|  |
| --- |
| **BỘ GIÁO DỤC VÀ ĐÀO TẠO**  **TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP.HỒ CHÍ MINH** |
| **KHOA CÔNG NGHỆ THÔNG TIN**  ****  **Logo ĐH Sư Phạm Kỹ Thuật TP Hồ Chí Minh-HCMUTE** |

|  |
| --- |
| **BÁO CÁO ĐỀ TÀI CUỐI KỲ** |
| **Môn học: KHAI PHÁ DỮ LIỆU**  **Mã lớp học phần: DAMI330484\_23\_2\_02**  **GVHD: Trần Trọng Bình** |
| **Nhóm sinh viên thực hiện: Nhóm 2**   |  |  | | --- | --- | | Dương Hoàng Khang | 21133046 | | Đào Nguyễn Hoàng Ngân | 21110553 | | Lương Tường Vy | 21133093 | |

**TP. Hồ Chí Minh, 30 tháng 05 năm 2024**

**DANH SÁCH THÀNH VIÊN THAM GIA**

**THỰC HIỆN ĐỀ TÀI VÀ VIẾT BÁO CÁO**

**Môn: Khai phá dữ liệu -** *HỌC KÌ II – NĂM HỌC 2023 – 2024*

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **HỌ VÀ TÊN** | **MSSV** | **TỶ LỆ ĐÓNG GÓP** |
| 1 | Dương Hoàng Khang | 21133046 | 100% |
| 2 | Đào Nguyễn Hoàng Ngân | 21110553 | 100% |
| 3 | Lương Tường Vy | 21133093 | 100% |

**Nhận xét của giảng viên:**

**……………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………**

*Ngày … tháng 05 năm 2024*

*Giảng viên chấm điểm*

**LỜI CẢM ƠN**

Lời đầu tiên, nhóm chúng em xin gửi lời cảm ơn chân thành đến **Trường Đại học Sư phạm Kỹ thuật TPHCM** đã đưa môn học **Khai phá dữ liệu** vào chương trình giảng dạy. Đặc biệt, chúng em xin gửi lời cảm ơn sâu sắc đến giảng viên bộ môn – **Th.S Trần Trọng Bình** đã dạy dỗ, truyền đạt những kiến thức quý báu cho chúng em trong suốt thời gian học tập vừa qua. Trong thời gian tham gia lớp học, nhóm em đã có thêm cho mình nhiều kiến thức bổ ích, tinh thần học tập hiệu quả, nghiêm túc. Đây chắc chắn sẽ là những kiến thức quý báu, là hành trang để chúng em có thể vững bước sau này. Bộ môn Khai phá dữ liệu là môn học thú vị, vô cùng bổ ích và có tính thực tế cao. Đảm bảo cung cấp đủ kiến thức, gắn liền với nhu cầu thực tiễn của sinh viên. Mặc dù chúng em đã cố gắng hết sức nhưng chắc chắn bài đồ án này khó có thể tránh khỏi những thiếu sót và nhiều chỗ còn chưa chính xác, kính mong thầy xem xét và góp ý để bài đồ án của chúng em được hoàn thiện hơn.

Nhóm chúng em xin chân thành cảm ơn!

**MỤC LỤC**

[**CHƯƠNG I: KHAI PHÁ LUẬT KẾT HỢP** 4](#_Toc167995536)

[**1. Thông tin về tập dữ liệu** 4](#_Toc167995537)

[**2. Phân tích về kỹ thuật khai phá áp dụng** 4](#_Toc167995538)

[**2.1. Khai phá luật kết hợp** 4](#_Toc167995539)

[**2.2. Thuật toán Apriori** 5](#_Toc167995540)

[**a.** **Tổng quan** 5](#_Toc167995541)

[**b.** **Khái niệm thuật toán:** 6](#_Toc167995542)

[**c.** **Các thành phần chính:** 6](#_Toc167995543)

[**2.3. Ưu nhược điểm của kỹ thuật khai phá đối với tập dữ liệu** 7](#_Toc167995544)

[**2.4. Ý nghĩa của kết quả khai phá** 8](#_Toc167995545)

[**3. Hệ thống gợi ý** 9](#_Toc167995546)

[**3.1. MATRIX FACTORIZATION** 9](#_Toc167995547)

[**3.2 Ưu nhược điểm của hệ thống gợi ý đối với tập dữ liệu** 10](#_Toc167995548)

[**3.3. Ý nghĩa của kết quả khai phá** 11](#_Toc167995549)

[**CHƯƠNG II: K NEAREST NEIGBBORS** 13](#_Toc167995550)

[**1. Tổng quan về hệ thống** 13](#_Toc167995551)

[**1.1. Khái niệm** 13](#_Toc167995552)

[**1.2. Các loại hệ thống đề xuất** 13](#_Toc167995553)

[**1.3. Thuật toán K Nearest Neighbors** 14](#_Toc167995554)

[**2. Thông tin về tập dữ liệu** 15](#_Toc167995555)

[**2.1. Nguồn dữ liệu** 15](#_Toc167995556)

[**2.2. Mô tả chi tiết tập dữ liệu** 15](#_Toc167995557)

[**3. Khám phá và tiền xử lý dữ liệu** 16](#_Toc167995558)

[**3.1. Giới thiệu** 16](#_Toc167995559)

[**3.2. Tiền xử lý dữ liệu** 17](#_Toc167995560)

[**4. Khai phá dữ liệu** 17](#_Toc167995561)

[**4.1. Phân tích dữ liệu** 17](#_Toc167995562)

[**4.2. Trực quan hóa dữ liệu bằng biểu đồ** 18](#_Toc167995563)

[**4.3. Triển khai** 18](#_Toc167995564)

[**CHƯƠNG III: KẾT LUẬN** 20](#_Toc167995565)

[**1.** **Kết quả đạt được** 20](#_Toc167995566)

[**2.** **Hạn chế** 20](#_Toc167995567)

[**3.** **Quá trình làm việc nhóm** 20](#_Toc167995568)

# **CHƯƠNG I: KHAI PHÁ LUẬT KẾT HỢP**

# **1. Thông tin về tập dữ liệu**

Tập dữ liệu được tham khảo từ Grouplens.org.

Đường dẫn tập dữ liệu: [MovieLens](https://grouplens.org/datasets/movielens/)

Ở đây tập dữ liệu gồm có 2 file csv

* File “movies.csv” gồm có 9742 dòng dữ liệu và 3 cột:
* movieId: Mã của phim
* title: Tên của phim
* genres: Thể loại tương ứng với phim đó
* File “ratings.csv” gồm có 100836 dòng dữ liệu và 3 cột:
* userId: Mã của người dùng
* movieId: Mã của phim
* rating: Đánh giá của người dùng (trên thang điểm 1-5)

# **2. Phân tích về kỹ thuật khai phá áp dụng**

## **2.1. Khai phá luật kết hợp**

Mục đích của bài toán phát hiện luật kết hợp là tìm ra mối quan hệ giữa các tập mục dữ liệu trong các cơ sở dữ liệu lớn và các mối quan hệ này là có ích trong hỗ trợ quyết định.

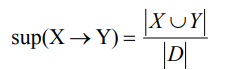
Ví dụ: trong cơ sở dữ liệu siêu thị, việc phát hiện được quan hệ "78% số khách hàng mua sữa và đường cũng mua bơ" sẽ rất có ích cho quyết định kinh doanh, chẳng hạn, quyết định về số lượng nhập các mặt hàng này hoặc bố trí chúng tại các ngăn hàng liền kề nhau. Trong cơ sở dữ liệu dân số, quan hệ "60% số người lao động ở độ tuổi trung niên có thu nhập thấp hơn mức thu nhập bình quân" sẽ rất có ích cho việc điều chỉnh chính sách thu nhập.

Khái niệm: Luật kết hợp (Association Rule) và phát hiện luật kết hợp (Association Rule Mining) được Rakesh Agrawal và cộng sự đề xuất lần đầu tiên vào năm 1993 nhằm phát hiện các mẫu có giá trị trong cơ sở dữ liệu tác vụ (transaction database) tại siêu thị. Bài toán này được phát biểu hình thức như sau: Kí hiệu I = {i1, i2,..., in} là tập các mục dữ liệu (mỗi mặt hàng trong siêu thị chính là một mục dữ liệu, và cũng có thể xem nó là một thuộc tính nhận giá trị nhị phân, khi đó I là các thuộc tính của CSDL); tập X ⊂ I được gọi là tập mục dữ liệu hoặc tập mục (itemset); và O = {t1, t2,..., tm} là tập định danh của các tác vụ (mỗi vụ mua hàng được xem là một tác vụ). Quan hệ D ⊆ I×O được gọi là cơ sở dữ liệu tác vụ. Mỗi tác vụ t được biểu diễn như một véc tơ nhị phân, trong đó t[k] = 1 nếu mặt hàng ik xuất hiện trong t và ngược lại t[k] = 0.

Cho một tập mục dữ liệu X ⊆ I, độ hỗ trợ của tập X, kí hiệu là sup(X), được định nghĩa là số (hoặc phần trăm) tác vụ trong D chứa X.

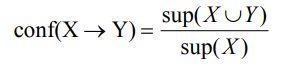
Luật kết hợp (association rule) được định nghĩa hình thức là biểu diễn mối quan hệ giữa hai tập mục dưới dạng X →Y, trong đó X ⊆ I, Y ⊆ I, X ∩Y = ∅. X được gọi là phần tiền đề (antecedent) và Y được gọi là phần hệ quả (consequent) của luật.

Độ hỗ trợ (support) của luật X → Y, kí hiệu là sup(X → Y), được định nghĩa là số (hoặc phần trăm) tác vụ trong D chứa X∪Y.



Theo Agrawal R. và cộng sự, luật kết hợp được phát hiện cần đáp ứng ràng buộc độ hỗ trợ (support constraint), theo đó, độ hỗ trợ của tập mục W = X∪Y (hợp tập tiền đề và tập hệ quả của luật) phải vượt qua (không nhỏ thua) một ngưỡng hỗ trợ tối thiểu do người dùng đưa vào. Mọi tập W có tính chất nói trên được gọi là tập phổ biến (frequent itemset) và còn được gọi là tập mục lớn (large itemset).

Độ tin cậy (confidence) của luật X → Y, kí hiệu là conf(X →Y), được định nghĩa là số (hoặc phần trăm) tác vụ trong D chứa X cũng chứa Y.



Luật kết hợp được phát hiện cần có tính tin cậy, theo đó nó cần có độ tin cậy vượt qua (không nhỏ thua) một ngưỡng tin cậy tối thiểu do người dùng đưa vào.

Luật đáp ứng ràng buộc độ hỗ trợ và có tính tin cậy được gọi là luật mạnh (strong association rule).

## **2.2. Thuật toán Apriori**

1. **Tổng quan**

Thuật toán Apriori được sử dụng để tính toán sự liên kết giữa các đối tượng. **Trả lời cho câu hỏi hai hoặc nhiều đối tượng đang xét có mối gắn kết với nhau hay không?**. Hay nói cách khác, chúng ta có thể nói rằng thuật toán Apriori là một luật kết hợp dựa trên phân tích hành động: Những người mua sản phẩm A cũng mua sản phẩm B. (Thuật toán thuộc giai đoạn I)

Mục tiêu chính của thuật toán Apriori là **tìm ra các tập phổ biến** giữa các đối tượng khác nhau. Quy tắc kết hợp mô tả cách hai hoặc nhiều đối tượng có liên quan đến đối tượng khác. Thuật toán Apriori còn được gọi là **frequent pattern mining**. Nhìn chung, thuật toán vận hành trên cơ sở dữ liệu bao gồm một số lượng lớn giao dịch.

**Lợi ích**: Giả sử khách hàng đến một siêu thị nổi tiếng tên là A và mua các sản phẩm theo combo. Nó giúp khách hàng mua sản phẩm của họ một cách dễ dàng và tăng hiệu suất bán hàng của siêu thị A.

**Tình huống giả định:** Người bán cửa hàng Pizza làm một combo bánh pizza, nước ngọt và bánh mì cùng nhau. Anh ấy cũng cung cấp giảm giá cho khách hàng mua combo này. **Bạn có bao giờ nghĩ tại sao anh ấy lại làm như vậy không?** Anh ấy nhận thấy rằng khách hàng mua pizza cũng mua nước ngọt và bánh mì que. Vì vậy bằng cách thực hiện các combo, anh ấy đã tạo điều kiện thuận lợi cho khách hàng. Đồng thời, anh cũng nâng cao hiệu suất bán hàng của mình. Tương tự, khi chúng ta đến siêu thị A và chúng sẽ tìm thấy bánh quy, khoai tây chiên và Sôcôla được gói cùng nhau. Nó cho thấy người bán hàng tạo điều kiện thuận lợi cho khách hàng mua những sản phẩm này ở cùng một nơi.

1. **Khái niệm thuật toán:**

Thuật toán Apriori được sử dụng để khai thác **tập phổ biến** hay **luật kết hợp**. Nói chung, thuật toán apriori hoạt động trên cơ sở dữ liệu chứa số lượng giao dịch khổng lồ. Ví dụ: các mặt hàng dành cho khách hàng tại siêu thị A.

1. **Các thành phần chính:**

* **Support**
* **Confident**
* **Lift**

Để hiểu rõ hơn ta có tình huống giả định sau: Trong số 4000 giao dịch, 400 giao dịch chứa Bánh quy, trong khi 600 giao dịch chứa Sôcôla và 600 giao dịch này bao gồm 200 giao dịch bao gồm Bánh quy và sôcôla. Sử dụng dữ liệu này, chúng ta sẽ tìm ra **Support**, **Confident** và **Lift**.

* **Support:**

Đề cập đến độ phổ biến của sản phẩm. Giá trị được tìm thấy bằng cách tính dưới dạng thương số của việc chia số lượng giao dịch( bao gồm sản phẩm đó) cho tổng số giao dịch.

Ta có ví dụ:

**Support**(Bánh quy) = (Giao dịch liên quan đến bánh quy) / (Tổng số giao dịch) = 400/4000 = 10 phần trăm.

* **Confident:**

Đề cập đến xác xuất khách hàng mua cả hai sản phẩm với nhau (bánh quy và sôcôla). Công thức được tính bằng cách chia số lượng giao dịch bao gồm cả bánh quy và sôcôla cho tổng số giao dịch

**Confident** = (Giao dịch liên quan đến cả bánh quy và Sôcôla) / (Tổng số giao dịch liên quan đến Bánh quy) = 200/400 = 50 %. Điều đó có nghĩa là 50% khách hàng mua bánh quy cũng mua sôcôla.

* **Lift:**

**Lift** đề cập đến sự gia tăng tỷ lệ bán của một sản phẩm (sôcôla) khi bán một sản phẩm khác (bánh quy).

**Ta có công thức sau:**

Lift = (**Confident**(Bánh quy - sôcôla) / (**Support**(Bánh quy) = 50/10 = 5

**Kết luận:** Điều đó có nghĩa là xác suất để mọi người mua cả bánh quy và sôcôla cùng nhau gấp 5 lần so với xác suất mua một mình bánh quy. Nếu giá trị mức tăng dưới một, nó yêu cầu mọi người khó có thể mua cả hai mặt hàng cùng nhau. Giá trị càng lớn thì sự kết hợp càng tốt.

## **2.3. Ưu nhược điểm của kỹ thuật khai phá đối với tập dữ liệu**

Ưu điểm của kỹ thuật khai phá luật kết hợp (association rule mining) khi áp dụng vào tập dữ liệu MovieLens như đã mô tả là:

1. **Phân tích quan hệ giữa các thể loại phim**: Kỹ thuật này cho phép phân tích mối quan hệ giữa các thể loại phim một cách chi tiết. Ví dụ, thông qua việc áp dụng kỹ thuật Apriori, ta có thể tìm ra các cặp thể loại phim thường đi kèm nhau, như Adventure và Children, hoặc Comedy và Romance. Điều này giúp hiểu sâu hơn về sở thích của người xem và cách họ liên kết các thể loại phim lại với nhau.
2. **Dễ dàng tìm hiểu dữ liệu**: Kỹ thuật này cung cấp một cách tiếp cận trực quan và dễ hiểu để khám phá dữ liệu. Bằng cách hiển thị các quy tắc liên kết dưới dạng bảng và biểu đồ, người dùng có thể nhanh chóng nhận thấy mối quan hệ giữa các thể loại phim và hiểu rõ hơn về dữ liệu.
3. **Áp dụng linh hoạt**: Kỹ thuật khai phá luật kết hợp có thể áp dụng vào nhiều loại dữ liệu khác nhau, không chỉ là dữ liệu về phim. Điều này mang lại tính linh hoạt cao cho kỹ thuật này và làm cho nó trở thành một công cụ hữu ích trong việc khám phá các mối quan hệ trong dữ liệu.

Tuy nhiên, cũng có một số nhược điểm:

1. **Dữ liệu giới hạn**: Dữ liệu từ tập dữ liệu MovieLens có thể bị giới hạn, không phản ánh đầy đủ sở thích của người xem. Do đó, các quy tắc liên kết có thể không phản ánh chính xác mối quan hệ giữa các thể loại phim trong thực tế.
2. **Sự hạn chế của chỉ số đánh giá**: Mặc dù các chỉ số như lift, confidence, và conviction được sử dụng để đánh giá sự hữu ích của các quy tắc, nhưng chúng cũng có thể không phản ánh đầy đủ về mối quan hệ thực sự giữa các thể loại phim. Điều này có thể dẫn đến việc hiểu sai về mối quan hệ giữa các thể loại phim.
3. **Yêu cầu về xử lý dữ liệu**: Trước khi áp dụng kỹ thuật này, dữ liệu cần phải được tiền xử lý một cách cẩn thận để loại bỏ dữ liệu không đáng tin cậy và chuẩn bị dữ liệu cho quá trình phân tích. Điều này có thể tốn thời gian và công sức.

Tóm lại, kỹ thuật khai phá luật kết hợp là một công cụ hữu ích để phân tích mối quan hệ giữa các yếu tố trong dữ liệu, nhưng cần được sử dụng cẩn thận và phản ánh đúng về dữ liệu và mục tiêu của phân tích.

## **2.4. Ý nghĩa của kết quả khai phá**

Dựa vào kết quả khai phá luật kết hợp từ tập dữ liệu MovieLens, chúng ta có thể đi vào chi tiết về những quy tắc và mối quan hệ được phát hiện giữa các thể loại phim. Dưới đây là một số kết quả có thể được thu được từ quá trình phân tích:

1. **Các cặp thể loại phim phổ biến**: Kết quả có thể bao gồm các cặp thể loại phim thường đi kèm nhau. Ví dụ, có thể phát hiện ra rằng các bộ phim thuộc thể loại "Action" thường được xem kèm với "Adventure" hoặc "Sci-Fi". Điều này chỉ ra rằng có một mối quan hệ mạnh mẽ giữa các thể loại này và người xem thường có xu hướng thích những loại phim có chứa các yếu tố hành động, phiêu lưu và khoa học viễn tưởng.
2. **Thể loại phim ít liên kết**: Từ kết quả, chúng ta cũng có thể nhận thấy các thể loại phim có ít liên kết với nhau. Ví dụ, có thể phát hiện ra rằng "Musical" và "Horror" thường không được xem cùng nhau. Điều này có thể cho thấy rằng người xem có sở thích rõ ràng và thường không kết hợp giữa các thể loại phim hoàn toàn khác nhau như nhạc kịch và kinh dị.
3. **Quy tắc có độ tin cậy và hữu ích**: Các quy tắc có thể được đánh giá dựa trên các chỉ số như confidence và lift để xác định độ tin cậy và mức độ hữu ích của chúng. Ví dụ, một quy tắc có confidence cao và lift lớn giữa hai thể loại phim cho thấy mối quan hệ mạnh mẽ và có ý nghĩa giữa chúng.
4. **Quy tắc phức tạp và đa chiều**: Ngoài các quy tắc đơn giản liên kết hai thể loại phim, cũng có thể phát hiện được các quy tắc phức tạp hơn và có nhiều chiều. Ví dụ, có thể có các quy tắc liên kết ba hoặc nhiều thể loại phim với nhau, cho thấy một mối quan hệ phức tạp giữa các thể loại phim.

Tóm lại, kết quả khai phá luật kết hợp từ tập dữ liệu MovieLens cung cấp cái nhìn sâu sắc về cách mà người xem liên kết các thể loại phim với nhau và giúp hiểu rõ hơn về sở thích của họ trong việc xem phim.

# **3. Hệ thống gợi ý**

## **3.1. MATRIX FACTORIZATION**

Đây là phương pháp phổ biến nhất trong model-based CF

* Kỹ thuật: ý tưởng là dùng để thu hẹp không gian cho ít chiều lại

A math equations with numbers and symbols

Description automatically generated with medium confidence

*M,N: số items và số users*

Chú ý rằng, x được xây dựng dựa trên thông tin mô tả của *item* và quá trình xây dựng này độc lập với quá trình đi tìm hệ số phù hợp cho mỗi *user*. Như vậy, việc xây dựng *item profile* đóng vai trò rất quan trọng và có ảnh hưởng trực tiếp lên hiệu năng của mô hình. Thêm nữa, việc xây dựng từng mô hình riêng lẻ cho mỗi *user* dẫn đến kết quả chưa thực sự tốt vì không khai thác được đặc điểm của những *users* gần giống nhau.

Bây giờ, giả sử rằng ta không cần xây dựng từ trước các *item profile* x mà vector đặc trưng cho mỗi *item* này có thể được huấn luyện đồng thời với mô hình của mỗi *user* (ở đây là 1 vector hệ số). Điều này nghĩa là, biến số trong bài toán tối ưu là cả X và W; trong đó, X là ma trận của toàn bộ *item profiles*, mỗi **hàng** tương ứng với 1 *item*, W là ma trận của toàn bộ *user models*, mỗi **cột** tương ứng với 1 *user*

Hệ thống gợi ý dựa trên phương pháp Matrix Factorization với Singular Value Decomposition (SVD) là một trong những phương pháp phổ biến, hiệu quả và được sử dụng trong ứng dụng gợi ý phim.

Hệ thống gợi ý dựa trên Matrix Factorization với SVD hoạt động như sau:

1. **Xây dựng ma trận đánh giá**: Trước tiên, dữ liệu đánh giá từ người dùng được biểu diễn dưới dạng ma trận, trong đó mỗi hàng đại diện cho một người dùng, mỗi cột đại diện cho một mục (phim), và mỗi ô chứa xếp hạng mà người dùng đã đưa ra cho mục tương ứng.
2. **Phân tích ma trận đánh giá bằng SVD**: SVD được áp dụng cho ma trận đánh giá để phân tích nó thành ba ma trận con: ma trận người dùng, ma trận tiện ích (mục), và ma trận đặc trưng ẩn. Mục tiêu của quá trình này là tìm ra các đặc trưng ẩn, ví dụ như sở thích của người dùng và thuộc tính của các mục, dựa trên các xếp hạng đã được cung cấp.
3. **Học các vector đặc trưng ẩn**: SVD học các vector đặc trưng ẩn cho người dùng và các mục trong không gian đặc trưng thấp. Các vector đặc trưng này biểu diễn sở thích của người dùng và thuộc tính của các mục theo cách mà không gian chiều dữ liệu thấp hơn.
4. **Dự đoán xếp hạng**: Sau khi đã học các vector đặc trưng, hệ thống có thể dự đoán xếp hạng cho các cặp người dùng-mục chưa được đánh giá. Điều này được thực hiện bằng cách tính toán tích vô hướng của vector đặc trưng của người dùng và của mục, sau đó áp dụng một hàm kích hoạt (ví dụ: sigmoid, ReLU) để đưa ra dự đoán cuối cùng.
5. **Gợi ý các mục cho người dùng**: Cuối cùng, các mục được gợi ý cho người dùng dựa trên các dự đoán xếp hạng đã tính toán. Các mục được sắp xếp theo xếp hạng dự đoán từ cao đến thấp, và sau đó có thể được hiển thị cho người dùng để họ có thể chọn xem.

Tóm lại, Matrix Factorization với SVD là một phương pháp mạnh mẽ để xây dựng các hệ thống gợi ý, bằng cách học các đặc trưng ẩn từ dữ liệu đánh giá và sử dụng chúng để dự đoán xếp hạng cho các cặp người dùng-mục.

## **3.2 Ưu nhược điểm của hệ thống gợi ý đối với tập dữ liệu**

Dựa vào cách hoạt động của hệ thống gợi ý dựa trên Matrix Factorization với SVD và tập dữ liệu được cung cấp, ta có thể đánh giá các ưu nhược điểm như sau:

**Ưu điểm:**

1. **Hiệu suất cao:** Phương pháp Matrix Factorization với SVD thường cho hiệu suất tốt trên các tập dữ liệu lớn, đặc biệt là khi có một số lượng lớn người dùng và mục.
2. **Có thể xử lý mất dữ liệu:** Matrix Factorization cho phép xử lý dữ liệu thưa (sparse data), nghĩa là nó có thể đối phó tốt với các tập dữ liệu mà không phải tất cả các người dùng đều đánh giá tất cả các mục.
3. **Tính tổng quát:** Hệ thống gợi ý dựa trên Matrix Factorization không phụ thuộc vào thông tin mô tả về người dùng hoặc các mục. Thay vào đó, nó dựa trên thông tin về cách người dùng đánh giá các mục, làm cho nó có tính tổng quát cao và không bị giới hạn bởi các thuộc tính cụ thể.

**Nhược điểm:**

1. **Cần nhiều dữ liệu:** Mặc dù SVD có thể hoạt động tốt trên các tập dữ liệu lớn, nhưng đôi khi nó cần một lượng lớn dữ liệu đánh giá để học các đặc trưng ẩn một cách chính xác.
2. **Khó khăn với dữ liệu thưa:** Mặc dù SVD có thể xử lý dữ liệu thưa, nhưng nếu có quá ít dữ liệu, có thể gây ra hiện tượng "cold start" cho các mục mới hoặc người dùng mới, nghĩa là hệ thống gợi ý sẽ gặp khó khăn trong việc dự đoán cho những người dùng hoặc mục chưa có đủ dữ liệu đánh giá.
3. **Không xử lý được các biến cố đặc biệt:** Matrix Factorization không thể xử lý các biến cố đặc biệt như sự thay đổi nhanh chóng trong sở thích của người dùng hoặc xu hướng mới trong dữ liệu. Điều này có thể làm cho hệ thống gợi ý trở nên lỗi thời nhanh chóng.

Tóm lại, mặc dù Matrix Factorization với SVD có nhiều ưu điểm về hiệu suất và tính tổng quát, nhưng cũng có nhược điểm liên quan đến lượng dữ liệu cần thiết và khả năng xử lý dữ liệu thưa.

## **3.3. Ý nghĩa của kết quả khai phá**

Kết quả khai phá dữ liệu từ mô hình gợi ý dựa trên Matrix Factorization với SVD mang lại một số ý nghĩa quan trọng:

1. **Dự đoán xếp hạng cho các bộ phim chưa được xem:** Một trong những ứng dụng chính của hệ thống gợi ý là dự đoán xếp hạng cho các mục mà người dùng chưa từng tương tác trước đó. Kết quả từ mô hình SVD giúp dự đoán xếp hạng cho những bộ phim mà người dùng có thể quan tâm nhưng chưa biết đến.
2. **Cung cấp gợi ý cá nhân hóa:** Bằng cách sử dụng lịch sử đánh giá của người dùng cũng như thông tin về các bộ phim, mô hình có thể tạo ra các gợi ý cá nhân hóa phù hợp với sở thích và quan tâm của từng người dùng cụ thể.
3. **Tối ưu hóa trải nghiệm người dùng:** Kết quả từ mô hình gợi ý có thể được sử dụng để cải thiện trải nghiệm người dùng trên các nền tảng phim trực tuyến. Bằng cách cung cấp các gợi ý chính xác và phù hợp, nền tảng có thể giúp người dùng khám phá nhiều nội dung mới và tăng cường sự hài lòng của họ.
4. **Tiết kiệm thời gian và nỗ lực của người dùng:** Thay vì phải tìm kiếm qua hàng nghìn bộ phim để tìm ra những bộ phim phù hợp, hệ thống gợi ý có thể giảm bớt gánh nặng này bằng cách cung cấp các gợi ý được tùy chỉnh dựa trên sở thích của người dùng.

Tóm lại, kết quả khai phá từ mô hình gợi ý dựa trên Matrix Factorization với SVD không chỉ giúp cải thiện trải nghiệm người dùng mà còn có thể tạo ra giá trị kinh doanh bằng cách tăng cường sự tương tác và hài lòng của người dùng trên các nền tảng phim trực tuyến

# **CHƯƠNG II: K NEAREST NEIGBBORS**

1. **Tổng quan về hệ thống** 
   1. **Khái niệm**

Hệ thống đề xuất (Recommender Systems) là một công cụ mạnh mẽ trong việc hỗ trợ người dùng tìm kiếm và lựa chọn các sản phẩm hoặc dịch vụ phù hợp với sở thích cá nhân của họ. Các hệ thống này sử dụng các thuật toán và kỹ thuật học máy để phân tích dữ liệu người dùng và gợi ý các mục mà họ có thể quan tâm. Hệ thống đề xuất được ứng dụng rộng rãi trong nhiều lĩnh vực, từ thương mại điện tử, giải trí, đến các dịch vụ trực tuyến và mạng xã hội.

Một trong những ví dụ điển hình của hệ thống đề xuất là Netflix. Netflix sử dụng hệ thống đề xuất để gợi ý các bộ phim và chương trình truyền hình dựa trên lịch sử xem của người dùng. Hệ thống này phân tích dữ liệu về các phim mà người dùng đã xem, đánh giá và các thông tin khác để đưa ra các gợi ý phù hợp. Kết quả là người dùng có thể dễ dàng tìm thấy các nội dung mà họ yêu thích, từ đó tăng cường sự hài lòng và gắn kết với nền tảng.

Các hệ thống đề xuất cũng được sử dụng rộng rãi trong thương mại điện tử. Amazon, chẳng hạn, sử dụng hệ thống đề xuất để gợi ý các sản phẩm cho người dùng dựa trên lịch sử mua sắm, các sản phẩm đã xem và các đánh giá của họ. Hệ thống này giúp người dùng tìm thấy các sản phẩm phù hợp nhanh chóng và hiệu quả, đồng thời giúp Amazon tối ưu hóa doanh số bán hàng và tăng cường sự hài lòng của khách hàng.

Ngoài ra, hệ thống đề xuất còn được ứng dụng trong các mạng xã hội như Facebook và Twitter, nơi chúng gợi ý các kết nối bạn bè, các bài viết và nội dung mà người dùng có thể quan tâm. Các hệ thống này sử dụng dữ liệu về hành vi và sở thích của người dùng để tạo ra các gợi ý cá nhân hóa, giúp người dùng kết nối với những người và nội dung mà họ thực sự quan tâm.

* 1. **Các loại hệ thống đề xuất**
* Hệ thống dựa trên nội dung (Content-based): Hệ thống dựa trên nội dung gợi ý các mục dựa trên các đặc trưng của mục đó và sở thích của người dùng. Các đặc trưng này có thể được trích xuất từ nội dung của các mục hoặc từ thông tin về người dùng. Một trong những lợi thế lớn của hệ thống dựa trên nội dung là khả năng gợi ý các mục mới mà chưa có người dùng nào khác tương tác. Điều này đặc biệt hữu ích trong các trường hợp khi không có đủ dữ liệu về người dùng hoặc mục đó, hay còn gọi là vấn đề "người dùng mới" và "mục mới". Tuy nhiên, hệ thống này cũng có những hạn chế, như khả năng gợi ý bị giới hạn bởi những gì người dùng đã thích trong quá khứ, và khó khăn trong việc xác định các đặc trưng phù hợp nếu thông tin về mục không đầy đủ.
* Hệ thống lọc cộng tác (Collaborative Filtering): Hệ thống lọc cộng tác gợi ý các mục dựa trên sự tương đồng giữa người dùng hoặc các mục. Hệ thống này không yêu cầu thông tin chi tiết về nội dung của các mục mà dựa vào hành vi của người dùng. Có hai loại chính của hệ thống lọc cộng tác: dựa trên người dùng (user-based) và dựa trên mục (item-based). Hệ thống lọc cộng tác dựa trên người dùng tìm các người dùng có sở thích giống nhau và gợi ý các mục mà họ đã thích. Lợi thế của hệ thống lọc cộng tác là khả năng gợi ý các mục đa dạng và khám phá các mục mà người dùng có thể chưa biết tới. Tuy nhiên, hệ thống này cũng đối mặt với vấn đề "người dùng mới" hoặc "mục mới" khi không có đủ dữ liệu về người dùng hoặc mục đó.
* Hệ thống lai (Hybrid): Hệ thống lai kết hợp các phương pháp khác nhau để cải thiện độ chính xác và hiệu quả của gợi ý. Ví dụ, hệ thống lai có thể kết hợp giữa hệ thống dựa trên nội dung và hệ thống lọc cộng tác để tận dụng các ưu điểm của cả hai phương pháp. Bằng cách kết hợp các phương pháp, hệ thống lai có thể cung cấp các gợi ý chính xác hơn và phong phú hơn. Hệ thống lai có thể được triển khai theo nhiều cách khác nhau, bao gồm kết hợp các gợi ý từ các hệ thống khác nhau, sử dụng một phương pháp làm đầu vào cho phương pháp khác, hoặc xây dựng một mô hình duy nhất kết hợp các đặc trưng của các phương pháp khác nhau. Mặc dù hệ thống lai phức tạp hơn trong triển khai và bảo trì, nhưng chúng thường mang lại kết quả tốt hơn và đáng tin cậy hơn.
  1. **Thuật toán K Nearest Neighbors**

K Nearest Neighbors (KNN) là một thuật toán học máy không có tham số, được sử dụng chủ yếu trong các bài toán phân loại và hồi quy. KNN hoạt động dựa trên nguyên lý đơn giản: một đối tượng được phân loại dựa trên "đa số phiếu" của những đối tượng lân cận gần nhất của nó trong không gian đặc trưng. KNN là một trong những thuật toán đơn giản nhất nhưng lại rất mạnh mẽ và hiệu quả trong nhiều ứng dụng. Thuật toán này không yêu cầu giai đoạn huấn luyện phức tạp và có thể dễ dàng hiểu và triển khai. Tuy nhiên, KNN cũng có những hạn chế, đặc biệt là khi làm việc với các tập dữ liệu lớn hoặc có độ chiều cao.

KNN hoạt động dựa trên các bước chính sau:

* Phương pháp tính khoảng cách: Khoảng cách giữa hai điểm dữ liệu thường được tính bằng khoảng cách Euclidean, nhưng cũng có thể sử dụng các phương pháp khác như khoảng cách Manhattan hoặc khoảng cách Minkowski. Việc chọn phương pháp tính khoảng cách phụ thuộc vào đặc trưng của dữ liệu và bài toán cụ thể.
* Lựa chọn số lượng hàng xóm (K): Số lượng hàng xóm (K) là một tham số quan trọng ảnh hưởng đến hiệu suất của KNN. Một giá trị K nhỏ có thể dẫn đến mô hình quá khớp (overfitting), trong khi một giá trị K lớn có thể dẫn đến mô hình dưới khớp (underfitting). Việc lựa chọn K phù hợp cần dựa vào thực nghiệm và đánh giá hiệu quả của mô hình.

Trong bài toán phân loại, KNN gán nhãn cho một điểm dữ liệu mới dựa trên nhãn của K điểm dữ liệu lân cận nhất. Nhãn của điểm dữ liệu mới sẽ là nhãn chiếm đa số trong K điểm lân cận. Trong bài toán hồi quy, giá trị của điểm dữ liệu mới sẽ là trung bình cộng của các giá trị của K điểm lân cận.

# **Thông tin về tập dữ liệu**

## **2.1. Nguồn dữ liệu**

Nhóm sử dụng Tập dữ liệu TMDB 5000 Movie được lấy từ trang web Kaggle (kaggle.com).

Đường dẫn tập dữ liệu: [TMDB 5000 Movie](https://www.kaggle.com/datasets/tmdb/tmdb-movie-metadata)

## **2.2. Mô tả chi tiết tập dữ liệu**

**tmdb\_5000\_credits.csv**

Số dòng: 4814 dòng

Số cột: 4 cột

Tập dữ liệu này chứa thông tin về dàn diễn viên và đội ngũ làm phim cho mỗi bộ phim.

|  |  |
| --- | --- |
| **Tên cột** | **Ý nghĩa** |
| movie\_id | Mã định danh duy nhất cho mỗi bộ phim |
| title | Tên của bộ phim |
| cast | Một chuỗi JSON chứa thông tin chi tiết về các diễn viên trong phim |
| crew | Một chuỗi JSON chứa thông tin chi tiết về đội ngũ làm phim |

**tmdb\_5000\_movies.csv**

Số dòng: 4804 dòng

Số cột: 20 cột

Tập dữ liệu này chứa thông tin chi tiết về các bộ phim. Nó bao gồm các cột sau:

|  |  |
| --- | --- |
| **Tên cột** | **Ý nghĩa** |
| budget | Ngân sách của bộ phim |
| genres | Một chuỗi JSON chứa các thể loại của bộ phim |
| homepage | URL của trang chủ bộ phim |
| id | Mã định danh duy nhất cho mỗi bộ phim |
| keywords | Một chuỗi JSON chứa các từ khóa liên quan đến bộ phim |
| original\_language | Ngôn ngữ gốc của bộ phim |
| original\_title | Tên gốc của bộ phim |
| overview | Tóm tắt nội dung của bộ phim |
| popularity | Điểm phổ biến của bộ phim |
| production\_companies | Một chuỗi JSON chứa các công ty sản xuất tham gia làm phim |
| production\_countries | Một chuỗi JSON chứa các quốc gia nơi bộ phim được sản xuất |
| release\_date | Ngày phát hành của bộ phim |
| revenue | Doanh thu của bộ phim |
| runtime | Thời lượng của bộ phim tính bằng phút |
| spoken\_languages | Một chuỗi JSON chứa các ngôn ngữ được sử dụng trong bộ phim |
| status | Trạng thái phát hành của bộ phim (ví dụ: Released) |
| tagline | Khẩu hiệu của bộ phim |
| title | Tên của bộ phim |
| vote\_average | Điểm đánh giá trung bình của bộ phim |
| vote\_count | Số lượt đánh giá bộ phim đã nhận được |

1. **Khám phá và tiền xử lý dữ liệu** 
   1. **Giới thiệu**

Bộ dữ liệu sử dụng trong báo cáo này bao gồm hai tệp chính: tmdb\_5000\_movies.csv và tmdb\_5000\_credits.csv. Dữ liệu này bao gồm thông tin chi tiết về các bộ phim như tên phim, thể loại, đạo diễn, diễn viên, và điểm đánh giá. Bộ dữ liệu này được lấy từ TMDB (The Movie Database), một cơ sở dữ liệu lớn và phong phú về các bộ phim và chương trình truyền hình.

Các thuộc tính chính của bộ dữ liệu bao gồm:

* Tên phim (title): Tên của bộ phim.
* Thể loại (genres): Các thể loại mà bộ phim thuộc về, được lưu trữ dưới dạng JSON.
* Đạo diễn (director): Tên của đạo diễn của bộ phim.
* Diễn viên (cast): Danh sách các diễn viên tham gia bộ phim, được lưu trữ dưới dạng JSON.
* Điểm đánh giá (rating): Điểm đánh giá trung bình của bộ phim dựa trên đánh giá của người dùng.
  1. **Tiền xử lý dữ liệu**
* Xử lý dữ liệu thiếu: Kiểm tra và xử lý các giá trị thiếu trong bộ dữ liệu. Các giá trị thiếu có thể được xử lý bằng cách loại bỏ các hàng dữ liệu không đầy đủ hoặc thay thế các giá trị thiếu bằng các giá trị trung bình hoặc giá trị mặc định.
* Chuyển đổi dữ liệu: Chuyển đổi các thuộc tính lưu trữ dưới dạng JSON sang định dạng chuỗi hoặc các đặc trưng khác có thể sử dụng trong phân tích. Ví dụ, thuộc tính genres và cast có thể được chuyển đổi từ JSON sang chuỗi.
* Làm sạch dữ liệu: Xóa các ký tự đặc biệt, khoảng trắng dư thừa, và chuẩn hóa dữ liệu để đảm bảo tính nhất quán và dễ dàng trong xử lý.
* Tính toán đặc trưng: Tạo ra các đặc trưng mới từ dữ liệu ban đầu, ví dụ như số lượng thể loại mà một bộ phim thuộc về, hay số lượng diễn viên tham gia trong một bộ phim. Các bước tiền xử lý dữ liệu này rất quan trọng để đảm bảo dữ liệu đầu vào cho hệ thống gợi ý là đầy đủ và chất lượng, từ đó cải thiện độ chính xác của dự đoán và gợi ý.

1. **Khai phá dữ liệu**
   1. **Phân tích dữ liệu**

* Số lượng phim theo thể loại: Thể loại phim thường được lưu trữ dưới dạng JSON, do đó cần chuyển đổi dữ liệu này thành các chuỗi thể loại riêng biệt trước khi tiến hành phân tích. Sau khi chuyển đổi, cần đếm số lượng phim thuộc mỗi thể loại để xác định thể loại phim phổ biến. Chúng ta sẽ đếm số lượng phim thuộc mỗi thể loại và biểu diễn chúng bằng các thống kê mô tả.
* Điểm đánh giá trung bình: Điểm đánh giá của các phim là tính toán các thống kê mô tả cho điểm đánh giá như giá trị trung bình, trung vị, độ lệch chuẩn, và phạm vi giá trị (từ điểm thấp nhất đến điểm cao nhất). Điều này giúp hiểu rõ hơn về phân phối của điểm đánh giá và xác định xem có phim nào được đánh giá rất cao hoặc rất thấp hay không.
* Các đạo diễn phổ biến: Phân tích tần suất xuất hiện của các đạo diễn trong bộ dữ liệu giúp xác định các đạo diễn có nhiều phim nhất trong cơ sở dữ liệu và có thể phát hiện ra các đạo diễn có ảnh hưởng lớn hoặc có nhiều phim được đánh giá cao.
  1. **Trực quan hóa dữ liệu bằng biểu đồ**
* Biểu đồ thanh (Bar Chart): Sử dụng biểu đồ thanh để biểu diễn độ phổ biến của các thuộc tính.
* Sử dụng Word Cloud để hiển thị các từ khoá cụ thể nhất để mô tả về bộ phim
  1. **Triển khai**

Thuật toán K Nearest Neighbors (KNN) là một trong những thuật toán đơn giản và hiệu quả cho bài toán gợi ý phim.

Mô tả chi tiết các bước triển khai KNN:

* Tính toán độ tương đồng: Tính toán độ tương đồng giữa các phim dựa trên các đặc trưng đã chọn (thể loại, đạo diễn, diễn viên). Có nhiều cách để đo độ tương đồng, nhưng phổ biến nhất là sử dụng khoảng cách Euclidean hoặc cosine similarity.
* Xác định số lượng hàng xóm (K): Số lượng hàng xóm K là một tham số quan trọng ảnh hưởng đến hiệu suất của mô hình. Giá trị K có thể được xác định thông qua quá trình thử nghiệm và đánh giá mô hình. Một giá trị K quá nhỏ có thể dẫn đến quá khớp (overfitting), trong khi giá trị K quá lớn có thể dẫn đến dưới khớp (Underfitting).
* Thực hiện dự đoán điểm và gợi ý phim: Hệ thống dựa vào điểm đánh giá của các phim hàng xóm để dự đoán điểm đánh giá cho phim mới và gợi ý các phim tương tự.

Tính toán độ tương đồng giữa các phim:

* Khoảng cách Euclidean: Khoảng cách Euclidean giữa hai điểm dữ liệu (hai phim) được tính bằng công thức:

*Trong đó p và q là các vector đặc trưng của hai phim.*

* Cosine Similarity: Cosine similarity đo lường độ tương đồng giữa hai vector bằng cách tính cosine của góc giữa chúng:

*Giá trị cosine similarity nằm trong khoảng từ -1 đến 1, trong đó 1 có nghĩa là hai vector hoàn toàn giống nhau, và -1 là hoàn toàn khác nhau.*

# **CHƯƠNG III: KẾT LUẬN**

* 1. **Kết quả đạt được**

Xây dựng được hệ thống chấm điểm và gợi ý phim sử dụng *mô hình thuật toán K Nearest Neighbors (KNN)*.

Xây dựng được mô hình dự đoán đánh giá của user về các bộ phim chưa coi và sắp xếp chúng theo thứ tự từ cao đến thấp để có thể gợi ý cho người dùng những phim phù hợp với sở thích của người dùng khi áp dụng *phương pháp khai phá luật kết hợp*.

* 1. **Hạn chế**

Trong quá trình thực hiện khai phá dữ liệu, nhóm chúng em không tránh khỏi gặp một số những khó khăn vì còn hạn chế về mặt kiến thức và thời gian. Tập dữ liệu còn nhiều dữ kiện chưa được khai thác hết. Những thực hiện của chúng em chưa đạt được đến mức nâng cao để áp dụng được tất cả các mô hình dự đoán.

* 1. **Quá trình làm việc nhóm**

Bảng phân công nhiệm vụ:

|  |  |  |  |
| --- | --- | --- | --- |
|  | Ngân | Khang | Vy |
| Tìm tập dữ liệu |  | x | x |
| + MovieLens |  | x |  |
| + TMDB 5000 Movie |  |  | x |
| Chương I: Khai phá luật kết hợp |  | x |  |
| Demo Khai phá luật kết hợp |  | x |  |
| Chương II: KNN | x |  | x |
| + Tổng quan về hệ thống | x |  |  |
| + Thông tin về tập dữ liệu |  |  | x |
| + Khám phá và tiền xử lý dữ liệu | x |  | x |
| + Khai phá dữ liệu | x |  | x |
| Demo KNN | x |  | x |
| Chương III: Kết luận |  |  | x |