

**TRƯỜNG ĐẠI HỌC SƯ PHẠM KỸ THUẬT TP. HỒ CHÍ MINH**

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

**🙞🕮🙜**



**BÁO CÁO ĐỒ ÁN 3**

**ĐỀ TÀI:**

**TÌM HIỂU VỀ THUẬT TOÁN RECOMMENDATION**

**GVHD : Thầy Huỳnh Xuân Phụng**

**SVTH :**

**Nguyễn Thành Như 17110202**

**Võ Ngọc Thuận 17110234**

**TP. Hồ Chí Minh, tháng 12 năm 2020**

|  |  |
| --- | --- |
|  | **CỘNG HÒA XÃ HỘI CHỦ NGHĨA VIỆT NAM**  **Độc lập – Tự do – Hạnh phúc**  **\*\*\*\*\*\*\*** |

PHIẾU NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

Họ và tên Sinh viên: Nguyễn Thành Như MSSV: 17110202

Họ và tên Sinh viên: Võ Ngọc Thuận MSSV: 17110234  
Ngành: Công nghệ thông tin  
Tên đề tài: Tìm hiểu về thuật toán Recommendation  
Họ và tên Giảng viên hướng dẫn: thầy Huỳnh Xuân Phụng

NHẬN XÉT1. Về nội dung đề tài & khối lượng thực hiện:

2. Ưu điểm:

3. Khuyết điểm:

4. Đánh giá loại:

5. Điểm: (Bằng chữ: )

*Tp. Hồ Chí Minh, ngày tháng 01 năm 2021.*

Giảng viên hướng dẫn

*(Ký & ghi rõ họ tên)*

**LỜI CẢM ƠN**

Trong quá trình thực hiện, cả nhóm chúng em đã rất may mắn khi nhận được sự hướng dẫn, giúp đỡ từ thầy cô và bạn bè. Nay em xin được phép gửi lời cảm ơn chân thành này đến thầy Huỳnh Xuân Phụng, người đã trực tiếp hỗ trợ em trong suốt quá trình định hướng chọn đề tài, hướng dẫn, nhận xét và góp ý cũng như cung cấp tài liệu tham khảo. Nếu không có những lời hướng dẫn, những kinh nghiệm thực tiễn của cô thì em nghĩ rằng đề tài này sẽ khó có thể hoàn thiện và hoàn thành đúng thời hạn được. Một lần nữa, em xin cảm ơn thầy.

Em cũng xin chân thành cảm ơn các quý thầy cô trong khoa Đào tạo Chất lượng cao đã giúp đỡ hỗ trợ kiến thức cũng như giải đáp thắc mắc của em. Cùng với đó, em xin được gửi cảm ơn đến các bạn cùng khóa đã cung cấp nhiều thông tin và kiến thức hữu ích giúp cho em hoàn thiện đề tài hơn.

Khoảng thời gian có hạn, cùng với kiến thức còn hạn chế và còn nhiều bỡ ngỡ khác do đó thiếu sót là điều không thể tránh khỏi nên em rất mong nhận được những ý kiến đóng góp quý báo từ các quý Thầy Cô để kiến thức của em được hoàn thiện hơn sau này. Em xin chân thành cảm ơn.

Thành phố Hồ Chí Minh, ngày tháng 12 năm 2020

**Sinh viên thực hiện**

Nguyễn Thành Như Võ Ngọc Thuận

**LỜI NÓI ĐẦU**

Trong những năm gần đây, chúng ta có thể thấy rõ được sự phát triển nhanh chóng của công nghệ, giai đoạn “Cách mạng Công nghiệp 4.0” tập trung chủ yếu vào sản xuất thông minh dựa trên sự phát triển đột phá của cách ngành nghề công nghệ thông tin, công nghệ sinh học, công nghệ nano, ... Mỗi một cuộc cách mạng công nghệ đều sẽ mang đến một bước ngoặt lớn với cách thức chúng ta sản xuất, lao động, giúp thế giới xung quanh ta đang thay đổi từng ngày một cách mạnh mẽ như thế nào. Thế giới đang đi những bước dài mỗi ngày, góp một phần không nhỏ trong đó chính là công nghệ thông tin, và cụ thể hơn, một trong các công nghệ góp phần vào bước phát triển của công nghệ thông tin, chính là Machine Learning. Machine Learning (ML) là một trong các cụm từ được nhắc đến khá nhiều gần đây. Không chỉ trong lĩnh vực trí tuệ nhân tạo, mà rất, rất nhiều lĩnh vực khác cũng đang có sự tham gia của ML.

Cùng các lĩnh vực được áp dụng như đã nêu ở trên thì việc áp dụng ML vào kinh doanh trực tuyến cũng là một điều đương nhiên có. Với việc công nghệ phát triển nhanh chóng như hiện nay, các trang thương mại điện tử cũng phải cập nhật, nâng cấp cho mình những công nghệ mới nhất và một trong số đó là việc tăng trải nghiệm người dùng. Chính vì thế cả nhóm đã quyết định tìm hiểu về thuật toán Recommendation, một thuật toán khuyến nghị sản phẩm đến đúng khách hàng cần mua, tăng hiệu quả của việc kinh doanh.

MỤC LỤC

[DANH MỤC CÁC BẢNG 2](#_Toc60763163)

[DANH MỤC CÁC HÌNH 3](#_Toc60763164)

[CHƯƠNG 1. PHẦN MỞ ĐẦU 4](#_Toc60763165)

[1. Lý do chọn đề tài 4](#_Toc60763166)

[2. Mục tiêu đề tài 4](#_Toc60763167)

[3. Phương pháp nghiên cứu 4](#_Toc60763168)

[4. Giới thiệu đề tài 4](#_Toc60763169)

[CHƯƠNG 2. GIỚI THIỆU TỔNG QUAN VỀ RECOMMENDATION 5](#_Toc60763170)

[1. Khái niệm 5](#_Toc60763171)

[2. Tổng quan về Recommendation 5](#_Toc60763172)

[3. Mục đích sử dụng 6](#_Toc60763173)

[4. Tìm hiểu quy trình, các bước để xây dựng một hệ thống Recommendation 7](#_Toc60763174)

[5. Các phương pháp trong thuật toán gợi ý Recommendation. 8](#_Toc60763175)

[CHƯƠNG 3. UTILITY MATRIX 9](#_Toc60763176)

[1. Khái niệm 9](#_Toc60763177)

[2. Ví dụ minh họa 9](#_Toc60763178)

[CHƯƠNG 4. CONTENT – BASED RECOMMENDATION SYSTEM 11](#_Toc60763179)

[1. Giới thiệu 11](#_Toc60763180)

[2. Item profiles 11](#_Toc60763181)

[3. Xây dựng hàm mất mát 13](#_Toc60763182)

[4. Ví dụ về hàm mất mát cho user Nam 14](#_Toc60763183)

[5. Ưu điểm 14](#_Toc60763184)

[6. Nhược điểm 14](#_Toc60763185)

[CHƯƠNG 5. COLLABORATIVE FILTERING 15](#_Toc60763186)

[1. Giới thiệu 15](#_Toc60763187)

[2. Similarity functions 15](#_Toc60763188)

[3. RMSE (Root Mean Squared Error) hoặc MAE (Mean Absolute Error) 16](#_Toc60763189)

[4. Lập trình Collaborative Filtering trên Python 17](#_Toc60763190)

[4.1. Class Cosine 18](#_Toc60763191)

[4.2. Class CollaborativeFiltering 18](#_Toc60763192)

[4.3. Đánh giá hiệu quả của dự đoán 20](#_Toc60763193)

[CHƯƠNG 6: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN 22](#_Toc60763194)

[1. Tổng kết 22](#_Toc60763195)

[2. Hạn chế 22](#_Toc60763196)

[3. Hướng phát triển đề tài: 22](#_Toc60763197)

[TÀI LIỆU THAM KHẢO 23](#_Toc60763198)

# DANH MỤC CÁC BẢNG

[**Bảng 1. Ví dụ về Utility Matrix** 11](#_Toc60762813)

[**Bảng 2.** 14](#_Toc60762814)

# DANH MỤC CÁC HÌNH

[Hình 1. Ví dụ minh họa về Recommendation 7](#_Toc60762715)

[Hình 2. Trích - A New Similarity Measure Based on Mean Measure of Divergence for Collaborative Filtering in Sparse Environment 18](#_Toc60762716)

# 

# CHƯƠNG 1. PHẦN MỞ ĐẦU

## 1. Lý do chọn đề tài

Ngày nay việc nhu cầu sử dụng internet của con người ngày càng cao, thời gian con người dành thời gian cho nó ngày càng nhiều. Từ đó, ta nhận thấy việc sử dụng tìm kiếm được đúng những nội dung phù hợp với bản thân ngày càng được quan trọng đặc biệt là trong lĩnh vực kinh doanh thương mại. Chính vì những điều vừa nói, chúng em đã quyết định lựa chọn đề tài tìm hiểu về thuật toán Recommendation để ứng dụng Machine Learning giúp cho người dùng có được trải nghiệm tốt và chính xác nhất.

## 2. Mục tiêu đề tài

Tìm hiểu và nắm được khái niệm, lợi ích, nguyên lý hoạt động của hệ thống Recommendtion. Xây dựng thành công một sản phẩm demo áp dụng thuật toán đã tìm hiểu.

## 3. Phương pháp nghiên cứu

Vận dụng các kiến thức đã học được, tham khảo các nguồn tài liệu, giáo trình được cung cấp. Ngoài ra còn tham khảo thêm các nguồn tài liệu khác trên mạng internet và vận dụng tất cả để thực hiện đề tài nghiên cứu một cách thành công nhất.

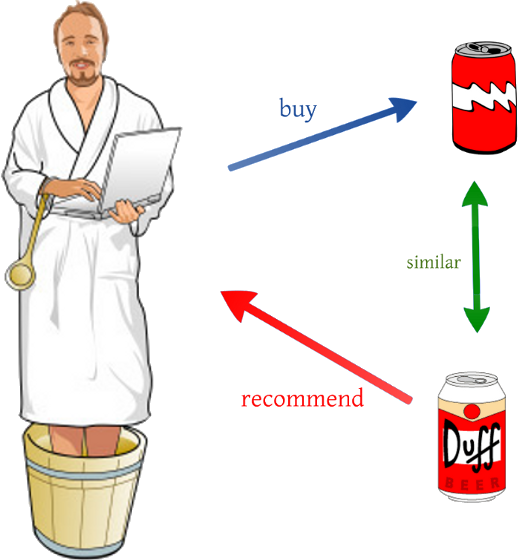
## 4. Giới thiệu đề tài

Ở đề tài này chúng em tiến hành tìm hiểu chung về thuật toán Recommendation. Thuật toán này giúp gợi ý nội dung đến đúng đối tượng người dùng. Hiện nay, thuật toán đang được áp dụng rất nhiều trong các lĩnh vực công nghệ. Để hiểu chi tiết hơn, chúng ta sẽ xem ở phần nội dung bên dưới.

# CHƯƠNG 2. GIỚI THIỆU TỔNG QUAN VỀ RECOMMENDATION

## 1. Khái niệm

Thuật toán về Recommendation là gì? Ngay từ cái tên của nó tã cũng đã thấy được ý nghĩa của nó là gì, Recommendation dịch ra có nghĩa là gợi ý rằng một cái gì đó tốt hoặc phù hợp cho một mục đích hoặc công việc cụ thể. Nó là một dạng của hệ hỗ trợ ra quyết định, cung cấp giải pháp mang tính cá nhân hóa mà không phải trải qua quá trình tìm kiếm phức tạp. Hệ gợi ý học từ người dùng và gợi ý các sản phẩm tốt nhất trong số các sản phẩm phù hợp. Nói một cách gần gũi thì Recommendation giống như một người mai mối giúp khách hàng và sản phẩm tìm được nhau.[1]



Hình 1. Ví dụ minh họa về Recommendation

## 2. Tổng quan về Recommendation

Hệ thống Recommendation là một dạng của hệ hỗ trợ ra quyết định, cung cấp giải pháp mang tính cá nhân hóa mà không phải trải qua quá trình tìm kiếm phức tạp. Hệ gợi ý học từ người dùng và gợi ý các sản phẩm tốt nhất trong số các sản phẩm phù hợp. Thuật toán Recommendation sẽ sử dụng các tri thức khai phá học được từ hành vi con người dùng để đưa ra các gợi ý về sản phẩm mà họ thích trong hàng ngàn hàng vạn sản phẩm có trong hệ thống. Các website thương mại điện tử, ví dụ như sách, phim, nhạc, báo...sử dụng hệ thống gợi ý để cung cấp các thông tin giúp cho người sử dụng quyết định sẽ lựa chọn sản phẩm nào. Nhiệm vụ chính của một Recommendation System là tối ưu hóa lượng thông tin khổng lồ nhằm đưa đến cho người dùng những thứ phù hợp nhất với họ. Điểm mấu chốt ở đây, là nó sẽ phải tìm ra được mối quan hệ của từng người dùng với từng sản phẩm (item) dựa trên những thói quen, lựa chọn của họ trong quá khứ, thuộc tính của sản phẩm đó,.. để có thể gợi ý cho họ những thứ phù hợp trong tương lai. Ví dụ như một người dùng có sở thích là về các video lịch sử sẽ có khả năng cao rằng sẽ thích thú với một video về lịch sử khác hoặc một video về nội dung giáo dục hơn là một bộ phim hành động. Những điều tương tự như thế đều sẽ là kết quả của một hệ thống gợi ý. Các sản phẩm được gợi ý dựa trên số lượng sản phẩm đó đã được bán, dựa trên các thông tin cá nhân của người sử dụng, dựa trên sự phân tích hành vi mua hàng trước đó của người sử dụng để đưa ra các dự đoán về hành vi mua hàng trong tương lai của chính khách hàng đó. Các dạng gợi ý bao gồm: gợi ý các sản phẩm tới người tiêu dùng, các thông tin sản phẩm mang tính cá nhân hóa, tổng kết các ý kiến cộng đồng, và cung cấp các chia sẻ, các phê bình, đánh giá mang tính cộng đồng liên quan tới yêu cầu, mục đích của người sử dụng đó.

## 3. Mục đích sử dụng

Mục đích cơ bản nhất của Recommendation System đó là giúp tăng doanh thu cho trang web, tăng trải nghiệm của người dùng bằng việc gợi ý hiệu quả những sản phẩm (item) cho tuỳ từng người dùng, nó sẽ giúp người dùng cảm thấy thuận tiện, thích thú hơn. Ngoài ra, Recommendation System mang lại cho người dùng những điều sau:

* *Sự liên quan*: Người dùng sẽ truy cập/ sử dụng/ mua những thứ liên quan tới sở thích, thói quen của họ.
* *Sự mới lạ*: Một hệ gợi ý hiệu quả sẽ đưa ra được những sản phẩm liên quan tới người dùng mà họ chưa từng thấy bao giờ trong quá khứ. Việc gợi ý những sản phẩm hot, trending sẽ ít mang lại cảm giác mới lạ cho người sử dụng và dẫn tới việc mất cân bằng và sự phong phú về mặt sản phẩm của hệ thống.
* *Sự bất ngờ*: Một mức độ cao hơn của mới lạ, khi mà sản phẩm được gợi ý không chỉ chưa từng được người dùng biết đến mà còn gây bất ngờ cho họ. Có thể hiểu đơn giản hơn là những sự liên quan mà người dùng không hề biết tới.
* *Sự phong phú*: Trước đây, các hệ thống gợi ý thường sẽ đưa ra một danh sách các sản phẩm rất liên quan tới nhau để gợi ý cho người dùng. Điều này có nguy cơ dẫn tới một khả năng là người dùng sẽ không thích bất cứ một cái nào ở trong danh sách trên. Nói cách khác, sự phong phú tăng xác suất trong việc người dùng chắc chắn sẽ thích ít nhất một sản phẩm từ danh sách đó, không gây nhàm chán cho họ bởi những sản phẩm giống nhau lặp đi lặp lại.

Với những gì Recommendation System đem lại như trên nó không những giúp ích cho khách hàng mà qua đó giúp tăng số lượng sản phẩm bán được cho website.

## 4. Tìm hiểu quy trình, các bước để xây dựng một hệ thống Recommendation

Các bước để xây dựng một hệ thống Recommendation là:

* ***Thu thập là lưu trữ thông tin***: như chúng ta đã biết thì Recommendation là một thuật toán áp dụng Machine Learning, thì thứ đầu tiên phải có đó chính là việc thu thập dữ liệu để tiến hành các bước sau như là đánh giá, phân tích. Những dữ liệu chúng ta tiến hành thu thập đó là các hành động/tương tác của người dùng trên hệ thống đối với một đối tượng/sản phẩm, chẳng hạn như: mua,số lượt xem, đánh giá,…
* ***Phân tích/chuẩn hóa dữ liệu***: theo nhóm tìm hiểu được thì sau khi hoàn thành bước thu thập thông tin, dữ liệu thì chúng được lưu trữ lại trong database, tuy nhiên lúc này những thông tin chúng ta thu thập được sẽ là một ma trận thưa (trống nhiều vị trí, ví dụ một người A có thể đã xem sản phẩm A, E, F nhưng họ không xem các sản phẩm B, C, D thì lúc này dữ liệu ở người A với sản phẩm B, C, D không có).  Chính vì lý do đó nên trước khi đi tính toán độ tương tự của các user, chúng ta cần phải điền đầy đủ các dấu hỏi đó mà nó **không làm ảnh hưởng đến độ tương đương** của ma trận. Sau khi điền đầy đủ thông tin, dữ liệu thì chúng ta mới có thể áp dụng được các thuật toán để tính toán độ tương tự vào.
* ***Lọc và đưa ra gợi ý phù hợp:*** chọn được mô hình phù hợp từ những dữ liệu đã được chuẩn hóa, chúng ta tiến hành chạy mô hình rồi phân tích để đưa ra gợi ý phù hợp nhất cho người dùng.

## 5. Các phương pháp trong thuật toán gợi ý Recommendation.

Các phương pháp trong thuật toán Recommendation:

* ***Content - Based Recommendation System (Gợi ý dựa vào nội dung):*** Ví dụ: một *user* đọc rất nhiều sách về kỹ năng, vậy thì gợi ý một cuốn sách trong cơ sở dữ liệu có chung đặc tính *kỹ năng* tới *user* này, ví dụ sách *Muôn kiếp nhân sinh*. Cách tiếp cận này yêu cầu việc sắp xếp các *items* vào từng nhóm hoặc đi tìm các đặc trưng của từng *item*. Tuy nhiên, có những *items* không có nhóm cụ thể và việc xác định nhóm hoặc đặc trưng của từng *item* đôi khi là bất khả thi.
* ***Collaborative Filtering (Lọc cộng tác):*** hệ thống gợi ý *items* dựa trên sự tương quan (similarity) giữa các *users* hoặc *items*. Có thể hiểu rằng ở nhóm này một *item* được *recommended* tới một *user* dựa trên những *users* có *hành vi* tương tự. Ví dụ: *users A, B, C* đều thích các sách sách của nhà văn Nguyễn Nhật Ánh. Ngoài ra, hệ thống biết rằng *users B, C* cũng thích các sách của nhà văn Nguyễn Ngọc Thạch nhưng chưa có thông tin về việc liệu user A có thích Nguyễn Ngọc Thạch hay không. Dựa trên thông tin của những users tương tự là B và C, hệ thống có thể dự đoán rằng A cũng thích Nguyễn Ngọc Thạch và gợi ý tác phẩm của tác giả này tới *A*.
* ***Session - Based Recommendation System (Gợi ý theo chuỗi hoạt động):*** dựa vào chuỗi hoạt động của một người dùng để đưa ra gợi ý. Ví dụ: trên trang Tiki, khi click mua/xem một món hàng nào đó, ở phía dưới sẽ hiển thị những sản phẩm thường được mua cùng kèm theo.
* ***Clustering User (Phân cụm người dùng):*** phân cụm và gợi ý người dùng theo nhóm.

Trong các Recommendation Systems thường được chia thành hai nhóm lớn chính đó là Content - Based Recommendation System và Collaborative Filtering.

# CHƯƠNG 3. UTILITY MATRIX

## 1. Khái niệm

Đây là ma trận biểu diễn mức độ quan tâm (rating) của người dùng (user) với mỗi sản phẩm (item). Ma trận này được xây dựng từ dữ liệu mà hệ thống thu thập được từ người dùng. Nhưng không phải mỗi người dùng đều quan tâm đến tất cả những sản phẩm trong hệ thống, chính vì thế trong ma trận sẽ có những ô rỗng. Nhiệm vụ của hệ thống là dựa vào các ô đã có giá trị trong ma trận trên (dữ liệu thu được từ trong quá khứ), thông qua mô hình đã được xây dựng, dự đoán các giá trị của các ô còn trống (của user hiện hành), sau đó sắp xếp kết quả dự đoán và chọn ra Top-N items theo thứ tự rating giảm dần, từ đó gợi ý chúng cho người dùng.

## 2. Ví dụ minh họa

**Bảng 1. Ví dụ về Utility Matrix**

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
|  | **An** | **Linh** | **Trang** | **Nam** |
| Mắt biếc | 5 | 1 | 4 | ? |
| Tôi thấy hoa vàng trên cỏ xanh | 4 | 3 | 5 | 3 |
| Gia đình là số 1 | 2 | 4 | 2 | ? |
| Về nhà đi con | ? | 4 | 3 | 5 |
| Kính vạn hoa | ? | 2 | 5 | 2 |

Bảng trên thể hiện về mức độ quan tâm của người dùng đến với 5 bộ phim và hai bộ phim được phân thành hai nhóm, nhóm 1 là các bộ phim được chuyển thể từ tác giả Nguyễn Nhật Ánh, nhóm 2 là các bộ phim về gia đình. Từ bảng trên ta thấy An, Trang có mức độ quan tâm cao đến với phim Mắt biếc và Tôi thấy hoa vàng trên cỏ xanh. Ngoài ra ta thấy Linh đang quan tâm cao đến với phim Kính vạn hoa, từ mức độ tương đồng ở trên ta cũng có thể gợi ý cho An đến với bộ phim Kính vạn hoa. Tương tự như vậy, ta thấy Nam và Linh cùng quan tâm cao đến với phim Về nhà đi con, cùng với mức độ quan tâm đến với phim Kính vạn hoa đều rất thấp từ đó ta có thể thấy được có thể Nam sẽ thích phim Gia đình là số 1.

Thông thường, có rất nhiều users và items trong hệ thống, và mỗi user thường chỉ rate một số lượng rất nhỏ các item, thậm chí có những user không rate item nào (với những users này thì cách tốt nhất là gợi ý các items phổ biến nhất). Vì vậy, lượng ô màu xám của utility matrix trong các bài toán đó thường là rất lớn, và lượng các ô đã được điền là một số rất nhỏ.

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

Những gì ta có thể thấy ở trên đều dựa vào nhận thức của chúng ta, vậy làm sao để cho hệ thống biết được và giới thiệu cho người dùng? Thì để hệ thống có thể làm được những điều trên thì ta phải tiến hành qua các bước xử lý, huấn luyện cho hệ thống để có thể tiến hành thu thập thông tin và phân tích để sau cùng có thể đưa ra gợi ý chính xác.

# 

# CHƯƠNG 4. CONTENT – BASED RECOMMENDATION SYSTEM

## 1. Giới thiệu

[Content-based Recommendation Systems](https://machinelearningcoban.com/2017/05/17/contentbasedrecommendersys/) là một Hệ thống gợi ý sản phẩm đơn giản dựa trên đặc trưng của mỗi item. Đặc điểm của Content-based Recommendation Systems là việc xây dựng mô hình cho mỗi user không phụ thuộc vào các users khác mà phụ thuộc vào profile của mỗi items. Việc làm này có lợi thế là tiết kiệm bộ nhớ và thời gian tính toán. Đồng thời, hệ thống có khả năng tận dụng các thông tin đặc trưng của mỗi item như được mô tả trong bản mô tả (description) của mỗi item. Bản mô tả này có thể được xây dựng bởi nhà cung cấp hoặc được thu thập bằng cách yêu cầu users gắn tags cho items. Việc xây dựng feature vector cho mỗi item thường bao gồm các kỹ thuật Xử lý ngôn ngữ tự nhiên (Natural Language Processing - NLP)

## 2. Item profiles

Trong các hệ thống content-based, chúng ta cần xây dựng một bộ hồ sơ (profile) cho mỗi item. *Profile* này được biểu diễn dưới dạng toán học là một feature vector. Trong những trường hợp đơn giản, *feature vector* được trực tiếp trích xuất từ *item*. Ví dụ, xem xét các *features* của một bộ phim mà có thể được sử dụng trong các Recommendation Systems:

* Diễn viên: Cùng là một thể loại phim hành động nhưng có người thích phim của diễn viên Ma Dong-seok.
* Đạo diễn: Cùng là một thể loại nhưng có người thích các bộ phim của đạo diễn Nguyễn Quang Dũng.
* Thể loại: Điều này quá rõ ràng, phim tình cảm, phim hài, phim hành động, phim kinh dị, mỗi thể loại sẽ thu hút một nhóm đối tượng khác nhau.

Sau đây ta xây dựng một feature vector hai chiều cho mỗi bộ phim ở bảng 1: chiều thứ nhất là mức độ Tình cảm, chiều thứ hai là mức độ Gia đình của bộ phim đó. Đặt các feature vector cho mỗi bộ phim là x1, x2, x3, x4, x5:

**Bảng 2.**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | **An** | **Linh** | **Trang** | **Nam** | **Item’s feature vector** |
| Mắt biếc | 5 | 1 | 4 | ? | X1=[0.99, 0.3] |
| Tôi thấy hoa vàng trên cỏ xanh | 4 | 3 | 5 | 3 | X1=[0.95, 0.2] |
| Gia đình là số 1 | 2 | 4 | 2 | ? | X1=[0.15, 0.9] |
| Về nhà đi con | ? | 4 | 3 | 5 | X1=[0.3, 0.92] |
| Kính vạn hoa | ? | 2 | 5 | 2 | X1=[0.12, 0.35] |
| User’s model | θ1 | θ2 | θ3 | θ4 |  |

Giả sử feature vector cho mỗi sản phẩm được cho trong cột cuối cùng. Với mỗi người dùng, chúng ta cần tìm một mô hình θi tương ứng sao cho mô hình thu được là tốt nhất. Ở đây, chúng ta tạm coi các vector này đã được xác định bằng một cách nào đó. Tương tự như thế, hành vi của mỗi user cũng có thể được mô hình hoá dưới dạng tập các tham số θ. Dữ liệu huấn luyện để xây dựng mỗi mô hình θu là các cặp (item profile, rating) tương ứng với các item mà user đó đã đánh giá. Việc điền các giá trị còn thiếu trong ma trận utility chính là việc dự đoán mức độ quan tâm khi áp dụng mô hình θu lên chúng. Đầu ra này có thể được viết dưới dạng một hàm f(θu, xi). Việc lựa chọn dạng của f(θu, xi) tuỳ thuộc vào mỗi bài toán. Bài toán đi tìm mô hình θi cho mỗi *user* có thể được coi là một bài toán Regression trong trường hợp *ratings* là một dải giá trị, hoặc bài toán *Classification* trong trường hợp *ratings* là một vài trường hợp cụ thể, như *like/dislike* chẳng hạn. Dữ liệu training để xây dựng mỗi mô hình θi là các cặp (*item profile, ratings*) tương ứng với các *items* mà *user* đó đã *rated*. Việc *điền* các giá trị còn thiếu trong ma trận Utility chính là việc dự đoán đầu ra cho các *unrated items* khi áp dụng mô hình θi lên chúng. Việc lựa chọn mô hình Regression/Classification nào tùy thuộc vào ứng dụng.

## 3. Xây dựng hàm mất mát

Giả sử rằng ta có thể tìm được một mô hình cho mỗi user, được minh hoạ bởi một vector cột hệ số **w**n ∈ ℝd và bias bn sao cho mức độ quan tâm của một user tới một item có thể tính được bằng một hàm tuyến tính:

ymn = xm + bn (17.1)

Xét một *user* thứ *n* bất kỳ, nếu ta coi tập huấn luyện là tập hợp các thành phần đã được điền của yn (cột thứ n của ma trận Y), ta có thể xây dựng hàm mất mát tương tự như *ridge regression* (với *l2 regularization*) như sau:

(wn, bn) = (17.2)

trong đó, thành phần thứ hai là regularization và λ là một tham số dương; sn là số lượng các item mà user thứ n đã đánh giá, là tổng các phần tử trên cột thứ n của ma trận R, tức sn =. Chú ý rằng regularization thường không được áp dụng lên bias bn.

Vì biểu thức hàm mất mát (17.2) chỉ phụ thuộc vào các item đã được đánh giá, ta có thể rút gọn nó bằng cách đặt ∈ là vector con của yn, được xây dựng bằng cách trích các thành phần khác dấu ‘?’ ở cột thứ n của Y. Đồng thời, đặt ∈ là ma trận con của ma trận đặc trưng X, được tạo bằng cách trích các cột tương ứng với các item đã được đánh giá bởi user thứ n. (Xem ví dụ phía dưới để hiểu rõ hơn). Khi đó, biểu thức hàm mất mát của mô hình cho user thứ n được viết gọn thành:

(wn, bn) = (17.3)

trong đó, en là vector cột với tất cả các thành phần là 1. Đây chính xác là hàm mất mát của ridge regression. Cặp nghiệm wn, bn có thể được tìm thông qua các thuật toán gradient descent. Trong chương này, chúng ta sẽ trực tiếp sử dụng class Ridge trong sklearn .linear\_model. Có một điểm đáng lưu ý ở đây là wn chỉ được xác định nếu user thứ n đã đánh giá ít nhất một sản phẩm.

## 4. Ví dụ về hàm mất mát cho user Nam

Với ví dụ trong Bảng 2, ma trận đặc trưng cho các item (mỗi cột tương ứng với một item) là : (17.4)

Xét trường hợp của user Nam với n = 5, y5 = ⇒ r5 = . Vì Nam mới chỉ đánh giá item thứ hai, thứ tư và thứ năm nên s5 = 3. Hơn nữa,

, , (17.5)

Khi đó, hàm mất mát cho hệ số tương ứng với user Nam là:

(w5, b5) = (17.6)

## 5. Ưu điểm

Việc xây dựng mô hình cho mỗi user không phụ thuộc vào các user khác mà phụ thuộc vào profile của các item. Việc làm này có lợi thế là tiết kiệm bộ nhớ và thời gian tính toán.

## 6. Nhược điểm

Cách làm này có hai nhược điểm cơ bản:

* Thứ nhất, khi xây dựng mô hình cho một user, các hệ thống content-based không tận dụng được thông tin từ các user khác. Những thông tin này thường rất hữu ích vì hành vi mua hàng của các user thường được nhóm thành một vài nhóm đơn giản. Nếu biết hành vi mua hàng của một vài user trong nhóm, hệ thống nên có khả năng suy luận ra hành vi của những user còn lại.
* Thứ hai, không phải lúc nào chúng ta cũng có thể xây dựng profile cho mỗi ite

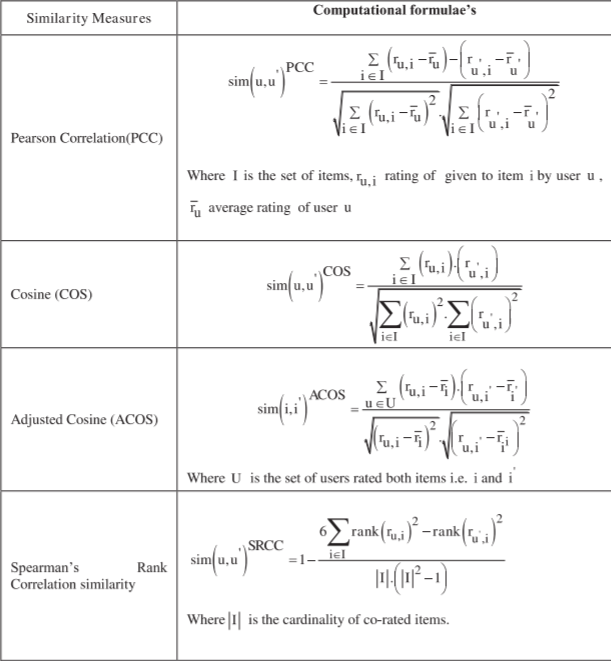
# CHƯƠNG 5. COLLABORATIVE FILTERING

## 1. Giới thiệu

Ý tưởng của Collaborative Filtering (CF) là xác định mức độ quan tâm của một user tới một item dựa trên hành vi của các user khác gần giống với user này. Việc gần giống nhau giữa các user có thể được xác định thông qua mức độ quan tâm (similarity) của các user này tới các item khác mà hệ thống đã biết.

## 2. Similarity functions

Để đo *similarity* giữa hai *users*, cách thường làm là xây dựng *feature vector* cho mỗi *user* rồi áp dụng một hàm có khả năng đo *similarity* giữa hai vectors đó. Các vectors này được xây dựng trực tiếp dựa trên Utility matrix chứ không dùng dữ liệu ngoài. Với mỗi user, thông tin duy nhất chúng ta biết là các *ratings* mà *user* đó đã thực hiện, tức cột tương ứng với *user* đó trong Utility matrix. Tuy nhiên, khó khăn là các cột này thường có rất nhiều *missing ratings* vì mỗi *user* thường chỉ *rated* một số lượng rất nhỏ các *items*. Cách khắc phục là bằng cách nào đó, ta *giúp* hệ thống *điền* các giá trị này sao cho việc điền không làm ảnh hưởng nhiều tới *sự giống nhau* giữa hai vector. Việc *điền* này chỉ phục vụ cho việc tính *similarity* chứ không phải là *suy luận* ra giá trị cuối cùng. Sau khi dữ liệu đã được chuẩn hóa xong thì chúng ta sẽ tiến hành sử dụng một vài *similarity function* sau:

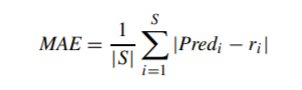


Hình 2. Trích - A New Similarity Measure Based on Mean Measure of Divergence for Collaborative Filtering in Sparse Environment

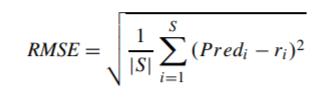
## 3. RMSE (Root Mean Squared Error) hoặc MAE (Mean Absolute Error)

Hiệu quả và độ chính xác của các dự đoán được xác định dựa trên tham số đó chính là RMSE (Root Mean Squared Error) hoặc MAE (Mean Absolute Error).

MAE sẽ được tính bằng công thức sau:



RMSE sẽ được tính bằng công thức sau:



RMSE là một số thực không âm sẽ có giá trị từ 0.0 và 0.0 là trường hợp tốt nhất khi mà dự đoán không có sai lệch.

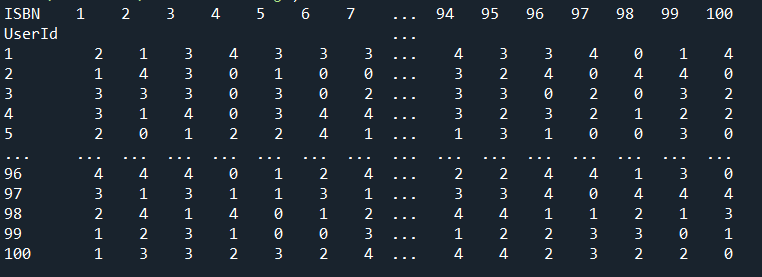
Để thực hiện tính toán 2 tham số này trong python chúng ta có thể sử dụng 2 thư viện là **from** **sklearn.metrics** **import** mean\_absolute\_error để tính MAE

và **from** **sklearn.metrics** **import** mean\_squared\_error và set squared = False để tính RMSE

## 4. Lập trình Collaborative Filtering trên Python

Khởi tạo class Collaborative Filtering

Dữ liệu đầu vào của hàm khởi tạo class Collaborative Filtering là ma trận df\_ustd chứa các đánh giá của người dùng về các cuốn sách có dạng như sau

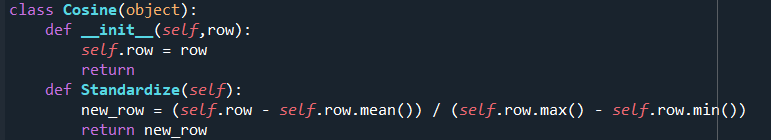


Để cho đơn giản, giả sử rằng không có users hay items mới và cũng không có ratings nào bị thay đổi.

### 4.1. Class Cosine

Với việc sử dụng độ tương đồng cosine similarity ta tiến hành chuẩn hóa dữ liệu ratings bằng cách sau:

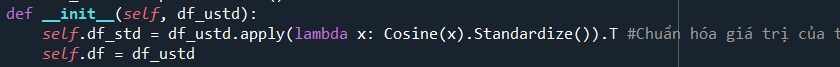
Xây dựng 1 class Cosine và 1 phương thức để chuẩn hóa



Hàm standardize sẽ chịu trách nhiệm tính toán chuẩn hóa dữ liệu cho từng dòng ratings.

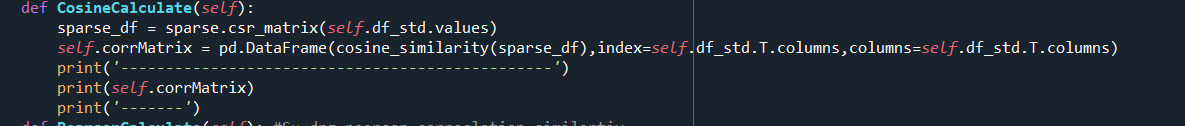
### 4.2. Class CollaborativeFiltering

Ngay tại hàm khởi tạo class CollaborativeFiltering chúng ta chuẩn hóa ma trận df\_ustd thành ma trận df\_std để sử dụng



Tạo ra ma trận Corr để thực hiện predict.

Với độ tương đồng cosine ta xây dựng hàm sau



Hàm cosine\_similarity được import từ thư viện sklearn.metrics.pairwise

**Với độ tương đồng pearson correclation similartiy**



**Với độ tương đồng spearman correclation similarity**



**Với độ tương đồng kendall correclation similarity**



Self.df chính là ma trận chưa chuẩn hóa và ta sử dụng hàm corr(method = ) để tính toán ma trận theo các độ tương đồng.

**Hàm Get\_Similar\_Score**

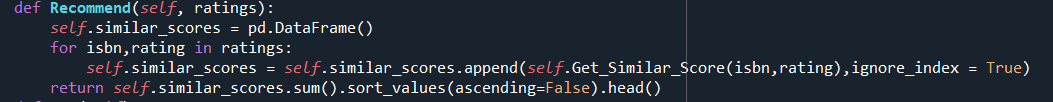
Sau khi có ma trận độ tương đồng, việc tiếp theo cần làm đó là tính toán độ tương đồng similar\_score, đầu vào của hàm sẽ bao gồm isbn của sách và rating là đánh giá của sách.



Sử dụng ma trận corr để tính toán similar\_score cho cuốn sách được đánh giá mới, rating mới giảm 2.5 vì ở đây chúng ta sử dụng max rating là 5.

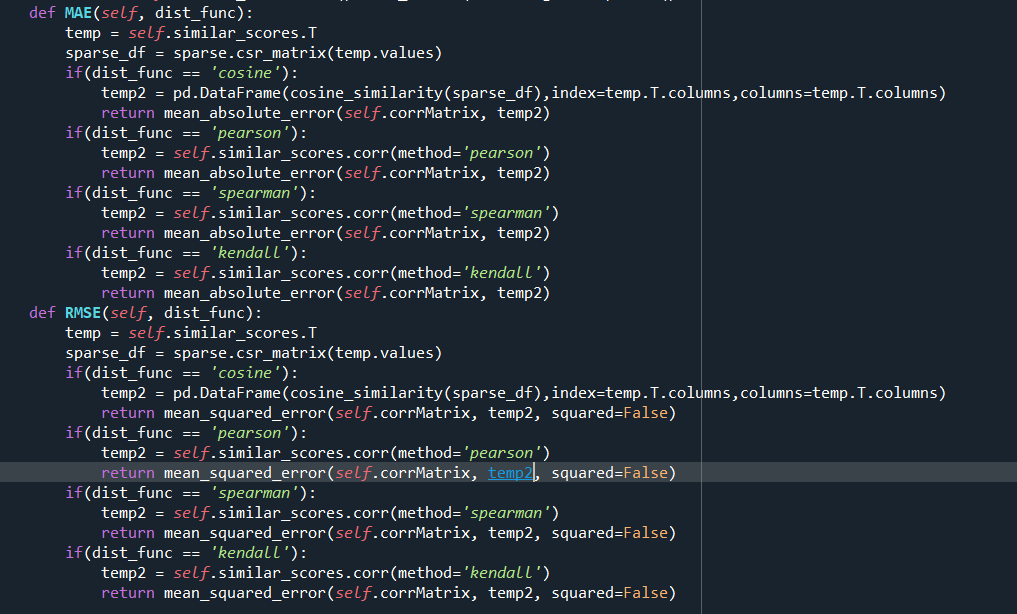
**Hàm Recommend**

Đầu vào là một mảng ratings các object chứa isbn của sách và rating của nó, hàm sẽ thực hiện tính toán một ma trận similar\_scores dựa vào hàm Get\_Similar\_Score đã cài đặt ở trên và mảng ratings, tiến hành sum theo index và trả về kết quả đã sort giảm dần lấy 5 phần tử đầu tiên.



### 4.3. Đánh giá hiệu quả của dự đoán

Để đánh giá độ hiệu quả của dự đoán chúng ta dựa vào 2 tham số MAE và RMSE để đưa ra đánh giá, cài đặt 2 hàm MAE và RMSE



Để tính MAE ta sử dụng hàm mean\_absolute\_error được cung cấp từ thư viện sklearn.metrics. RMSE ta sử dụng hàm mean\_squared\_error cũng được cung cấp từ thư viện trên.

Đầu vào của 2 hàm là tên độ tương đồng được sử dụng.

Tiến hành chạy trên data ratings với 100 books và 100 users

Bảng RMSE:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Độ tương đồng  Input(\*) | Cosine | PCC | SRC | KCC |
| [(1,5),(2,3)] | 0.864013166 | 0.992422459 | 0.992388307 | 0.992132368 |
| [(1, 5), (2, 3), (99, 3)] | 0.786548007 | 0.822964501 | 0.75913859 | 0.707449617 |
| [(1,5), (2,3), (99,3), (15,1)] | 0.651896751 | 0.668561654 | 0.62243002 | 0.546451714 |
| [(1, 5), (2, 3), (99, 3), (15, 1), (3, 1), (4, 2), (5, 5)] | 0.4821970539 | 0.500455717 | 0.4506441783 | 0.365305420 |

Bảng MAE:

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Độ tương đồng  Input(\*) | Cosine | PCC | SRC | KCC |
| [(1,5),(2,3)] | 0.828427508 | 0.982503062 | 0.982456214 | 0.98395545 |
| [(1, 5), (2, 3), (99, 3)] | 0.739198638 | 0.776280001 | 0.708659951 | 0.619183981 |
| [(1,5), (2,3), (99,3), (15,1)] | 0.582095922 | 0.597239261 | 0.542030451 | 0.4482414391 |
| [(1, 5), (2, 3), (99, 3), (15, 1), (3, 1), (4, 2), (5, 5)] | 0.407573271 | 0.425591340 | 0.369698292 | 0.2968220654 |

**Kết luận:** khi demo với dataset gồm 100 cuốn sách và 100 người dùng qua các phép thử trên ta thấy rằng sử dụng độ sai lệch SRC (Spearman Rank Coefficient) cho ra các kết quả dự đoán có sai lệch thấp nhất kế tiếp là KCC và cuối cùng là Cosine và PCC. Cùng với đó RMSE và MAE sẽ thay đổi tùy vào số lượng record, số lượng record càng lớn thì độ chính xác sẽ càng cao.

# CHƯƠNG 6: KẾT LUẬN VÀ HƯỚNG PHÁT TRIỂN

## 1. Tổng kết

Sau khi hoàn thành đề tài cơ bản nhóm đã đạt được những mục tiêu đề ra ban đầu. Nhóm đã hiểu và học được cơ về một hệ thống Recommendation gồm những gì và cách xây dựng ra sao, tìm hiểu và sử dụng Collaborative Filtering để đưa ra kết quả với tệp dữ liệu 100 người dùng và 100 sách.

## **2. Hạn chế**

Vẫn còn chưa hiểu sâu và chưa tìm hiểu được tất cả các phương pháp khác trong Recommendation. Chưa tích hợp được vào một hệ thống thực tế.

## 3. Hướng phát triển đề tài:

Nhóm sẽ cải thiện những nội dung sau:

* Tìm hiểu các phương pháp để xây dựng hệ thống Recommendation.
* Tích hợp vào hệ thống thực tế.
* Cải thiện hiệu suất của hệ thống.

# TÀI LIỆU THAM KHẢO

**[1]** “A New Similarity Measure Based on Mean Measure of Divergence for Collaborative Filtering in Sparse Environment”

**[2]** [“ Machine Learning cơ bản”](https://redux.js.org/) – Vũ Hữu Tiệp

**[3]** https://machinelearningcoban.com/

**[4]** https://viblo.asia/p/gioi-thieu-ve-he-thong-goi-y-recommender-systems-hoac-recommendation-systems-maGK78yOZj2