền tảng đào tạo trực tuyến. Theo McKinsey, 35% doanh thu của Amazon được tạo ra từ các tương tác với hệ thống khuyến nghị của hãng này. Một thống kê khác cũng cho thấy 75% thời lượng xem phim trên Netflix được thực hiện nhờ các khuyến nghị được cá nhân hoá .

1.3 Hệ thống khuyến nghị phim theo sở thích

Xuất phát từ nhu cầu giải trí của người dùng trên toàn thế giới về phim ảnh, các trang web, các ứng dụng xem phim trực tuyến ngày càng xuất hiện nhiều hơn. Với sự xuất hiện của các trang web này đã đáp ứng được các tiêu chí của người dùng như tiện lợi, nhanh chóng, miễn phí và di động:

- Tiện lợi và nhanh chóng:
 - o Người dùng có thể truy cập nhanh chóng vào nhiều bộ phim từ mọi nơi có kết nối internet.
 - o Không cần tải xuống, người dùng có thể xem ngay trên trình duyệt hoặc ứng dụng di động.
- Miễn phí và thanh toán theo nhu cầu:

- Nhiều trang web và ứng dụng cung cấp dịch vụ xem phim miễn phí, tạo ra sự thuận tiện cho người dùng.
- Mô hình thanh toán theo yêu cầu cũng trở thành một lựa chọn phổ biến, nơi người dùng chỉ trả tiền cho những nội dung họ chọn xem.
- Đa dạng về nội dung :
 - Người dùng có nhiều lựa chọn từ các thể loại phim khác nhau, từ phim hành động, tình cảm đến phim khoa học viễn tưởng và nhiều hơn nữa.
 - Nhiều trang web cũng cung cấp các loại nội dung đặc sắc như series truyền hình, phim ngắn, và nhiều loại show giải trí khác.
- Di động và dễ truy cập:
 - Ứng dụng di động cho phép người dùng xem phim bất cứ nơi đâu, tăng tính di động và linh hoạt.
 - Giao diện thân thiện với người dùng và dễ sử dụng giúp cải thiện trải nghiệm xem phim.
- Chia sẻ và tương tác xã hội:
 - Các tính năng chia sẻ trên mạng xã hội giúp người dùng chia sẻ suy nghĩ và đánh giá về phim với bạn bè.
 - Bình luận và đánh giá từ cộng đồng có thể giúp người dùng quyết định xem phim nào.

Và trên hầu hết các trang web xem phim trực tuyến ví dụ như Netflix, Disney, Amazon Prime Video, HBO Max... đều áp dụng hệ thống gợi ý phim.

Bằng cách áp dụng các kiến thức của khai thác dữ liệu, phân tích dữ liệu, trí tuệ nhân tạo và hệ thống khuyến nghị, bài báo cáo sẽ tiến hành phân tích đầu vào, đầu ra của hệ thống khuyến nghị.

Bộ dữ liệu được sử dụng trong bài báo cáo là bộ dữ liệu movie-lens 1M DataSet được tạo ra bởi GroupLens Research là một nhóm nghiên cứu thuộc Đại học Minnesota và được phát hành vào tháng 2 năm 2003.

Bộ dữ liệu này chứa thông tin của 1.000.209 đánh giá ẩn danh của khoảng 3.900 phim được tạo bởi 6.040 người dùng MovieLens vào năm 2000. Bao gồm

3 tệp dữ liệu: movies.dat chứa thông tin của các bộ phim bao gồm: MoviesId, Title(tên phim) và Genres(thể loại phim), users.dat chứa thông tin của UserId, Giới tính, Tuổi, Nghề nghiệp, mã zip, ratings.dat chứa thông tin của UserId, MovieId, Rating(số điểm đánh giá) và Timestamp(Thời gian đánh giá).

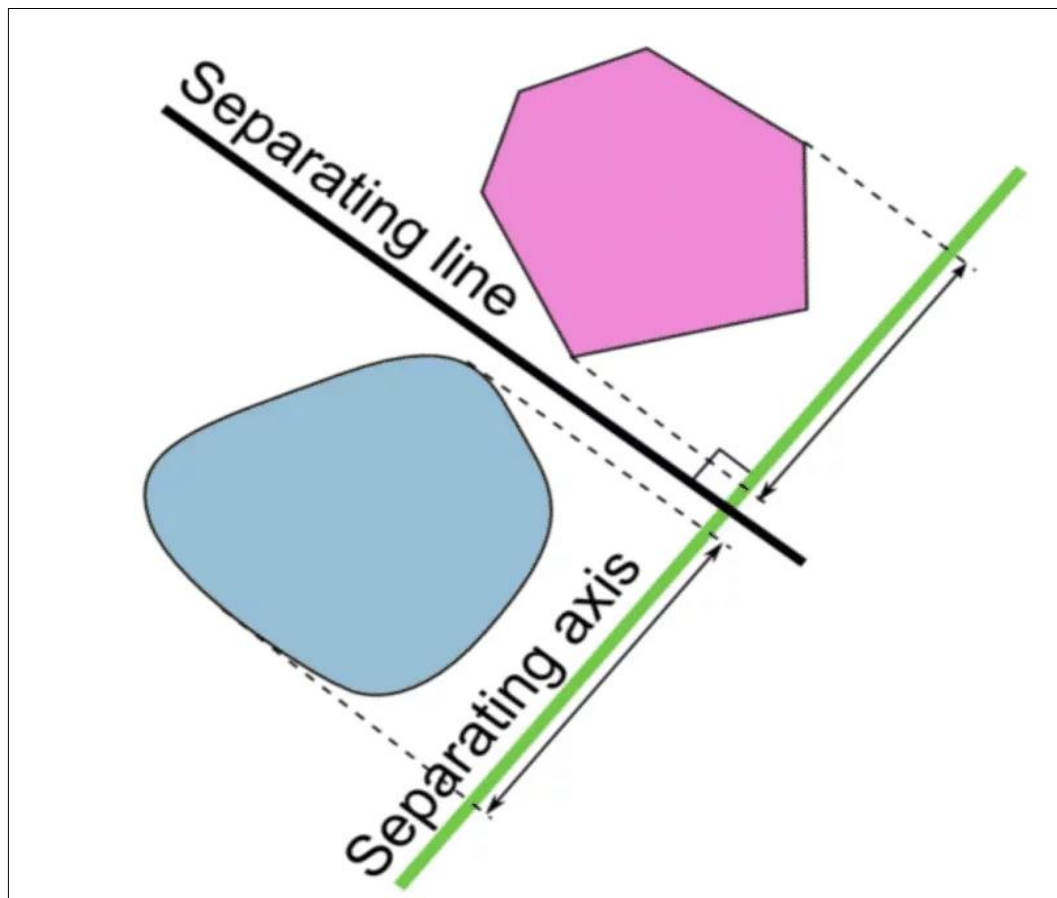
Đầu vào của bài toán là tệp dữ liệu ratings.dat sẽ bao gồm UserID là Id của người dùng, MoviesID là Id của phim, Rating và Timestamp. Mô hình bài toán sẽ dựa vào số điểm đánh giá của người dùng, người dùng có cùng sở thích và từ đó hệ thống sẽ tìm được phim nào có lượt đánh giá cao nhất và thể loại phim tương ứng với nó.

Đầu ra sẽ là các phim có cùng thể loại với phim người dùng đánh giá số điểm cao nhất đó.

CHƯƠNG 2. CÁC KỸ THUẬT GIẢI QUYẾT BÀI TOÁN

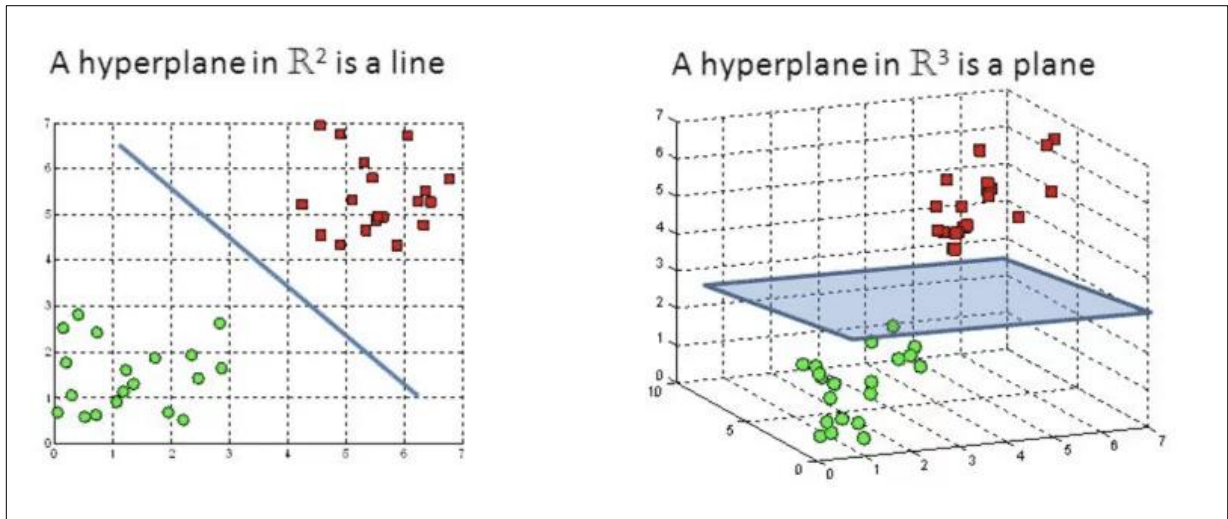
2.1 Thuật toán SVM(Support Vector Machine)

SVM là một thuật toán học có giám sát (supervised learning), nó có thể sử dụng cho cả bài toán phân lớp hoặc hồi quy. Mục tiêu của SVM là tìm ra một siêu phẳng trong không gian N chiều (ứng với N đặc trưng) chia dữ liệu thành hai phần tương ứng với lớp của chúng. Nói theo ngôn ngữ của đại số tuyến tính, siêu phẳng này phải có lề cực đại và phân chia hai bao lồi và cách đều chúng.[4]



Hình 2-1 Minh họa thuật toán SVM

Siêu phẳng tạo ra một biên giới phân chia 2 lớp của dữ liệu:



Hình 2-2 Thuật toán SVM trong không gian 2 chiều (R^2) và không gian 3 chiều (R^3)

Là một kỹ thuật phân lớp khá phổ biến, SVM thể hiện được nhiều ưu điểm trong số đó có việc tính toán hiệu quả trên các tập dữ liệu lớn. Có thể kể thêm một số.

Ưu điểm của phương pháp này như:

- Xử lý trên không gian số chiều cao: SVM là một công cụ tính toán hiệu quả trong không gian chiều cao, trong đó đặc biệt áp dụng cho các bài toán phân loại văn bản và phân tích quan điểm nơi chiều có thể cực kỳ lớn.
- Tiết kiệm bộ nhớ: Do chỉ có một tập hợp con của các điểm được sử dụng trong quá trình huấn luyện và ra quyết định thực tế cho các điểm dữ liệu mới nên chỉ có những điểm cần thiết mới được lưu trữ trong bộ nhớ khi ra quyết định.
- Tính linh hoạt - phân lớp thường là phi tuyến tính. Khả năng áp dụng Kernel mới cho phép linh động giữa các phương pháp tuyến tính và phi tuyến tính từ đó khiến cho hiệu suất phân loại lớn hơn.

Nhược điểm:

- Bài toán số chiều cao: Trong trường hợp số lượng thuộc tính (p) của tập dữ liệu lớn hơn rất nhiều so với số lượng dữ liệu (n) thì SVM cho kết quả khá tồi.

- Chưa thể hiện rõ tính xác suất: Việc phân lớp của SVM chỉ là việc cố gắng tách các đối tượng vào hai lớp được phân tách bởi siêu phẳng SVM. Điều này chưa giải thích được xác suất xuất hiện của một thành viên trong một nhóm là như thế nào.

Tóm lại, SVM là một phương pháp hiệu quả cho bài toán phân lớp dữ liệu. Nó là một công cụ đặc lực cho các bài toán về xử lý ảnh, phân loại văn bản, phân tích quan điểm. Một yếu tố làm nên hiệu quả của SVM đó là việc sử dụng Kernel function khiến cho các phương pháp chuyển không gian trở nên linh hoạt hơn.

2.2 Thuật toán Naïve Bayes

Naive Bayes là một thuật toán phân loại cho các vấn đề phân loại nhị phân (hai lớp) và đa lớp. Kỹ thuật này dễ hiểu nhất khi được mô tả bằng các giá trị đầu vào nhị phân hoặc phân loại.[5]

Thuật toán Naive Bayes tính xác suất cho các yếu tố, sau đó chọn kết quả với xác suất cao nhất.

Thuật toán này là một thuật toán mạnh mẽ trong các bài toán:

- Dự đoán với thời gian thực.
- Phân loại Text/ Lọc thư rác.
- Hệ thống Recommendation.

Một vài ưu điểm của thuật toán Naïve Bayes:

- Dễ hiểu và triển khai: Naive Bayes có cách thức hoạt động đơn giản và dễ hiểu, điều này làm cho nó trở thành một lựa chọn phù hợp cho các ứng dụng thực tế và cả cho người mới bắt đầu trong lĩnh vực học máy.
- Hiệu suất tốt với dữ liệu lớn: Mặc dù giả định về sự độc lập giữa các đặc trưng không phải lúc nào cũng chính xác trong thực tế, nhưng Naive Bayes thường cho kết quả tốt, đặc biệt là với các tập dữ liệu lớn.

- Dự đoán thời gian thực: Với quá trình huấn luyện đơn giản và chi phí tính toán thấp, Naive Bayes thích hợp cho các ứng dụng đòi hỏi dự đoán thời gian thực.
- Hiệu suất tốt với văn bản: Naive Bayes thường được sử dụng cho các bài toán phân loại văn bản như lọc thư rác hay phân loại văn bản theo chủ đề vì nó xử lý dữ liệu văn bản tốt.
- Hiệu suất tốt khi dữ liệu hợp lý với giả định: Nếu giả định về sự độc lập giữa các đặc trưng là hợp lý với bài toán cụ thể, Naive Bayes có thể cho kết quả rất tốt.

Mặc dù Naive Bayes có nhiều ưu điểm, nhưng cũng tồn tại một số nhược điểm cần xem xét khi sử dụng thuật toán này:

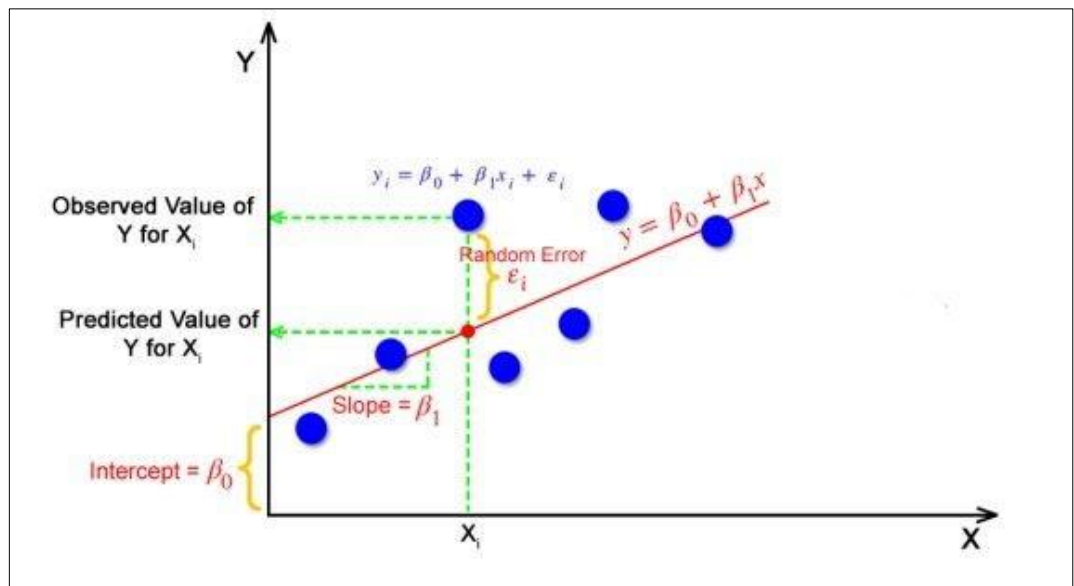
- Giả định về sự độc lập: Giả định rằng các đặc trưng là độc lập với nhau có thể không phản ánh đúng mối quan hệ giữa chúng trong thực tế, làm giảm độ chính xác của mô hình.
- Khả năng xử lý thông tin liên tục: Naive Bayes thường hoạt động tốt với dữ liệu rời rạc hoặc dạng nhị phân, nhưng nếu dữ liệu là liên tục, cần phải sử dụng các biến thể như Gaussian Naive Bayes, và trong trường hợp này, giả định về phân phối Gaussian có thể không chính xác.
- Dữ liệu thiếu: Nếu có một đặc trưng trong tập kiểm tra mà chưa xuất hiện trong tập huấn luyện, xác suất của lớp sẽ trở thành 0. Điều này có thể dẫn đến việc dự đoán không chính xác.
- Ảnh hưởng của biến liên tục có thể lớn: Trong mô hình Gaussian Naive Bayes, các giá trị ngoại lệ (outliers) có thể có ảnh hưởng lớn đến ước lượng của các tham số phân phối Gaussian.
- Yếu tố mất mát thông tin giữa các đặc trưng: Do giả định về sự độc lập, Naive Bayes có thể bỏ qua mối quan hệ quan trọng giữa các đặc trưng, đặc biệt là khi chúng có sự tương quan.
- Chênh lệch class prior: Nếu có chênh lệch lớn giữa số lượng mẫu trong các lớp, có thể dẫn đến việc mô hình tập trung nhiều vào lớp có số lượng mẫu lớn hơn.

Tóm lại, Naive Bayes thường là một lựa chọn tốt trong những tình huống đơn giản và với dữ liệu có tính độc lập giữa các đặc trưng. Tuy nhiên, cần cẩn trọng khi áp dụng nó cho các tình huống phức tạp hơn hoặc khi giả định về sự độc lập không chính xác.

2.3 Mô hình hồi quy tuyến tính

Hồi quy tuyến tính là một phương pháp thống kê để hồi quy dữ liệu với biến phụ thuộc có giá trị liên tục trong khi các biến độc lập có thể có một trong hai giá trị liên tục hoặc là giá trị phân loại. [6]

Nói cách khác "Hồi quy tuyến tính" là một phương pháp để dự đoán biến phụ thuộc (Y) dựa trên giá trị của biến độc lập (X). Nó có thể được sử dụng cho các trường hợp chúng ta muốn dự đoán một số lượng liên tục. Ví dụ, dự đoán giao thông ở một cửa hàng bán lẻ, dự đoán thời gian người dùng dừng lại một trang nào đó hoặc số trang đã truy cập vào một website nào đó v.v...



Hình 2-3 Minh họa mô hình hồi quy tuyến tính

Ưu điểm của mô hình hồi quy tuyến tính:

- Dễ hiểu và triển khai: Mô hình hồi quy tuyến tính rất dễ hiểu và đơn giản để triển khai. Điều này làm cho nó trở thành một công cụ phổ

biến trong nhiều lĩnh vực và cho cả người mới bắt đầu trong học máy.

- Hiệu suất tốt với dữ liệu dạng tuyến tính: Khi mối quan hệ giữa biến độc lập và phụ thuộc là tuyến tính, mô hình hồi quy tuyến tính cho kết quả rất tốt và dự đoán ổn định.
- Dễ giải quyết và đánh giá: Các phương pháp ước lượng tham số cho mô hình hồi quy tuyến tính đã được phát triển rất tốt. Có nhiều phương pháp kiểm tra và đánh giá mô hình, giúp xác định độ chính xác và độ tin cậy.
- Ít cần tham số: Mô hình hồi quy tuyến tính thường ít đòi hỏi tham số so với các mô hình phức tạp hơn, điều này giúp giảm nguy cơ quá mức điều chỉnh (overfitting) khi có ít dữ liệu.

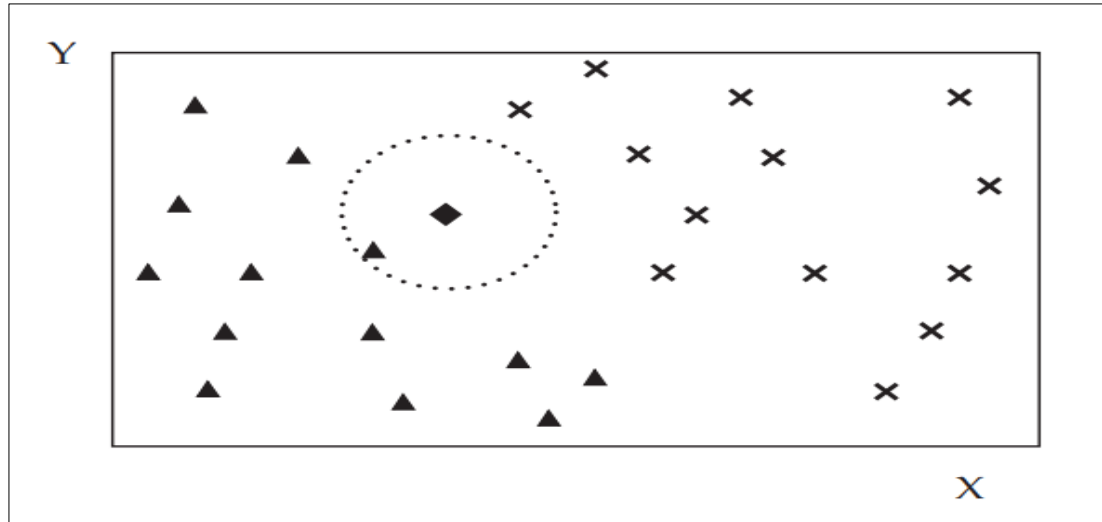
Nhược điểm:

- Giả định về tuyến tính: Mô hình này giả định rằng mối quan hệ giữa biến độc lập và phụ thuộc là tuyến tính. Nếu mối quan hệ này không đúng, mô hình sẽ không biểu diễn đúng và dự đoán kém.
- Nhạy cảm với nhiễu (noise) và biến động: Mô hình hồi quy tuyến tính có thể bị ảnh hưởng nhiều bởi các điểm dữ liệu nhiễu hoặc biến động, dẫn đến dự đoán không ổn định.
- Không giải quyết được mối quan hệ phi tuyến tính: Mô hình hồi quy tuyến tính không thể mô tả được mối quan hệ phi tuyến tính giữa các biến, điều này có thể làm mất mát thông tin quan trọng.
- Overfitting với dữ liệu lớn: Khi có nhiều biến độc lập, mô hình có thể dễ dàng trở nên quá mức điều chỉnh dữ liệu huấn luyện, làm giảm khả năng tổng quát hóa cho dữ liệu mới.
- Khả năng thiếu linh hoạt: Mô hình hồi quy tuyến tính có thể bị hạn chế trong việc mô hình các mối quan hệ phức tạp và không tuyến tính giữa các biến.

2.4 Thuật toán K-Nearest Neighbors (KNN)

Thuật toán K-Nearest Neighbors hay còn được gọi là thuật toán K- láng giềng gần nhất được sử dụng rộng rãi trong các lĩnh vực của học máy hay khai phá dữ liệu. Ý tưởng của thuật toán cũng giống với tên gọi của chính nó, là tìm

điểm chung của k láng giềng gần nhất với đối tượng cần phân lớp, phân cụm sau đó dựa vào những đặc điểm giống nhau đó có thể kết luận rằng dữ liệu mới này giống với các láng giềng gần với nó.



Hình 2-4 Minh họa thuật toán KNN

Ưu điểm:

- Đơn giản và dễ hiểu: KNN là một thuật toán đơn giản và dễ hiểu, không yêu cầu nhiều giả định phức tạp.
- Không yêu cầu giả định về phân phối dữ liệu: KNN không đặt giả định về phân phối dữ liệu, do đó nó có khả năng làm việc tốt với các dữ liệu phi tuyến tính và không đồng nhất
- Độ phức tạp của quá trình training gần bằng 0
- Việc dự đoán kết quả của dữ liệu mới đơn giản

Nhược điểm:

- Tốn nhiều thời gian tính toán: KNN có độ phức tạp tính toán cao, đặc biệt là khi số lượng điểm dữ liệu lớn. Vì phải tính toán khoảng cách giữa điểm mới và tất cả các điểm trong tập huấn luyện.

- Yêu cầu lưu toàn bộ tập dữ liệu huấn luyện trong bộ nhớ: KNN yêu cầu lưu toàn bộ tập dữ liệu huấn luyện trong bộ nhớ, điều này có thể tạo ra vấn đề với các tập dữ liệu lớn.

Ứng dụng:

- Phân loại hình ảnh: KNN có thể được sử dụng để phân loại hình ảnh dựa trên các đặc trưng hoặc pixel giống nhau
- Gợi ý sản phẩm: KNN được sử dụng trong các hệ thống gợi ý để đề xuất sản phẩm dựa trên lịch sử mua sắm của người dùng và các người dùng khác có sở thích giống nhau
- Y tế và dự đoán bệnh: KNN có thể dự đoán nguy cơ bệnh dựa trên dữ liệu bệnh nhân có đặc điểm tương tự.
- Dự đoán giá trị: Trong lĩnh vực tài chính, KNN có thể dự đoán giá trị chứng khoán dựa trên biểu đồ giá trị trước đó...

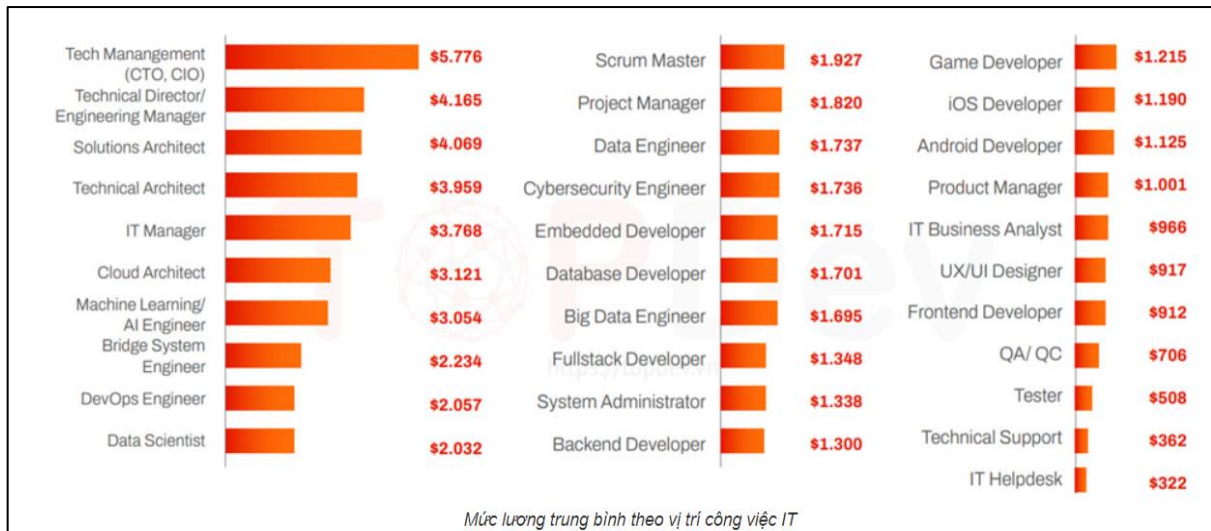
2.5 Phương pháp Collaborative Filtering và Content-Based Filtering trong khai phá dữ liệu

2.5.1 Khai phá dữ liệu

Khai phá dữ liệu hay Data Mining là một lĩnh vực đang dần trở nên phổ biến trong những năm gần đây không chỉ ở Việt Nam mà còn trên toàn thế giới.

Data Mining là kỹ thuật dùng để tìm kiếm, phân tích và đưa ra các mẫu tiềm ẩn có tính hợp lệ, mới lạ, có ích và có thể hiểu được trong một khối dữ liệu lớn thông qua các quá trình như làm tiền xử lý dữ liệu, phân tích, rút gọn các giá trị không cần thiết, khử nhiễu...

Data Mining là một kỹ thuật đóng vai trò thiết yếu trong các lĩnh vực như Data Engineer, Data Analyst, Data Science



Hình 2-5 Thống kê mức lương trung bình theo vị trí công việc IT

Tầm quan trọng của Data Mining:

- Khai thác dữ liệu là một phần quan trọng đối với sự thành công của bất kỳ sáng kiến phân tích nào.
- Các doanh nghiệp có thể sử dụng quy trình khai phá kiến thức để tăng niềm tin của khách hàng, tìm kiếm nguồn doanh thu mới và thu hút khách hàng quay lại.
- Quy trình khai thác dữ liệu hiệu quả hỗ trợ trong nhiều khía cạnh khác nhau của việc lập kế hoạch kinh doanh và quản lý hoạt động cũng như rất nhiều lĩnh vực khác.

Một vài ứng dụng của Data Mining:

- Phân tích thẻ tín dụng/ Dự báo chứng khoán
- Phân tích khiếu nại/ gian lận
- Phân tích các bản ghi cuộc gọi
- Quản lý giao vận/ hậu cần
- Phân tích hiệu quả điều trị/ Dự báo/ Phân tích DNA
- Phân tích thị trường giáo dục/ Quản lý chất lượng
- Dự đoán bảo trì/ Phát hiện lỗi/ Lập lịch/ Hỗ trợ ra quyết định
- Phân tích, dự báo thị trường/ Tiếp thị lan truyền ...
- Phân tích, dự báo năng lượng sử dụng

Phân biệt Data Mining và Data Analyst

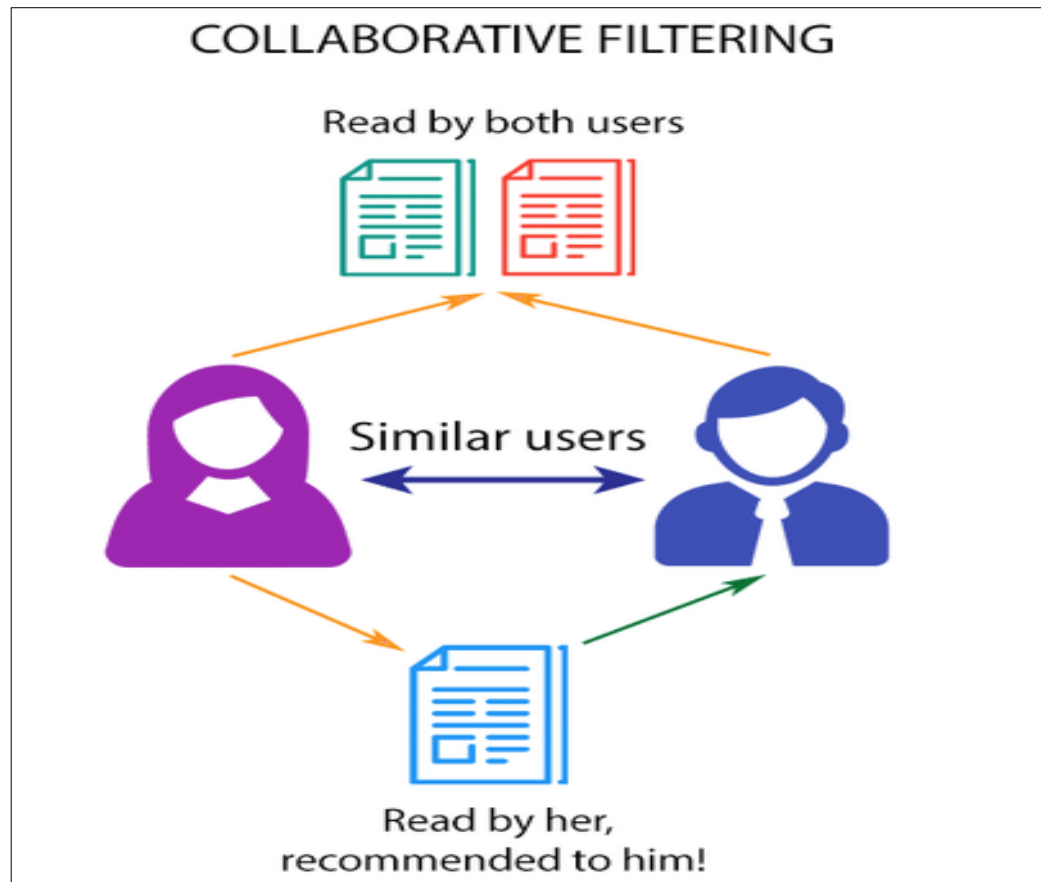
Data Mining	Data Analytics
-------------	----------------

<p>Một cách khác để nói về Data Mining là “Khám phá kiến thức trong cơ sở dữ liệu” (KDD). Đây là một quá trình phức tạp và liên tục, bao gồm nhiều bước như tiền xử lý, phân tích, trình bày và đánh giá dữ liệu. Mục tiêu của Data Mining là tìm ra những thông tin có giá trị, hữu ích và mới mẻ từ một lượng dữ liệu lớn.</p>	<p>Data Analytics là một quá trình phức tạp và toàn diện, bao gồm nhiều bước khác nhau để xử lý, biến đổi, kiểm tra và trích xuất những thông tin có ý nghĩa từ dữ liệu. Data Analytics kết hợp khoa học dữ liệu, kiến thức về lĩnh vực cụ thể, và công nghệ để khám phá thông tin từ dữ liệu và tạo ra các báo cáo, biểu đồ, và dự đoán dựa trên dữ liệu.</p>
<p>Data Mining xây dựng thuật toán để tìm cấu trúc trong dữ liệu và giải thích cụ thể về cấu trúc đó. Thuật toán này dựa trên kiến thức toán học và khoa học, giúp tổ chức thu thập dữ liệu một cách rõ ràng và chính xác</p>	<p>Đối với Data Analytics sẽ thực hiện trên dữ liệu có cấu trúc, bán cấu trúc hoặc không có cấu trúc. Các chuyên gia được giao nhiệm vụ phát hiện các mẫu trong dữ liệu và sử dụng chúng để tóm tắt cho khách hàng và áp dụng nó vào chiến lược của doanh nghiệp.</p>

2.5.2 Collaborative Filtering

Lọc cộng tác hay Collaborative Filtering là phương pháp phân tích dữ liệu người dùng để tìm ra mối tương quan giữa các đối tượng người dùng. Lọc cộng tác hoạt động bằng cách xây dựng một cơ sở dữ liệu, lưu trữ dưới dạng ma trận người dùng (users) - sản phẩm (items) và mỗi dòng của nó là một vector.

Sau đó, phân tích dữ liệu, tính toán sự tương đồng giữa các users với nhau để đưa ra gợi ý. Ý tưởng quan trọng của phương pháp này là những người dùng tương tự có xu hướng sử dụng những sản phẩm tương tự.



Hình 2-6 Minh họa phương pháp Collaborative Filtering

Ví dụ: Nếu khách hàng A thích các sản phẩm tương tự khách hàng B thì “Collaborative Filtering” sẽ đoán rằng khách hàng A có khả năng sẽ thích các sản phẩm khác mà khách hàng B đã thích/mua và ngược lại.

Collaborative Filtering là thuật toán lọc tương tác tức là tìm ra sản phẩm mà khách hàng có khả năng ưa thích nhất dựa vào những sản phẩm mà những khách hàng khác có hành vi tương tự đã lựa chọn. Thuật toán sẽ không cần sử dụng thông tin sản phẩm là đầu vào cho dự báo rating. Đầu vào của thuật toán là một ma trận tiện ích (utility matrix) chứa giá trị rating của các cặp (user, item). Mỗi cột là các rating mà một user đã rate và mỗi dòng là các rating của một item được rate. Có 2 phương pháp chính được sử dụng trong collaborative filtering bao gồm: Neighborhood-based collaborative Filtering và Matrix Factorization.

2.5.2.1 Neighborhood-based collaborative Filtering

Ở phương pháp này ta sẽ cần xây dựng ma trận hệ số tương quan của véc tơ rating của các users để tìm ra nhóm users có cùng sở thích. Hệ số tương quan giữa các users càng lớn thì sở thích của họ càng giống nhau và trái lại thì họ càng có sở thích khác biệt. Thuật toán sẽ dự đoán giá trị rating tại một cặp (user, item) chưa được rate bằng cách tính tổng có trọng số các giá trị rating của users tương quan nhất với user đó mà đã rate item trên. Trọng số thông thường sẽ bằng chính hệ số tương quan.

Để xây dựng một thuật toán Neighborhood-based collaborative Filtering chúng ta cần trải qua các bước cơ bản bên dưới.

	u_0	u_1	u_2	u_3	u_4	u_5	u_6
i_0	5	5	2	0	1	?	?
i_1	4	?	?	0	?	2	?
i_2	?	4	1	?	?	1	1
i_3	2	2	3	4	4	?	4
i_4	2	0	4	?	?	?	5
↓ ↓ ↓ ↓ ↓ ↓ ↓ ↓							
\bar{u}_j	3.25	2.75	2.5	1.33	2.5	1.5	3.33

a) Original utility matrix \mathbf{Y} and mean user ratings.

	u_0	u_1	u_2	u_3	u_4	u_5	u_6
i_0	1.75	2.25	-0.5	-1.33	-1.5	0	0
i_1	0.75	0	0	-1.33	0	0.5	0
i_2	0	1.25	-1.5	0	0	-0.5	-2.33
i_3	-1.25	-0.75	0.5	2.67	1.5	0	0.67
i_4	-1.25	-2.75	1.5	0	0	0	1.67

b) Normalized utility matrix $\bar{\mathbf{Y}}$.

	u_0	u_1	u_2	u_3	u_4	u_5	u_6
u_0	1	0.83	-0.58	-0.79	-0.82	0.2	-0.38
u_1	0.83	1	-0.87	-0.40	-0.55	-0.23	-0.71
u_2	-0.58	-0.87	1	0.27	0.32	0.47	0.96
u_3	-0.79	-0.40	0.27	1	0.87	-0.29	0.18
u_4	-0.82	-0.55	0.32	0.87	1	0	0.16
u_5	0.2	-0.23	0.47	-0.29	0	1	0.56
u_6	-0.38	-0.71	0.96	0.18	0.16	0.56	1

c) User similarity matrix \mathbf{S} .

	u_0	u_1	u_2	u_3	u_4	u_5	u_6
i_0	1.75	2.25	-0.5	-1.33	-1.5	0.18	-0.63
i_1	0.75	0.48	-0.17	-1.33	-1.33	0.5	0.05
i_2	0.91	1.25	-1.5	-1.84	-1.78	-0.5	-2.33
i_3	-1.25	-0.75	0.5	2.67	1.5	0.59	0.67
i_4	-1.25	-2.75	1.5	1.57	1.56	1.59	1.67

d) $\hat{\mathbf{Y}}$

Predict normalized rating of u_1 on i_1 with $k = 2$

Users who rated i_1 : $\{u_0, u_3, u_5\}$

Corresponding similarities: $\{0.83, -0.40, -0.23\}$

\Rightarrow most similar users: $\mathcal{N}(u_1, i_1) = \{u_0, u_5\}$

with **normalized ratings** $\{0.75, 0.5\}$

$$\Rightarrow \hat{y}_{i_1, u_1} = \frac{0.83 \cdot 0.75 + (-0.23) \cdot 0.5}{0.83 + |-0.23|} \approx 0.48$$

e) Example

	u_0	u_1	u_2	u_3	u_4	u_5	u_6
i_0	5	5	2	0	1	1.68	2.70
i_1	4	3.23	2.33	0	1.67	2	3.38
i_2	4.15	4	1	-0.5	0.71	1	1
i_3	2	2	3	4	4	2.10	4
i_4	2	0	4	2.9	4.06	3.10	5

f) Full \mathbf{Y}

Hình 2-7 Các bước xây dựng thuật toán Neighborhood-based Collaborative Filtering

Hình 1: Sơ đồ các bước thực hiện thuật toán Neighborhood-based collaborative Filtering. Bao gồm các bước: a) Lấy trung bình các cột, b) Chuẩn hóa ma trận bằng cách trừ đi trung bình, c) Tính hệ số tương quan của ma trận chuẩn hóa, d) Dự đoán trên ma trận chuẩn hóa các vị trí chưa được rate, e) Diễn giải công thức dự báo rating, f) Chuyển đổi sang giá trị rating thực tế.

Thuật toán sẽ trải qua lần lượt các step sau đây:

1. Chuẩn hóa dữ liệu ở ma trận tiện ích Y bằng cách trừ đi ở mỗi cột (là các rating của cùng 1 user) trung bình giá trị rating của cột. Việc này là để loại bỏ sự khác biệt về mức độ cho điểm của các user. Vì ví dụ: Có một số user khó tính có thể cho điểm cao nhất là 3 nhưng user dễ tính thì điểm thấp nhất là 3. Khi đó nếu nhận định user khó tính không thích item (nếu ta coi 3 là điểm thấp) hoặc user dễ tính yêu thích item (nếu ta coi 3 là điểm cao) là không chuẩn xác. Chuẩn hóa giá trị rating nhằm mục đích đưa trung bình rating của các user sau khi chuẩn hóa về 0. Giá trị rating dương thể hiện user ưa thích item và trái lại âm sẽ là không thích, bằng 0 là trung lập.

2. Tính ma trận hệ số tương quan giữa các véc tơ cột. Ma trận tương quan thể hiện sự tương đồng trong hành vi mua sắm giữa các user. Từ ma trận tương quan ta có thể xác định ra các users có sở thích tương đồng nhất với một user xác định. Hệ số tương quan dương và càng gần 1 chứng tỏ 2 users có sở thích giống nhau. Hệ số tương quan âm là 2 users có hành vi trái ngược.

3. Dự báo rating của một user u cho một item i bằng cách xác định trên ma trận hệ số tương quan một tập $S(u, k|i)$ gồm k users có giá trị tương quan lớn nhất đối với user u mà đã rate item i . Giá trị dự báo rating của user u sẽ được tính bằng tổng có trọng số của các rating trong tập k users tương quan nêu trên theo công thức bên dưới:

$$\hat{y}_{i,u} = \frac{\sum_{u_j \in S(u, k|i)} \bar{y}_{i,u_j} \text{sim}(u, u_j)}{\sum_{u_j \in S(u, k|i)} |\text{sim}(u, u_j)|}$$

Chuyển giá trị dự báo ở ma trận chuẩn hóa sang giá trị dự báo rating bằng cách cộng các giá trị ở ma trận chuẩn hóa với giá trị trung bình của mỗi cột.

Hạn chế của phương pháp collaborative filtering:

Thường phải lưu một ma trận hệ số tương quan với kích thước rất lớn. Việc này dẫn tới tốn tài nguyên lưu trữ và thời gian tính toán.

Ở trên ta đã lựa chọn việc chuẩn hóa theo chiều user. Ngoài ra, ta cũng có thể lựa chọn chuẩn hóa theo chiều item mà không làm thay đổi phương pháp bằng cách chuyển vị ma trận tiện ích Y . Việc lựa chọn chuẩn hóa theo chiều nào sẽ căn cứ trên kích thước theo chiều nào là lớn hơn. Thông thường số lượng user sẽ nhiều hơn item. Khi đó chuẩn hóa theo item sẽ có lợi thế hơn bởi: Kích thước ma trận hệ số tương quan giữa các user là nhỏ hơn nên tốn ít tài nguyên. Thêm nữa khi một user rating một item mới thì giá trị thay đổi về trung bình trên mỗi cột item là nhỏ hơn so với trường hợp chuẩn hóa theo user. Điều này dẫn tới ma trận hệ số tương quan ít thay đổi hơn và tần suất cập nhật cũng ít hơn.

Bên cạnh thuật toán Neighborhood-based collaborative Filtering, một thuật toán khác cũng thuộc lớp các bài toán collaborative filtering đó là matrix factorization. Thuật toán này thường mang lại độ chính xác cao hơn và đồng thời yêu cầu ít tài nguyên lưu trữ hơn.

2.5.2.2. Matrix factorization

Ngoài ra còn một phương pháp collaborative filtering khác dựa trên một phép phân rã ma trận (matrix factorization). Tức là chúng ta sẽ phân tích ma trận tiện ích thành tích của các ma trận items và ma trận users.

$$Y \approx \begin{bmatrix} \mathbf{x}_1 \mathbf{w}_1 & \mathbf{x}_1 \mathbf{w}_2 & \dots & \mathbf{x}_1 \mathbf{w}_M \\ \mathbf{x}_2 \mathbf{w}_1 & \mathbf{x}_2 \mathbf{w}_2 & \dots & \mathbf{x}_2 \mathbf{w}_M \\ \dots & \dots & \ddots & \dots \\ \mathbf{x}_N \mathbf{w}_1 & \mathbf{x}_N \mathbf{w}_2 & \dots & \mathbf{x}_N \mathbf{w}_M \end{bmatrix} = \begin{bmatrix} \mathbf{x}_1 \\ \mathbf{x}_2 \\ \dots \\ \mathbf{x}_N \end{bmatrix} [\mathbf{w}_1 \quad \mathbf{w}_2 \quad \dots \quad \mathbf{w}_M] = \mathbf{XW}$$

Khi đó mỗi dòng của ma trận X đại diện cho một véc tơ nhân tố ẩn của một item, đó là những nhân tố bất kì, rất trừu tượng mà chúng ta không nên đặt tên cho chúng. Mỗi cột của ma trận W đại diện cho một véc tơ các hệ số thể hiện mức độ yêu thích của user đối với các nhân tố ẩn. Số lượng nhân tố ẩn thông thường là một số có giá trị rất nhỏ so với số lượng user và item nên dung lượng cần lưu trữ đối với 2 ma trận X và W là rất nhỏ so với lưu trữ toàn bộ ma trận Y .

Sau khi tìm được các ma trận items X và ma trận users W , giá trị ước lượng rating của một user j lên một item i sẽ chính bằng tích:

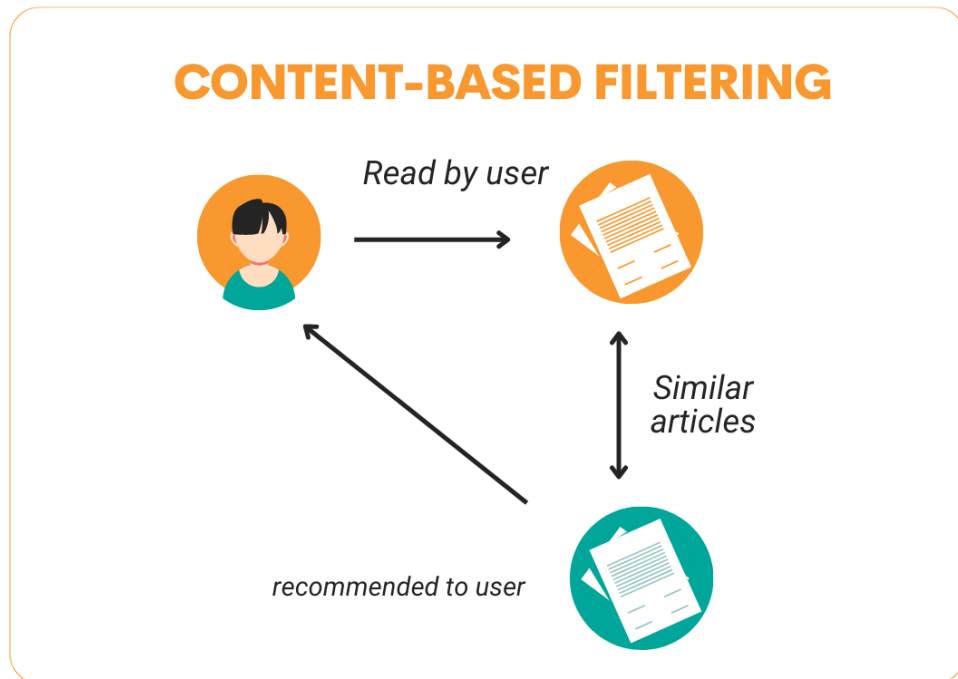
$$\hat{y}_{ij} = \mathbf{x}_i^T \mathbf{w}_j$$

Như vậy giá trị dự báo được tính toán đơn giản hơn so với Neighborhood-based collaborative Filtering vì chỉ cần thực hiện phép nhân véc tơ mà không cần phải cộng với trung bình cột để chuyển về giá trị gốc.

Quá trình dự báo hệ số cho mô hình hồi qui của mỗi user tương tự như phương pháp Content-Based Filtering. Nhưng có sự kết hợp giữa tìm nghiệm tối ưu của ma trận items và ma trận users. Quá trình này được thực hiện xen kẽ nhau nên không chỉ tận dụng được các thông tin là đầu vào của users mà còn tận dụng được sự giống nhau trong sở thích của các users. Chính vì thế phương pháp mới được xếp vào nhóm Collaborative Filtering.

2.5.3 Content-Based Filtering

Phương pháp gợi ý dựa trên nội dung là phương pháp dựa vào dữ liệu về các sản phẩm mà khách hàng đã thích trong quá khứ để tính độ tương tự với các sản phẩm trong hệ thống. Từ đó, gợi ý những sản phẩm tương tự với sản phẩm mà khách hàng đã thích, đã xem, đã mua trong quá khứ.

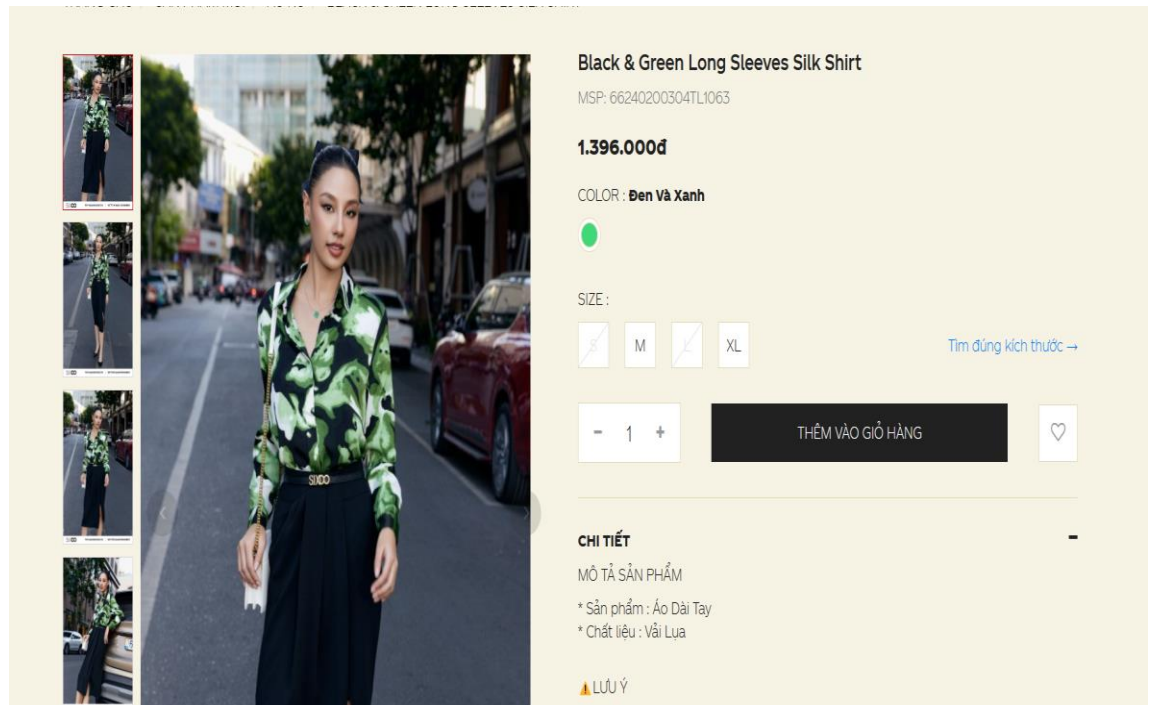


Hình 2-8 Minh họa phương pháp Content-based Filtering

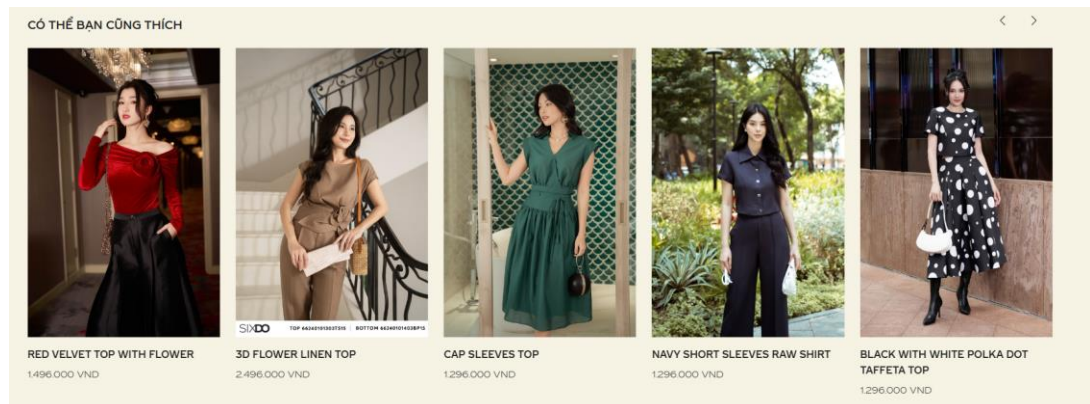
Ý tưởng đằng sau lọc cộng tác dựa trên nội dung nếu khách hàng thích một sản phẩm A, khách cũng có thể thích sản phẩm tương tự với A là B.

Ứng dụng của Content-based Filtering vào hệ thống gợi ý Recommend Systems:

- Như ví dụ đã trình bày cùng với đặc điểm của Content-based Filtering là dự đoán các sản phẩm có tính tương đồng với nhau nên trong các hệ thống Recommend System – RS như sách thì hệ thống sẽ gợi ý các sách có cùng thể loại hoặc cùng tác giả; với quần áo hệ thống sẽ gợi ý các mẫu sản phẩm có cùng một bộ sưu tập hay cùng một nhà thiết kế...
- Ứng dụng rộng rãi trong các trang web thương mại điện tử:
Ví dụ trang web chuyên kinh doanh các thể loại quần áo váy đầm <https://sixdo.vn/>. Người dùng chọn sản phẩm muốn mua là Black & Green Long Sleeves Silk Shirt.



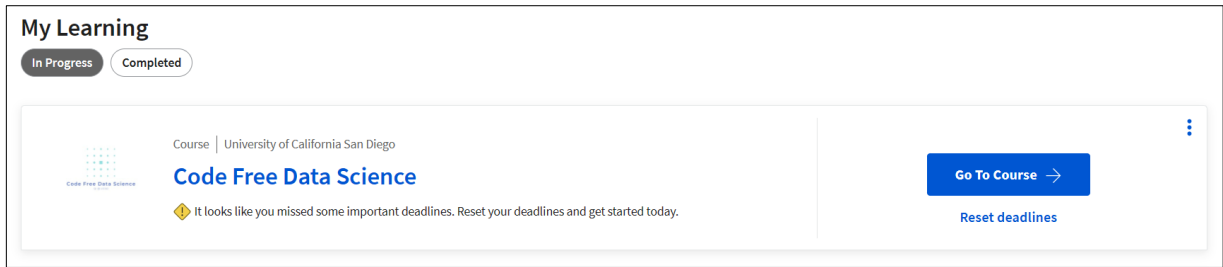
Hình 2-9 Minh họa sản phẩm của website Sixdo.vn



Hình 2-10 Ứng dụng của Content-based Filtering trên website Sixdo.vn

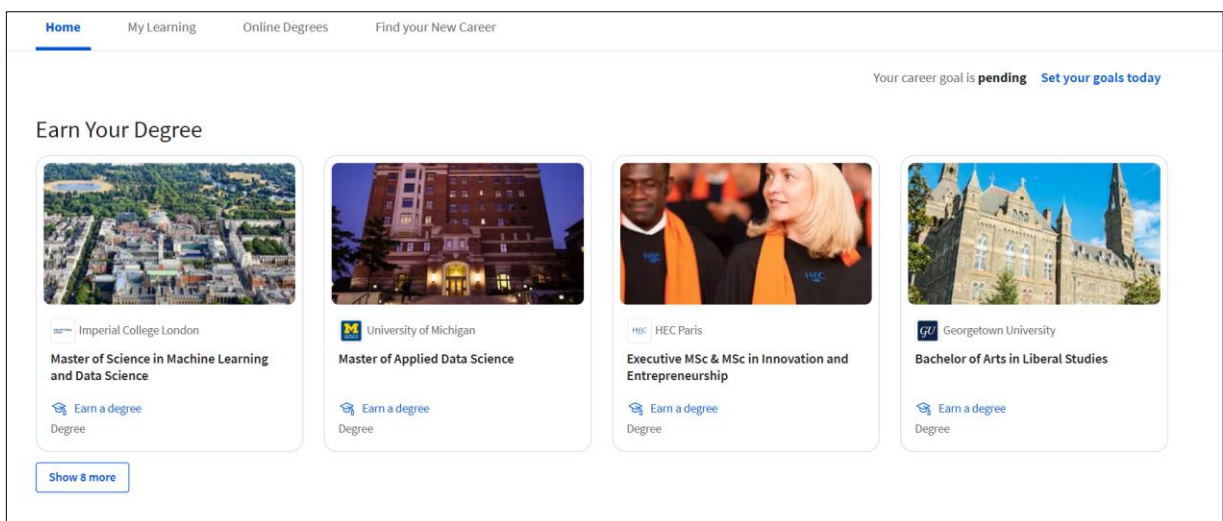
Với ứng dụng của RS thì hệ thống sẽ gợi ý ra các sản phẩm cùng thể loại như hình trên.

- Ứng dụng trong các website học trực tuyến ví dụ như <https://www.coursera.org/>. Người dùng chọn học một khóa học có liên quan đến Data Science, cụ thể là Code Free Data Science :



Hình 2-11 Thông tin các khóa học trên Coursera.org

Thì khi quay trở về Home, Coursera sẽ gợi ý cho người dùng này một số khóa học có liên quan ví dụ như Master of Science in Machine Learning



Hình 2-12 Ứng dụng của Content-based Filtering trên Coursera.org

Qua các ứng dụng cụ thể, có thể thấy Content-based Filtering nói riêng và các phương pháp recommend trong lĩnh vực Khai phá dữ liệu nói chung đều có tính thiết thực giúp cải thiện chất lượng dịch vụ, nâng cao đời sống cá nhân cũng như tập thể lên.

Ngoài ra, các ứng dụng này còn giúp tiết kiệm thời gian một cách đáng kể, giúp các cá nhân có thể dễ dàng tìm kiếm các khóa học, sản phẩm tương tự như sản phẩm họ muốn mua hoặc đã mua; giúp các tổ chức, doanh nghiệp nâng cao khả năng dịch vụ, tăng doanh thu hay sự nổi tiếng của thương hiệu với mọi người.

CHƯƠNG 3. THỰC NGHIỆM

3.1 Bộ dữ liệu

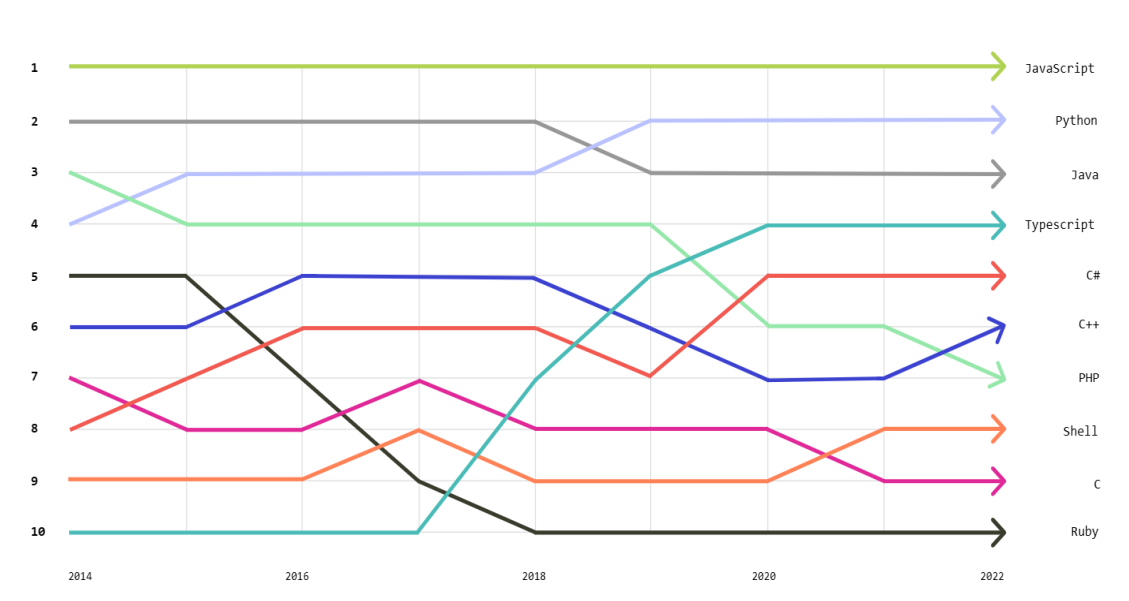
Bộ dữ liệu được sử dụng trong bài báo cáo là bộ dữ liệu MovieLens 1M DataSet được tạo ra bởi GroupLens Research là một nhóm nghiên cứu thuộc Đại học Minnesota và được phát hành vào tháng 2 năm 2003.

Bộ dữ liệu này chứa thông tin của 1.000.209 đánh giá ẩn danh của khoảng 3.900 phim được tạo bởi 6.040 người dùng MovieLens vào năm 2000. Bao gồm 3 tệp dữ liệu: movies.dat chứa thông tin của các bộ phim bao gồm: MoviesId, Title(tên phim) và Genres(thể loại phim), users.dat chứa thông tin của UserId, Giới tính, Tuổi, Nghề nghiệp, mã zip, ratings.dat chứa thông tin của UserId, MovieId, Rating(số điểm đánh giá) và Timestamp(Thời gian đánh giá).

3.2 Ngôn ngữ sử dụng

Bài báo cáo sử dụng ngôn ngữ lập trình chính là Python trên IDE là Pycharm phiên bản 2023.1.

Python là một trong các ngôn ngữ đang dần trở nên phổ biến ở những năm gần đây không chỉ ở Việt Nam mà còn trên toàn thế giới.



Hình 3-1 Thống kê tần suất sử dụng các ngôn ngữ lập trình tính từ 2014-2022

Dựa vào thống kê theo Stack Overflow, dựa trên khảo sát hơn 70,000 nhà phát triển từ 180 quốc gia, cung cấp thông tin chi tiết về các ngôn ngữ lập trình được sử dụng phổ biến nhất, có thể thấy Python vẫn phát triển chậm rãi từ những năm 2014 và những năm tiếp theo bắt đầu phát triển mạnh vượt từ vị trí top 4 lên top 2 những ngôn ngữ được sử dụng phổ biến nhất.

Lý do đơn giản để hiểu tại sao các Developer lại sử dụng nhiều Python đến vậy chính là sự phát triển của AI và các ngành Data như Data Analyst, Business Analyst... Các thư viện mạnh mẽ và phổ biến của Python như TensorFlow, PyTorch, Pandas, Scikit-learn làm cho Python trở thành ngôn ngữ lập trình lý tưởng cho các nhà phân tích dữ liệu lớn.

Bên cạnh đó, về cú pháp câu lệnh, biến, hàm của Python dễ tiếp cận với những người mới bắt đầu. Python là ngôn ngữ hỗ trợ cả hướng cấu trúc và hướng đối tượng nên nó có thể áp dụng cho nhiều kiểu lập trình khác nhau tùy thuộc vào kỹ năng của các Developer.

Không chỉ hỗ trợ mạnh mẽ trong các lĩnh vực AI hay Data, Python cũng được ứng dụng trong các website dựa trên 2 framework chính là Flask và Django.

	Flask	Django
Khái niệm chung	Flask là một micro-framework nhẹ nhàng và linh hoạt, được thiết kế để cung cấp các công cụ và thư viện cần thiết để xây dựng các ứng dụng web, mà không áp đặt quá nhiều cấu trúc hoặc các thành phần mặc định. Flask tuân theo mô hình WSGI (Web Server Gateway Interface) và cung cấp một số công cụ cơ bản để xử lý các yêu cầu HTTP và quản lý các tuyến đường URL.	Django là một framework web mạnh mẽ và đầy đủ tính năng, được thiết kế để giúp các nhà phát triển xây dựng các ứng dụng web phức tạp một cách nhanh chóng và dễ dàng. Django tuân theo mô hình MVT (Model-View-Template) và đi kèm với nhiều tính năng tích hợp sẵn, chẳng hạn như hệ thống quản lý người dùng, quản lý cơ sở dữ liệu, form, và khả năng quản lý URL.
Đặc điểm	Nhẹ nhàng và linh hoạt: Flask rất nhẹ và không đi kèm với nhiều	Tích hợp đầy đủ: Django đi kèm với tất cả các tính năng cần thiết

chính	<p>thành phần mặc định, giúp các nhà phát triển có sự tự do tối đa để quyết định cách tổ chức ứng dụng của mình.</p> <p>Tiện ích mở rộng: Flask có một hệ sinh thái lớn các tiện ích mở rộng (extensions) mà bạn có thể sử dụng để thêm các tính năng như ORM, xác thực, và nhiều hơn nữa.</p> <p>Dễ học và sử dụng: Flask có một cú pháp đơn giản và dễ học, phù hợp cho các dự án nhỏ và trung bình, hoặc khi bạn cần xây dựng một nguyên mẫu nhanh chóng</p>	<p>để xây dựng một ứng dụng web, từ xác thực người dùng đến quản lý cơ sở dữ liệu.</p> <p>Bảo mật: Django có các biện pháp bảo mật mạnh mẽ được tích hợp sẵn để bảo vệ ứng dụng khỏi các lỗ hổng phổ biến như SQL injection và cross-site scripting (XSS).</p> <p>Quản lý cơ sở dữ liệu ORM: Django sử dụng một ORM (Object-Relational Mapping) mạnh mẽ, giúp dễ dàng tương tác với cơ sở dữ liệu mà không cần viết SQL trực tiếp.</p> <p>Quản lý URL: Django cung cấp một hệ thống quản lý URL mạnh mẽ và linh hoạt.</p>
-------	---	---

3.3 Công cụ sử dụng

PyCharm 2023.1 là một môi trường phát triển tích hợp (Integrated Development Environment - IDE) phổ biến được phát triển bởi JetBrains dành cho lập trình Python. Đây là phiên bản mới nhất của PyCharm tính đến thời điểm hiện tại.



Hình 3-2 IDE Pycharm





Dưới đây là một số tính năng quan trọng trong PyCharm 2023.1:

- **Cải thiện hiệu suất:** PyCharm 2023.1 được cải tiến về hiệu suất tổng thể, giúp tăng tốc quá trình phân tích mã, đánh dấu cú pháp và gợi ý thông minh.
- **Hỗ trợ Python 3.10:** PyCharm 2023.1 hỗ trợ đầy đủ cho Python 3.10, bao gồm các tính năng mới như PEP 634 (Structural Pattern Matching) và PEP 636 (Variadic Generics).
- **Tích hợp công cụ nâng cao:** PyCharm 2023.1 tích hợp các công cụ phổ biến như Docker, WSL (Windows Subsystem for Linux), và SSH (Secure Shell) để giúp phát triển và triển khai ứng dụng Python một cách dễ dàng và tiện lợi.
- **Cải thiện công cụ kiểm tra và gỡ lỗi:** PyCharm 2023.1 cung cấp các công cụ mạnh mẽ để kiểm tra và gỡ lỗi mã nguồn Python, bao gồm tích hợp với công cụ kiểm tra tự động như pytest và unittest.
- **Cải thiện tích hợp Git:** Phiên bản mới này cung cấp tích hợp Git nâng cao, cho phép bạn quản lý mã nguồn, nhánh và thực hiện các hoạt động Git một cách dễ dàng từ giao diện người dùng của PyCharm.
- **Cải thiện trình chỉnh sửa mã:** PyCharm 2023.1 cung cấp các cải tiến cho trình chỉnh sửa mã, bao gồm gợi ý thông minh, kiểm tra cú pháp và định dạng tự động, giúp bạn viết mã Python nhanh chóng và chính xác hơn.

3.4 Quá trình thực nghiệm

3.4.1 Tiền xử lý dữ liệu

Bộ dữ liệu khi mới tải về:

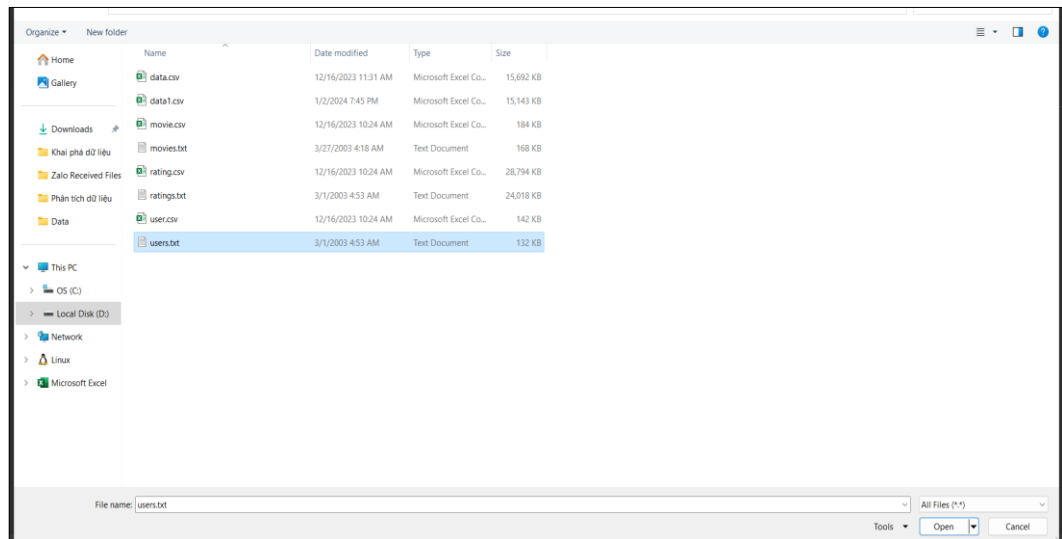
 movies.dat	5/14/2024 3:42 PM	DAT File	168 KB
 ratings.dat	5/14/2024 3:42 PM	DAT File	24,018 KB
 README	5/14/2024 3:42 PM	File	6 KB
 users.dat	5/14/2024 3:42 PM	DAT File	132 KB

Hình 3-3 File dữ liệu gốc khi mới tải về

Do trong Pycharm không đọc trực tiếp được các file .dat nên cần phải chuyển về định dạng .txt bằng File Explorer: Tiến hành trực tiếp đổi tên file trong các công cụ quản lý file trên windows.

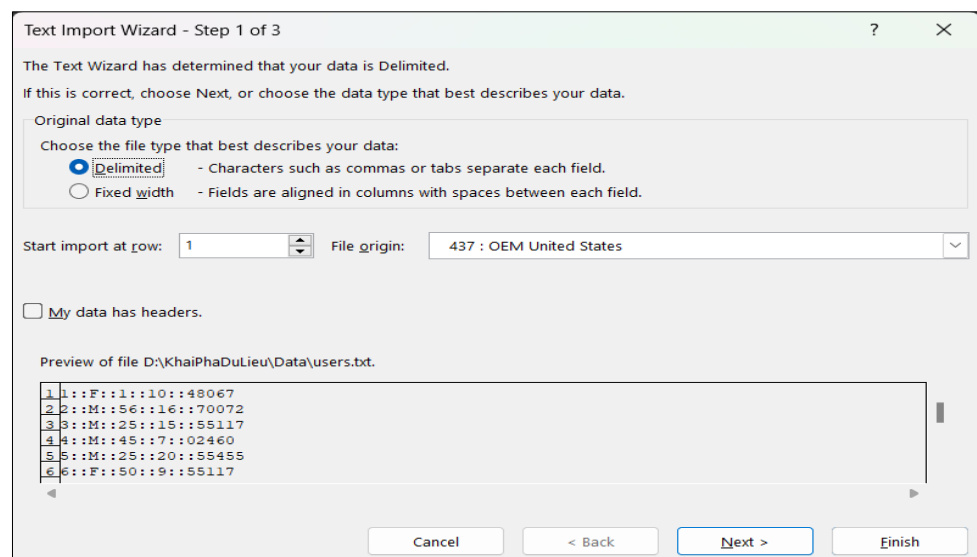
Sử dụng excel mở các file định dạng .txt theo các bước:

- Mở excel chọn open
- Tìm kiếm file định dạng .txt(chọn mục All file)



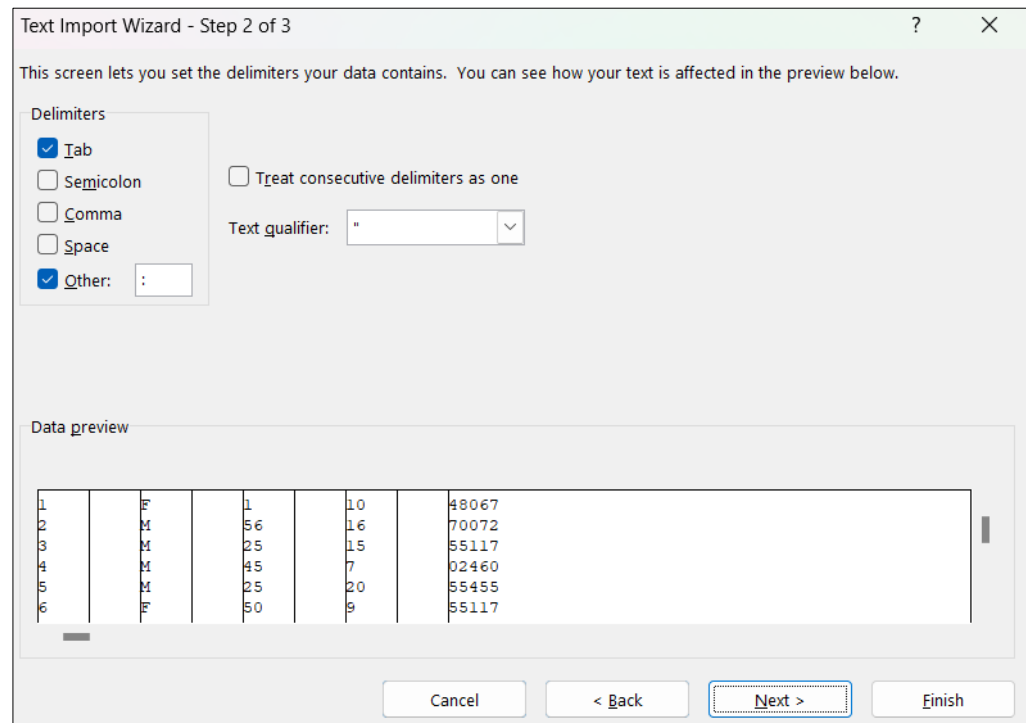
Hình 3-4 Chọn file dữ liệu cần chuyển định dạng

- Chọn Open và chọn Delimited



Hình 3-5 Cửa sổ Text Import Wizard Step 1 of 3

- Chọn Next và chọn mục other, gõ ký tự “:” vì các dữ liệu trong bộ movie-len 1m ở các file .txt ngăn cách nhau bởi dấu “:”



Hình 3-6 Cửa sổ Text Import Wizard Step 2 of 3

- Chọn Finish
- Tiến hành xóa các cột dữ liệu rỗng, và lưu lại dưới định dạng .csv (lựa chọn csv MS-DOS)

Dữ liệu sau khi đã được tiền xử lý sẽ ở dạng .csv như sau:

User.csv:

	A	B	C	D	E	F	G
1		UserID	Gender	Age	Occupatio	Zip-code	
2	0	1 F		1	10	48067	
3	1	2 M		56	16	70072	
4	2	3 M		25	15	55117	
5	3	4 M		45	7	2460	
6	4	5 M		25	20	55455	
7	5	6 F		50	9	55117	
8	6	7 M		35	1	6810	
9	7	8 M		25	12	11413	
10	8	9 M		25	17	61614	
11	9	10 F		35	1	95370	
12	10	11 F		25	1	4093	
13	11	12 M		25	12	32793	
14	12	13 M		45	1	93304	
15	13	14 M		35	0	60126	
16	14	15 M		25	7	22903	
17	15	16 F		35	0	20670	
18	16	17 M		50	1	95350	
19	17	18 F		18	3	95825	
20	18	19 M		1	10	48073	
21	19	20 M		25	14	55113	
22	20	21 M		18	16	99353	
23	21	22 M		18	15	53706	
24	22	23 M		35	0	90049	
25	23	24 F		25	7	10023	
26	24	25 M		18	4	1500	

Hình 3-7 File user.dat sau khi đã chuyển định dạng về user.csv

Movie.csv:

A	B	C	D	E	F	G	H
	MovieID	Title	Genres				
0	1	Toy Story (1995)	Animation Children's Comedy				
1	2	Jumanji (1995)	Adventure Children's Fantasy				
2	3	Grumpier Old Men (1995)	Comedy Romance				
3	4	Waiting to Exhale (1995)	Comedy Drama				
4	5	Father of the Bride Part II (1995)	Comedy				
5	6	Heat (1995)	Action Crime Thriller				
6	7	Sabrina (1995)	Comedy Romance				
7	8	Tom and Huck (1995)	Adventure Children's				
8	9	Sudden Death (1995)	Action				
9	10	GoldenEye (1995)	Action Adventure Thriller				
10	11	American President, The (1995)	Comedy Drama Romance				
11	12	Dracula: Dead and Loving It (1995)	Comedy Horror				
12	13	Balto (1995)	Animation Children's				
13	14	Nixon (1995)	Drama				
14	15	Cutthroat Island (1995)	Action Adventure Romance				
15	16	Casino (1995)	Drama Thriller				
16	17	Sense and Sensibility (1995)	Drama Romance				
17	18	Four Rooms (1995)	Thriller				
18	19	Ace Ventura: When Nature Calls (1995)	Comedy				
19	20	Money Train (1995)	Action				
20	21	Get Shorty (1995)	Action Comedy Drama				
21	22	Copycat (1995)	Crime Drama Thriller				
22	23	Assassins (1995)	Thriller				
23	24	Powder (1995)	Drama Sci-Fi				
24	25	Twister (1996)	Drama Romance				

Hình 3-8 File Movie.dat sau khi đã chuyển định dạng về movie.csv

Rating.csv:

A	B	C	D	E	F
	UserID	MovieID	Rating	Timestamp	
0	1	1193	5	9.78E+08	
1	1	661	3	9.78E+08	
2	1	914	3	9.78E+08	
3	1	3408	4	9.78E+08	
4	1	2355	5	9.79E+08	
5	1	1197	3	9.78E+08	
6	1	1287	5	9.78E+08	
7	1	2804	5	9.78E+08	
8	1	594	4	9.78E+08	
9	1	919	4	9.78E+08	
10	1	595	5	9.79E+08	
11	1	938	4	9.78E+08	
12	1	2398	4	9.78E+08	
13	1	2918	4	9.78E+08	
14	1	1035	5	9.78E+08	
15	1	2791	4	9.78E+08	
16	1	2687	3	9.79E+08	
17	1	2018	4	9.78E+08	
18	1	3105	5	9.78E+08	
19	1	2797	4	9.78E+08	
20	1	2321	3	9.78E+08	
21	1	720	3	9.78E+08	
22	1	1270	5	9.78E+08	
23	1	527	5	9.79E+08	
24	1	2349	3	9.78E+08	

Hình 3-9 File Rating.dat sau khi đã chuyển định dạng thành Rating.csv

3.4.2 Quy trình thực nghiệm

Đầu tiên, ta tiến hành import các thư viện cần thiết cho bài toán mô hình khuyến nghị phim theo sở thích

```
1 import pandas as pd
2 from sklearn.neighbors import NearestNeighbors
3 from sklearn.metrics.pairwise import cosine_similarity
4 from sklearn.feature_extraction.text import TfidfVectorizer
5 from sklearn.metrics.pairwise import linear_kernel
6
```

Tiếp đến, tiến hành đọc dữ liệu bằng các sử dụng thư viện pandas và phương thức `read_csv()`

```
# Đọc dữ liệu
ratings_df = pd.read_csv("Data/rating.csv")
movies_df = pd.read_csv("Data/movie.csv")
users_df = pd.read_csv("Data/user.csv")
```

Sau đó tiến hành kết hợp các bảng dữ liệu để tạo ra một bảng dữ liệu mới chứa thông tin về đánh giá phim và thông tin người dùng liên quan

```
# Merge dữ liệu
movie_ratings_df = pd.merge(ratings_df, movies_df[['MovieID', 'Title', 'Genres']], on='MovieID', how='left')
user_movie_ratings_df = pd.merge(movie_ratings_df, users_df[['UserID', 'Gender', 'Age', 'Occupation', 'Zip-code']],
                                on='UserID', how='left')
```

Cụ thể, đoạn mã sử dụng thư viện pandas để thực hiện các thao tác kết hợp dữ liệu. Đầu tiên, nó kết hợp bảng dữ liệu "ratings_df" và "movies_df" trên cột "MovieID" để thêm thông tin về tiêu đề và thể loại của các bộ phim. Kết quả của phép kết hợp này được lưu vào bảng "movie_ratings_df".

Tiếp theo, đoạn mã kết hợp bảng "movie_ratings_df" với bảng "users_df" trên cột "UserID" để thêm thông tin về giới tính, độ tuổi, nghề nghiệp và mã vùng của người dùng. Kết quả cuối cùng được lưu vào bảng "user_movie_ratings_df".

Với việc kết hợp các bảng dữ liệu này, có thể tạo ra một bảng dữ liệu hoàn chỉnh chứa thông tin về đánh giá phim cùng với thông tin người dùng liên quan.

Tiếp đến, tiến hành tạo ma trận đánh giá và xử lý giá trị thiếu

```
# Tạo ma trận đánh giá (UserID x MovieID)
rating_matrix = user_movie_ratings_df.pivot_table(index='UserID', columns='MovieID', values='Rating')

# Xử lý giá trị thiếu
rating_matrix = rating_matrix.fillna(0)
```

Đoạn mã trên tạo ma trận đánh giá (rating matrix) dựa trên bảng dữ liệu "user_movie_ratings_df". Ma trận đánh giá là một ma trận có kích thước UserID x MovieID, trong đó mỗi hàng tương ứng với một người dùng và mỗi cột tương ứng với một bộ phim. Giá trị trong ma trận đại diện cho đánh giá mà người dùng đã đưa ra cho bộ phim tương ứng.

Để tạo ma trận đánh giá, đoạn mã sử dụng phương thức `pivot_table()` của pandas. Cụ thể, cột 'UserID' được sử dụng làm chỉ mục (index), cột 'MovieID' được sử dụng làm cột và cột 'Rating' được sử dụng để điền vào các ô của ma trận. Kết quả của phép chuyển đổi này được gán cho biến `rating_matrix`.

Để xử lý giá trị thiếu trong ma trận, đoạn mã sử dụng phương thức `fillna()` của pandas và điền giá trị 0 vào các ô có giá trị thiếu. Điều này có thể được thực hiện để thay thế các ô không có đánh giá bằng giá trị 0, giả sử rằng các giá trị thiếu đại diện cho việc người dùng chưa đưa ra đánh giá cho bộ phim tương ứng.

Bài toán gợi ý (recommendation) thường không rơi vào nhóm phân loại (classification) hoặc phân cụm (clustering) một cách truyền thống. Thay vào đó, nó thường được gọi là một bài toán "học không giám sát" hoặc "học hướng dẫn" (unsupervised learning).

Trong bài toán gợi ý, mục tiêu là dự đoán sự quan tâm của một người dùng đối với một sản phẩm (phim, sách, sản phẩm thương mại, v.v.) dựa trên lịch sử đánh giá của người dùng hoặc dựa trên sự tương đồng giữa người dùng và sản phẩm.

Cụ thể hơn, trong bài báo cáo này sẽ trình bày 3 phương pháp thường được sử dụng trong các bài toán gợi ý:

Collaborative Filtering (Lọc cộng tác) kết hợp cùng thuật toán KNN:

- Trong lọc cộng tác (Collaborative Filtering), khi ta nhập một người dùng cụ thể (ví dụ, người dùng 1), thuật toán sẽ tìm các người dùng giống nhau với người dùng này dựa trên lịch sử đánh giá của họ cho các sản phẩm (trong trường hợp này là các bộ phim).
- Các người dùng giống nhau này sẽ được sử dụng để tạo ra một danh sách các sản phẩm mà họ đã đánh giá cao và người dùng cụ thể chưa xem. Danh sách này sau đó được sắp xếp theo độ giống nhau và độ đánh giá cao để đưa ra gợi ý.
- Tiến hành khởi tạo knn_model

```
# Lọc cộng tác: Sử dụng KNN để tìm người dùng giống nhau
knn_model = NearestNeighbors(metric='cosine', algorithm='brute')
knn_model.fit(rating_matrix.values)
```

Để tạo mô hình KNN, đoạn mã sử dụng lớp NearestNeighbors từ module sklearn.neighbors. Trong đó, tham số metric được đặt là 'cosine', cho biết phương pháp tính độ tương đồng cosine sẽ được sử dụng để đo khoảng cách giữa các điểm dữ liệu. Tham số algorithm được đặt là 'brute', cho biết thuật toán 'brute force' sẽ được sử dụng để tìm các hàng láng giềng gần nhất.

Sau khi mô hình KNN được tạo ra, nó được khớp với giá trị của ma trận đánh giá bằng cách sử dụng phương thức fit(). Điều này có nghĩa là mô hình sẽ học cách xác định các hàng láng giềng gần nhất dựa trên độ tương đồng cosine giữa các hàng của ma trận đánh giá.

Tiếp đến ta tiến hành xây dựng hàm recommend_movies với tham số truyền vào là một user_id (là người dùng muốn gợi ý phim)


```

usage
def recommend_movies(user_id):
    if user_id not in rating_matrix.index:
        print(f"User {user_id} not found in the dataset.")
        return []

    _, similar_users = knn_model.kneighbors([rating_matrix.loc[user_id]])

    # Kiểm tra xem có người dùng giống nhau hay không
    if len(similar_users) == 0 or similar_users.flatten()[0] not in rating_matrix.index:
        print(f"No similar users found for user {user_id}.")
        return []

    # Lấy danh sách phim mà những người dùng giống nhau đã đánh giá cao
    recommended_movies = rating_matrix.loc[similar_users.flatten()].mean(axis=0).sort_values(ascending=False)

    # Loại bỏ các phim đã được người dùng đánh giá
    user Rated movies = rating_matrix.loc[user_id][rating_matrix.loc[user_id] > 0].index
    final_recommendations = recommended_movies[~recommended_movies.index.isin(user Rated movies)].index

    return final_recommendations

```

Hàm `recommend_movies(user_id)` trên được định nghĩa để đề xuất danh sách các bộ phim được khuyến nghị cho một người dùng cụ thể.

Đầu tiên, hàm kiểm tra xem `user_id` có tồn tại trong chỉ mục của ma trận đánh giá `rating_matrix` hay không. Nếu không tìm thấy, hàm sẽ in ra thông báo và trả về một danh sách rỗng.

Nếu `user_id` tồn tại trong ma trận đánh giá, hàm sử dụng mô hình KNN `knn_model` để tìm các hàng láng giềng gần nhất cho người dùng đó. Các hàng láng giềng này được lưu trong biến `similar_users`.

Tiếp theo, hàm kiểm tra xem có hàng láng giềng gần nhất hay không và xem hàng láng giềng đầu tiên có tồn tại trong chỉ mục của ma trận đánh giá không. Nếu không tìm thấy, hàm sẽ in ra thông báo và trả về một danh sách rỗng.

Nếu có hàng láng giềng gần nhất, hàm tính trung bình điểm đánh giá của các hàng láng giềng đó và sắp xếp giảm dần để lấy ra danh sách các phim được đề xuất dựa trên đánh giá của những người dùng giống nhau. Danh sách này được lưu trong biến `recommended_movies`.

Tiếp theo, hàm lọc bỏ các phim đã được người dùng đánh giá từ danh sách `recommended_movies`, bằng cách lấy danh sách các phim đã được người dùng đánh giá từ hàng tương ứng với `user_id` trong ma trận đánh giá. Danh sách phim cuối cùng được lưu vào biến `final_recommendations`.

Cuối cùng, hàm trả về danh sách phim cuối cùng `final_recommendations` là danh sách các bộ phim được khuyến nghị cho người dùng `user_id`.

Tiếp đến, ta tiến hành nhập người dùng cần gợi ý từ bàn phím

```
# Thực hiện gợi ý cho một người dùng
user_id_to_recommend = int(input("Nhập người dùng cần gợi ý phim: "))
recommendations = recommend_movies(user_id_to_recommend)

# Hiển thị các phim được gợi ý
recommended_movies = movies_df[movies_df['MovieID'].isin(recommendations)][['MovieID', 'Title', 'Genres']]
print(recommended_movies.to_string())

print("\nDự đoán của mô hình:")
predicted_ratings = pd.Series([5] * len(recommendations), index=recommendations)
print(predicted_ratings)
```

Đoạn mã trên thực hiện gợi ý các bộ phim cho một người dùng cụ thể và hiển thị danh sách các bộ phim được gợi ý cùng với dự đoán điểm đánh giá của mô hình cho từng bộ phim.

Đầu tiên, người dùng được yêu cầu nhập `user_id_to_recommend` bằng cách sử dụng hàm `input()`. Giá trị này được truyền vào hàm `recommend_movies()` để lấy danh sách các bộ phim được gợi ý cho người dùng đó.

Tiếp theo, danh sách các bộ phim được gợi ý được lưu vào biến `recommendations`. Đoạn mã sử dụng DataFrame `movies_df` để lấy thông tin về tiêu đề và thể loại của các bộ phim tương ứng với danh sách

recommendations. Thông tin này được hiển thị bằng cách chọn các cột 'MovieID', 'Title', và 'Genres' từ DataFrame `recommended_movies` và sử dụng phương thức `to_string()` để hiển thị kết quả.

Sau đó, dự đoán điểm đánh giá của mô hình được tạo ra bằng cách tạo một Series `predicted_ratings` với giá trị 5 cho mỗi bộ phim trong danh sách recommendations. Điều này đơn giản chỉ là một dự đoán giả định của mô hình và có thể được thay đổi tùy ý.

Cuối cùng, danh sách các bộ phim được gợi ý và dự đoán điểm đánh giá của mô hình được hiển thị ra màn hình.

Nhập người dùng cần gợi ý phim: 3

	MovieID	Title	Genres
0	1	Toy Story (1995)	Animation Children's Comedy
1	2	Jumanji (1995)	Adventure Children's Fantasy
2	3	Grumpier Old Men (1995)	Comedy Romance
3	4	Waiting to Exhale (1995)	Comedy Drama
4	5	Father of the Bride Part II (1995)	Comedy
5	6	Heat (1995)	Action Crime Thriller
6	7	Sabrina (1995)	Comedy Romance
7	8	Tom and Huck (1995)	Adventure Children's
8	9	Sudden Death (1995)	Action
9	10	GoldenEye (1995)	Action Adventure Thriller
10	11	American President, The (1995)	Comedy Drama Romance
11	12	Dracula: Dead and Loving It (1995)	Comedy Horror
12	13	Balto (1995)	Animation Children's
13	14	Nixon (1995)	Drama
14	15	Cutthroat Island (1995)	Action Adventure Romance
15	16	Casino (1995)	Drama Thriller

Content-Based Filtering (Lọc dựa trên nội dung):

- Content-Based Filtering: Phương pháp này giả định rằng những người dùng có sở thích tương tự sẽ thích những bộ phim có nội dung tương tự.
- Phương pháp Content-Based Filtering tập trung vào các đặc trưng nội dung của các bộ phim và không phụ thuộc vào thông tin về hành vi hoặc sự tương tác của người dùng. Điều này giúp phương pháp này có thể gợi ý các bộ phim mới hoặc các bộ phim chưa được xem

bởi người dùng. Tuy nhiên, một hạn chế của Content-Based Filtering là nó không thể gợi ý các bộ phim ngoài phạm vi nội dung đã biết.

- Tính toán các vector đặc trưng của mỗi phim dựa trên thông tin Genres(thể loại)

```
# Tính toán vector đặc trưng của mỗi phim dựa trên thông tin nội dung (Genres)
tfidf_vectorizer = TfidfVectorizer(stop_words='english')
genres_matrix = tfidf_vectorizer.fit_transform(movies_df['Genres'].fillna(''))
```

Một vector đặc trưng của các phim được tính toán bằng cách sử dụng TfidfVectorizer từ sklearn.feature_extraction.text.

Các từ dừng (stop words) được loại bỏ và vector đặc trưng của các phim dựa trên thông tin thể loại (Genres) được tính toán từ DataFrame movies_df. Kết quả được lưu trữ trong biến genres_matrix.

Tiếp theo, độ tương tự cosine giữa các phim dựa trên vector đặc trưng được tính toán bằng cách sử dụng linear_kernel từ thư viện sklearn.metrics.pairwise. Kết quả được lưu trữ trong biến content_similarity.

```
# Tính toán độ tương tự cosine giữa các phim dựa trên vector đặc trưng
content_similarity = linear_kernel(genres_matrix, genres_matrix)
```

Sau đó, hàm content_based_recommendation được định nghĩa để thực hiện gợi ý dựa trên Content-Based Filtering cho một phim cụ thể. Hàm này lấy movie_id của phim và tính toán độ tương tự cosine giữa phim này và tất cả các phim khác. Kết quả được sắp xếp theo độ tương tự giảm dần và danh sách các phim được gợi ý (loại bỏ phim đầu tiên vì đó là chính phim đang xét) được lưu trữ trong biến recommended_movies.

```
def content_based_recommendation(movie_id):
    # Lấy index của phim trong DataFrame movies_df
    movie_idx = movies_df[movies_df['MovieID'] == movie_id].index[0]

    # Tính toán độ tương tự cosine giữa phim này và tất cả các phim khác
    similarity_scores = list(enumerate(content_similarity[movie_idx]))

    # Sắp xếp các phim theo độ tương tự giảm dần
    sorted_movies = sorted(similarity_scores, key=lambda x: x[1], reverse=True)

    # Lấy danh sách các phim được gợi ý (loại bỏ phim đầu tiên vì đó là chính phim đang xét)
    recommended_movies = [movies_df.iloc[movie[0]] for movie in sorted_movies[1:11]]

    return recommended_movies
```

Tiếp theo, người dùng được yêu cầu nhập `movie_id_to_recommend` bằng cách sử dụng hàm `input()`. Giá trị này được truyền vào hàm `content_based_recommendation()` để lấy danh sách các bộ phim được gợi ý dựa trên Content-Based Filtering cho phim đó.

```
5
6 # Thực hiện gợi ý cho một phim
7 movie_id_to_recommend = int(input("Nhập phim cần gợi ý (lọc theo nội dung): "))
8 content_based_recommendations = content_based_recommendation(movie_id_to_recommend)
9
10 # Hiển thị các phim được gợi ý
11 print("Các phim được gợi ý dựa trên Content-Based Filtering:")
12 for movie in content_based_recommendations:
13     print(f"{movie['Title']} - Genres: {movie['Genres']}")
14
```

Kết quả của lọc theo nội dung

```

Nhập phim cần gợi ý ( lọc theo nội dung): 3
Các phim được gợi ý dựa trên Content-Based Filtering:
Sabrina (1995) - Genres: Comedy|Romance
Clueless (1995) - Genres: Comedy|Romance
Two if by Sea (1996) - Genres: Comedy|Romance
French Twist (Gazon maudit) (1995) - Genres: Comedy|Romance
Vampire in Brooklyn (1995) - Genres: Comedy|Romance
If Lucy Fell (1996) - Genres: Comedy|Romance
Boomerang (1992) - Genres: Comedy|Romance
Pie in the Sky (1995) - Genres: Comedy|Romance
French Kiss (1995) - Genres: Comedy|Romance
Forget Paris (1995) - Genres: Comedy|Romance

```

Hybrid Methods (Phương pháp kết hợp) :

- Là phương pháp kết hợp của lọc cộng tác và lọc theo nội dung

```

# Hàm dự đoán dựa trên lọc cộng tác và lọc dựa trên nội dung
1 usage
def hybrid_recommendation(user_id, movie_id):
    # Lọc cộng tác: Tìm người dùng giống nhất
    _, similar_users = knn_model.kneighbors([rating_matrix.loc[user_id]])

    # Lọc dựa trên nội dung: Tìm phim giống nhất
    similar_movies = content_similarity[movie_id].argsort()[::-6:-1]

    # Kết hợp các gợi ý từ cả hai phương pháp
    combined_recommendations = set(similar_users.flatten()) | set(similar_movies)

    # Lọc bỏ các phim đã được người dùng đánh giá
    user Rated movies = rating_matrix.loc[user_id][rating_matrix.loc[user_id] > 0].index
    final_recommendations = [movie for movie in combined_recommendations if movie not in user Rated movies]

    return final_recommendations

# Thực hiện gợi ý cho một người dùng và một phim cụ thể
user_id_to_recommend = int(input("Nhập người dùng cần gợi ý phim: "))
movie_id_to_base_recommendation = int(input("Nhập phim cần gợi ý: "))
recommendations = hybrid_recommendation(user_id_to_recommend, movie_id_to_base_recommendation)

# Hiển thị các phim được gợi ý
recommended_movies = movies_df[movies_df['MovieID'].isin(recommendations)][['MovieID', 'Title', 'Genres']]
print(recommended_movies.to_string())

```

Đoạn mã trên thực hiện việc kết hợp lọc cộng tác và lọc dựa trên nội dung để gợi ý phim cho một người dùng và một phim cụ thể.

Trước tiên, hàm `hybrid_recommendation` được định nghĩa để thực hiện gợi ý kết hợp. Hàm này lấy `user_id` của người dùng và `movie_id` của phim làm đầu vào. Đầu tiên, lọc cộng tác được thực hiện bằng cách tìm các người dùng giống nhất với `user_id` thông qua mô hình KNN (`knn_model`). Các người dùng giống nhất được lưu trữ trong `similar_users`.

Tiếp theo, lọc dựa trên nội dung được thực hiện bằng cách tìm các phim giống nhất với `movie_id` thông qua ma trận `content_similarity`. Các phim giống nhất được lưu trữ trong `similar_movies`.

Sau đó, các gợi ý từ cả hai phương pháp được kết hợp lại thành một tập hợp `combined_recommendations`. Tập hợp này chứa các người dùng giống nhất và các phim giống nhất, không có các phim mà người dùng đã đánh giá.

Cuối cùng, các phim được gợi ý được lọc bỏ các phim đã được người dùng đánh giá và được lưu trữ trong `final_recommendations`.

Cuối cùng, danh sách các phim được gợi ý được hiển thị ra màn hình bằng cách lấy các phim từ DataFrame `movies_df` dựa trên các giá trị `recommended_movies`. Các cột "MovieID", "Title" và "Genres" được hiển thị.

Nhập người dùng cần gợi ý phim: 3

Nhập phim cần gợi ý: 4

	MovieID	Title	Genres
1	2	Jumanji (1995)	Adventure Children's Fantasy
474	478	Jimmy Hollywood (1994)	Comedy
1934	2003	Gremlins (1984)	Comedy Horror
1935	2004	Gremlins 2: The New Batch (1990)	Comedy Horror
2111	2180	Torn Curtain (1966)	Thriller
2298	2367	King Kong (1976)	Action Adventure Horror
2930	2999	Man of the Century (1999)	Comedy
3429	3498	Midnight Express (1978)	Drama
3430	3499	Misery (1990)	Horror

3.5 Kết quả thực nghiệm

Trong bài báo cáo này, một mô hình gợi ý phim theo sở thích được xây dựng bằng cách kết hợp phương pháp Collaborative Filtering cùng với thuật toán K-Nearest Neighbors (KNN) và lọc dựa trên nội dung (Content-based Filtering). Mục tiêu là gợi ý các bộ phim cho một người dùng dựa trên sở thích của người dùng và các thông tin nội dung của các bộ phim.

Thuật toán KNN được sử dụng để tìm các người dùng giống nhau với người dùng hiện tại. Đầu tiên, một ma trận đánh giá được tạo ra từ các đánh giá của người dùng cho các bộ phim.

Sau đó, mô hình KNN được huấn luyện trên ma trận đánh giá để tìm các người dùng có sở thích tương tự. Kết quả là danh sách các người dùng giống nhau.

Tiếp theo, lọc dựa trên nội dung được thực hiện bằng cách tính toán độ tương tự cosine giữa các bộ phim dựa trên thông tin nội dung của chúng, trong trường hợp này là thể loại (genres) của phim. Độ tương tự này được tính bằng cách sử dụng vector đặc trưng TF-IDF.

Sau đó, hai phương pháp gợi ý được kết hợp lại để tạo ra danh sách các bộ phim được gợi ý. Các phim từ cả lọc cộng tác và lọc dựa trên nội dung được kết hợp và sau đó lọc bỏ các phim mà người dùng đã đánh giá. Danh sách các bộ phim cuối cùng này là kết quả gợi ý cho người dùng.

Đoạn mã cũng cung cấp một ví dụ về cách sử dụng mô hình. Người dùng được yêu cầu nhập một người dùng và một phim cụ thể, sau đó mô hình sẽ gợi ý danh sách các bộ phim dựa trên sở thích của người dùng và thông tin nội dung của phim đã cho.

Tóm lại, bài báo cáo này triển khai một mô hình gợi ý phim theo sở thích bằng cách kết hợp thuật toán KNN và lọc dựa trên nội dung. Mô hình này có thể

được sử dụng để gợi ý các bộ phim cho người dùng dựa trên sở thích của họ và các thông tin nội dung của các bộ phim.

CHƯƠNG 4. XÂY DỰNG SẢN PHẨM DEMO

4.1 Công cụ sử dụng

Trong bài báo cáo này, phần xây dựng sản phẩm demo là 1 website đơn giản sử dụng Flask cùng với HTML và Bootstrap để xây dựng giao diện.

4.1.1 Flask

Như bài báo cáo đã nói trong phần 3.2 Ngôn ngữ sử dụng, là Python có đề cập đến Flask.

Flask là một micro-framework được viết bằng ngôn ngữ lập trình Python dùng cho các nhà phát triển web. Nó được phát triển bởi Armin Ronacher, người dẫn đầu một nhóm những người đam mê Python quốc tế có tên là Pocco. [8]



Hình 4-1 Flask

Flask dựa trên bộ công cụ Werkzeug WSGI và template engine Jinja2. Cả hai đều là các dự án của Pocco. Micro ở đây không có nghĩa là framework này thiếu các chức năng mà thể hiện ở việc nó sẽ cung cấp những chức năng “cốt lõi” nhất

cho các ứng dụng web và có khả năng mở rộng, người dùng cũng có thể mở rộng bất cứ lúc nào vì Flask hỗ trợ rất nhiều các tiện ích mở rộng như tích hợp CSDL, hệ thống upload, xác thực, template, email... Việc là một micro-framework cũng giúp cho flask có một môi trường xử lý độc lập và ít phải sử dụng các thư viện bên ngoài, điều này giúp nó nhẹ và ít gặp phải các lỗi hơn, việc phát hiện và xử lý các lỗi cũng dễ dàng và đơn giản hơn.

Ưu điểm của Flask:[8]

- Siêu nhỏ nhẹ, là một công cụ tối giản.
- Tốc độ hoạt động cực nhanh.
- Có khả năng hỗ trợ NoQuery.
- Tương đối đơn giản (so với các framework có cùng chức năng khác)
- Mang lại khả năng kết nối với các tiện ích mở rộng bởi không có ORM.
- Trình duyệt được nhúng sẵn trình gỡ rối.
- Sử dụng các mã ngắn, đơn giản.
- Ngăn chặn các rủi ro về bảo mật khi lập trình web do ít phụ thuộc vào bên thứ ba.
- Có khả năng kiểm soát mọi vấn đề khi dùng Flask.
- Cho phép biên dịch module, thư viện, giúp việc lập trình nhanh chóng, dễ dàng hơn và không cần gõ code bậc thấp.

Nhược điểm của Flask: Chính vì siêu nhỏ nhẹ và tối giản, Flask không phải là một lựa chọn tốt nếu lập trình viên muốn một framework có đầy đủ các tính năng. Thay vào đó, lập trình viên sẽ phải tự gọi các tiện ích mà mình có nhu cầu sử dụng vì nó không được tích hợp sẵn trong framework, và đôi khi việc này trở nên bất tiện và khiến cho khối lượng công việc phải làm tăng lên đáng kể.

4.1.2 HTML

HTML hay Hyper Text Markup Language, được hiểu là ngôn ngữ đánh dấu siêu văn bản. Về bản chất HTML không phải là một ngôn ngữ lập trình mà là ngôn ngữ dùng để xây dựng cấu trúc cho một website bất kỳ.



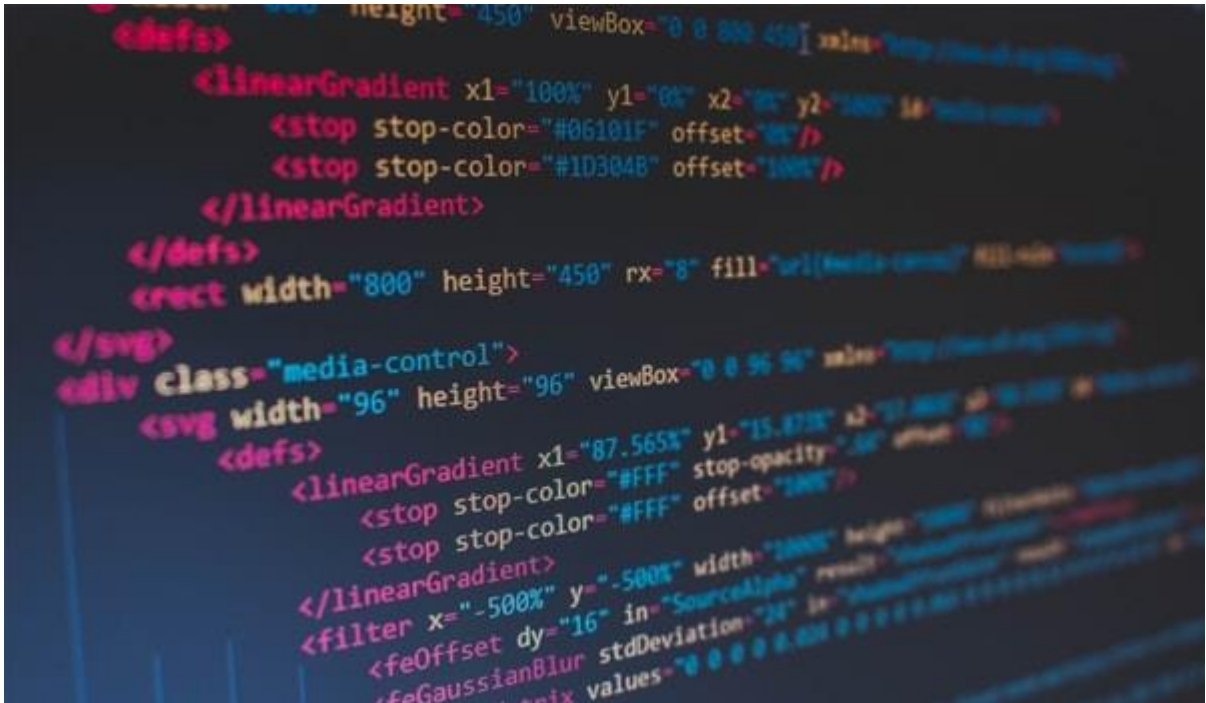
Hình 4-2 HTML

HTML được tạo ra bởi Tim Berners-Lee, một nhà vật lý học của trung tâm nghiên cứu CERN ở Thụy Sĩ. Hiện nay, HTML đã trở thành một chuẩn Internet được tổ chức W3C (World Wide Web Consortium) vận hành và phát triển.[9]

Phiên bản đầu tiên của HTML xuất hiện năm 1991, gồm 18 tag HTML. Phiên bản HTML 4.01 được xuất bản năm 1999. Sau đó, các nhà phát triển đã thay thế HTML bằng XHTML vào năm 2000.

Đến năm 2014, HTML được nâng cấp lên chuẩn HTML5 với nhiều tag được thêm vào markup, mục đích là để xác định rõ nội dung thuộc loại là gì (ví dụ như: <article>, <header>, <footer>,...).

Theo Mozilla Developer Network thì HTML Element Reference hiện nay có khoảng hơn 140 tag. Tuy nhiên một vài tag trong số đó đã bị tạm ngưng (do không được hỗ trợ bởi các trình duyệt hiện hành).



Hình 4-3 Một đoạn code bằng HTML

4.1.3 Bootstrap

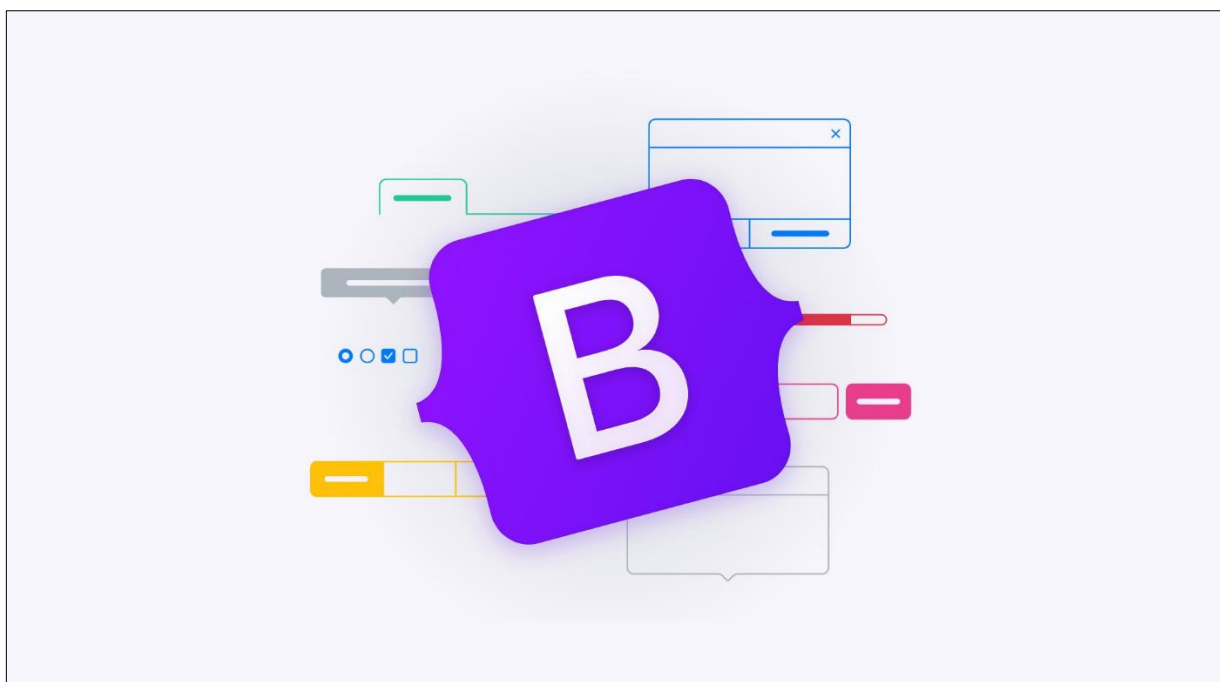
Bootstrap là sản phẩm của Mark Otto và Jacob Thornton tại Twitter. Nó được xuất bản như là một mã nguồn mở vào ngày 19/8/2011 trên GitHub. Tên gọi ban đầu là Twitter Blueprint.[10]



Hình 4-4 Bootstrap với giao diện lúc đầu

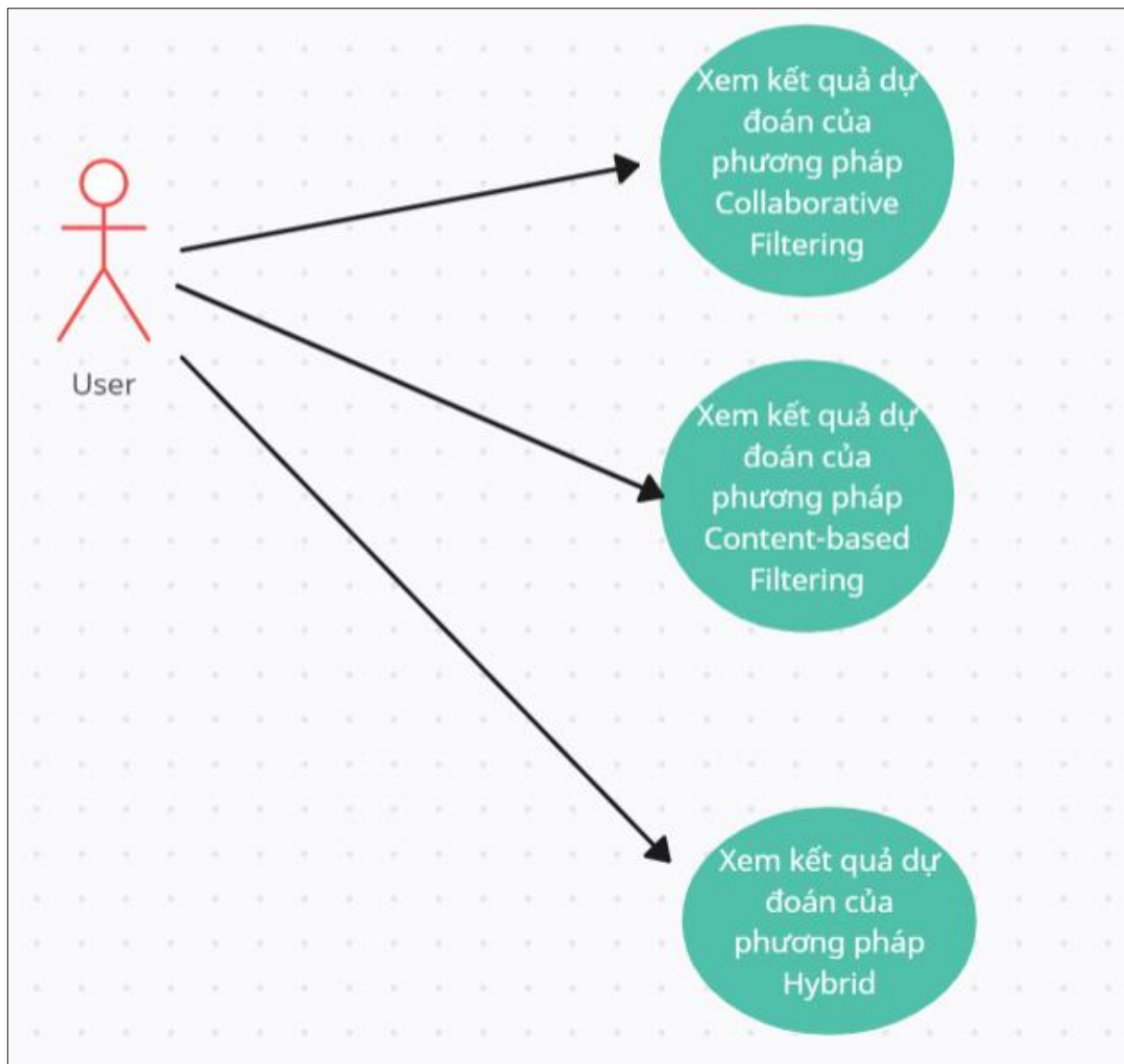
Một số đặc điểm chính của Bootstrap:

- **Responsive Design:** Bootstrap sử dụng hệ thống lưới (grid system) để tạo ra các thiết kế responsive, giúp trang web tự động điều chỉnh kích thước và hiển thị tốt trên các thiết bị khác nhau, từ máy tính để bàn đến điện thoại di động.
- **Pre-styled Components:** Cung cấp một loạt các thành phần giao diện người dùng (UI components) như nút, biểu mẫu, bảng, điều hướng, thẻ và nhiều hơn nữa, giúp tăng tốc quá trình phát triển và đảm bảo tính nhất quán trong thiết kế.
- **Customization:** Bootstrap cho phép tùy chỉnh dễ dàng thông qua các biến SASS, giúp bạn thay đổi màu sắc, kích thước, khoảng cách và nhiều thuộc tính khác mà không cần viết lại toàn bộ mã CSS.
- **Cross-browser Compatibility:** Đảm bảo trang web hiển thị tốt trên nhiều trình duyệt khác nhau, từ các trình duyệt phổ biến như Chrome, Firefox, Safari, đến các phiên bản cũ của Internet Explorer.
- **JavaScript Components:** Cung cấp các plugin JavaScript tích hợp sẵn để thêm chức năng động vào trang web như modal, tooltip, popover, carousel, và nhiều thành phần khác mà không cần phải viết nhiều mã JavaScript.



Hình 4-5 Bootstrap với giao diện mới

4.2 Phân tích thiết kế hệ thống



Hình 4-6 Use-case hệ thống

4.2.1 Use case Xem kết quả dự của phương pháp Collaborative Filtering

1. Tên use case: Xem kết quả dự đoán của phương pháp Collaborative Filtering.
2. Mô tả vắn tắt

Use case này cho phép người dùng xem kết quả dự đoán của phương pháp Collaborative Filing.

3. Luồng sự kiện

3.1. Luồng cơ bản

3.1.1. Use case này bắt đầu khi người dùng chọn 1 user cụ thể. Hệ thống sẽ hiển thị drop-down menu các Id của từng user tương ứng.

3.1.2. Khi người dùng kích nút “Gợi ý Collaborative”, hệ thống sẽ lọc danh sách các phim được gợi ý và hiển thị ra màn hình. Use case kết thúc.

3.2. Luồng rẽ nhánh: Không có.

4. Điều kiện đặc biệt: Không có.

5. Tiền điều kiện: Không có.

6. Hậu điều kiện: Không có.

7. Điểm mở rộng: Không có.

4.2.2 Use case Xem kết quả dự đoán của phương pháp Content-based Filtering

1. Tên use case: Xem kết quả dự đoán của phương pháp Content-based Filtering.

2. Mô tả vắn tắt

Use case này cho phép người dùng xem kết quả dự đoán của phương pháp Content-based Filing.

3. Luồng sự kiện

3.1. Luồng cơ bản

3.1.1. Use case này bắt đầu khi người dùng chọn tên 1 phim cụ thể. Hệ thống sẽ hiển thị drop-down menu tên phim tương ứng với từng id.

3.1.2. Khi người dùng kích nút “Gợi ý Content-Based”, hệ thống sẽ lọc danh sách các phim được gợi ý và hiển thị ra màn hình. Use case kết thúc.

3.2. Luồng rẽ nhánh: Không có.

4. Điều kiện đặc biệt: Không có.
5. Tiền điều kiện: Không có.
6. Hậu điều kiện: Không có.
7. Điểm mở rộng: Không có.

4.2.3 Use case Xem kết quả dự đoán của phương pháp Hybrid

1. Tên use case: Xem kết quả dự đoán của phương pháp Hybrid Filtering.
2. Mô tả vắn tắt

Use case này cho phép người dùng xem kết quả dự đoán của phương pháp Hybrid Filtering.

3. Luồng sự kiện

3.1. Luồng cơ bản

3.1.1. Use case này bắt đầu khi người dùng chọn id của một user và tên 1 phim cụ thể. Hệ thống sẽ hiển thị drop-down menu id của các user và tên phim.

3.1.2. Khi người dùng kích nút “Gợi ý Hybrid”, hệ thống sẽ lọc danh sách các phim được gợi ý và hiển thị ra màn hình. Use case kết thúc.

- 3.2. Luồng rẽ nhánh: Không có.
4. Điều kiện đặc biệt: Không có.
5. Tiền điều kiện: Không có.
6. Hậu điều kiện: Không có.
7. Điểm mở rộng: Không có.

4.3 Quy trình xây dựng giao diện

Đầu tiên, để sử dụng Flask cần phải tải Flask về project thông qua lệnh sau:

```
pip install flask
```

Sau đó import Flask vào project hiện tại và sử dụng:

```
from flask import Flask, request, render_template
```

Sau đó chuyển các function của 3 phương pháp Filtering là Collaborative Filtering, Content-based Filtering và Hybrid Filtering về dạng API(Application Programming Interface) để sử dụng cho phần front-end:

```
@app.route('/')
def index():
    movies = movies_df[['MovieID', 'Title']].to_dict(orient='records')
    users = users_df['UserID'].tolist()
    return render_template('index.html', movies=movies, users=users)

@app.route('/recommend/collaborative', methods=['POST'])
def recommend_collaborative():
    user_id = int(request.form['user_id'])
    recommendations = recommend_movies(user_id)
    collab_movies = movies_df[movies_df['MovieID'].isin(recommendations)][['MovieID', 'Title', 'Genres']]
    return render_template('index.html', collab_movies=collab_movies.to_dict(orient='records'), movies=movies_df[['MovieID', 'Title']].to_dict(orient='records'), users=users_df['UserID'].tolist())

@app.route('/recommend/content', methods=['POST'])
def recommend_content():
    movie_id = int(request.form['movie_id'])
    recommendations = content_based_recommendation(movie_id)
    return render_template('index.html', content_movies=[movie.to_dict() for movie in recommendations], movies=movies_df[['MovieID', 'Title']].to_dict(orient='records'), users=users_df['UserID'].tolist())

@app.route('/recommend/hybrid', methods=['POST'])
def recommend_hybrid():
    user_id = int(request.form['hybrid_user_id'])
    movie_id = int(request.form['hybrid_movie_id'])
    recommendations = hybrid_recommendation(user_id, movie_id)
    hybrid_movies = movies_df[movies_df['MovieID'].isin(recommendations)][['MovieID', 'Title', 'Genres']]
    return render_template('index.html', hybrid_movies=hybrid_movies.to_dict(orient='records'), movies=movies_df[['MovieID', 'Title']].to_dict(orient='records'), users=users_df['UserID'].tolist())

if __name__ == '__main__':
    app.run(debug=True)
```

Với phần index.html sẽ cần 3 form lần lượt chứa 3 tag <select> để lựa chọn id người dùng cần dự đoán, tên phim cần dự đoán, và cả id và tên phim muốn dự đoán:

```

<div class="container mt-3">
  <!-- Collaborative Filtering Form -->
  <form method="post" action="/recommend/collaborative">
    <div class="form-group">
      <label for="user_id">Select User ID for Collaborative Filtering</label>
      <select class="form-control" id="user_id" name="user_id" required>
        {% for user in users %}
          <option value="{{ user }}">{{ user }}</option>
        {% endfor %}
      </select>
    </div>
    <button type="submit" class="btn btn-primary">Gợi ý Collaborative</button>
  </form>

  <!-- Content-Based Filtering Form -->
  <form method="post" action="/recommend/content" class="mt-4">
    <div class="form-group">
      <label for="movie_id">Select Movie for Content-Based Filtering</label>
      <select class="form-control" id="movie_id" name="movie_id" required>
        {% for movie in movies %}
          <option value="{{ movie.MovieID }}">{{ movie.Title }}</option>
        {% endfor %}
      </select>
    </div>
    <button type="submit" class="btn btn-primary">Gợi ý Content-Based</button>
  </form>

```

```

<!-- Hybrid Filtering Form -->
<form method="post" action="/recommend/hybrid" class="mt-4">
  <div class="form-group">
    <label for="hybrid_user_id">Select User ID for Hybrid Filtering</label>
    <select class="form-control" id="hybrid_user_id" name="hybrid_user_id" required>
      {% for user in users %}
        <option value="{{ user }}">{{ user }}</option>
      {% endfor %}
    </select>
  </div>
  <div class="form-group">
    <label for="hybrid_movie_id">Select Movie for Hybrid Filtering</label>
    <select class="form-control" id="hybrid_movie_id" name="hybrid_movie_id" required>
      {% for movie in movies %}
        <option value="{{ movie.MovieID }}">{{ movie.Title }}</option>
      {% endfor %}
    </select>
  </div>
  <button type="submit" class="btn btn-primary">Gửi ý Hybrid</button>
</form>

```

```

<!-- Collaborative Filtering Results -->
{% if collab_movies %}
<h2 class="mt-5">Collaborative Filtering Recommendations</h2>
<ul class="list-group">
  {% for movie in collab_movies %}
    <li class="list-group-item">{{ movie.Title }} - Genres: {{ movie.Genres }}</li>
  {% endfor %}
</ul>
{% endif %}

<!-- Content-Based Filtering Results -->
{% if content_movies %}
<h2 class="mt-5">Content-Based Recommendations</h2>
<ul class="list-group">
  {% for movie in content_movies %}
    <li class="list-group-item">{{ movie.Title }} - Genres: {{ movie.Genres }}</li>
  {% endfor %}
</ul>
{% endif %}

<!-- Hybrid Filtering Results -->
{% if hybrid_movies %}
<h2 class="mt-5">Hybrid Recommendations</h2>
<ul class="list-group">
  {% for movie in hybrid_movies %}
    <li class="list-group-item">{{ movie.Title }} - Genres: {{ movie.Genres }}</li>
  {% endfor %}
</ul>
{% endif %}

```

4.4 Giao diện sản phẩm

Movie Recommendations

Bộ dữ liệu được sử dụng trong bài báo cáo là bộ dữ liệu movie-lens 1M DataSet được tạo ra bởi GroupLens Research là một nhóm nghiên cứu thuộc Đại học Minnesota và được phát hành vào tháng 2 năm 2003.

Bộ dữ liệu này chứa thông tin của 1.000.209 đánh giá ẩn danh của khoảng 3.900 phim được tạo bởi 6.040 người dùng MovieLens vào năm 2000. Bao gồm 3 tập dữ liệu: movies.dat chứa thông tin của các bộ phim bao gồm: MovieId, Title(tên phim) và Genres(thể loại phim), users.dat chứa thông tin của UserId, Giới tính, Tuổi, Nghề nghiệp, mã zip, Ratings.dat chứa thông tin của UserId, MovieId, Rating(số điểm đánh giá) và Timestamp(Thời gian đánh giá).

Select User ID for Collaborative Filtering

1

Gợi ý Collaborative

Select Movie for Content-Based Filtering

Toy Story (1995)

Gợi ý Content-Based

Select User ID for Hybrid Filtering

1

Select Movie for Hybrid Filtering

Toy Story (1995)

Gợi ý Hybrid

Hình 4-7 Màn hình chính của sản phẩm

Movie Recommendations

Bộ dữ liệu được sử dụng trong bài báo cáo là bộ dữ liệu movie-lens 1M DataSet được tạo ra bởi GroupLens Research là một nhóm nghiên cứu thuộc Đại học Minnesota và được phát hành vào tháng 2 năm 2003.

Bộ dữ liệu này chứa thông tin của 1.000.209 đánh giá ẩn danh của khoảng 3.900 phim được tạo bởi 6.040 người dùng MovieLens vào năm 2000. Bao gồm 3 tập dữ liệu: movies.dat chứa thông tin của các bộ phim bao gồm: MovieId, Title(tên phim) và Genres(thể loại phim), users.dat chứa thông tin của UserId, Giới tính, Tuổi, Nghề nghiệp, mã zip, Ratings.dat chứa thông tin của UserId, MovieId, Rating(số điểm đánh giá) và Timestamp(Thời gian đánh giá).

Select User ID for Collaborative Filtering

1

1
2
3
4
5
6
7
8
9
10
11
12
13
14
15
16
17
18
19

Hình 4-8 Màn hình chọn người dùng muốn dự đoán phim

Collaborative Filtering Recommendations	
Twelve Monkeys (1995) - Genres: Drama Sci-Fi	
Star Wars: Episode IV - A New Hope (1977) - Genres: Action Adventure Fantasy Sci-Fi	
Blade Runner (1982) - Genres: Film-Noir Sci-Fi	
Godfather, The (1972) - Genres: Action Crime Drama	
Casablanca (1942) - Genres: Drama Romance War	
Gone with the Wind (1939) - Genres: Drama Romance War	
Star Wars: Episode V - The Empire Strikes Back (1980) - Genres: Action Adventure Drama Sci-Fi War	
Boat, The (Das Boot) (1981) - Genres: Action Drama War	
Back to the Future (1985) - Genres: Comedy Sci-Fi	
Star Trek: The Wrath of Khan (1982) - Genres: Action Adventure Sci-Fi	

Hình 4-9 Danh sách các phim được hệ thống gợi ý bằng Collaborative Filtering

The screenshot shows a web application interface for movie recommendations. At the top, there's a navigation bar with icons for back, forward, refresh, and home. Below this, a dropdown menu is open, displaying a list of movies including 'Toy Story (1995)', 'Jumanji (1995)', 'Grumpier Old Men (1995)', 'Waiting to Exhale (1995)', 'Father of the Bride Part II (1995)', 'Heat (1995)', 'Sabrina (1995)', 'Tom and Huck (1995)', 'Sudden Death (1995)', 'GoldenEye (1995)', 'American President, The (1995)', 'Dracula: Dead and Loving It (1995)', 'Balto (1995)', 'Nixon (1995)', 'Cutthroat Island (1995)', 'Casino (1995)', 'Sense and Sensibility (1995)', 'Four Rooms (1995)', 'Ace Ventura: When Nature Calls (1995)', and 'Money Train (1995)'. Below the list, there's a button labeled 'Gợi ý Content-Based'. Underneath that, there are two dropdown menus for 'Hybrid Filtering'. The first is labeled 'Select User ID for Hybrid Filtering' and has '1' selected. The second is labeled 'Select Movie for Hybrid Filtering' and has 'Toy Story (1995)' selected.

Hình 4-10 Màn hình lựa chọn tên phim muốn gợi ý

Content-Based Recommendations

Assassins (1995) - Genres: Thriller
Unforgettable (1996) - Genres: Thriller
Jade (1995) - Genres: Thriller
Mute Witness (1994) - Genres: Thriller
Safe (1995) - Genres: Thriller
Tie That Binds, The (1995) - Genres: Thriller
Dream Man (1995) - Genres: Thriller
Hideaway (1995) - Genres: Thriller
Poison Ivy II (1995) - Genres: Thriller
Shallow Grave (1994) - Genres: Thriller

Hình 4-11 Danh sách các phim được gợi ý bằng Content-based Filtering

Select User ID for Hybrid Filtering

14

Select Movie for Hybrid Filtering

Casino (1995)

Gợi ý Hybrid

Hình 4-12 Màn hình lựa chọn ID người dùng và tên phim muốn gợi ý

Hybrid Recommendations

Balto (1995) - Genres: Animation Children's
Little Princess, A (1995) - Genres: Children's Drama
Pretty Woman (1990) - Genres: Comedy Romance
Homeward Bound II: Lost in San Francisco (1996) - Genres: Adventure Children's
Asfour Stah (1990) - Genres: Drama
Next Step, The (1995) - Genres: Drama
In Search of the Castaways (1962) - Genres: Adventure Children's
Those Who Love Me Can Take the Train (Ceux qui m'aiment prendront le train) (1998) - Genres: Drama

Hình 4-13 Danh sách các phim được gợi ý bằng Hybrid Filtering

KẾT LUẬN

Qua các chương trong bài báo cáo cụ thể là 4 chương: Chương 1 – Giới thiệu chung về phim, ảnh, hệ thống gợi ý; Chương 2 – Các kỹ thuật giải quyết bài toán gợi ý phim; Chương 3 – Quy trình xây dựng bài toán gợi ý phim bằng Collaborative Filtering và Content-based Filtering thông qua ngôn ngữ lập trình Python trên IDE Pycharm version 2023.1 và Chương 4 – chương cuối của bài báo cáo đã xây dựng một website đơn giản sử dụng framework Flask kết hợp cùng với HTML và Bootstrap để phần nào làm rõ được cách các đoạn code đã trình bày ở Chương 3 ứng dụng vào bài toán thực tế.

Bài báo cáo có ý nghĩa quan trọng trong việc xây dựng một hệ thống gợi ý phim nói riêng và các bài toán hệ thống gợi ý khác nói chung.

Mô hình gợi ý phim theo sở thích được ứng dụng rộng rãi trong các trang web phim ảnh nổi tiếng như Netflix.com, Amazon Prime Video, Hulu, Disney+, IMDb TV... và nhiều dịch vụ khác. Các trang web này sử dụng các thuật toán và phương pháp khác nhau để gợi ý phim dựa trên sở thích và hành vi xem phim của người dùng. Ví dụ, với Netflix sử dụng một hệ thống gợi ý phức tạp dựa trên thuật toán máy học để đề xuất nội dung cho người dùng. Họ thu thập thông tin về lịch sử xem phim, đánh giá, đánh dấu yêu thích và các thông tin khác để hiểu sở thích cá nhân của người dùng. Sau đó, họ sử dụng các thuật toán gợi ý như Collaborative Filtering (Lọc cộng tác) và Content-Based Filtering (Lọc theo nội dung) để đề xuất các bộ phim tương tự hoặc liên quan mà người dùng có thể quan tâm...

Trong bản báo cáo đã trình bày được một phần cơ bản nhất của mô hình gợi ý phim theo sở thích bằng các phương pháp lọc cộng tác, lọc theo nội dung và phương pháp lọc kết hợp cùng với thuật toán KNN là một thuật toán được sử dụng rộng rãi trong các bài toán đưa ra gợi ý, đề xuất nói chung và bài toán gợi ý phim nói riêng.

Tuy nhiên trong bài báo cáo với đoạn mã nguồn đã phân tích ở chương 3, phần 3.4.2 thì do mô hình yêu cầu phải truyền vào người dùng nào muốn dự đoán nên đôi khi sẽ không được thuận tiện để áp dụng cho các trang web lớn, vì lúc đó số lượng người đăng nhập rất nhiều. Lúc đó có thể sẽ xảy ra sai sót trong quá trình dự đoán vì mô hình trên chỉ là một phần nhỏ trong các bài toán lớn. Ngoài ra, do không phải bài toán phân lớp hay phân cụm nên mô hình gợi ý đã phân tích thường khó xác định đúng được các phim đó có thực sự người dùng sẽ thích hay không.

Để cải thiện mô hình, cần phải xem xét lại trên nhiều yếu tố như cách chọn số láng giềng gần nhất, bộ dữ liệu cần thêm các thông tin như tên người dùng, thời lượng xem phim trung bình trên 1 ngày... và các phương pháp liên quan đến tiền xử lý dữ liệu để giúp mô hình dự đoán một cách chính xác hơn.

Cuối cùng, em xin chân thành cảm ơn đến cô Đặng Quỳnh Nga đã cung cấp những kiến thức chuyên môn liên quan đến lĩnh vực khai thác dữ liệu, quá trình thực hiện báo cáo và đặc biệt là những gợi ý về nội dung tổng quan về các phần trong bài báo cáo này.

TÀI LIỆU THAM KHẢO

- [1] Website giới thiệu điện ảnh là gì, cơ hội nghề nghiệp trong ngành điện ảnh. URL: <https://www.careerlink.vn/cam-nang-viec-lam/tu-van-nghe-nghiep/dien-anh-la-gi-co-hoi-viec-lam-trong-nganh-dien-anh>. Lần truy cập gần nhất ngày: 20/03/2024
- [2] Website giới thiệu phim là gì. URL: <https://thuvienphapluat.vn/hoi-dap-phap-luat/201C9-hd-phim-la-gi.html>. Lần truy cập gần nhất ngày: 20/03/2024
- [3] Website giới thiệu hệ thống khuyến nghị. URL: <https://digital.fpt.com/linh-vuc/he-thong-khuyen-nghi.html>. Lần truy cập gần nhất ngày: 21/03/2024
- [4] Website giới thiệu Thuật toán SVM. URL: <https://viblo.asia/p/gioi-thieu-ve-support-vector-machine-svm-6J3ZgPVEImB>. Lần truy cập gần nhất ngày: 21/03/2024
- [5] Website giới thiệu Thuật toán Naïve Bayes. URL: <https://trituenhantao.io/kien-thuc/phan-1-phan-loai-naive-bayes-ly-thuyet/>. Lần truy cập gần nhất ngày: 22/03/2024
- [6] Website giới thiệu Linear Regression trong Machine Learning. URL: <https://viblo.asia/p/linear-regression-hoi-quy-tuyen-tinh-trong-machine-learning-4P856akRIY3>. Lần truy cập gần nhất ngày: 22/03/2024
- [7] Website thống kê top các ngôn ngữ lập trình được sử dụng trong năm 2022. URL: <https://octoverse.github.com/2022/top-programming-languages>. Lần truy cập gần nhất ngày: 15/04/2024
- [8] Website giới thiệu Flask là gì. URL: <https://www.mcivietnam.com/blog-detail/flask-python-la-gi-so-sanh-flask-va-django/>. Lần truy cập gần nhất ngày: 20/04/2024
- [9] Website giới thiệu HTML. URL: https://wiki.matbao.net/html-la-gi-nen-tang-lap-trinh-web-cho-nguoi-moi-bat-dau/?gad_source=1&gclid=CjwKCAjwo6GyBhBwEiwAzQTmcwZZMZSB3eU

5H-MNLxMzEHRJI2KQJfziMXD3cfjeGLwO4YpQ_pfDgRoC8lgQAvD_BwE

Lần truy cập gần nhất ngày: 21/04/2024

[10] Website giới thiệu Bootstrap. URL: https://wiki.matbao.net/bootstrap-la-gi-cai-dat-bootstrap-web-chuan-responsive/?gad_source=1&gclid=CjwKCAjwo6GyBhBwEiwAzQTmcw_kP6WWDpAtFO8Kpb2jLAJ6cic8AZIGxHAhuBmDCS8_kI227b4lWRoCKhEQAvD_BwE Lần truy cập gần nhất ngày: 21/04/2024