**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA HỆ THỐNG THÔNG TIN**

****

****

**ĐỀ XUẤT GIẢI PHÁP GIẢI QUYẾT HIỆN TƯỢNG “LONG TAIL” TRONG KINH DOANH**

**Giáo viên hướng dẫn: ThS. Thái Bảo Trân**

TP. Hồ Chí Minh ngày 10 tháng 4 năm 2018

**ĐẠI HỌC QUỐC GIA THÀNH PHỐ HỒ CHÍ MINH**

**ĐẠI HỌC CÔNG NGHỆ THÔNG TIN**

**KHOA HỆ THỐNG THÔNG TIN**

****

****

**ĐỀ XUẤT GIẢI PHÁP GIẢI QUYẾT HIỆN TƯỢNG “LONG TAIL” TRONG KINH DOANH**

**Sinh viên thực hiện:**

Nguyễn Văn Vinh 15521018

Vũ Văn Thái 15520784

**LỜI CẢM ƠN**

Đầu tiên, nhóm tác giả xin gửi lời cảm ơn chân thành đến toàn thể quý thầy cô Trường Đại học Công nghệ Thông tin- Đại học Quốc giá TP.HCM và quý thầy cô khoa Hệ thống Thông tin đã tận tình truyền đạt những kiến thức cơ bản làm nền tảng cho nhóm thực hiện đề tài này.

Đặc biệt, nhóm xin gửi lời cảm ơn tới cô Thái Bảo Trân đã trực tiếp hướng dẫn tận tình, sửa chữa và góp ý giúp nhóm hoàn thành nghiên cứu của mình.

Đồng thời, nhóm cảm ơn anh Trịnh Hoài Chương (KHMT – Đại học Bách Khoa HCM) đã góp ý, hỗ trợ nhóm trong quá trình thực hiện đề tài.

Trong thời gian thực hiện đề tài, nhóm đã vận dụng những kiến thức tiếp thu được kết hợp với việc học hỏi và nghiên cứu những kiến thức mới để hoàn thiện nghiên cứu một cách tốt nhất.

Tuy nhiên do hạn chế về mặt thời gian và kiến thức nên đề tài không tránh khỏi những thiếu sót. Vì thế, nhóm mong muốn nhận được những lời góp ý của Quí Thầy Cô để nhóm hoàn thiện kiến thức hơn, nhằm cải thiện nghiên cứu cũng như kiến thức của nhóm trong tương lai. Đó cũng là hành trang quan trọng để nhóm bước ra môi trường làm việc thực tế.

Xin chân thành cảm ơn!

**Nhóm tác giả,**

Nguyễn Văn Vinh, Vũ Văn Thái

**NHẬN XÉT CỦA GIẢNG VIÊN**

………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………………

**MỤC LỤC**

[**MỞ ĐẦU** 7](#_Toc511065110)

[**CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI** 1](#_Toc511065111)

[**1.1 Tên đề tài** 1](#_Toc511065112)

[**1.2 Mục tiêu đề tài** 1](#_Toc511065113)

[**1.3 Nội dung, phạm vi nghiên cứu** 1](#_Toc511065114)

[**1.3.1 Nội dung nghiên cứu** 1](#_Toc511065115)

[**1.3.2 Phạm vi nghiên cứu** 1](#_Toc511065116)

[**1.3.3 Kết quả dự kiến** 2](#_Toc511065117)

[**CHƯƠNG 2: CƠ SỞ LÝ THUYẾT** 3](#_Toc511065118)

[**2.1 Hệ thống khuyến nghị** 3](#_Toc511065119)

[**2.1.1 Giới thiệu về hệ thống khuyến nghị** 3](#_Toc511065120)

[**2.1.2 Ưu điểm của hệ thống khuyến nghị** 3](#_Toc511065121)

[**2.1.3 Vấn đề cần lưu ý trong hệ thống khuyến nghị** 3](#_Toc511065122)

[**2.2 Hướng tiếp cận trong hệ thống khuyến nghị** 3](#_Toc511065123)

[**2.2.1. Khuyến nghị dựa vào đặc tính sản phẩm (Content-Based)** 4](#_Toc511065124)

[**2.2.1.1 Ma trận Utility Matrix** 4](#_Toc511065125)

[**2.2.1.2 Xây dựng ma trận Utility Matrix** 5](#_Toc511065126)

[**2.2.1.3 Xây dựng Item Profile** 6](#_Toc511065127)

[**2.2.1.4 Xây dựng hàm mất mát.** 6](#_Toc511065128)

[**2.2.2. Khuyến nghị dựa vào sự tương quang (Colaborative- Filtering)** 8](#_Toc511065129)

[**2.2.2.1. Khuyến nghị dựa vào sự tương quan của người dùng (User-User Neighborhood-Based)** 9](#_Toc511065130)

[**2.2.2.1.1. Hàm tương quan (similarity fuctions)** 9](#_Toc511065131)

[**2.2.2.1.2. Hàm dự đoán (rating prediction)** 13](#_Toc511065132)

[**2.2.2.2. Khuyến nghị dựa vào sự tương quan của sản phẩm (Item-Item Neighborhood-Based)** 14](#_Toc511065133)

[**2.2.2.3 Phân tích ma trận thành nhân tử (Matrix Factorization Collaborative Filtering)** 15](#_Toc511065134)

[**2.2.3.1 Hàm mất mát (loss function)** 17](#_Toc511065135)

[**CHƯƠNG 3: Cài đặt, triển khai hệ thống** 18](#_Toc511065136)

[**3.1 Kiến trúc hệ thống** 18](#_Toc511065137)

[**3.2 Dữ liệu được sử dụng để thử nghiệm** 18](#_Toc511065138)

[**3.3 Yêu cầu chức năng** 21](#_Toc511065139)

[**3.3.1 Chức năng tính toán:** 21](#_Toc511065140)

[**3.3.2 Phi chức năng** 21](#_Toc511065141)

[**3.3.3 Chức năng của của việc sử dụng API** 21](#_Toc511065142)

[**3.4 Thông tin phiên bản** 23](#_Toc511065143)

[**3.5 Cài đặt hệ thống** 23](#_Toc511065144)

[**3.5.1 Cài đặt Python** 23](#_Toc511065145)

[**3.5.2 Cài đặt Nodejs** 23](#_Toc511065146)

[**CHƯƠNG 4: KẾT QUẢ ĐẠT ĐƯỢC** 24](#_Toc511065147)

[**4.1. Kết quả đạt được** 24](#_Toc511065148)

[**4.2. Khó khăn và hạn chế** 24](#_Toc511065149)

[**4.3. Kết luận:** 25](#_Toc511065150)

[**4.3.1 Kết quả và nhận xét về Content-Based khi áp dụng vào bộ dữ liệu.** 25](#_Toc511065151)

[**4.3.2 Kết quả và nhận xét User-User Neighborhood khi áp dụng vào bộ dữ liệu.** 26](#_Toc511065152)

[**4.3.3 Kết quả và nhận xét User-User Neighborhood khi áp dụng vào bộ dữ liệu.** 27](#_Toc511065153)

[**4.3.4 Kết quả và nhận xét Matrix Factorization khi áp dụng vào bộ dữ liệu.** 27](#_Toc511065154)

[**4.4. Hướng phát triển** 28](#_Toc511065155)

# **MỞ ĐẦU**

Ngày nay, khi công nghệ thông tin ngày càng phát triển, các cửa hàng thực tế dường như không thể đáp ứng được nhu cầu của các công ty hay doanh nghiệp khi không gian trưng bày của sản phẩm thì có hạn nhưng mẫu mã cũng như chủng loại sản phẩm thì không ngừng tăng lên gây ra vấn đề nan giải về không gian trưng bày cho doanh nghiệp. Từ đó các trang website bán hàng online, các trang thương mại điện tử được mở ra để tạo ra một không gian trưng bày gần như vô tận cho việc trưng bày sản phẩm.

Tuy nhiên, lại có vấn đề phát sinh khi hầu hết doanh thu lại tập trung vào số ít các sản phẩm, các sản phẩm còn lại có doanh thu rất ít hoặc gần như bằng không. Đây chính là hiện tượng “long tail” (đuôi dài) trong kinh doanh. Nguyên nhân chủ yếu gây ra việc này là do các mặt hàng phù hợp với từng người dùng không đến được với tay người dùng. Vì các sản phẩm được thấy nhiều là những sản phẩm mang tính chất phổ biến, nhiều người cùng dùng hoặc thích. Và các sản phẩm này không chắc sẽ phù hợp với từng người dùng do nhu cầu về màu sắc, kích thước hay thể loại khác nhau. Ngoài ra việc có quá nhiều sản phẩm làm người dùng cảm thấy không thể tìm kiếm được sản phẩm mình mong muốn mặc dù sản phẩm này tồn tại trong kho.

Đó là lý do để nhóm thực hiện đề tài: “Đề xuất giải pháp giải quyết hiện tượng “long tail” (đuôi dài) trong kinh doanh. Với mong muốn có thể đưa những sản phẩm phù hợp nhất đến với tay khách hàng. Đồng thời có thể nâng doanh thu của các trang thương mại điện tử nhằm hạn chế được chuỗi hàng hóa, sản phẩm có doanh thu gần như bằng không ở phía sau.

Báo cáo gồm 4 chương:

Chương 1: Tổng quan đề tài

Giới thiệu mục tiêu, ý nghĩa, đối tượng, phạm vi và phương pháp nghiên cứu đề tài.

Chương 2: Cơ sở lý thuyết

Trình bày các cơ sở lý thuyết của đề tài bao gồm khái niệm về hệ thống khuyến nghị, các thuật toán liên quan cũng như các công cụ để xây dựng được ứng dụng.

Chương 3: Cài đặt, triển khai hệ thống

Trình bày về cấu trúc và cách cài đặt hệ thống.

Chương 4: Kết quả đạt được

Tổng kết các kết quả đạt được, xác định được hạn chế của hệ thống. Đồng thời có thể đưa ra các hướng phát triển hệ thống trong tương lai

# **CHƯƠNG 1: TỔNG QUAN ĐỀ TÀI**

**1.1 Tên đề tài**

Đề xuất giải pháp giải quyết hiện tượng “long tail” trong kinh doanh.

**1.2 Mục tiêu đề tài**

* Nghiên cứu và ứng dụng một số thuật toán và xây dựng được mô hình minh họa.
* Các bước thực hiện:

+ Bước 1: Thu thập và đánh giá dữ liệu

+ Bước 2: Nghiên cứu phương pháp tiếp cận, phân tích thuật toán

+ Bước 3: Đánh giá về điểm mạnh và yếu của từng thuật toán

+ Bước 4: Xây dựng mô hình

**1.3 Nội dung, phạm vi nghiên cứu**

**1.3.1 Nội dung nghiên cứu**

-Tìm hiểu về hệ thống khuyến nghị

- Tìm hiểu về các thuật toán có thể áp dụng trong hệ thống khuyến nghị

- Xây dựng website hỗ trợ tương tác người dùng.

**1.3.2 Phạm vi nghiên cứu**

Do thời gian không cho phép nên nhóm tập trung vào phân tích những vấn đề sau:

* Các thuật toán có thể áp dụng vào hệ thống khuyến nghị bao gồm:

+ Phân tích đặc tính sản phẩm

+ Phân tích dựa vào đặc điểm của sản phẩm hoặc người dùng

* Tìm hiểu và triển khai mô hình bằng ngôn ngữ Python.
* Đối tượng dữ liệu được sử dụng: Phim
* Đối tượng khuyến nghị: Người dùng có trong hệ thống
* Nội dung khuyến nghị: dự đoán phim phù hợp với người dùng.

**1.3.3 Kết quả dự kiến**

**-** Áp dụng được các phương pháp tiêu biểu của hệ thống khuyến nghị

**-** Áp dụng vào tập dữ liệu có sẵn với thuật toán phù hợp

**-** Xây dựng được mô hình:

**+ Đầu vào:** Người dùng chọn bộ dữ liệu đã được chuẩn bị (ở đây chúng ta xem như các bước tiền xử lý dữ liệu đã được làm xong). Đồng thời nhập các đặc tính của sản phẩm

**+ Đầu ra:** khuyến nghị các phim phù hợp đến người dùng.

# **CHƯƠNG 2: CƠ SỞ LÝ THUYẾT**

**2.1 Hệ thống khuyến nghị**

**2.1.1 Giới thiệu về hệ thống khuyến nghị**

Hệ thống khuyến nghị (Recommendation System) là một hệ thống tập hợp các kỹ thuật xử lý dữ liệu đã có sẵn để gợi ý cho người dùng trong hệ thống.

Các ứng dụng nổi tiếng: Youtube, Facebook, Lazada, Tiki, …

**2.1.2 Ưu điểm của hệ thống khuyến nghị**

- Tăng được số lượng sản phẩm bán ra so với các hệ thống không được hỗ trợ hệ thống khuyến nghị nhờ vào việc phân tích được các đặc tính tiềm ẩn liên quan giữa các sản phẩm hoặc người dùng.

- Tăng được sự hài lòng của khách hàng đối với doanh nghiệp khi hệ thống có thể đáp ứng được nhu cầu về sản phẩm, đồng thời có thể đem những sản phẩm tốt nhất, phù hợp nhất đến tay người dùng.

**2.1.3 Vấn đề cần lưu ý trong hệ thống khuyến nghị**

Dữ liệu: Đây là vấn đề quan trọng nhất trong dữ liệu, có một nguyên tắc trong phân tích dữ liệu “garbage in garbage out”, nguồn dữ liệu cần được đảm bảo về độ chính xác cũng như mức độ “sạch”. Đồng thời, nguồn dữ liệu phải đảm bảo được các tính chất cần thiết của một bộ dữ liệu chuẩn: Khối lượng dữ liệu phải đủ lớn, tốc độ phát sinh dữ liệu, mức độ đa dạng của dữ liệu, độ tin cậy của dữ liệu và giá trị của dữ liệu.

Hành vi người dùng: nếu người dùng liên tục thay đổi hành vi sẽ ảnh hưởng đến kết quả của hệ thống khuyến nghị

**2.2 Hướng tiếp cận trong hệ thống khuyến nghị**

Các nhóm chính của hệ thống recommendation:

**Content-based systems:** đánh giá dựa trên các đặc tính của sản phẩm được khuyến nghị.

Ví dụ: một khách hàng muốn mua áo sơ mi cổ tàu. Như vậy hệ thống sẽ gợi ý thêm những mẫu khác của loại sản phẩm đó như áo sơ mi nhưng không phải cổ tàu hoặc là áo cổ tàu nhưng không phải áo sơ mi. Tuy nhiên đối với một số loại sản phẩm khó xác định rõ ràng các đặc trưng thì thuật toán này trở nên bất khả thi.

**Collaborative filtering:** đánh giá dựa trên sự tương quan giữa các sản phẩm với nhau hoặc giữa các khách hàng với nhau.

Ví dụ: khách hàng A B thích áo sơ mi cổ tàu, đồng thời khách hàng A thích áo thun cổ tàu nhưng không biết B có thích hay không, nhưng vì A B đều có những đặc tính tương quan với nhau nên hệ thống sẽ đoán B cũng thích áo thun cổ tàu và sẽ gợi ý cho khách hàng B.

**2.2.1. Khuyến nghị dựa vào đặc tính sản phẩm (Content-Based)**

Với nhóm thuật toán thứ nhất này, chúng ta quan tâm đến sự đánh giá của user dành cho items, mức độ quan tâm này được đánh giá thông qua một giá trị, tạm gọi là rating. Tập hợp mức độ đánh giá này, kể cả những giá trị chưa biết được gọi là Utility Matrix.

#### **2.2.1.1 Ma trận Utility Matrix**

Xét ví dụ về utility matrix với: hệ thống “Gợi ý bài hát”. Các bài hát được người dùng đánh giá theo mức độ từ 0 đến 5 sao. Các dấu '?' ứng với việc dữ liệu chưa tồn tại trong cơ sở dữ liệu. Recommendation Systems cần phải tự điền các giá trị này.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E | F |
| Vợ người ta | 5 | 5 | 0 | 0 | 1 | ? |
| Lạc trôi | 5 | ? | ? | 0 | ? | ? |
| Người lạ ơi | ? | 4 | 1 | ? | ? | 1 |
| Đắp mộ cuộc tình | 1 | 1 | 4 | 4 | 4 | ? |
| Duyên phận | 1 | 0 | 5 | ? | ? | ? |

Hình 1: Ví dụ về utility matrix với hệ thống “Gợi ý bài hát”.

Trong ví dụ được trình bày ở trên, ta thấy rằng A, B, C, D, E, F và 5 bài hát. Các ô có số được điền vào thể hiện mức độ “yêu thích” từ 0 (không thích) đến 5 (rất thích) của của user lên item. Các ô màu xám là các ô chưa xác định được và công việc cần làm của hệ thống khuyến nghị là cố gắng điền được số thích hợp vào dầu “?”. Ta có thể xem đây là bài toán hoàn thiện ma trận (Matrix Completion).

Từ ví dụ đơn giản nêu trên, 3 bài hát đầu là nhạc trẻ, 2 bài hát sau là nhạc bolero. Ta dễ dàng nhận thấy được A và B thích nhạc trẻ và C, D, E, F thích thể loại nhạc bolero. Từ đó, nếu một hệ thống khuyến nghị tốt sẽ gợi ý bài “Lạc trôi” cho A và “Người lạ ơi” cho B. Tương tự cho những users khác. Khi có một bài hát mới được thêm vào, ta chỉ cần xác định thể loại của nó thì hệ thống có thể gợi ý bài hát đó đến người dùng phù hợp.

Nếu càng nhiều ô được điền thì hệ thống sẽ càng gợi ý chính xác.Tuy nhiên, trong thực tế, không phải ai cũng sẵn sàng đánh giá sản phẩm của mình đã mua nên ma trận Utility trong các mô hình thực tế thường trống rất nhiều, đôi khi có những users không rate(đánh giá) cho bất cứ sản phẩm nào. Điều này gây ra nhiều khó khăn cho hệ thống.

#### **2.2.1.2 Xây dựng ma trận Utility Matrix**

Ma trận Utility matrix là vô cùng quan trọng trong hệ thống khuyên nghị vì nếu không có, gần như không thể gợi ý được sản phẩm đến được người dùng. Có 2 cách để xây dựng được ma trận

* Nhờ người dùng đánh giá sản phẩm, cách này có hạn chế vì người dùng thường ít khi đánh giá sản phẩm. Đồng thời cách này không được khách quan vì đó quan điểm của từng người
* Hướng khác là dựa vào hành vi của khách hàng, giống như việc Facebook thường làm là hay giới thiệu những tin mới của các trang mà ta hay like, đôi khi nút “dislike” cũng là một cách để chúng ta dễ dàng xác định được hành vi của khách hàng.

#### **2.2.1.3 Xây dựng Item Profile**

Trong hệ thống Content-Based, chúng ta cần xây dựng một bộ hồ sơ cho mỗi item. Bộ hồ sơ này biểu diễn dưới dạng toán học là một feture vecto. Trong một số trường hợp khó xác định cụ thể, nếu không thường thì các yếu tố để sử dụng làm vecto được xác định một cách rõ ràng. Và với vecto được tạo ra, ta cần tìm mô hình phù hợp cho mỗi users với item đó. Nếu rating là một dải các giá trị thì đây sẽ là bài toán regression, nếu rating là các điểm thì đây là bài toàn classification

Xét ví dụ được nếu trên:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | A | B | C | D | E | F | Item’s feature vectos |
| Vợ người ta | 5 | 5 | 0 | 0 | 1 | ? | x =[0.97, 0.02] |
| Lạc trôi | 5 | ? | ? | 0 | ? | ? | x =[0.91, 0.13] |
| Người lạ ơi | ? | 4 | 1 | ? | ? | 1 | x =[0.92, 0.06] |
| Đắp mộ cuộc tình | 1 | 1 | 4 | 4 | 4 | ? | x =[0.01, 0.98] |
| Duyên phận | 1 | 0 | 5 | ? | ? | ? | x =[0.05, 0.91] |
| User’s model | ? | ? | ? | ? | ? | ? |  |

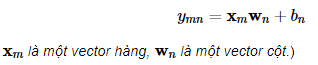
Hình 2: Giả sử feature vector cho mỗi *item* được cho trong cột cuối cùng. Với mỗi *user*, chúng ta cần tìm một mô hình tương ứng sao cho mô hình thu được là *tốt nhất*.

Trong ví dụ ở hình 1, chúng ta đơn giản hóa bài toán bằng việc xây dựng một feature vecto 2 chiều: nhạc trẻ và bolero. Giả sử các feature vecto cho mỗi bài hát được cho sẵn như trong hình 2. Mục tiêu cần đạt là xác định được các mô hình ở dâu “?” sao cho mô hình tìm được tốt nhất có thể.

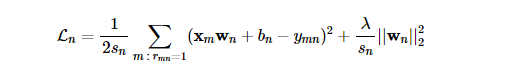
#### **2.2.1.4 Xây dựng hàm mất mát.**

Ma trận Utility U được xây dựng bởi N user và M item. Thành phần nằm ở dòng n, cột m chính là rating của user n dành cho item m mà hệ thống ghi nhận được. Trong thực tế, ma trận này bị khuyết rất nhiều giá trị tương ứng với đánh giá của user dành cho item. Một vài trường hợp, ta đặt them ma trân R là ma trân thể hiện việc một user đã rate cho item hay chưa được biểu diễn bởi 1 là đã rate và 0 là chưa rate.

Giả sử mô hình tìm được cho mỗi user được minh họa bởi một hàm tuyến tính, lấy một hàm tuyến tính đơn giản:



Nếu ta xét một user bất kỳ n và xem tập dữ liệu training set là tập hợp những thành phần đã được điền của  ta có thể xây dựng hàm mất mát như sau:

(1)

(1) : Công thức được xây dựng dựa trên hàm mất mát của Ridge Regression được tham khảo của *Andrew Ng’s course*

Trong đó: là tham số dương

 là số lượng các item mà user thứ n hay nói cách khác, chính là tổng các phần tử cột thứ n của ma trận R

Ngoài ra, vì hàm mất mát này chỉ phụ thuộc vào các item đã được rated, nên ta có thể rút gọn lại bằng cách đặt  là vecto của y bằng cách trích các thành phần khác dấu “?” ở cột thứ n, tức đã được rate bởi user n trong Utility Matrix Y. Đặt là ma trận của vecto X, được tạo thành bằng cách trích các hàng tương ứng với các item đã được rate bởi user n. Khi đó hàm mất mát sẽ còn được viết gọn lại theo một cách khác nữa l:



Trong đó:  là vecto cột chứa  phần tử 1

Cặp nghiệm ( ) có thể tìm được thông qua Stochastic Gradient Descent (SGD) hoặc là Mini-bacth (GD). Tuy nhiên, hiện nay một số thư viện đã hỗ trợ tính toán cho các thuật toán như thế này và một trong số đó là class Ridege trong sklearn.linear\_model.

Tổng kết lại, ta thấy Content-Based là một phương pháp nền tảng và cơ bản nhất trong hệ thống recommendation system. Và mục đích chủ yếu của phương pháp này là xây dựng một mô hình cho mỗi user sao cho không phụ thuộc vào user khác. Việc xây dựng mô hình này có thể được coi là regression hoặc classification với dữ liệu mà đã được huấn luyện. Hồ sơ của sản phẩm (Item profile) không phụ thuộc vào user mà phụ thuộc vào đặc tính của user hoặc có thể là do người dùng gắn tag.

**2.2.2 Khuyến nghị dựa vào sự tương quang (Colaborative- Filtering)**

Ở nhóm thứ nhất, Content-Based chủ yếu dựa vào các đặc tính của item để gợi ý cho khách hàng và việc xây dựng mô hình cho mỗi user không phụ thuộc vào các user khác mà chỉ phụ thuộc vào đặc tính của item. Việc này có thể tiết kiệm được bộ nhớ và thời gian tính toán cho hệ thống. Content-Based đã tận dụng được các thông tin đặc tính của item trong bản mô tả của nhà cung cấp hoặc từ việc gắn tag hoặc label cho item

Tuy nhiên, Content-Based lại có những nhược điểm riêng. Đầu tiên là hệ thống sẽ không tận dụng được thông tin từ các user khác. Thông thường, các thông này rất hữu ích cho việc phân tích sở thích của người dùng vì hành vi mua hàng của user thường có xu hướng được nhóm thành các nhóm đơn giản và chỉ cần biết được hành vi của một vài user, hệ thống có thể dự đoán được hành vi của các user còn lại. Và ngoài ra, việc nhờ user gắn tag cho item không phải là việc làm đơn giản vì không phải user nào cũng sẵn sàng làm việc này và sẽ mang tính chủ quan cho các đặc tính của item.

Những nhược điểm được nêu ở trên có thể được giải quyết bằng Collaborative Filtering (CF). Ý tưởng cơ bản của cách làm này là xác định sự quan tâm của một user lên một item dựa vào một user khác gần giống với user cần xác định. Việc xác định gần giống này thông qua mực độ quan tâm của user này tới item khác mà hệ thống đã ghi nhận được. Ta có thể lấy ví dụ như sau: A và B đều thích dạng áo cổ tàu và đã rate cho item này 5 sao, và A thích áo thun cổ tàu, nhiều khả năng B cũng thích áo thun cổ tàu.

Có 2 câu hỏi quan trọng cần được xác định trong hệ thống CF:

* Một là làm thế nào để xác định được sự giống nhau trong hành vi hay nói cách khác là sự giống nhau giữa các user
* Hai là khi đã xác định được sự giống nhau giữa các user làm thế nào để dự đoán được mức độ quan tâm của một user lên một item chưa được rate.

Việc xác định mức độ quan tâm của mỗi user tới một item dựa vào mức độ quan tâm của các user gần giống được gọi là user-user collaborative filtering. Ngoài ra, còn có một hướng khác là dựa vào item-item collaborative filtering, hướng tiếp cận này thay vì dùng đến những user gần giống nhau thì dùng đến những item gần giống nhau. Mặc dù đều xuất phát dựa vào sự giống nhau giữa các đối tượng quan trọng trong Utility Matrix nhưng item-item lại tỏ ra có lợi thế hơn so với user-user.

#### **2.2.2.1 Khuyến nghị dựa vào sự tương quan của người dùng (User-User Neighborhood-Based)**

##### **2.2.2.1.1 Hàm tương quan (similarity fuctions)**

Công việc quan trọng trong thuật toán này là phải xác định được sự giống nhau giữa các users với nhau. Dữ liệu duy nhất chúng ta có được là ma trận Utility. Vậy nên sự tương quan phải được xác định bằng các cột trong ma trận. Ta xét ví dụ sau để dễ dàng quan sát hơn:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | U0 | U1 | U2 | U3 | U4 | U5 | U6 |
| I0 | 5 | 5 | 2 | 0 | 1 | ? | ? |
| I1 | 4 | ? | ? | 0 | ? | ? | ? |
| I2 | ? | 4 | 1 | ? | ? | 1 | 2 |
| I3 | 2 | 2 | 3 | 4 | 4 | 4 | 4 |
| I4 | 2 | 0 | 4 | ? | ? | ? | 5 |

Hình 3: Ví dụ về utility matrix dựa trên số sao một user rate cho một item. Một cách trực quan, hành vi của U0 giống với U1 hơn là U2, U3, U4, U5, U6. Từ đó có thể dự đoán rằng U0 sẽ quan tâm tới I2 vì U1 cũng quan tâm tới I2.

Trong ví dụ trên, U0 đến U6 là các users và I0 đến I4 là các items, mỗi ô vuông thể hiện đánh giá của users lên item đó. Các dấu “?” là các đánh giá còn thiếu của uses lên item. Ta đặt cho sự giống nhau giữa 2 users là sim (Ui, Uj). Quan sát trên ví dụ ta có thể thấy được U0 và U1 tương đối thích I0, I1, I2 và không thích I3, I4. Và điều ngược lại xảy ra với nhóm còn lại. Vậy nên nếu một hàm similarity tốt cần đảm bảo

Sim (U0, U1) > Sim(U0, Ui) với i > 1

Để đo được similarity giữa 2 users, ta thường xây dựng feature vecto rồi dùng một hàm gián tiếp để đo sự giống nhau giữa 2 vecto đó. Việc xây dựng feature vecto này khác với thuật toán Content-Based vì feature vecto được xây dựng dựa vào ma trận Utility chứ không phải từ dữ liệu bên ngoài. Khó khăn ở đây là trong ma trận Utility thường có rất nhiều khoảng trống không có dữ liệu.

Vậy nên ta nên thay mỗi dấu “?” bằng một giá trị tương đối để hạn chế việc sai lệch quá lớn. Có nhiều cách có thể thực hiện ở đây như lấy giá trị ở giữa, nhưng điều này lại gây ra vấn đề với những users dễ tính và users khó tính vì đặc tính của 2 users này quyết định nhiều đến số sao được rate, users dễ tính thường rate từ 3 đến 5 sao cho các sản phẩm từ không thích đến thích nhưng đối với users khó tính thì thường từ 1 đến 3 sao cho điều tương tự. Vậy nên có một cách tốt hơn là lấy trung bình rating của users đó lên sản phẩm, cách làm này sẽ tránh được việc nêu trên.

Bước cần làm tiếp theo là chuẩn hóa dữ liệu: ta vẫn xét ví dụ ở trên

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | U0 | U1 | U2 | U3 | U4 | U5 | U6 |
| I0 | 5 | 5 | 2 | 0 | 1 | ? | ? |
| I1 | 4 | ? | ? | 0 | ? | ? | ? |
| I2 | ? | 4 | 1 | ? | ? | 1 | 2 |
| I3 | 2 | 2 | 3 | 4 | 4 | 4 | 4 |
| I4 | 2 | 0 | 4 | ? | ? | ? | 5 |
| Đánh giá trung bình | 3.25 | 2.75 | 2.5 | 1.33 | 2.5 | 2.5 | 3.6 |

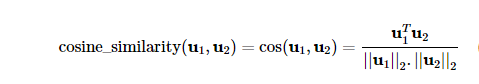
Hình 4:Ví dụ về đánh giá của user dành cho item.

Khi có được các giá trị trung bình, ta lấy các giá trị trừ đi giá trị trung bình và thay các dấu “?” bằng 0 thì ta có ma trận chuẩn hóa (normalized utility matrix). Việc làm này khá quan trọng vì ta có thể xác định được users thích sản phẩm nào hoặc không thích sản phẩm nào, số 0 tượng trưng cho việc chưa xác định được. Ngoài ra, về kĩ thuật, số lương users và items rất lớn nên sẽ rất khó khăn trong vấn đề về bộ nhớ nên chúng ta nên lưu ma trân này dưới dạng sparse matrix (chỉ lưu những giá trị khác 0). Nhờ vậy việc tính toán cũng như vấn đề về bộ nhớ sẽ được giải quyết tốt hơn.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | U0 | U1 | U2 | U3 | U4 | U5 | U6 |
| I0 | 1.75 | 2.25 | -0.5 | -1.33 | -1.5 | 0 | 0 |
| I1 | 0.75 | 0 | 0 | -1.33 | 0 | 0 | 0 |
| I2 | 0 | 1.25 | -1.5 | 0 | 0 | -1.5 | -1.6 |
| I3 | -1.25 | -0.75 | 0.5 | 2.67 | 1.5 | 1.5 | 0.4 |
| I4 | -1.25 | -2.75 | 1.5 | 0 | 0 | 0 | 1.4 |

Hình 5: Utility Matrix đã được chuẩn hoá

Sau khi có được ma trận chuẩn hóa, tiếp theo chúng ta sử dụng một trong các hàm similiraty function để tính sự tương quan giữa 2 vecto với nhau. Một trong số đó hay được sử dụng nhất là cosine similiraty và cũng khá quen thuộc khi được nhắc đên vì đã được tiếp xúc từ những năm học trung học cơ sở.

Công thức: 

Trong đó:  là vectors tương ứng với users 1, 2 đã được chuẩn hóa.Ở đây chúng ta sử một hàm được viết sẵn trong thư viện python để tính toán và có kết quả như ở bảng dưới

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | U0 | U1 | U2 | U3 | U4 | U5 | U6 |
| U0 | 1 | 0.77 | -0.408 | -0.707 | -0.816 | -0.408 | -0.408 |
| U1 | 0.77 | 1 | -0.707 | -0.272 | -0.471 | -0.236 | -0.707 |
| U2 | -0.408 | -0.707 | 1 | 0 | 0 | 0.5 | 1 |
| U3 | -0.707 | -0.272 | 0 | 1 | 0.866 | 0.577 | 0 |
| U4 | -0.816 | -0.471 | 0 | 0.866 | 1 | 0.5 | 0 |
| U5 | -0.408 | -0.236 | 0.5 | 0.577 | 0.5 | 1 | 0.5 |
| U6 | -0.408 | -0.707 | 1 | 0 | 0 | 0.5 | 1 |

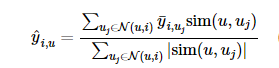
Hình 6: Ma trận users similarity matrix

Độ giống nhau này nằm trong khoảng giá trị từ [-1, 1] (hàm cos có giá trị từ -1 đến 1). Nếu giá trị giữa 2 vector này bằng 1 chứng tỏ hoàn toàn giống nhau, ngược lại đối với -1. Từ bảng giá trị nêu trên, ta có thể rút ra được một vài nhận xét sau:

* U0 và U1, U6 và U2, ngoài ra còn 1 vài cặp nữa tương đối giống nhau khi có chỉ số tương đối là dương. Và điều này cũng hợp lý khi nhìn vào ma trận đã được chuẩn hóa. Điều này chứng tỏ ta đang tiếp cận đúng hướng.
* U0 gần U1 và xa hầu hết các users còn lại

##### **2.2.2.1.2 Hàm dự đoán (rating prediction)**

Số điểm dự đoán thường được sẽ được xác định là trung bình có trọng số của rating đã được chuẩn hóa. Công thức phổ biến và hay được dùng để tính toán là:



Trong đó:  là tập hợp số users trong neighborhood của u đã rate i.

Ở đây chúng ta sẽ tính ví dụ về điểm đánh giá của một users dựa vào đánh giá của users khác. Chúng ta sử dụng giá trị neighborhood ở đây là k =2, đây là lựa chọn thường được sử dụng và đưa ra dự đoán khá tốt cho kết quả.

Tính normalized rating của U0 cho I2 với k =2,

* Ta xác định được users đã rate cho I2 là U1, U2 và U5
* Xác định similarity của U0 với các users đã rate cho I2 là U1, U2, U5, ta có các giá trị 0.77, -0.408, -0.408. K =2 nên ta lấy 2 giá trị lớn nhất từ 3 giá trị tìm được. Ở đây có 2 giá trị giống nhau nên ta có thể chọn bất kì giá trị nào
* Ta xét tiếp giá trị của normalized rating của U1, U2 cho I2 là 1.25 và -1.5
* Áp dụng công thức ta có kết quả dự đoán là :1.33

Việc quy đổi về giá trị đúng bằng cách cộng kết quả dự đoán với đánh giá trung bình sẽ ra được kết quả cuối cùng. Ta có bảng giá trị dưới sau khi hoàn thành tất cả các phép dự đoán.

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | U0 | U1 | U2 | U3 | U4 | U5 | U6 |
| I0 | 5 | 5 | 2 | 0 | 1 | 1.5 | 3.6 |
| I1 | 4 | 3 | 2.5 | 0 | 1.986 | 1.914 | 3.6 |
| I2 | 4.25 | 4 | 1 | 0.33 | 1.5 | 1 | 2 |
| I3 | 2 | 2 | 3 | 4 | 4 | 4 | 4 |
| I4 | 2 | 0 | 4 | 1.33 | 2.5 | 3.5 | 5 |

Hình 6: Hoàn thiện ma trận sau khi thực hiện các bước trên

#### **2.2.2.2. Khuyến nghị dựa vào sự tương quan của sản phẩm (Item-Item Neighborhood-Based)**

Cách tiếp cận thuật toán này tương tự như cách tiếp cận thuật toán User- User, nhưng chúng ta có thể nhận thấy được số lượng sản phẩm tăng lên khó có thể bằng số lượng người dùng tăng. Và vì thế nên các vấn đề về bộ ở thuật toán User-User sẽ được giảm bớt khi dùng đối tượng là Item.

Ở đây, do hầu hết cách phân tích đều giống nhau chỉ khác nhau đối tượng phân tích nên các bước sẽ được trình bày ngắn gọn bằng hình ảnh. Chúng ta tiếp tục sử dụng ví dụ ở trên để có thể so sánh sự khác nhau khi thay đổi đối tượng được phân tích.

Sau khi tính các giá trị trung bình:

|  |  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- | --- |
|  | U0 | U1 | U2 | U3 | U4 | U5 | U6 | Giá trị TB |
| I0 | 5 | 5 | 2 | 0 | 1 | ? | ? | 2.6 |
| I1 | 4 | ? | ? | 0 | ? | ? | ? | 2 |
| I2 | ? | 4 | 1 | ? | ? | 1 | 2 | 2 |
| I3 | 2 | 2 | 3 | 4 | 4 | 4 | 4 | 3.29 |
| I4 | 2 | 0 | 4 | ? | ? | ? | 5 | 2.75 |

Chuẩn hóa dự liệu:

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | U0 | U1 | U2 | U3 | U4 | U5 | U6 |
| I0 | 2.4 | 2.4 | -0.6 | -2.6 | -1.6 | 0 | 0 |
| I1 | 2 | 0 | 0 | -2 | 0 | 0 | 0 |
| I2 | 0 | 2 | -1 | 0 | 0 | -1 | 0 |
| I3 | -1.29 | -1.29 | -0.29 | 0.71 | 0.71 | 0.71 | 0.71 |
| I4 | -0.75 | -2.75 | 1.25 | 0 | 0 | 0 | 2.25 |

Ma trận similarity:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
|  | I0 | I1 | I2 | I3 | I4 |
| I0 | 1 | 0.784 | 0.453 | -0.784 | -0.37 |
| I1 | 0.784 | 1 | 0 | -0.5 | 0 |
| I2 | 0.453 | 0 | 1 | -0.577 | -0.68 |
| I3 | -0.784 | -0.5 | -0.577 | 1 | 0.471 |
| I4 | -0.37 | 0 | -0.68 | 0.471 | 1 |

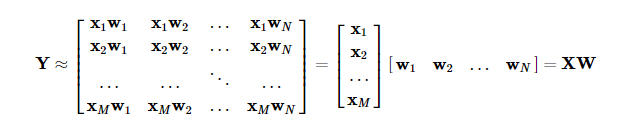
Ở đây, khi quan sát chúng ta sẽ thấy nhóm được in xanh là nhóm không âm, nhóm còn lại hầu như là âm, vậy vô tình chúng ta đã phân loại được các nhóm và thực hiện được việc item clustering, điều này khi ứng dụng vào thực tế mang lại kết quả vô cùng lớn cho việc phân tích về sau.

Cuối cùng là bảng dự đoán kết quả:

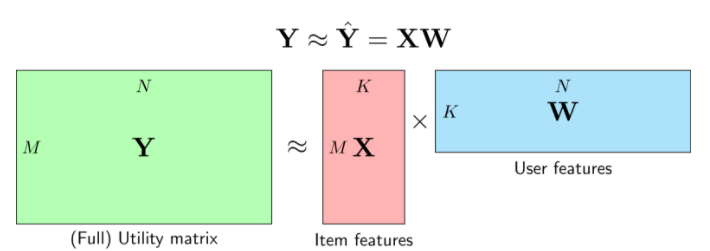
|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
|  | U0 | U1 | U2 | U3 | U4 | U5 | U6 |
| I0 | 5 | 5 | 2 | 0 | 1 | 2.97 | 1.7 |
| I1 | 4 | 3 | 2 | 0 | 1.39 | 2 | 2 |
| I2 | 3 | 4 | 1 | 1 | 1.56 | 1 | 2 |
| I3 | 2 | 2 | 3 | 4 | 4 | 4 | 4 |
| I4 | 2 | 0 | 4 | 2.75 | 3.19 | 3.34 | 5 |

#### **2.2.2.3 Phân tích ma trận thành nhân tử (Matrix Factorization Collaborative Filtering)**

Ở thuật toán Content-Based, mỗi item sẽ được mô tả bằng một vector item profile, sau đó chúng ta đi tìm một mô hình của từng user phù hợp với item profile đó. Tuy nhiên, cách làm này chưa thật sự tốt vì không tận dụng được sự tương quan giữa các users với nhau



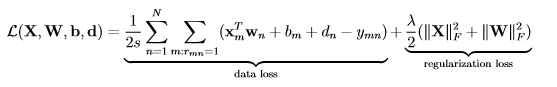
Với thuật toán Matrix Factorization, chúng ta không cần thiết phải xây dựng feature vecto một cách độc lập với mô hình. Chúng ta có thể huấn luyện đồng thời giữa feature vector và mô hình của mỗi user với mỗi item. Điều này có nghĩa là biến số của trong bài toán tối ưu là cả X và W thay vì mỗi W như ở thuật toán Content-Based. Trong đó X là ma trận của toàn bộ item profile và W là ma trận của toàn bộ user model.



Ý tưởng đằng sau thuật toán này là tồn tại các đặc trưng ẩn mô tả sự liên quan giữa các users và các item. Giống như việc một chiếc áo gồm nhiều tính chất ẩn như áo sơ mi, cổ tàu, vải cotton, …. Và mỗi tính chất ẩn mang một hệ số trong feature vecto, hệ số càng cao có nghĩa là item mang tính chất đó sẽ càng rõ. Điều tương tự cũng diễn ra với users

Thuật toán này được xếp vào dạng Collaborative-Filtering vì chúng ta đi tìm một biến số khi biến số kia đã được xác định, nghĩa là mỗi dòng của X(item) phụ thuộc chặt chẽ vào mỗi cột của W (user). Chính vì chúng có quan hệ chặt chẽ với nhau nên được xếp vào dạng nêu trên. Và điều quan trọng trong thuật toán này là hàm mất mát vì thuật toán này dựa trên việc “học” là chủ yếu thay vì phân tích độ giống nhau giống như 2 thuật toán trước

Hàm mất mát (loss function)có thể được xây dựng như sau:

**(2)**

(2): Hàm mất mát được tham khảo từ ebook *Vu Huu Tiep, Ebook “Machienlearning cơ bản”*

Trong đó:  = 1 nếu item thứ m đã được đánh giá bởi user n, s là số lượng rating,

* , lần lượt là bias của item m và user n, data loss là trung bình sai số của mô hình, thành phần thứ 2 là regularization loss, là một thành phần thêm vào để tránh hiện tượng overfitting.

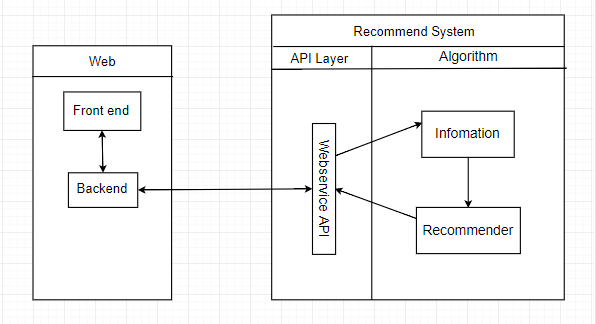
Tuy nhiên, việc tối ưu đồng thời W, X, b, d là tương đối phức tạp nên thay vào đó ta tối ưu từng cặp khi cố định cặp còn lại. Quá trình này cứ liên tục cho đến khi hàm được hội tụ.

Chú thích:

* Hệ số bias: là hệ số thể hiện sự đánh giá từ phía người dùng, tồn tại độc lập và có quan hệ chủ quan với người dùng.
* Overfitting: là hiện tượng xảy ra khi mô hình quá chính xác gây nên việc mô hình không thật sự tốt khi có dữ liệu mới.

# **CHƯƠNG 3: CÀ ĐẶT, TRIỂN KHAI HỆ THỐNG**

**3.1 Kiến trúc hệ thống**



* Hệ thống được chia làm 2 module:

+ Website tương tác giữa người dùng và hệ thống

+ Hệ thống khuyến nghị

* Website tương tác:

+ Front end: được phát triển bằng AngularJS

+ Back end được phát triển bằng ngôn ngữ lập trình Python

* Hệ thống khuyến nghị:

+ API layer được phát triển bằng ngôn ngữ Python

+ Algorithm được phát triển bằng ngôn ngữ lập trình Python, sử dụng thư viện hỗ trợ sklearn

**3.2 Dữ liệu được sử dụng để thử nghiệm**

MovieLens 100klà bộ dữ liệu được công bố vào năm 1998 do nhóm GroupLens thực hiện. Bộ dữ liệu này có 100,000 rating từ 943 users cho 1682 bộ phim. Bộ dữ liệu gồm nhiều tập tin, ở đây chúng ta sử dụng một trong số đó để kiểm tra cho những thuật toán được trình bày ở trên:

- u.data (user\_id, movie\_id, rating, unix\_time): chứa thông tin của 943 users rate cho 1682 item. Mỗi user rate ít nhất 20 sản phẩm

- ua.base, ua.test, ub.base, ub.test (user\_id, movie\_id, rating, unix\_time): được trích xuất từ tập data dùng để huấn luyện thuật toán và file test dùng để kiếm tra độ chính xác của thuật toán

- u.user (user\_id, age, sex, occupation, zip\_code) : chứa thông tin của users

|  |  |
| --- | --- |
| - | u.item(movie\_id, movie title, release date, video release, IMDb url,….(gồm |

19 thuộc tính về thể loại phim)): thông tin về mỗi bộ phim

- u.genre: Chứa tên của 19 thể loại phim: unknown, Action, Adventure, Animation, Children's, Comedy, Crime, Documentary, Drama, Fantasy, Film-Noir, Horror, Musical, Mystery, Romance, Sci-Fi, Thriller, War, Western

**Cấu trúc chi tiết của từng file**

|  |  |  |
| --- | --- | --- |
| **Tên tập tin** | **Thuộc tính** | **Ý nghĩa** |
| u.data, ua.base,ua.test  ub.base,ub.test | User\_id | ID người dùng |
| Movie\_id | ID phim |
| Rating | Đánh giá |
| Unix time | Thời gian đánh giá |
| u.user | User\_id | ID người dùng |
| Age | Tuổi |
| Sex | Giới tính |
| Occupation | Nghề nghiệp |
| Zip\_code | Mã bưu chính |
| u.item | Movie\_id | ID phim |
| Movie title | Tên phim |
| Release date | Ngày sản xuất |
| Video realease | Phát hành |
| IMDb url | Link phim |

**Cấu trúc chi tiết của thể loại phim**

|  |  |
| --- | --- |
| **Thể loại** | **Ý nghĩa** |
| unknown | Chưa biết |
| Action | Hành động |
| Adventure | Phiêu lưu |
| Animation | Hoạt hình |
| Children's | Trẻ em |
| Comedy | Hài kịch |
| Crime | Tội phạm |
| Documentary | Tài liệu |
| , Drama | Chính kịch |
| Fantasy | Tưởng tượng |
| Film-Noir | Thuộc về tội phạm |
| Horror | Kinh dị |
| Musical | Âm nhạc |
| Mystery | Điều tra |
| Romance | Lãng mạn |
| Sci-Fi | Khọc học viễn tưởng |
| Thriller | Tâm lý |
| War | Chiến tranh |
| Western | Miền tây Hoa Kì |

**3.3 Yêu cầu chức năng**

**3.3.1 Chức năng tính toán**

- Khuyến nghị theo đặc tính của sản phẩm (Content-Based)

- Khuyến nghị theo sự tương quan giữa người dùng hoặc sản phẩm (user- user, item- item neighborhood based).

- Khuyến nghị theo cách phân tích ma trận thành nhân tử (matrix factorization)

**3.3.2 Phi chức năng**

- Hệ thống sẽ giải quyết được mọi bài toán trong tất cả lĩnh vực khi bộ dữ liệu đầu vào thỏa mãn 1 nguyên tắc nhất định được lập trình sẵn

-Hiển thị những thông tin cần thiết cho người dùng

**3.3.3 Chức năng của của việc sử dụng API**

- Có thể tương thích trên nhiều loại ngôn ngữ khác nhau, dễ dàng mở rộng và phát triển khi áp dụng vào hệ thống thực tế

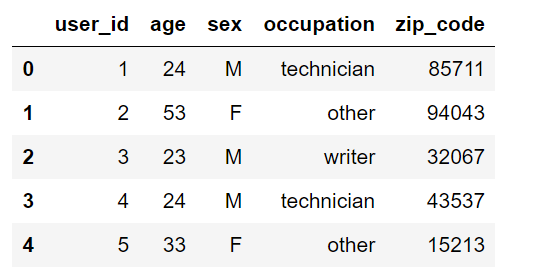
- Người dùng dẫn đường dẫn vào bộ dữ liệu có sẵn và nhập vào các thể loại cần thiết của sản phẩm theo đúng thứ tự trong file

- Gợi ý những phim khác mà người dùng có thể thích nhưng chưa thể tiếp cận được

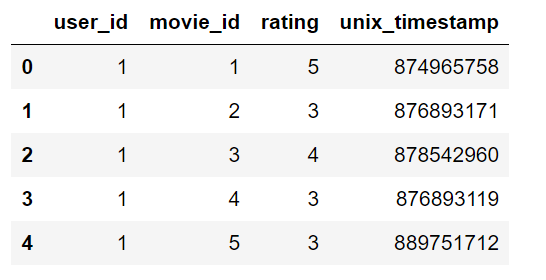
**3.3.4 Yêu cầu về dữ liệu**

Đặt tên đúng theo tên được đặt ra và các trường dữ liệu cần được tạo theo đúng thứ tự

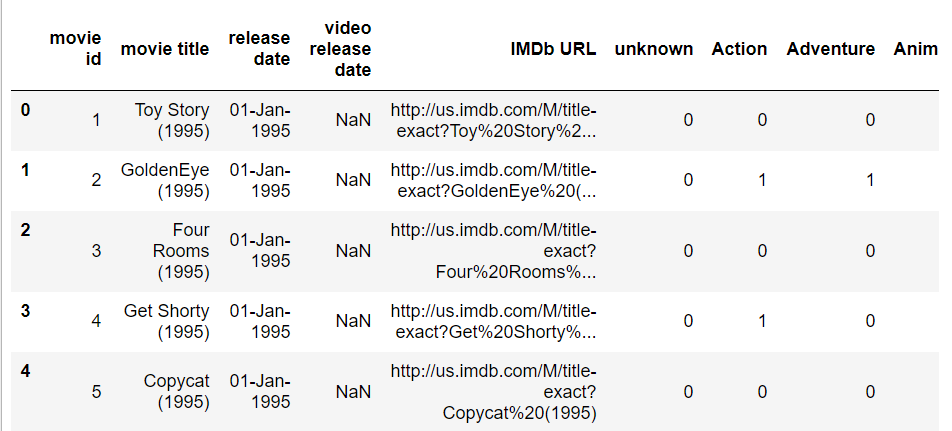
**File: u.user**



**File: u.data, ua.base,ub.base,ua.test,ub.test**



**File u.item** : các trường ban đầu cần được dữ đúng theo chuẩn, các trường thuộc tính sản phẩm có thể thay đổi tùy theo sản phẩm



**3.4 Thông tin phiên bản**

|  |  |  |
| --- | --- | --- |
| STT | Công cụ | Phiên bản |
| 1 | Python | 3.6.2 |
| 2 | Nodejs | V8.10.0 |
| 3 | Thư viện sklearn | 0.19.0 |

**3.5 Cài đặt hệ thống**

**3.5.1 Cài đặt Python**

- Link tải: [www.python.org](http://www.python.org)

- Cài đặt thư viện: dùng pip install

pip install --upgrade <package\_name>

Trong đó đã bao gồm cài đặt cả thư viện sklearn.

**3.5.2 Cài đặt Nodejs**

- Link tải : [www.nodejs.org](http://www.nodejs.org)

- Sau khi cài đặt Nodejs xong, vào thư mục của mã nguồn (source code), chạy lệnh : “npm install” để cài đặt các package cần thiết.

# **CHƯƠNG 4: KẾT QUẢ ĐẠT ĐƯỢC**

**4.1. Kết quả đạt được**

- Khi thực hiện đề tài này, nhóm đã đạt được một số kết quả như sau:

* Biết được các phương pháp tiếp cận hệ thống recommend
* Đánh giá được phương pháp Alternating Least Square thông qua Root Mean Square Error (RMSE).
* Sử dụng Webservice API cho việc kiểm thử thuật toán.
* Xây dựng website thao tác giữa người dùng và hệ thống.

- Ngoài ra, khi kiểm tra trên cùng 1 user, thu được những kết quả khác nhau, cụ thể như

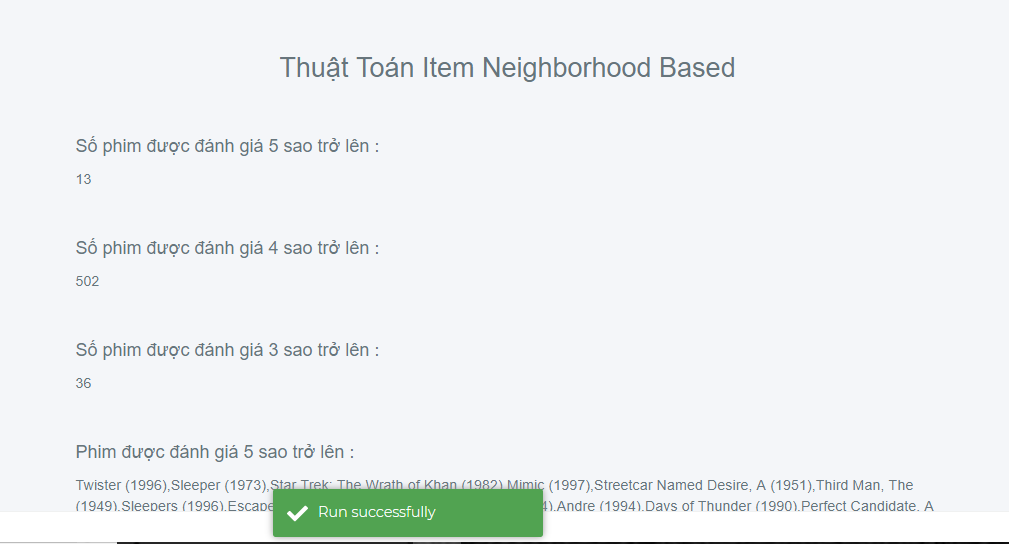
* **Content-Based.**



* **User-User Neighborhood-Based**



* **Item-Item Neighborhood-Based**



* **Matrix Factorization**



**4.2. Khó khăn và hạn chế**

- Bên cạnh những kết quả đạt được, trong quá trình thực hiện đề tài nhóm cũng gặp phải những khó khăn và hạn chế nhất định.

- Khó khăn:

* Thuật ngữ hệ thống khuyến nghị (Recommendation Systems) còn khá mới mẻ đối với nhóm. Thời gian đầu khi bắt đầu đề tài gặp nhiều vấn đề về thuật toán, xử lí dữ liệu, kiến trúc hệ thống khuyến nghị, …
* Tìm hiểu về các phương pháp để tối ưu hệ thống cũng như độ chính xác của thuật toán mất nhiều thời gian.

- Hạn chế:

* Độ chính xác của thuật toán chưa thực sự cao.
* Đối với hệ thống khuyến nghị cần phải có hạ tầng hệ thống mạnh để triển khai nhưng hạ tầng của nhóm đang sử dụng chỉ ở mức trung bình dẫn đến thời gian thao tác mất khá nhiều thời gian.

Tuy kết quả không được ấn tượng nhưng sẽ góp phần làm giảm được hiện tượng long tail trong kinh doanh

**4.3. Kết luận**

**4.3.1 Kết quả và nhận xét về Content-Based khi áp dụng vào bộ dữ liệu.**

Kết quả của thuật toán:



Từ hình vẽ ta có thể thấy được sự chênh lệch giữa đánh giá thực tế và đánh giá dự đoán của thuật toán, mức độ chênh lệch vẫn còn khá cao giữa thực tế và thuật toán

Đánh giá mô hình:



Để đánh giá mô hình tìm được, chúng ta sẽ sử dụng Root Mean Squared Error (RMSE), tức căn bậc hai của trung bình cộng bình phương của lỗi. Lỗi được tính là hiệu của true rating và predicted rating

Như vậy, với tập training, sai số vào khoảng 0.9 sao; với tập test, sai số lớn hơn một chút, rơi vào khoảng 1.3. Chúng ta thấy rằng kết quả này chưa thực sự tốt vì chúng ta đã đơn giản hoá mô hình đi quá nhiều.

Ưu điểm:

+ Không đòi hỏi quá nhiều dữ liệu người dùng để đưa ra được khuyến nghị những vẫn đạt kết quả chấp nhận được

+ Không ảnh hưởng về hành vi của người dùng hay nói cách khác là độc lập với người dùng

+Rõ ràng và minh bạch vì hệ thống sử dụng đặc tính của sản phẩm đã được cố định sẵn

+Tiếp cận khách hàng ngay lập tức.

Khuyết điểm:

+ Khi gặp những trường hợp khó xác định đặc tính sản phẩm gây ra rắc nhiều khó khăn trong việc khuyến nghị.

+ Khi chưa đủ dữ liệu của người dùng mới, hệ thống sẽ ánh không được như mong muốn của người dung.

**4.3.2 Kết quả và nhận xét User-User Neighborhood khi áp dụng vào bộ dữ liệu.**

**Đánh giá mô hình:**



Để đánh giá mô hình tìm được, chúng ta sẽ sử dụng cùng một phép đánh giá để so sánh giữa các các thuật toán với nhau

Như vậy, sai số vào khoảng 0.9 sao; Chúng ta thấy rằng kết quả này tuy chưa thực sự tốt nhưng sai số đã nhỏ hơn với thuật toán Content-Based.

- Ưu điểm: Khắc phục được những khó khăn của Content-Based

Khuyết điểm:

+ Vấn đề về bộ nhớ luôn là vấn đề khó khăn cho thuật toán này.Vì số lượng users không ngừng tăng lên theo thời gian

+ Ngoài ra, khi users thay đổi hành vi của mình bằng cách thay đổi rate hoặc rate thêm thì sẽ thay đổi giá trị của một vài bước nêu trên và việc tính toán lại rất mất thời gian và tốn bộ nhớ.

**4.3.3 Kết quả và nhận xét User-User Neighborhood khi áp dụng vào bộ dữ liệu.**

**Đánh giá mô hình:**



Tương tự như việc đánh giá User, ở đây chung ta cũng sử dụng chung một phương pháp.

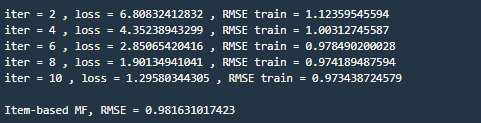
Như vậy, sai số vào khoảng 0.986 sao; Chúng ta thấy rằng kết quả này cho thấy sai số nhỏ hơn so với User-User Neighborhood, và kết quả này càng được rõ ràng hơn khi dữ liệu ban đầu là đủ lớn để có thể cho thấy được sự so khác giữa 2 thuật toán Neighborhood-Based.

- Ưu điểm: Hạn chế được những khó khăn của User-User Neighborhood-Based

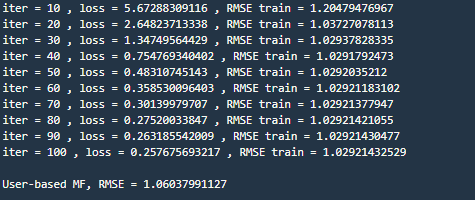
- Khuyết điểm: Vấn đề về bộ nhớ luôn vẫn luôn là vấn đề khó khăn cho dạng thuật toán này.

**4.3.4 Kết quả và nhận xét Matrix Factorization khi áp dụng vào bộ dữ liệu.**

Đánh giá mô hình dựa trên việc chuẩn hóa item(kết quả tốt hơn chuẩn hóa user):



Đánh giá mô hình dựa trên việc chuẩn hóa user:



Ở trên là kết quả khi áp dụng vào bộ dữ liệu, và kết quả được cải thiên rõ rệt qua tường lần train. Ở đây, sau 10 lần train, sai số đã giảm xuống còn 0.9734 so với giá trị chính xác, thuật toán Matrix Factorization thể hiện được khả năng “học” hơn hẵn những thuật toán khác.

- Ưu điểm: Tối ưu được bộ nhớ nhiều hơn khi làm việc với ma trận có kích thước nhỏ hơn nên sẽ có lợi thế lớn khi áp dụng với những dư liệu lớn

- Khuyết điểm: Phải huấn luyện nhiều lần và liên tục khi có dữ liệu mới vào.

**4.4. Hướng phát triển**

- Nhóm sẽ cố gắng tiếp cận với những mô hình thực tế từ những hộ kinh doanh online nhỏ lẻ với lượng dữ liệu vừa phải để nâng cao được độ chính xác trong việc khuyến nghị.

**TÀI LIỆU THAM KHẢO**

**Tài liệu tiếng Anh:**

[1] Francesco Ricci, Lior Rokach, Bracha Shapira, Paul B. Kantor, “Recommender Systems Handbook Second Edition”.

[2] Andrew Ng’s course, “Machine Learning”, <https://www.coursera.org/learn/machine-learning>

[3] Christopher R. Aberger, “Recommender: An Analysis of Collaborative Filtering Techniques”, Stanford university.

[4] Data school “Youtube”

**Tài liệu tiếng Việt:**

[4] Vu Huu Tiep, Ebook “Machienlearning cơ bản”

[5] Ong Xuan Hong ongxuanhong.wordpress.com