**ĐẠI HỌC HUẾ**

# **KHOA KỸ THUẬT VÀ CÔNG NGHỆ**



🙠🙟🕮🙝🙢

****

**BÁO CÁO**

**ĐỒ ÁN**

**NĂM HỌC 2023-2024**

**Giảng viên hướng dẫn: Quách Hải Thọ**

**Lớp: KHDL&TTNT**

|  |
| --- |
| Số phách  *(Do hội đồng chấm thi ghi)* |

**Thừa Thiên Huế, ngày …tháng…năm.....**

**ĐẠI HỌC HUẾ**



# KHOA KỸ THUẬT VÀ CÔNG NGHỆ

🙠🙟🕮🙝🙢

****

(MẪU BÌA PHỤ)

**BÁO CÁO ĐỒ ÁN**

**NĂM HỌC 2023-2024**

**Giảng viên hướng dẫn: Quách Hải Thọ**

**Lớp: KHDL&TTNT**

**Sinh viên thực hiện: Phan Thế Minh Châu**

**MSV: 20E1020067**

*(ký tên và ghi rõ họ tên)*

|  |
| --- |
| Số phách  *(Do hội đồng chấm thi ghi)* |

**Thừa Thiên Huế, ngày …tháng…năm.....**

ĐẠI HỌC HUẾ

# KHOA KỸ THUẬT VÀ CÔNG NGHỆ

🙠🙟🕮🙝🙢

**PHIẾU ĐÁNH GIÁ ĐỒ ÁN**

**Học kỳ I, năm học 2023 - 2024**

|  |  |
| --- | --- |
| **Cán bộ chấm thi 1** | **Cán bộ chấm thi 2** |
| **Nhận xét:**                      **Điểm đánh giá của CBCT1:**  Bằng số:  Bằng chữ: | **Nhận xét:**                      **Điểm đánh giá của CBCT2:**  Bằng số:  Bằng chữ: |

Điểm kết luận:

Bằng số:

Bằng chữ:

*Thừa Thiên Huế, ngày tháng năm 2023*

|  |  |
| --- | --- |
| **Cán bộ chấm thi 1**  *(Ký và ghi rõ họ và tên)* | **Cán bộ chấm thi 2**  *(Ký và ghi rõ họ và tên)* |

**Mục lục**

[Danh mục hình vẽ 5](#_Toc155431081)

[Danh mục các từ viết tắt 6](#_Toc155431082)

[Danh mục các thuật ngữ 7](#_Toc155431083)

[1. Giới thiệu 9](#_Toc155431084)

[1.1. Tổng quan về hệ thống khuyến nghị (Recommender System) 9](#_Toc155431085)

[1.2. Các phương pháp tiếp cận trong Hệ thống gợi ý (Recommender Systems) 10](#_Toc155431086)

[1.3. Ma trận Utility Matrix và bài toán gợi ý 12](#_Toc155431087)

[2. Phương pháp gợi ý dựa theo nội dung (Content-based Filtering) 12](#_Toc155431088)

[2.1. Ý tưởng 12](#_Toc155431089)

[2.2 Xây dựng Items Profile 13](#_Toc155431090)

[2.3. Áp dụng mô hình tuyến tính 14](#_Toc155431091)

[3. Phương pháp lọc cộng tác (Collaborative Filtering - CF) 15](#_Toc155431092)

[3.1. Ý tưởng 15](#_Toc155431093)

[3.2. User – User Collaborative Filtering 16](#_Toc155431094)

[3.3. Item – item Collaborative Filtering 20](#_Toc155431095)

[4. Triển khai thuật toán bằng Python 21](#_Toc155431096)

[4.1. Mô tả tập dữ liệu 21](#_Toc155431097)

[4.2. Content-based Filtering 21](#_Toc155431098)

[4.3. Collaborative Filtering 24](#_Toc155431099)

[4.4. So sánh các phương pháp 28](#_Toc155431100)

[5. Kết luận 29](#_Toc155431101)

[5.1. Tóm tắt các kết quả của báo cáo 29](#_Toc155431102)

[5.2. Hướng phát triển. 29](#_Toc155431103)

[6. Tài liệu tham khảo 30](#_Toc155431104)

# 

# **Danh mục hình vẽ**

[Hình 1: Minh họa RS 10](#_Toc155430282)

[Hình 2: ví dụ về Utility Matrix 11](#_Toc155430283)

[Hình 3: Minh họa Content-based Filtering 11](#_Toc155430284)

[Hình 4: Biểu diễn nhị phân 12](#_Toc155430285)

[Hình 5: Minh họa lọc cộng tác 15](#_Toc155430286)

[Hình 6: Utility matrix 16](#_Toc155430287)

[Hình 7: Trung bình cộng ratings của user 16](#_Toc155430288)

[Hình 8: Normalized Utility Matrix 17](#_Toc155430289)

[Hình 9: Similarity Matrix 17](#_Toc155430290)

[Hình 10: Ma trận dự đoán rating 18](#_Toc155430291)

[Hình 11: Ratings dự đoán theo thang 5 19](#_Toc155430292)

[Hình 12: Tóm tắt các bước thực hiện iiCF 20](#_Toc155430293)

[Hình 13: Dataset MoviesLens 21](#_Toc155430294)

[Hình 14: đồ thị MRSE khi lamda thay đổi 23](#_Toc155430295)

[Hình 15: Đồ thị MRES của CF khi N neighbors thay đổi 27](#_Toc155430296)

[Hình 16: Đồ thị so sánh RMSE các phương pháp 28](#_Toc155430297)

# **Danh mục các từ viết tắt**

|  |  |  |
| --- | --- | --- |
| STT | Tên viết tắt | Tên đầy đủ |
| 1 | RS | Recommender system ( Recommendation System) |
| 2 | CB | Content-based |
| 3 | CF | Collaborative Filtering |
| 4 | NBCF | Neighborhood-based Collaborative Filtering |
| 5 | UuCF | User-user Collaborative Filtering |
| 6 | IiCF | Item-item Collaborative Filtering |
| 7 | RMES | Root Mean Squared Error |
| 8 | MLs | Tập dữ liệu MovieLens |

# **Danh mục các thuật ngữ**

|  |  |  |
| --- | --- | --- |
| STT | Thuật ngữ | Ý nghĩa |
| 1 | Artificial Intelligence | Trí tuệ nhân tạo |
| 2 | Machine Learning | Học máy |
| 3 | Recommender system(s)  Recommendation System(s) | Hệ thống gợi ý |
| 4 | Content-based Filtering | Phương pháp gợi ý dựa trên nội dung |
| 5 | Collaborative Filtering | Phương pháp gợi ý dựa trên lọc cộng tác |
| 6 | Neighborhood-based Collaborative Filtering | Phương pháp gợi ý dựa trên hàng xóm gần nhấ |
| 7 | User-user Collaborative Filtering | Phương pháp gợi ý dựa trên sự tương đồng của các users (người dùng) |
| 8 | Item-item Collaborative Filtering | Phương pháp gợi ý dựa trên sự tương đồng của các items |
| 9 | Utility Matrix | Ma trận biểu diễn các giá trị ratings của các users cho các items |
| 10 | Normalized Utility Matrix | Ma trận Utility đã được chuẩn hóa |
| 11 | Cosine Similarity | Độ đo khoảng cách giữa các vector bằng hàm cosine |
| 12 | Pearson Corelation (Pearson) | Độ đo khoảng cách giữa các vector bằng hàm pearson |
| 13 | Similarity Matrix | Ma trận khoảng cách |
| 14 | Gradient Descent | Phương pháp tối ưu hàm mất má |
| 15 | Regression | Bài toán hồi quy |
| 16 | Classification | Bài toán phân lớp |
| 17 | Linear Regression | Mô hình hồi quy tuyến tính |
| 18 | Root Mean Squared Error | Sai số toàn phương trung bình |
| 19 | Precision | Độ chính xác |

# **1. Giới thiệu**

## 1.1. Tổng quan về hệ thống khuyến nghị (Recommender System)

Hệ thống khuyến nghị (Recommender System) là một công nghệ hiện đại, tận dụng sức mạnh của AI và ML nhằm kiến tạo những trải nghiệm được cá nhân hóa cho từng người dùng. Hoạt động cốt lõi của RS là phân tích hành vi, sở thích, đặc điểm và mối tương quan trong dữ liệu để dự đoán khả năng quan tâm của người dùng đối với các sản phẩm, dịch vụ hoặc nội dung mới.

Trong cuộc sống hàng ngày, chúng ta có thể dễ dàng bắt gặp những ứng dụng của Hệ thống gợi ý, chẳng hạn như:

* YouTube tự động chuyển các clip liên quan đến clip người dùng đang xem. Hệ thống gợi ý của YouTube sẽ phân tích các thông tin về clip đang xem, chẳng hạn như chủ đề, thể loại, độ dài,... để đưa ra các gợi ý về những clip tương tự mà họ có thể thích.
* Khi mua một món hàng trên Amazon, hệ thống sẽ tự động gợi ý “Frequently bought together”, hoặc nó biết khách hàng có thể thích món hàng nào dựa trên lịch sử mua hàng của họ. Hệ thống gợi ý của Amazon sẽ phân tích các thông tin về món hàng khách hàng vừa mua, chẳng hạn như nhãn hiệu, nhà sản xuất,... để đưa ra các gợi ý về những món hàng khác thường được mua cùng.
* Facebook hiển thị quảng cáo những sản phẩm có liên quan đến từ khoá người dùng vừa tìm kiếm. Hệ thống gợi ý của Facebook sẽ phân tích các thông tin về từ khoá vừa tìm kiếm, chẳng hạn như chủ đề, sản phẩm,... để đưa ra các gợi ý về những quảng cáo phù hợp.
* Facebook gợi ý kết bạn. Hệ thống gợi ý của Facebook sẽ phân tích các thông tin về bạn bè của người dùng, chẳng hạn như sở thích, mối quan hệ,... để đưa ra các gợi ý về những người bạn mới có thể phù hợp.
* Netflix tự động gợi ý phim cho người dùng. Hệ thống gợi ý của Netflix sẽ phân tích các thông tin về lịch sử xem phim của người dùng, chẳng hạn như thể loại, diễn viên, đạo diễn,... để đưa ra các gợi ý về những bộ phim mới có thể họ sẽ thích.

Hệ thống gợi ý có thể mang lại nhiều lợi ích cho cả người dùng và doanh nghiệp. Đối với người dùng, hệ thống gợi ý giúp họ dễ dàng tìm thấy những sản phẩm, nội dung phù hợp với sở thích và nhu cầu của mình, từ đó tiết kiệm thời gian và công sức tìm kiếm. Đối với doanh nghiệp, hệ thống gợi ý giúp họ tiếp cận đúng đối tượng khách hàng tiềm năng, nâng cao hiệu quả của các chiến dịch marketing.

**Hiện nay, Hệ thống gợi ý đang được ứng dụng rộng rãi trong nhiều lĩnh vực, bao gồm thương mại điện tử, giải trí, truyền thông,... và ngày càng trở nên phổ biến hơn trong cuộc sống của chúng ta.**

## 1.2. Các phương pháp tiếp cận trong Hệ thống gợi ý (Recommender Systems)

Các phương pháp tiếp cận chính trong Hệ thống gợi ý (Recommender Systems) có thể được chia thành hai nhóm chính:

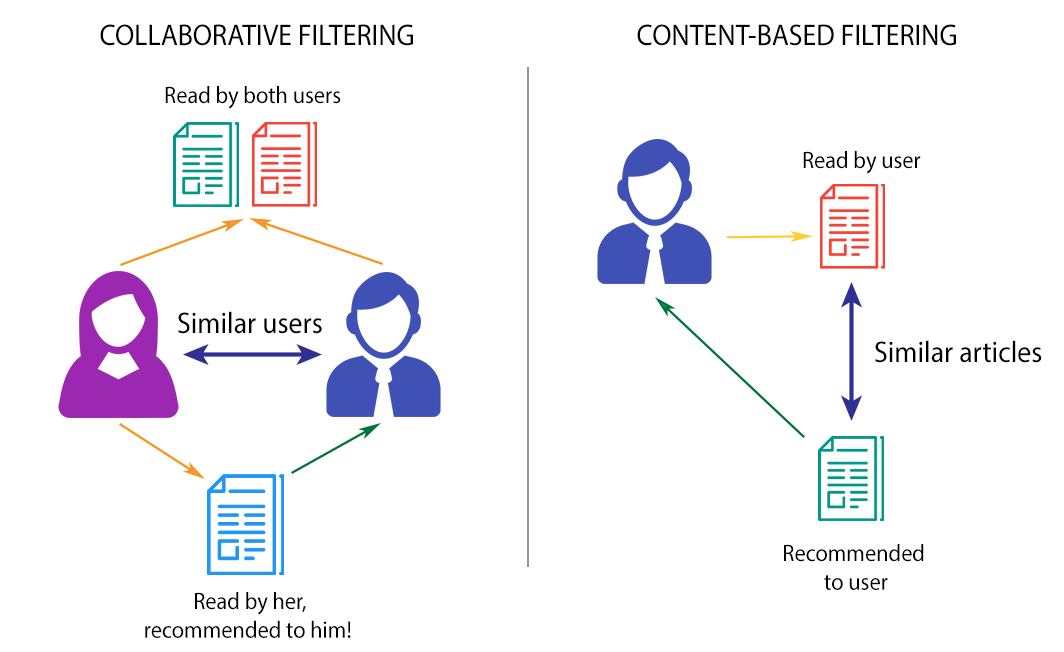
* Gợi ý dựa trên nội dung (Content-based recommendation): Phương pháp này dựa trên việc phân tích nội dung của các sản phẩm, nội dung để tìm ra những sản phẩm, nội dung có liên quan hoặc tương tự với nhau.
* Gợi ý dựa trên cộng tác (Collaborative filtering): Phương pháp này dựa trên việc phân tích lịch sử tương tác của người dùng với các sản phẩm, nội dung để tìm ra những người dùng có sở thích tương tự nhau.
* Kết hợp cả 2 phương pháp: Ngoài ra chúng ta cũng có thể sử dụng kết hợp cả 2 phương pháp trên để tạo thành một thuật toán kết hợp. Ưu điểm của phương pháp này đó là vừa tận dụng được các thông tin từ phía sản phẩm và các thông tin về hành vi mua sắm của người dùng.

**Content-based recommendation**: Gợi ý dựa trên nội dung là một phương pháp đơn giản và dễ thực hiện. Phương pháp này dựa trên việc phân tích các thông tin về sản phẩm, nội dung, chẳng hạn như chủ đề, thể loại, giá cả,... để tìm ra những sản phẩm, nội dung có liên quan hoặc tương tự với nhau.

Ví dụ, một hệ thống gợi ý dựa trên nội dung có thể sử dụng các thông tin về thể loại, diễn viên, đạo diễn,... để gợi ý những bộ phim tương tự cho người dùng.

**Collaborative filtering**: Gợi ý dựa trên cộng tác là một phương pháp phức tạp hơn nhưng có thể mang lại hiệu quả cao hơn. Phương pháp này dựa trên việc phân tích lịch sử tương tác của người dùng với các sản phẩm, nội dung để tìm ra những người dùng có sở thích tương tự nhau.

Ví dụ, một hệ thống gợi ý dựa trên cộng tác có thể sử dụng dữ liệu về lịch sử mua hàng của người dùng để gợi ý những sản phẩm mà những người dùng có sở thích tương tự đã mua.



Hình 1: Minh họa RS

Ngoài hai phương pháp chính trên, còn có một số phương pháp tiếp cận khác trong Hệ thống gợi ý, chẳng hạn như:

* Gợi ý dựa trên kết hợp (Hybrid recommendation): Phương pháp này kết hợp hai hoặc nhiều phương pháp tiếp cận khác nhau để mang lại hiệu quả tốt hơn.
* Gợi ý dựa trên tri thức (Knowledge-based recommendation): Phương pháp này sử dụng các kiến thức về thế giới thực, chẳng hạn như mối quan hệ giữa các sản phẩm, nội dung, để đưa ra các gợi ý.
* Gợi ý dựa trên phản hồi (Feedback-based recommendation): Phương pháp này sử dụng các phản hồi của người dùng về các sản phẩm, nội dung để cải thiện hiệu quả của hệ thống gợi ý.

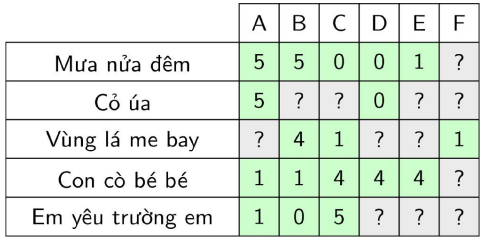
Lựa chọn phương pháp tiếp cận phù hợp phụ thuộc vào nhiều yếu tố, bao gồm:

* Tính chất của dữ liệu
* Yêu cầu về hiệu quả
* Khả năng tính toán

## 1.3. Ma trận Utility Matrix và bài toán gợi ý

Trong các hệ gợi ý, có 2 thực thể chính là users và items. Mỗi user sẽ có mức độ quan tâm (degree of preference) tới mỗi item là khác nhau. Mức độ quan tâm này, nếu đã biết trước, sẽ được gán cho một giá trị ứng với mỗi cặp user-item. Giả sử rằng mức độ quan tâm được đo bằng giá trị user rate cho item, tạm gọi giá trị này là rating. Khi đó, tập hợp tất cả các ratings, bao gồm cả những giá trị chưa biết cần được dự đoán, sẽ tạo nên một ma trận gọi là ma trận utility matrix:

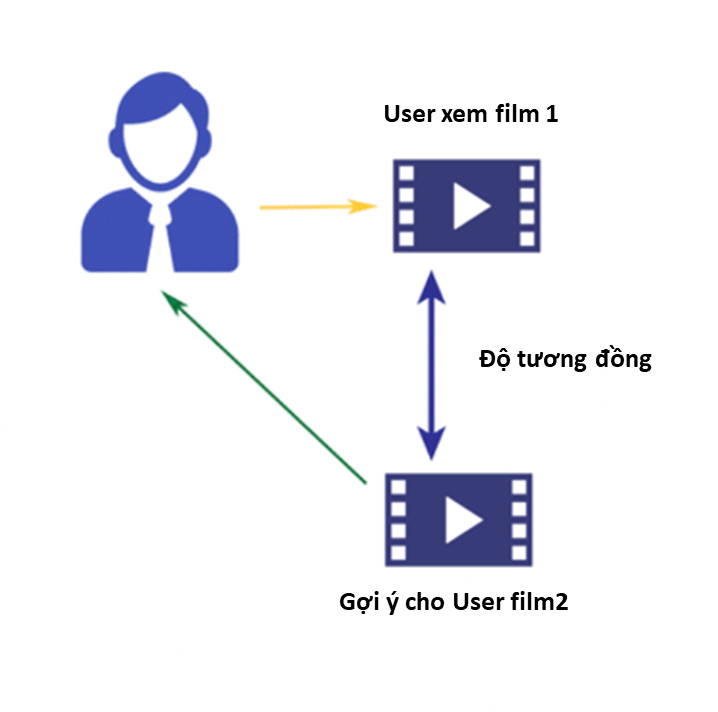
VD:



Hình 2: ví dụ về Utility Matrix

# **2. Phương pháp gợi ý dựa theo nội dung (Content-based Filtering)**

## 2.1. Ý tưởng



Hình 3: Minh họa Content-based Filtering

Ý tưởng của thuật toán này là, từ thông tin mô tả của item, biểu diễn item dưới dạng vec-tơ thuộc tính. Sau đó dùng các vec-tơ này để học mô hình của mỗi user, là ma trận trọng số của user với mỗi item.

Như vậy, thuật toán content-based gồm 2 bước:

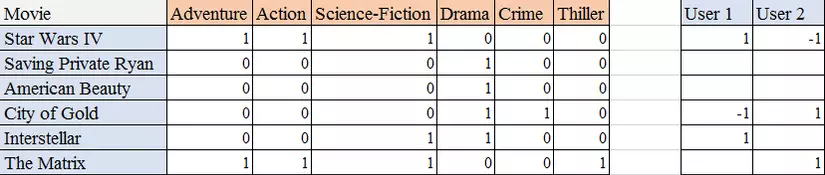
* Bước 1: Biểu diễn items dưới dạng vec-tơ thuộc tính - item profile
* Bước 2: Học mô hình của mỗi user

## 2.2 Xây dựng Items Profile

Trong hệ thống gợi ý content-based, mỗi item được biểu diễn dưới dạng vectơ đặc trưng (feature vector) n chiều. Vectơ này được xây dựng dựa trên nội dung của item, chẳng hạn như thể loại, đạo diễn, năm sản xuất, v.v. Trong trường hợp đơn giản, vectơ đặc trưng có thể được trích xuất trực tiếp từ item. Sau đó, các item có độ tương đồng cao với nhau sẽ được gợi ý cho người dùng.

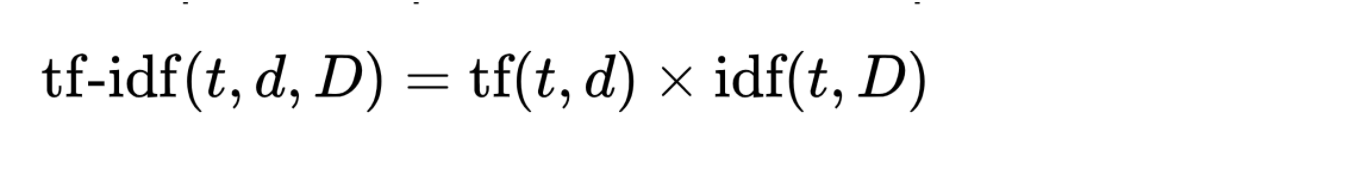
Phương pháp xây dựng feature vector là:

* Sử dụng biểu diễn nhị phân



Hình 4: Biểu diễn nhị phân

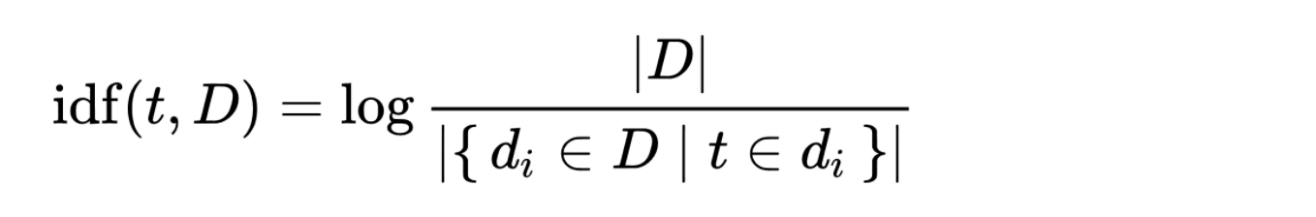
* Sử dụng TF-IDF (term frequency – inverse document frequency): tính toán mức độ quan trọng của mỗi từ dưạ trên tần suất xuất hiện và mối liên hệ với các từ khác trong toàn bộ đoạn văn bản.



**

Trong đó:

* tf(t, d): Tần suất xuất hiện của từ t trong văn bản d
* : Số lần xuất hiện của từ t trong văn bản d
* : Số từ trong văn bản d

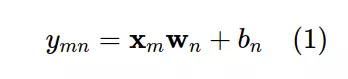


Trong đó:

* idf(t, D): giá trị idf của từ t trong tập văn bản
* |D|: Tổng số văn bản trong tập D
* |{d ∈ D: t ∈ d}|: thể hiện số văn bản trong tập D có chứa từ t.

## 2.3. Áp dụng mô hình tuyến tính

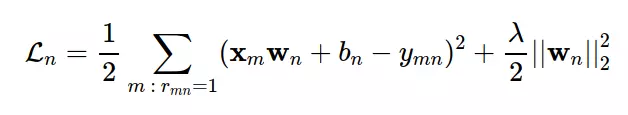
Giả sử rằng ta có thể tìm được một mô hình có thể tính được mức độ quan tâm của mỗi user với mỗi item bằng một hàm tuyến tính:



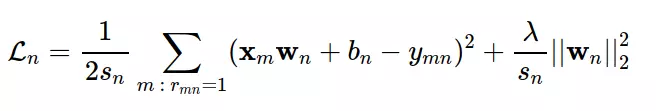
Trong đó, là vector đặc trưng của item m.

Mục tiêu của chúng ta sẽ là học ra mô hình của user, tức là tìm ra và .

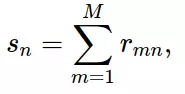
Xét một user thứ n bất kỳ, nếu ta coi training set là tập hợp các thành phần đã được điền của , ta có thể xây dựng hàm mất mát tương tự như sau:



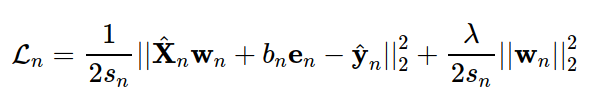
Trong đó, thành phần thứ hai là regularization term và λ là một tham số dương. Chú ý rằng regularization thường không được áp dụng lên . Trong thực hành, trung bình cộng của lỗi thường được dùng, và mất mát nên được viết lại thành:



Trong đó là số lượng các items mà user thứ n đã rated. Nói cách khác, là tổng các phần tử trên cột thứ n của ma trận rated or not R:



Vì hàm mục tiêu chỉ phụ thuộc vào các items đã được rated, ta có thể rút gọn nó bằng cách đặt là sub vector của y được xây dựng bằng cách trích các thành phần khác dấu? ở cột thứ n, tức đã được rated bởi user thứ n trong ma trận Y. Đồng thời, đặt là sub matrix của ma trận feature **X**, được tạo bằng cách trích các hàng tương ứng với các items đã được rated bởi user thứ n. Khi đó, biểu thức hàm mất mát của mô hình cho user thứ n được viết gọn thành công thức (\*):



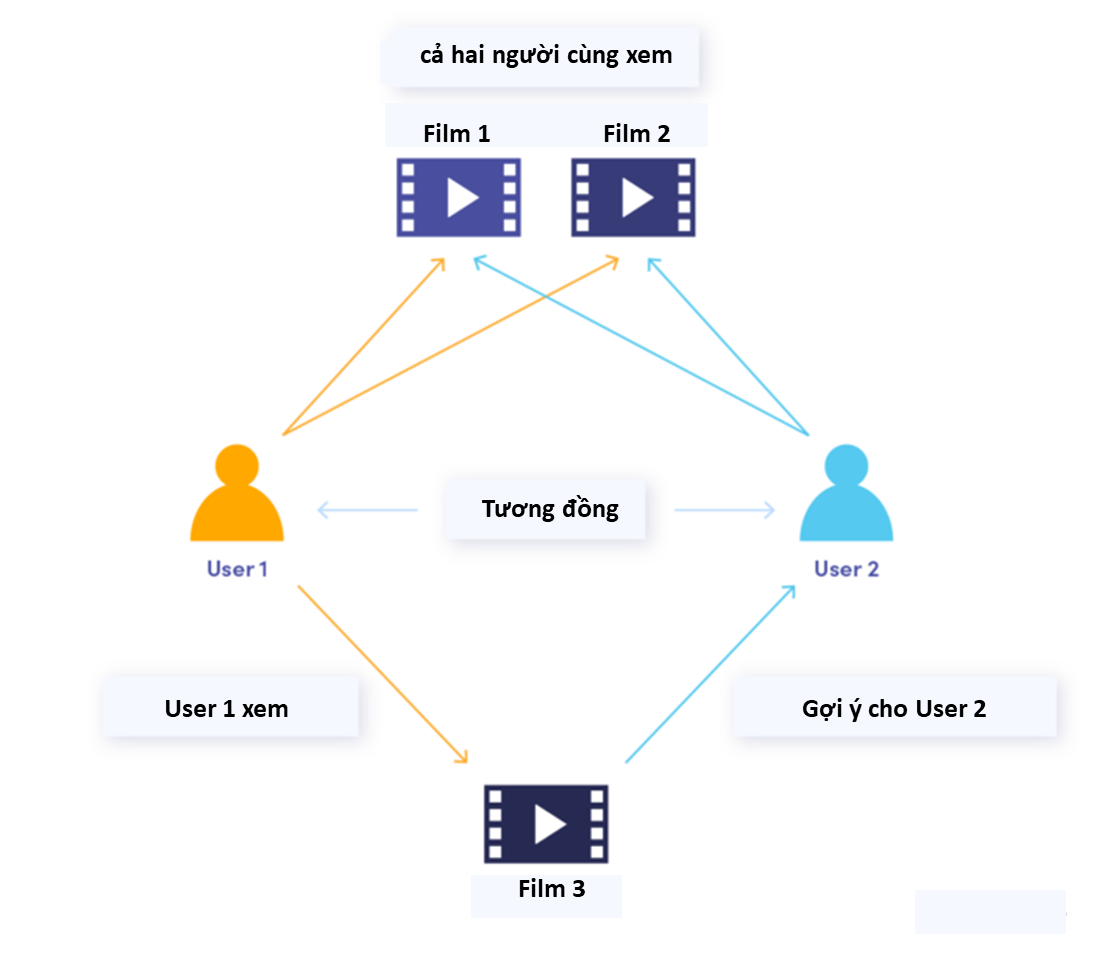
trong đó, là vector cột chứa phần tử 1.

Cuối cùng, chúng ta có thể sử dụng Stochastic Gradient Descent (SGD), hoặc Mini-batch GD để tìm ra và .

# **3. Phương pháp lọc cộng tác (Collaborative Filtering - CF)**

## 3.1. Ý tưởng

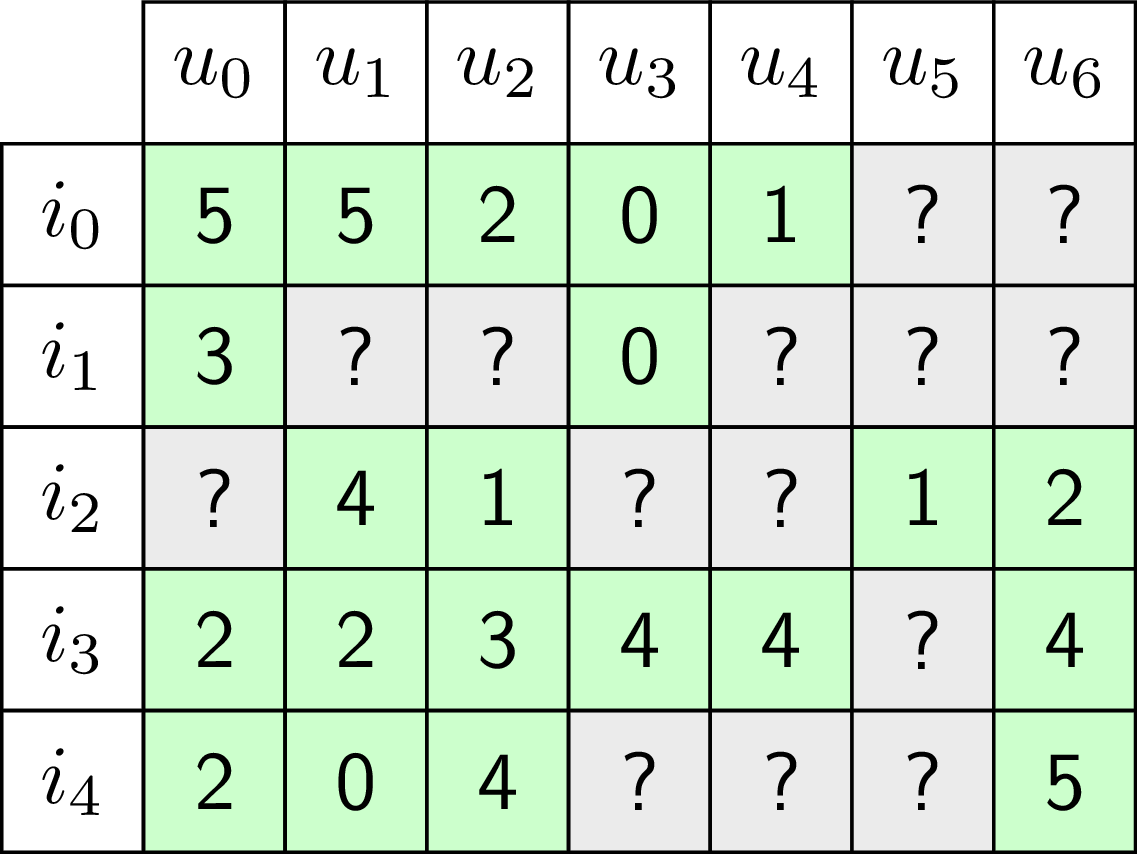
Lọc cộng tác gợi ý sản phẩm tới người dùng dựa trên lịch sử tương tác sản phẩm của chính người dùng đó kết hợp quan điểm của những người dùng khác. Đây là kỹ thuật mạnh và đã được áp dụng khá thành công trong các hệ thống thương mại lớn. Lọc cộng tác thu thập phản hồi từ nhiều người dùng đối với sản phẩm, lưu trữ phản hồi của người dùng dưới dạng ma trận người dùng (users) - sản phẩm (items). Mỗi dòng là một vectơ chứa giá trị phản hồi của người dùng đối với các sản phẩm. Sau đó, tính độ tương tự giữa các users hoặc giữa các items trong hệ thống, tìm ra mối tương quan và đưa ra gợi ý phù hợp. Độ tương tự giữa các user hoặc giữa các item được tính theo công thức Cosin hoặc Pearson.



Hình 5: Minh họa lọc cộng tác

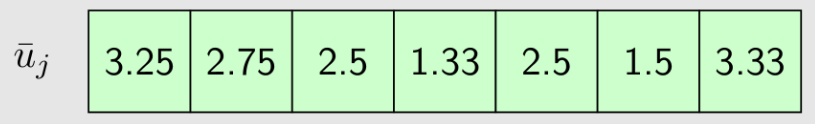
## 3.2. User – User Collaborative Filtering

Dữ liệu duy nhất chúng ta sử dụng cho phương pháp này là ma trận Utility:



Hình 6: Utility matrix

Tiếp theo ta sẽ tính giá trị trung bình cộng ratings của mỗi user.

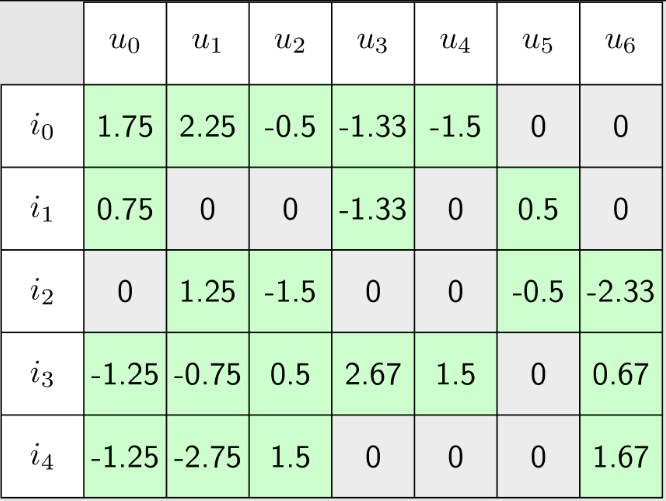


Hình 7: Trung bình cộng ratings của user

Như trong hình vẽ, ta có một ví dụ về ratings trung bình của mỗi users. Tuy nhiên, thay vì trực tiếp sử dụng các giá trị này thay cho các dấu ‘?’ của mỗi user. Ta sẽ trừ ratings của mỗi user cho giá trị trung bình ratings tương ứng của user đó và thay dấu ‘?’ bằng giá trị 0. Mục đích của cách xử lý này là:

* Phân loại ratings thành 2 loại: giá trị âm (user không thích item) và giá trị dương (user thích item). Các giá trị bằng 0, tương ứng với những item chưa được đánh giá.
* Số chiều của Utility matrix thường rất lớn, trong khi lượng ratings biết trước thường rất nhỏ so với kích thước của toàn bộ ma trận. Nếu thay dấu ‘?’ bằng ‘0’ thì chúng ta có thể sử dụng sparce matrix, tức ma trận chỉ lưu các giá trị khác 0 và vị trí của giá trị đó. Như vậy, việc lưu trữ sẽ tối ưu hơn.

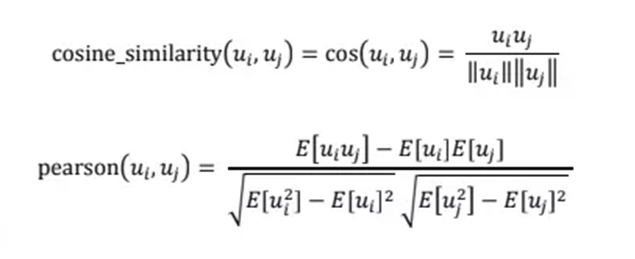
Ma trận sau khi chuẩn hóa được gọi là Normalized Utility Matrix:



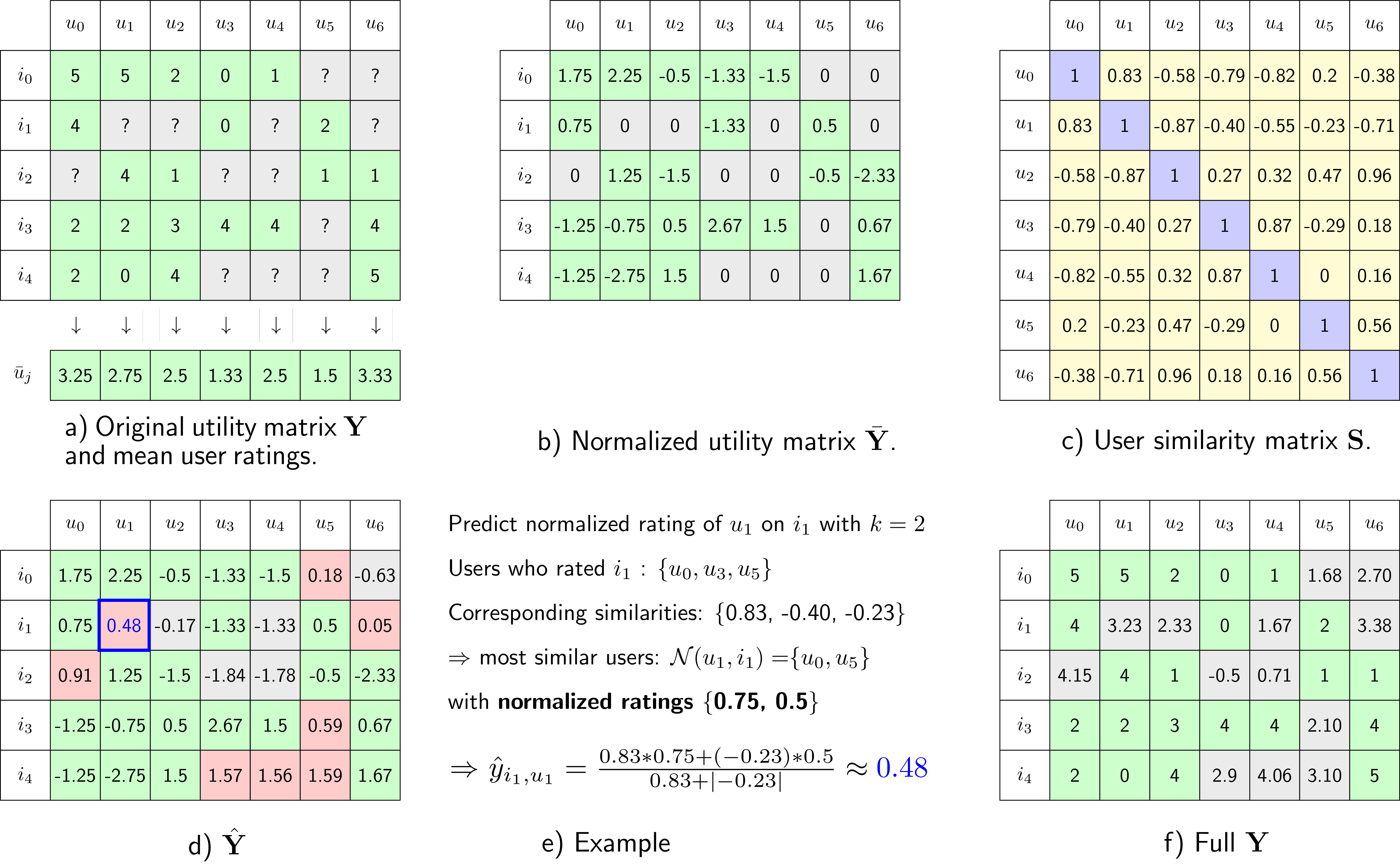
Hình 8: Normalized Utility Matrix

Sau khi chuẩn hóa ma trận Utility, chúng ta cần tính toán độ tương đồng giữa các users.

Một số similarity function thường được sử dụng là Cosine Similarity và Pearson Corelation.



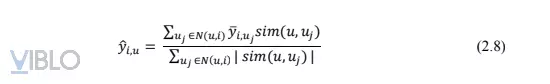
Tính độ tương đồng ta có được ma trận Similarity Matrix



Hình 9: Similarity Matrix

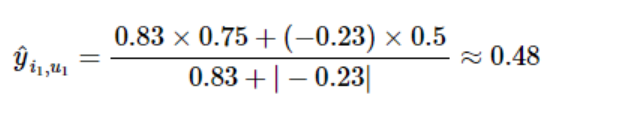
Cuối cùng sẽ dự đoán ratings của một user với mỗi item dựa trên k users gần nhất (neighbor users), tương tự như phương pháp K-nearest neighbors (KNN).

Công thức phổ biến được sử dụng để dự đoán rating của u cho i là:

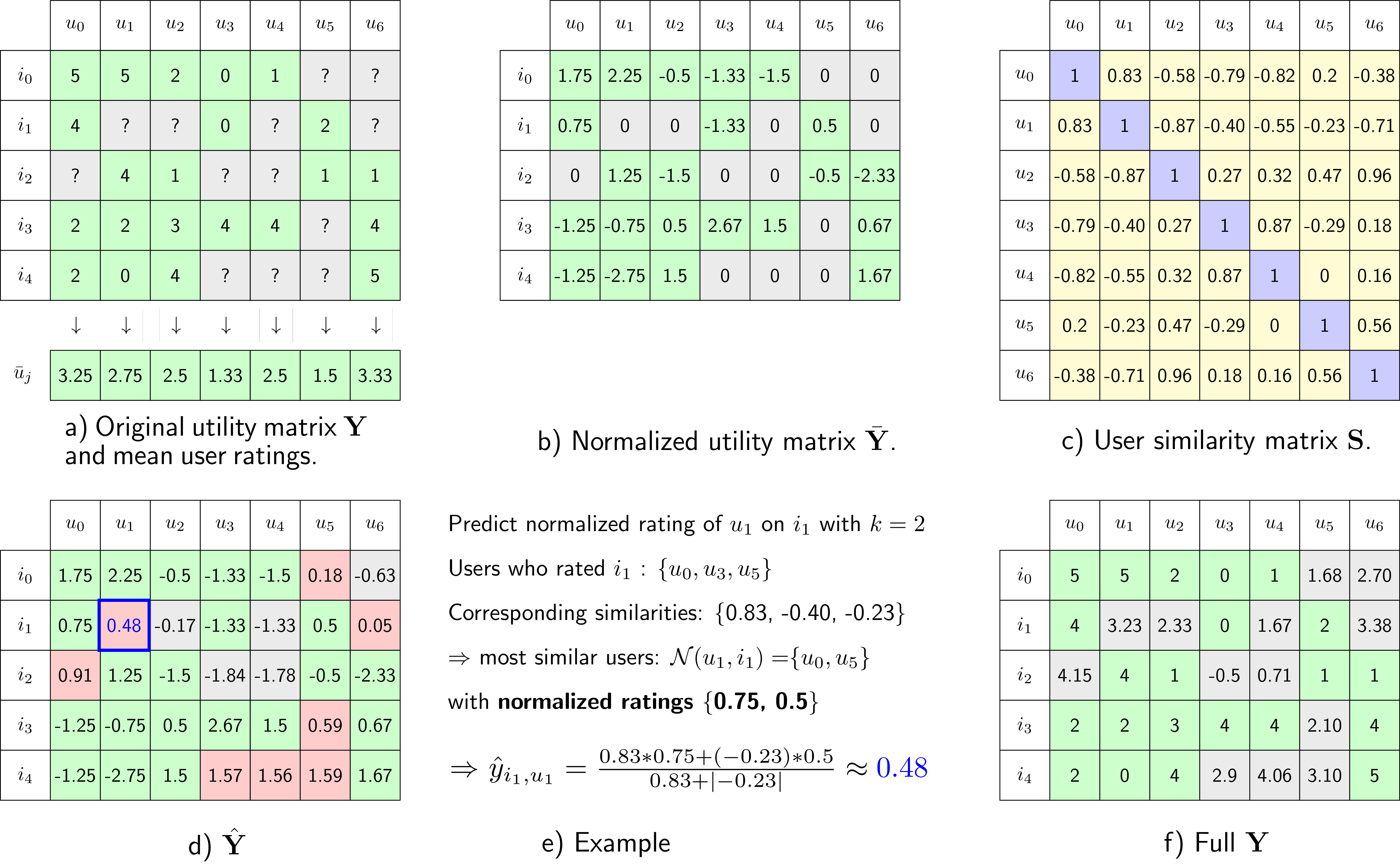


Trong đó N(u,i) là tập hợp k users trong neighborhood ( tức có similarity cao nhất ) của u mà đã rated i. Ví dụ về việc tính normalized rating của u1 cho i1 ở ví dụ trên được cho với số k = 2 các bước thực hiện là:

* Xác định các users đã rated i1, đó là u0 , u3 , u5
* Xác định similarities của u1 với các users này ta nhận được 0.83 , -0.40 , -0.23 . Hai ( k = 2 ) giá trị lớn - nhất là 0.83 và -0.23 tương ứng với u0 và u5.
* Xác định các normalized ratings của u0, u5 cho i1, ta thu được hai giá trị lần lượt là 0.75 và 0.5
* Dự đoán kết quả

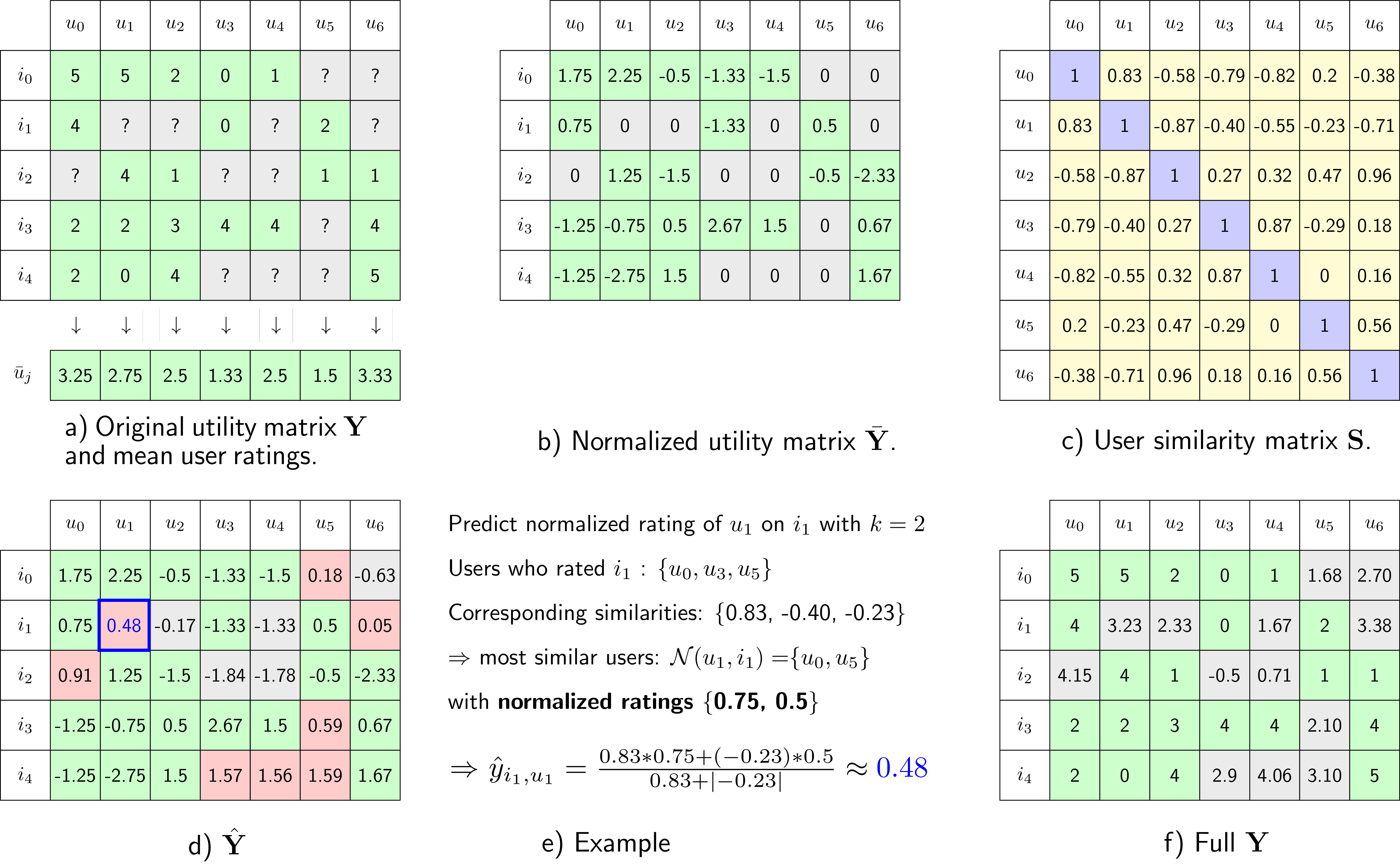


Tương tự đối với các user khác, ta thu được ma trận:



Hình 10: Ma trận dự đoán rating

Cuối cùng ta quy đổi các giá trị ratings đã chuẩn hoá về thang 5 có thể được thực hiện bằng cách cộng các cột của ma trận trên với giá trị rating trung bình đã tính ở trên, ta thu được kết quả dự đoán:



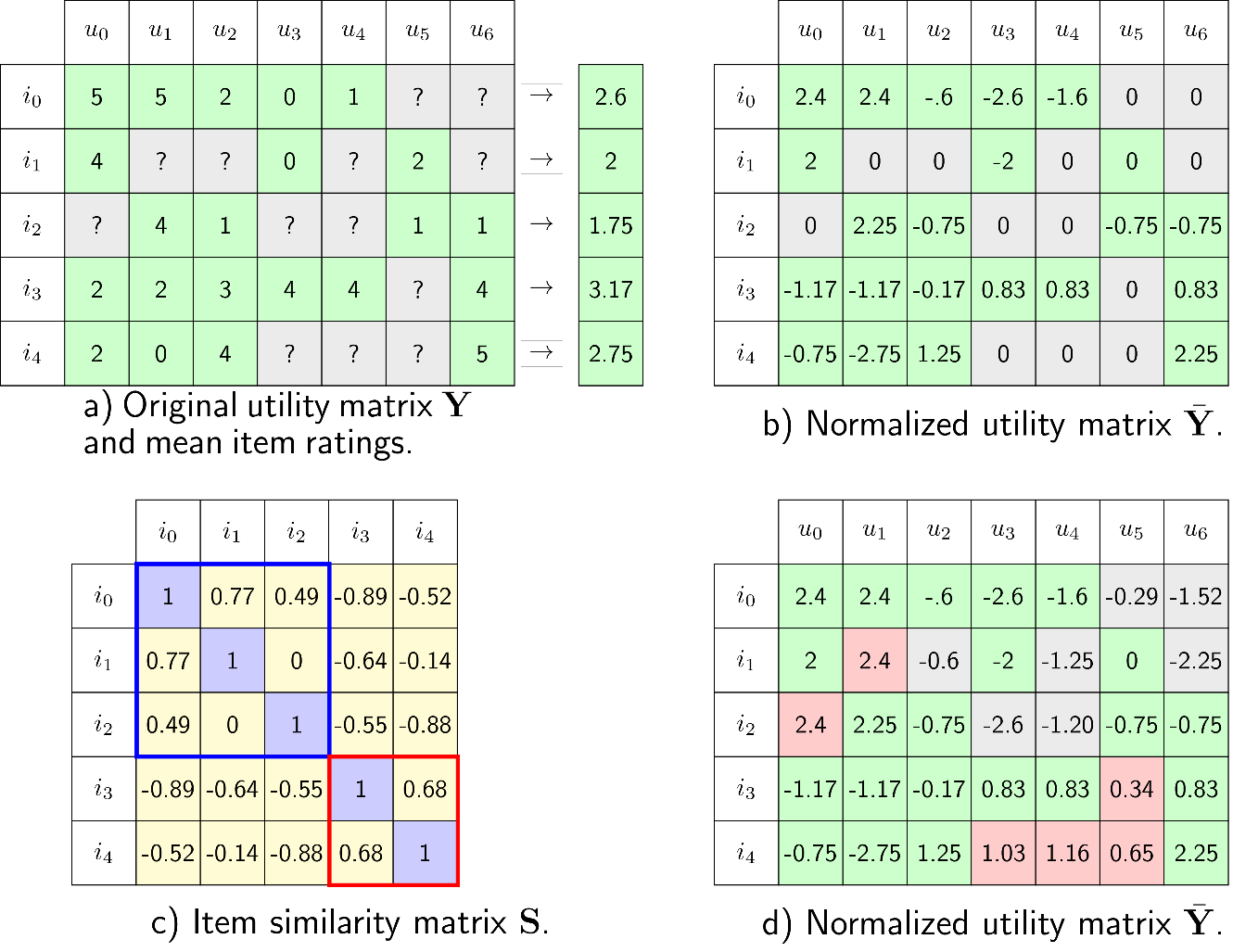
Hình 11: Ratings dự đoán theo thang 5

## 3.3. Item – item Collaborative Filtering

Trong phương pháp này, thay vì tính similarity giữa các users, chúng ta sẽ tính similarity giữa các items, rồi gợi ý cho users những items gần giống với item mà user đó đã thích. Lợi ích của phương pháp này là:

* Vì số lượng items thường rất nhỏ so với số lượng users nên kích thước ma trận Similarity sẽ nhỏ hơn rất nhiều, giúp tối ưu hơn cả về mặt lưu trữ và tốc độ tính toán. \
* Thường thì mỗi item được đánh giá bởi rất nhiều users, và con số này thường lớn hơn nhiều so với số items mà mỗi user đánh giá. Vì vậy, số giá trị đã biết trong một vector biểu diễn item sẽ lớn hơn nhiều so với một vector biểu diễn user. Trong trường hợp có thêm một số dữ liệu về ratings, giá trị trung bình ratings của iiCF sẽ ít thay đổi hơn so với uuCF, vì vậy sẽ ít phải cập nhật Similarity Matrix hơn.

Về mặt tính toán quy trình dự đoán missing ratings cũng tương tự như trong User-user CF tuy nhiên ở bước đầu tiên là chuẩn hóa ma trận Utility, thay vì tính trung bình cộng rating của các user, ta sẽ tính trung bình cộng rating của các items.

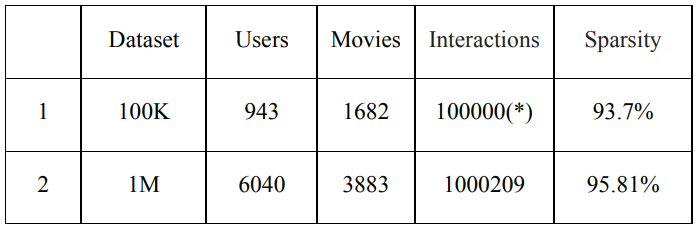


Hình 12: Tóm tắt các bước thực hiện iiCF

# **4. Triển khai thuật toán bằng Python**

## 4.1. Mô tả tập dữ liệu

GroupLens Reasearch đã thu thập và tạo ra các bộ dữ liệu có sẵn từ trang web MovieLens (http://movielens.org). Các tập dữ liệu này có tên là MovieLens (MLs). Có nhiều tập MLs khác nhau, tùy thuộc vào kích thước của tập hợp. Thống kê về các bộ dữ liệu MLs được sử dụng được trình bày trong Bảng 1 dưới đây:



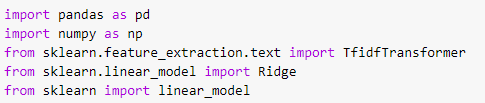
Hình 13: Dataset MoviesLens

Thành phần của các tập dữ liệu bao gồm các thông tin về: users, movies, ratings, genres và tags, etc.

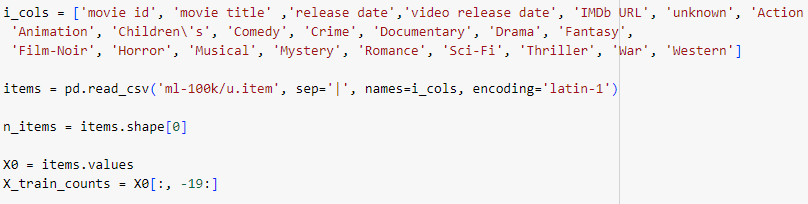
* Tập ratings bao gồm các trường: userID, itemID, rating, timestamp.
* Với tập users bao gồm các trường: ID, age, sex, job, zipcode.
* Tập genres thì bao gồm 3 trường: movieID, Tilte và Genres. Cụ thể thì các bộ phim trong bộ MLs được chia thành 19 thể loại khác nhau.

## 4.2. Content-based Filtering

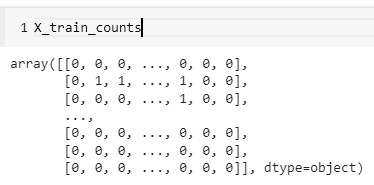
- **Khai báo các thư viện cần dùng**



**- Xây dựng item-profile:** ta sẽ get các vector nhị phân 19 chiều để mô tả cho mỗi bộ phim (do có 19 thể loại phim)

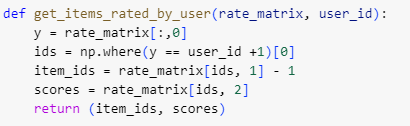


Ta được vector nhị phân như sau:

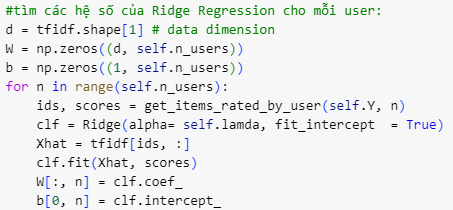


Ta có thể sử dụng trực tiếp các vector nhị phân này để làm feature vector. Nhưng ở đây, mình sẽ tiến hành tính tf-idf để sinh feature vector bằng thư viện TfidfTransformer 

**- Học mô hình:**

Đầu tiên chúng ta cần xây dựng hàm để get các bộ phim đã được rated bởi mỗi user: 

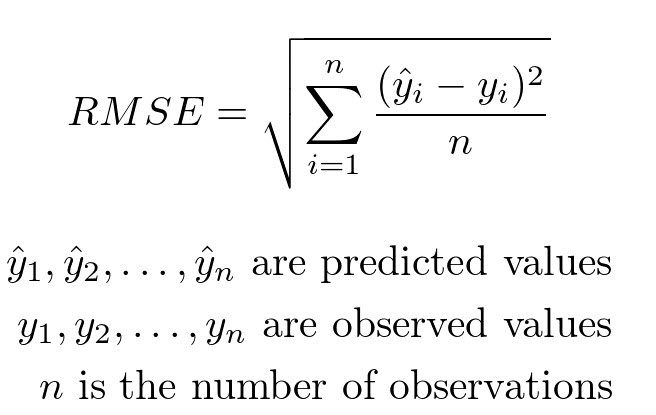
Với mỗi user, chúng ta sẽ sử dụng hàm get\_items\_rated\_by\_user để lấy mảng id các bộ phim đã được rated cùng điểm mà user đó đã rated cho mỗi bộ phim. Sau đó sử dụng các mảng đó để tính tính W và b, là các hệ số của Ridge Regression cho mỗi user.

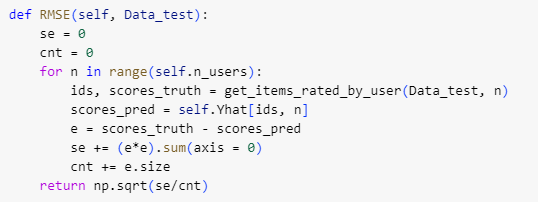


Sau khi tính được các hệ số W và b, ratings cho mỗi items được dự đoán bằng cách tính:

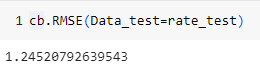


**- Đánh giá mô hình:** ta sẽ sử dụng độ đo RMSE để đánh giá mô hình



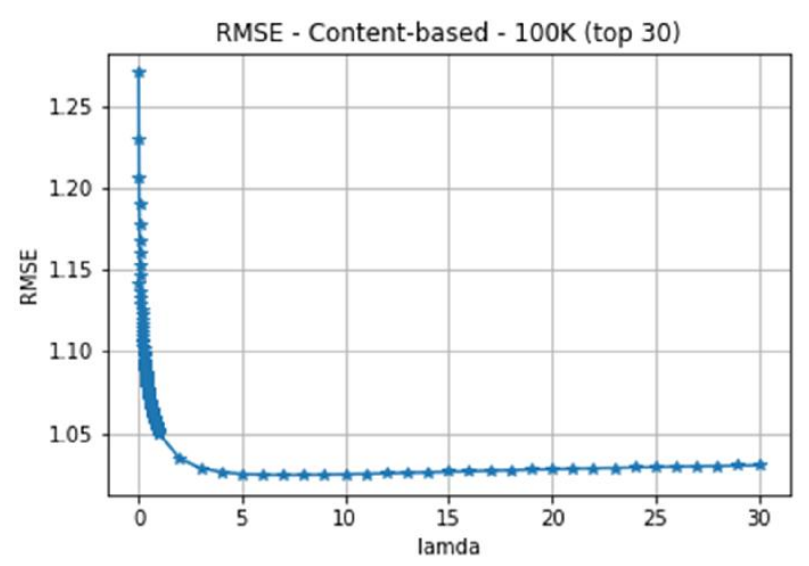


Đánh giá kết quả mô hình trên ta thu được MRSE ~ 1.245 với λ = 1



Ảnh hưởng của tham số λ tới mô hình:

Để phân tích ảnh hưởng của tham số λ đến mô hình Content-based, ta thiết lập tham số λ tăng dần từ 0.1 đến 30

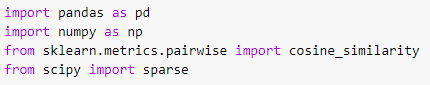


Hình 14: đồ thị MRSE khi lamda thay đổi

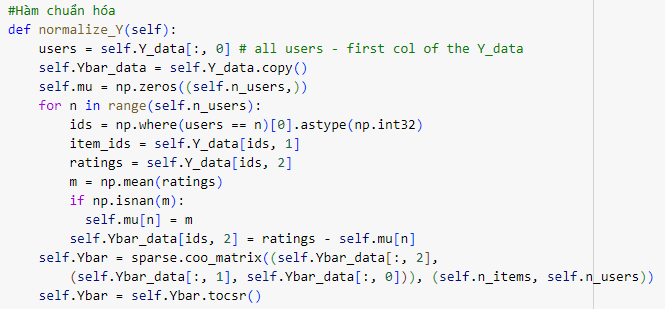
Quan sát Hình, ta có thể thấy được ảnh hưởng của tham số λ đến mô hình Content-based với các tập dữ liệu, thông qua độ đo RMSE. Với tập 100K, khi λ bắt đầu tăng từ 0.1 đến khoảng 1 thì RMSE giảm mạnh và tiến rất gần về giá trị 1.00. Khi λ tiếp tục tăng từ 1 đến 30 thì RMSE tăng không đáng kể.

## 4.3. Collaborative Filtering

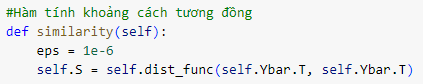
**- Khai báo các thư viện cần thiết**

****

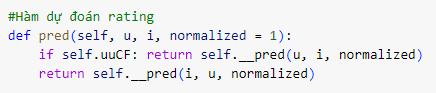
**- Chuẩn hóa vector**

****

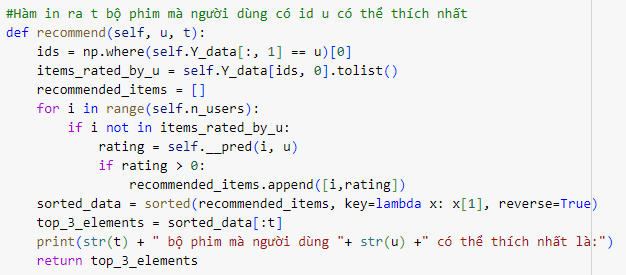
**- Tính khoảng cách tương đồng**

****

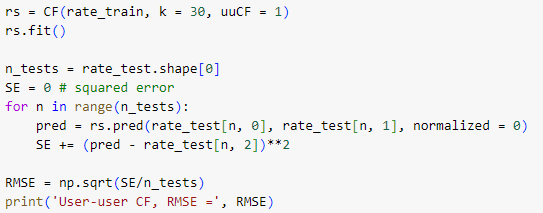
**- Dự đoán rating**

****

**- In ra top bộ phim mà người dùng có thể yêu thích nhất**

****

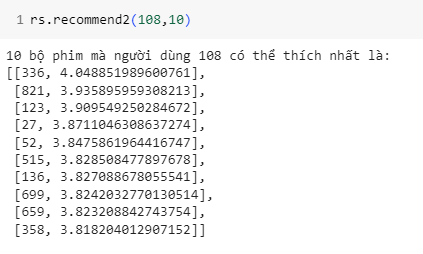
**- Kiểm thử và tính MRSE**

****

**Kết quả:**

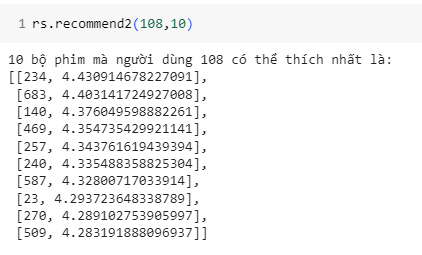
MRSE của user-user CF:





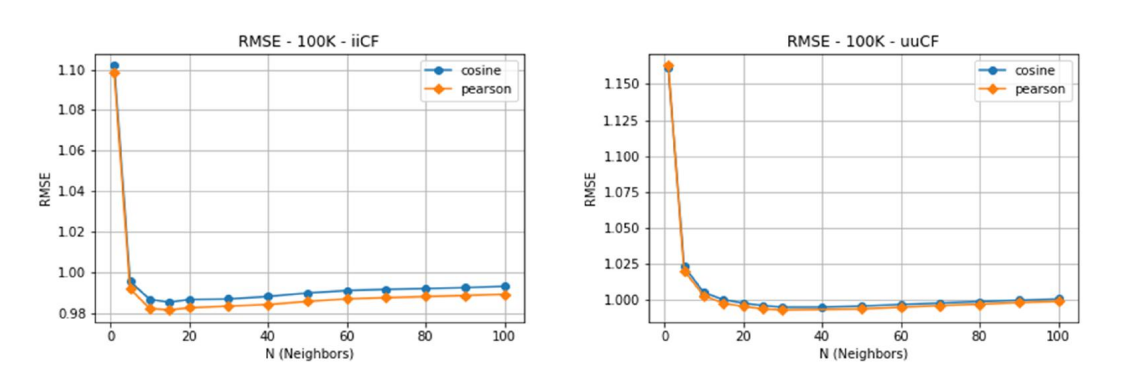
MRSE của item-item CF:





- **Ảnh hường của k neigh gần nhất và similarity functions:** Để phân tích ảnh hưởng của top N neighbors gần nhất và similarity function trong phương pháp NBCF (bao gồm cả iiCF và uuCF), ta thử nghiệm mô hình trên tập MLs 100K với bộ tham số:

* N tăng dần từ 10 đến 100: N ∈ {1, 5, 10 15, 20, 25} ∪ {30, 40 …, 90, 100}
* similarity\_function = {cosine\_similarity, pearson}



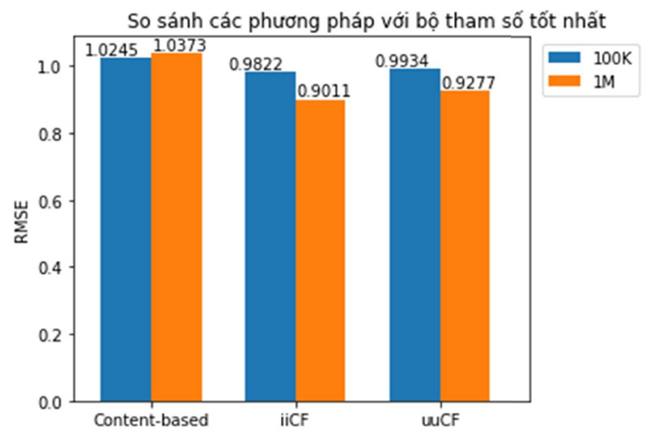
Hình 15: Đồ thị MRES của CF khi N neighbors thay đổi

Từ hình, ta có thể quan sát được ảnh hưởng của tham số N neighbors gần nhất và similarity function đến mô hình NBCF thông qua tham số RMSE. Đầu tiên, kết quả tạo ra khi sử dụng hai similarry function không chênh lệch nhiều với cả 2 mô hình (iiCF và uuCF). Đặc biệt, với iiCF, hai đường biểu diễn cho cosine và pearson gần như trùng khớp với nhau. Điều này chứng tỏ, hai hàm số này đem lại kết quả tương đối giống nhau trong việc xếp hạng các neighbors gần nhất.

Về ảnh hưởng của N neighbors gần nhất. Với cả 2 mô hình, ta thấy RMSE giảm nhanh khi N tăng từ 1 đế 5, sau đó tiếp tục giảm nhẹ khi N đến khoảng 20. Khi N lớn hơn 20, RMSE tăng chậm khi N tăng dần đến 100.

## 4.4. So sánh các phương pháp

Ta lựa chọn bộ tham số tốt nhất cho mỗi mô hình trên các tập dữ liệu MovieLens 100K và 1M(0.1) để so sánh hiệu quả của từng mô hình



Hình 16: Đồ thị so sánh RMSE các phương pháp

Quan sát Hình, ta thấy, mô hình Content-based cho kết quả kém nhất với RMSE trên cả hai tập dữ liệu đều lớn hơn 1. Hạn chế này là do mô hình thử nghiệm chỉ sử dụng thông tin về thể loại phim để xây dựng feature vector cho các items. Để thuật toán Content-based hiệu quả hơn, cần sử dụng tập dữ liệu cung cấp nhiều dữ liệu tường minh về items và feedback của users cho các items.

# **5. Kết luận**

## 5.1. Tóm tắt các kết quả của báo cáo

Báo cáo đã tập trung trình bày về hệ thống khuyến nghị và ứng dụng trong bài toán gợi ý phim. Sau phần giới thiệu tổng quan về hệ thống khuyến nghị và các ứng dụng của nó, báo cáo đi sâu vào các phương pháp tiếp cận (cụ thể là Content-based Filtering, user-user Collaborative và Item-item Collaborative Filtering) của hệ thống khuyến nghị, cách hoạt động của nó và ứng dụng của nó cho bài toán gợi ý phim.

Dựa trên những nội dung trình bày trong báo cáo, có thể tổng kết được một số kết quả như sau:

* Tìm hiểu về hệ gợi ý và bài toán gợi ý
* Tìm hiểu và thử nghiệm các phương pháp được sử dụng để giải quyết bài toán gợi ý, bao gồm: Content-based Filtering, Neighborhoods-based (gồm User-user Collaborative Filtering và Item-item Collaborative Filtering), trên tập dữ liệu MovieLens. Thu thập số liệu từ quá trình thử nghiệm để phân tích ảnh hưởng của các tham số đối với từng mô hình và lựa chọn bộ tham số tốt nhất cho từng mô hình.
* So sánh kết quả thu được khi thử nghiệm các mô hình. Mỗi mô hình đều có những ưu, nhược điểm khác nhau và có khả năng dự đoán kết quả khác nhau. Trong đồ án này, kết quả thu được cho thấy mô hình MF là tốt nhất.

## 5.2. Hướng phát triển.

Hướng phát triển tiếp theo của đồ án sẽ nhằm mục đích hạn chế các vấn đề còn tồn đọng trong đồ án hiện tại:

* Tiếp cận bài toán gợi ý các mô hình mới hơn như: View-enhanced eALS, Neural Matrix Factorization, Co-rating Model, etc.
* Mỗi phương pháp đều có những ưu và nhược điểm riêng. Vì vậy, ta có thể kết hợp các mô hình với nhau để thu được kết quả tối ưu hơn.

Sau đó, em mong muốn có thể áp dụng những kinh nghiệm và kiến thức thu được để áp dụng vào bài toán thực tế cụ thể, như mảng thương mại điện tử hoặc tin tức, giải trí trực tuyến. Áp dụng những kiến thức thu được trong đồ án này, cùng kiến thức về Web của bản thân để có thể tạo ra những trang Web đem lại trải nghiệm chân thật hơn tới người dùng

# **6. Tài liệu tham khảo**

[1] Nguyễn Hữu Tiệp, Machine Learning cơ bản, 2018

[2] Francesco Ricci, Lior Rokack, Bracha Shapira, Paul B. Kantor, Recommender systems handbook, Springer, 2015.

[3] Guy Shani and Asela Gunawardana, Evaluating recommendation systems, Recommender Systems Handbook, Springer, 2011

[4] Korea, Y., Bell, R., & Volinsky, Matrix Factorization techniques for recommender systems, IEEE Computer Society Press, 2009

[5] Ekstrand, Michael D., John T. Riedl, and Joseph A. Konstan, Collaborative filtering recommender systems, Foundations and Trends® in Human-Computer Interaction 4.2 (2011).

[6] Badrul Sarwar, George Karypis, Joseph Konstan, and John Riedl, Itembased collaborative filtering recommendation algorithms, ACM, 2001.

[7] Alexander Felfering, Michael Jeran, Gerald Ninaus, Florian Reinfrank, Stefan Reiterer and Martin Stettinger, Basic Approaches in Recommendation System, Recommendation Systems in Software Engineering, 2013.

**SOURCE CODE:** https://github.com/PhanTheMinhChau/TTNTCB