

**TRƯỜNG ĐẠI HỌC SƯ PHẠM TP.HCM**

**KHOA CÔNG NGHỆ THÔNG TIN**

**HỌC PHẦN:**

**Hệ tư vấn thông tin**

**Diagram

Description automatically generated with low confidence**

BÁO CÁO TÌM HIỂU ĐỀ TÀI

**Hệ thống gợi ý sách**

**Nhóm thực hiện: 5V1**

**Hoàng Trung Nam– MSSV: 44.01.104.145**

**Nguyễn Ngọc Danh– MSSV: 46.01.104.022**

**Trịnh Ngọc Huy – MSSV: 44.01.104.100**

**Đinh Phát Toàn– MSSV: 46.01.104.185**

**Nguyễn Văn Mạnh– MSSV: 44.01.104.137**

**Giảng viên hướng dẫn: Trần Thanh Nhã**

**TP. HCM, 2022**

Mục lục

[**Phần mở đầu** 3](#_Toc118574720)

[**Chương 1. Giới thiệu về Recommender system** 4](#_Toc118574721)

[**1.** **Giới thiệu** 4](#_Toc118574722)

[**2.** **Hệ thống gợi ý** 4](#_Toc118574723)

[**Chương 2. Các phương pháp đánh giá RS** 6](#_Toc118574724)

[**1.** **Tiêu chí định tính** 6](#_Toc118574725)

[**1.1.** **Tính mới của các gợi ý** 6](#_Toc118574726)

[**1.2.** **Tính đa dạng (Diversity) của các gợi ý** 7](#_Toc118574727)

[**1.3.** **Độ bao phủ (coverage) của các gợi ý** 8](#_Toc118574728)

[**1.4.** **Sự hài lòng của người sử dụng** 8](#_Toc118574729)

[**2. Tiêu chí định lượng** 8](#_Toc118574730)

[**2.1** **Đánh giá độ chính xác của các dự đoán** 8](#_Toc118574731)

[**2.1.1 MSE** 9](#_Toc118574732)

[**2.1.2 RMSE** 9](#_Toc118574733)

[**2.1.3 MAE** 9](#_Toc118574734)

[**2.2** **Đánh giá việc sử dụng các dự đoán** 10](#_Toc118574735)

[**2.2.1 Precision và Recall** 10](#_Toc118574736)

[**Chương 3. Đánh giá hệ thống gợi ý với cơ sở dữ liệu MovieLens 100k** 15](#_Toc118574737)

[**3.1 Cơ sở dữ liệu MovieLens 100k** 15](#_Toc118574738)

[**3.2. Xây dựng item profiles** 17](#_Toc118574739)

[**3.3. Tìm mô hình cho mỗi user** 18](#_Toc118574740)

[**3.4. Đánh giá mô hình** 20](#_Toc118574741)

[**Tài liệu tham khảo** 21](#_Toc118574742)

**Phần mở đầu**

1. **Lý do chọn đề tài**

Một hệ thống gợi ý không thể được triển khai nếu chưa qua đánh giá. Việc đánh giá một hệ thống gợi ý là một giai đoạn cần thiết vì hiệu quả của một hệ thống gợi ý không chỉ phụ thuộc vào đặc điểm dữ liệu mà còn phụ thuộc vào mục đích gợi ý (Herlocker J.L et al, 2004). Nghĩa là một hệ thống gợi ý với cùng một giải thuật thì kết quả gợi ý có thể có hiệu quả khác nhau trên những tập dữ liệu khác nhau. Liên quan đến mục đích của hệ thống, một vài hệ thống gợi ý chú trọng đến tính đa dạng của các mục dữ liệu trong danh sách gợi ý nhưng một số khác lại chú trọng đến tính mới của các mục dữ liệu. Tùy thuộc vào đặc trưng dữ liệu và mục đích của hệ thống gợi ý, các phương pháp đánh giá khác nhau có thể được sử dụng. Ngoài ra, nó còn phụ thuộc vào điều kiện để đánh giá hệ thống như dựa vào dữ liệu có sẵn để đánh giá (offline) hay triển khai hệ thống và đánh giá trực tuyến (online).

1. **Mục tiêu nghiên cứu**

* Tìm hiểu các phương pháp và kỹ thuật đánh giá hệ thống gợi ý để nắm bắt chức năng của hệ thống và đưa ra kết quả tối ưu
* Chỉ ra được cái nhìn tổng quan về hệ thống gợi ý cũng như phân tích chi tiết các vấn đề liên quan đến việc đánh giá một hệ thống gợi ý.
* Phân tích rõ ràng các phương pháp đánh giá để đánh giá chung về chất lượng của hệ thống.

1. **Phạm vi và phương pháp nghiên cứu**

* **Đối tượng nghiên cứu**

Các phương pháp đánh giá hệ thống gợi ý

* **Phạm vi nghiên cứu**

Tìm hiểu các tiêu chí đánh giá định tính và định lượng trong hệ thống nghiên cứu

1. **Phương pháp nghiên cứu**

Sử dụng tổng hợp một số phương pháp nghiên cứu khoa học như phân tích và tổng hợp, lịch sử.

1. **Kêt cấu của đề tài**

Chương 1. Giới thiệu về Recommender system

Chương 2. Các phương pháp đánh giá RS

Chương 3. Đánh giá hệ thống gợi ý với cơ sở dữ liệu MovieLens 100k

**Chương 1. Giới thiệu về Recommender system**

1. **Giới thiệu**

Ngày nay, hệ thống gợi ý được nhiều người biết đến như một công cụ hỗ trợ hữu ích để giúp người dùng tìm được nhiều thông tin liên quan và phù hợp trong một cơ sở dữ liệu lớn một cách nhanh chóng. Các hệ thống gợi ý được ứng dụng trong nhiều lĩnh vực như thương mại điện tử, giải trí, khoa học, tin tức… Trong lĩnh vực thương mại, người dùng sẽ được hệ thống gợi ý các sản phẩm phù hợp với nhu cầu của từng cá nhân. Ví dụ như hệ thống gợi ý bán hàng của Amazon, Ebay, …

Trong lĩnh vực giải trí, người dùng có thể được gợi ý các bộ phim, bài hát phù hợp mà người sử dụng không phải mất nhiều công sức tìm kiếm như hệ thống gợi ý phim MovieLens1 , last.fm2 , FilmConseil. Trong lĩnh vực khoa học, hệ thống gợi ý hỗ trợ người dùng tìm kiếm các bài báo khoa học như hệ thống tìm kiếm Citeseer3 hay sinh viên tìm kiếm các tài liệu học tập phù hợp với cá nhân như hệ thống School e-Guide của tác giả M. Almulla. Trong lĩnh vực tin tức, người đọc được hệ thống hỗ trợ gợi ý các bài báo phù hợp với từng người đọc riêng biệt ví dụ như netnews, yahoo news…

Một hệ thống gợi ý không thể được triển khai nếu chưa qua đánh giá. Việc đánh giá một hệ thống gợi ý là một giai đoạn cần thiết vì hiệu quả của một hệ thống gợi ý không chỉ phụ thuộc vào đặc điểm dữ liệu mà còn phụ thuộc vào mục đích gợi ý (Herlocker J.L et al, 2004). Nghĩa là một hệ thống gợi ý với cùng một giải thuật thì kết quả gợi ý có thể có hiệu quả khác nhau trên những tập dữ liệu khác nhau. Liên quan đến mục đích của hệ thống, một vài hệ thống gợi ý chú trọng đến tính đa dạng của các mục dữ liệu trong danh sách gợi ý nhưng một số khác lại chú trọng đến tính mới của các mục dữ liệu. Tùy thuộc vào đặc trưng dữ liệu và mục đích của hệ thống gợi ý, các phương pháp đánh giá khác nhau có thể được sử dụng. Ngoài ra, nó còn phụ thuộc vào điều kiện để đánh giá hệ thống như dựa vào dữ liệu có sẵn để đánh giá (offline) hay triển khai hệ thống và đánh giá trực tuyến (online). Trong phần hai, chúng tôi sẽ trình bày khái quát về một hệ thống gợi ý. Tiếp theo các nghi thức kiểm tra hệ thống, các phương pháp đánh giá một hệ thống gợi ý được trình bày chi tiết trong phần ba và bốn. Cuối cùng là phần kết luận.

1. **Hệ thống gợi ý**

Hệ thống gợi ý là hệ thống hỗ trợ ra quyết định nhằm gợi ý các thông tin liên quan đến người dùng một cách dễ dàng và nhanh chóng, phù hợp với từng người dùng (Adomavicius, G. and A. Tuzhilin, 2005). Ví dụ với trang web Amazon, một trong những trang web thương mại điện tử nổi tiếng nhất, khi người dùng truy cập vào trang web này họ sẽ được gợi ý những sản phẩm tiềm năng nhất từ hàng triệu sản phẩm trong hệ thống. Hệ thống gợi ý như một công cụ cung cấp những thông tin hữu ích và riêng biệt theo từng cá nhân trên một hệ thống chứa đựng một lượng lớn thông tin. Các hệ thống gợi ý được thiết kế nhằm cung cấp cho người dùng những đề nghị liên quan, những đề nghị hiệu quả nhất có thể từ thông tin của các mục dữ liệu, từ hồ sơ người sử dụng và từ mối liên hệ giữa những đối tượng này. Cấu trúc của một hệ thống gợi ý gồm có ba thành phần chính (Adomavicius, G. and A. Tuzhilin, 2005): tập hợp các người dùng U = {u1,…,up } bao gồm các thông tin của người dùng được lưu trên hệ thống; tập hợp các mục dữ liệu I = {i1,…,ip} bao gồm định danh và các thuộc tính của mục dữ liệu; tập hợp các “mối quan hệ” R = (Ui, Ij) giữa “người dùng” và “mục dữ liệu”, đây là tập hợp các giao dịch liên kết giữa tập hợp người dùng U và tập hợp mục dữ liệu I và những mô tả của mối liên kết này (Schafer J.B., et al, 2007).

Cụ thể một hệ thống gợi ý có thể được miêu tả như trong Hình giưới Tập hợp người dùng có thể là một người phụ nữ, một người đàn ông hay là một đứa trẻ. Người dùng này có thể mua, xem, chọn lựa, đọc hay đánh giá mục dữ liệu. “Người dùng” được xem như là tác nhân của hệ thống tác động lên các “mục dữ liệu”. “Mục dữ liệu” có thể là quần áo, phim ảnh, sách vở, bài báo, bài hát, cd, trang web, rượu… Mối quan hệ giữa người dùng và mục dữ liệu có thể là quan hệ yêu thích, mong muốn, mua, đọc… Sau đó, hệ thống sẽ cung cấp một danh sách các mục dữ liệu đề nghị cho người dùng. Những mục dữ liệu đề nghị phải dựa trên tiêu chí phù hợp với sở thích, thói quen của người dùng. Mục tiêu cuối cùng của một hệ thống gợi ý là đưa ra một danh sách các mục dữ liệu tiềm năng phù hợp với nhu cầu, mong muốn của người dùng

Graphical user interface

Description automatically generated

Sơ đồ tổng quan hệ thống gợi ý

**Chương 2. Các phương pháp đánh giá RS**

Nghiên cứu về 2 nhóm tiêu chí đánh giá “ định lượng ”,”định tính ” cũng như các phương pháp đánh giá của chúng.

1. **Tiêu chí định tính**

Trong những giai đoạn đầu phát triển thì hệ thống gợi ý chỉ sử dụng các độ đo chính xác định lượng như đã đề cập. Tuy nhiên, người dùng ngày càng có yêu cầu cao hơn và nhiều hơn về chất lượng của các gợi ý. Nếu chỉ xét độ chính xác thì không đủ để đánh giá hiệu quả của một hệ thống gợi ý nên cần đưa thêm thuộc tính chất lượng các gợi ý thay vì chỉ sử dụng độ chính xác của các gợi ý.

### **Tính mới của các gợi ý**

Việc đánh giá tính mới của gợi ý là hiển nhiên để đáp ứng nhu cầu của người sử dụng các sản phẩm mới được tạo ra liên tục. Khái niệm "sản phẩm mới" có thể có nhiều ý nghĩa khi đề cập đến hệ thống gợi ý. Tính mới của dữ liệu theo quan điểm thời gian (trong trường hợp xuất hiện sản phẩm mới) hoặc liên quan đến lịch sử của người sử dụng (một sản phẩm mà chưa bao giờ được mua). Điều này xảy ra như một trường hợp đặc biệt mà mục dữ liệu trong hệ thống chưa có thông tin liên quan đến người sử dụng, như thể hiện “sản phẩm mới” trong hình 3. Vấn đề “thiếu thông tin” (cold start problem).

Graphical user interface

Description automatically generated with medium confidence

Hình 1: Sự phân bố của sản phẩm

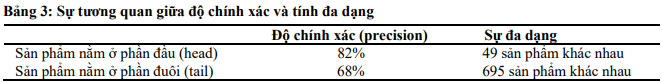
Một số hệ thống cung cấp rất chính xác gợi ý nhưng không hữu dụng trong thực tế vì không quan tâm đến các tiêu chí định tính trong quá trình xây dựng hệ thống. Ví dụ như hệ thống gợi ý “sữa tươi” cho khách hàng trong một siêu thị ở châu Âu. Đề nghị này là chính xác bởi vì hầu như tất cả các khách hàng đều mua sữa, nhưng nó không phải là hữu ích cho tất cả người dùng đã quen thuộc với sản phẩm này. Các nhà cung cấp nhận thức được điều này trong một thời gian dài và tổ chức các kệ đựng hàng hoá cho phù hợp. Do đó, việc giới thiệu cho người mua một thực phẩm mới có khả năng để làm hài lòng người mua mà họ không bao giờ nghĩ là cần thiết.

Thuộc tính mới được nhấn mạnh như là một chỉ số cần thiết để đánh giá tính hiệu quả của hệ thống gợi ý. Có 3 điểm quan trọng liên quan đến hiệu quả của hệ thống gợi ý.

1. Tính chính xác và tính mới lạ phải được tính đến để xây dựng được các gợi ý hiệu quả.
2. Yếu tố thời gian là điều cần thiết trong việc đánh giá tính mới của mục dữ liệu.
3. Danh sách gợi ý phù hợp nhất phải kết hợp một tỉ lệ các mục dữ liệu mới và các gợi ý phù hợp khác.

### **Tính đa dạng (Diversity) của các gợi ý**

Sự đa dạng của hệ thống gợi ý đo lường khả năng cung cấp một danh sách các mục dữ liệu được phân phối từ nhiều loại khác nhau. Có thể phân chia sự đa dạng của các gợi ý thành hai loại đa dạng: sự đa dạng cá nhân và đa dạng tổng thể. Loại đa dạng cá nhân quan tâm đến các khái niệm về đa dạng từ quan điểm của người sử dụng. Chỉ số này được tính toán dựa trên trung bình sự khác nhau giữa tất cả các cặp mục dữ liệu đã gợi ý. Ngược lại, sự đa dạng tổng thể là quan tâm đến các mục dữ liệu đã gợi ý hơn là quan tâm đến người dùng. Nếu sự đa dạng tổng thể của hệ thống giới thiệu là lớn, thì sự đa dạng của các gợi ý cá nhân cũng là rất lớn, nhưng điều này không đúng cho chiều ngược lại.



Hai yếu tố có tác động trực tiếp đến sự đa dạng của gợi ý là các thuật toán sử dụng để xây dựng hệ thống và các đặc tính của cơ sở dữ liệu. Mối quan hệ này được biểu diễn trong bảng trên:

* Bảng này cho thấy kết quả của hai trường hợp riêng biệt: các gợi ý được tạo ra, hoặc từ các mặt hàng phổ biến nhất **(i),** hoặc từ đuôi dài (long tail) **(ii).**
* Theo thông tin trong bảng này, sự đa dạng lớn hơn nhiều khi các gợi ý được tạo ra từ các mặt hàng thuộc đuôi dài. Trong trường hợp này, người ta cũng có thể nhận thấy rằng độ chính xác giảm. Điều đó chứng tỏ rằng sự đa dạng của các gợi ý sẽ tập trung vào các mục dữ liệu nằm thuộc phần đuôi dài, tuy nhiên cần phải có một tỉ lệ hợp lý để không làm giảm quá nhiều độ chính xác của hệ thống.

### **Độ bao phủ (coverage) của các gợi ý**

Độ bao phủ của hệ thống gợi ý là thước đo số lượng lĩnh vực mà danh sách các sản phẩm gợi ý được tạo ra thuộc về chúng, số lĩnh vực này có bao trùm được hệ thống hay không

* Cách dễ nhất để đo loại này là chọn một cách ngẫu nhiên cặp user/item, yêu cầu một dự đoán cho mỗi cặp, và đo tỷ lệ phần trăm mà dự đoán được cung cấp. Giống như chỉ số precision và recall phải được xem xét đồng thời, độ bao phủ (Coverage) thường được kết hợp với chỉ số “accuracy”, vì không thể tăng giá độ bao phủ mà không quan tâm đến việc tạo ra những gợi ý không thuộc hệ thống.
* Độ bao phủ được đo bằng sự phong phú của hồ sơ người dùng để đưa ra gợi ý. Ví dụ trong lọc cộng tác, người dùng phải đánh giá các item trước khi nhận các gợi ý. Việc đo lường này là một loại hình đặc trưng trong nghi thức đánh giá off-line.

### **Sự hài lòng của người sử dụng**

Sự hài lòng của người sử dụng là một khía cạnh hơi mơ hồ và phụ thuộc vào từng cá nhân khác nhau và do đó rất khó để đo lường. Sự hài lòng của người dùng được định nghĩa là mức độ mà một người dùng được hỗ trợ trong việc đối phó với các vấn đề quá tải thông tin.

Phương pháp đánh giá “rõ ràng” (Explicit) và “ngầm hiểu” (Implicit): phương pháp đánh giá một cách rõ ràng nghĩa là hệ thống đo độ hài lòng của người sử dụng bằng cách yêu cầu trực tiếp; phương pháp đánh giá ngầm hiểu thì cần phải đặt ra những giả định và dịch những quan sát được thành những giả định, ví dụ như sự gia tăng doanh số của một cửa hàng chứng tỏ sự hài lòng của khách hàng tăng lên.

* Kết quả so với quá trình: việc đánh giá có thể chỉ tập trung vào kết quả, nhưng nó cũng có thể tập trung vào quá trình áp dụng hệ thống gợi ý.

Đánh giá trong khoảng thời gian ngắn (Short term) và khoảng thời gian dài (Long term): đánh giá người sử dụng trong một khoảng thời gian ngắn có thể sẽ thiếu sót thông tin mà nó sẽ trở nên chính xác hơn sau một khoảng thời gian nhất định. Sở thích của người dùng cần phải xem xét đánh giá của người dùng qua một khoảng thời gian dài. Các nghiên cứu điều tra sự hài lòng của người dùng đối với hệ thống gợi ý là rất hiếm và nghiên cứu tập trung trên sự hài lòng của các gợi ý thì càng hiếm hơn.

## **2. Tiêu chí định lượng**

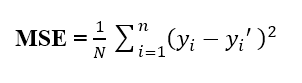
* 1. **Đánh giá độ chính xác của các dự đoán**

Việc đánh giá tính chính xác các dự đoán có thể sử dụng sai số bình phương trung bình (MSE - Mean Square Error), căn của sai số bình phương trung bình (RMSE - Root Mean Square Error), sai số tuyệt đối trung bình (MAE - Mean Absolute Error). Tính chính xác của các dự đoán được đo trên n quan sát, trong đó là giá trị dự đoán đánh giá của mục i và là giá trị đánh giá thực tế của mục i.

### **2.1.1 MSE**

MSE (Mean Square Error) có lẽ là một metric phổ biến nhất trong các bài toán hồi quy. Về cơ bản, nó tính trung bình của bình phương sai số giữa giá trị thực tế và giá trị dự đoán. MSE là thước đo chất lượng của một công cụ ước tính - nó luôn không âm và các giá trị càng gần 0 càng tốt.

Giả sử ta có một bài toán mà chắc hẳn ai đọc về Machine Learning cũng từng đọc qua, chính là bài toán dự đoán giá nhà. Coi giá trị thực tế của nhà thứ i là , còn giá trị dự đoán của căn nhà đó là . Vậy, MSE có thể được tính như sau:



MSE càng thấp thì dự báo càng tốt.

### **2.1.2 RMSE**

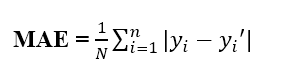
Root Mean Square Error (RMSE) hoặc Root Mean Square Deviation (RMSD) là căn bậc hai của mức trung bình của các sai số bình phương. RMSE là độ lệch chuẩn của các phần dư (sai số dự đoán).

Phần dư là thước đo khoảng cách từ các điểm dữ liệu đường hồi quy; RMSE là thước đo mức độ dàn trải của những phần dư này, nói cách khác, nó cho bạn biết mức độ tập trung của dữ liệu xung quanh đường phù hợp nhất.

RMSE luôn không âm và giá trị 0 (hầu như không bao giờ đạt được trong thực tế) sẽ chỉ ra sự phù hợp hoàn hảo với dữ liệu. Nói chung, RMSD thấp hơn sẽ tốt hơn RMSD cao hơn.

### **2.1.3 MAE**

Mean Squared Error (MSE) có lẽ là số liệu phổ biến nhất được sử dụng cho các bài toán hồi quy. Về cơ bản, nó tìm thấy sai số bình phương trung bình giữa các giá trị được dự đoán và thực tế. MSE là thước đo chất lượng của một công cụ ước tính - nó luôn không âm và các giá trị càng gần 0 càng tốt.



Trong đó n là số điểm dữ liệu, yᵢ là giá trị quan sát và là giá trị dự đoán.

Có thể diễn đạt MAE là tổng hòa của hai thành phần: Bất đồng về số lượng và Bất đồng về phân bổ.

MAE được biết đến là mạnh mẽ hơn đối với các yếu tố ngoại lai so (outliers) với MSE. Lý do chính là trong MSE bằng cách bình phương các sai số, các giá trị ngoại lai (thường có sai số cao hơn các mẫu khác) được chú ý nhiều hơn và chiếm ưu thế trong sai số cuối cùng và tác động đến các tham số của mô hình.

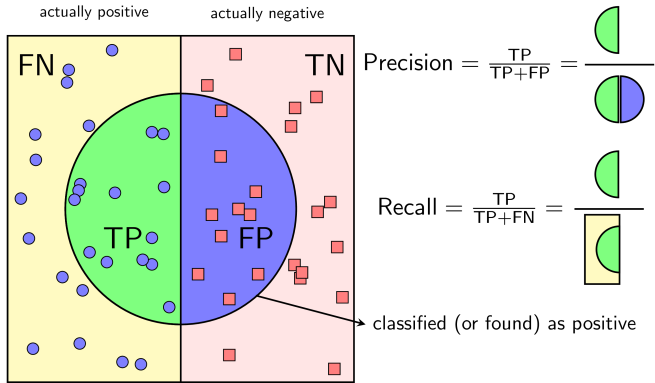
Các chỉ số này thích hợp cho một cơ sở dữ liệu không phải nhị phân và cho một giá trị dự đoán là số. Nó giúp đo lường mức độ sai số của các dự đoán. Các giá trị đo lường này bằng 0 khi hệ thống đạt được hiệu quả tốt nhất. Giá trị này càng cao thì hiệu quả của hệ thống càng thấp.

* 1. **Đánh giá việc sử dụng các dự đoán**

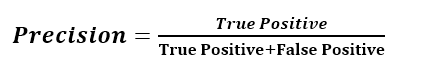
Ngoài việc đánh giá tính chính xác của các dự đoán, một số chỉ số khác như precision, recall và F1-score, Rscore được dùng để đánh giá việc sử dụng của các dự đoán trong trường hợp cơ sở dữ liệu nhị phân. Các chỉ số này đánh giá các gợi ý phù hợp cho mỗi người dùng thay vì đánh giá số điểm liên quan đến từng đề nghị. Đề nghị được coi là phù hợp khi người dùng chọn mục dữ liệu từ danh sách những đề nghị đã được gợi ý cho người dùng.

### **2.2.1 Precision và Recall**

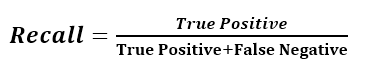
Giả sử ta có bài toán nhị phân với hai lớp là positive và negative.



Precision được định nghĩa là tỉ lệ **số điểm Positive mô hình dự đoán đúng**trên**tổng số điểm mô hình dự đoán là Positive.**



Recall được định nghĩa là tỉ lệ **số điểm Positive mô hình dự đoán đúng**trên**tổng số điểm thật sự là Positive**(hay**tổng số điểm được gán nhãn là Positive ban đầu).**



Precision càng cao, tức là số điểm mô hình dự đoán là positive đều là positive càng nhiều. Precision = 1, tức là tất cả số điểm mô hình dự doán là Positive đều đúng, hay không có điểm nào có nhãn là Negative mà mô hình dự đoán nhầm là Positive.

Recall càng cao, tức là số điểm là positive bị bỏ sót càng ít. Recall = 1, tức là tất cả số điểm có nhãn là Positive đều được mô hình nhận ra.

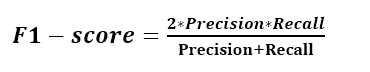
**2.2.2 F1-score**

Tuy nhiên, chỉ có Precision hay chỉ có Recall thì không đánh giá được chất lượng mô hình.

Chỉ dùng Precision, mô hình chỉ đưa ra dự đoán cho một điểm mà nó chắc chắn nhất. Khi đó Precision = 1, tuy nhiên ta không thể nói là mô hình này tốt.

Chỉ dùng Recall, nếu mô hình dự đoán tất cả các điểm đều là positive. Khi đó Recall = 1, tuy nhiên ta cũng không thể nói đây là mô hình tốt.

Khi đó F1-score được sử dụng. F1-score là trung bình điều hòa (harmonic mean) của precision và recall (giả sử hai đại lượng này khác 0). F1-score được tinh theo công thức:



Text, letter

Description automatically generatedRscore hay Breese score (Breese, J.S. and D. Heckerman, 1998) cũng là một trong những chỉ số đánh giá khả năng sử dụng dự đoán nhưng chỉ số này chính xác đến thứ tự của các gợi ý được xây dựng. Rscore đánh giá vị trí của sản phẩm được chọn bởi người dùng trong danh sách sản phẩm gợi ý được tạo ra bởi hệ thống.

Các chỉ số Precision, Recall và F\_score, Rscore thường được sử dụng đối với các hệ thống gợi ý trong lĩnh vực thương mại điện tử. Các chỉ số đánh giá, công thức tương ứng và một số hệ thống gợi ý/ nghiên cứu đã áp dụng các chỉ số tương ứng đó được tổng hợp trong bảng dưới đây:

Table

Description automatically generated with medium confidence

**Ví dụ cụ thể**

Ví dụ một bài toán phân loại ảnh đó là mèo hay không, trong dữ liệu dự đoán có 100 ảnh là mèo, 1000 ảnh không phải là mèo. Ở đây, kết quả dự đoán là như sau

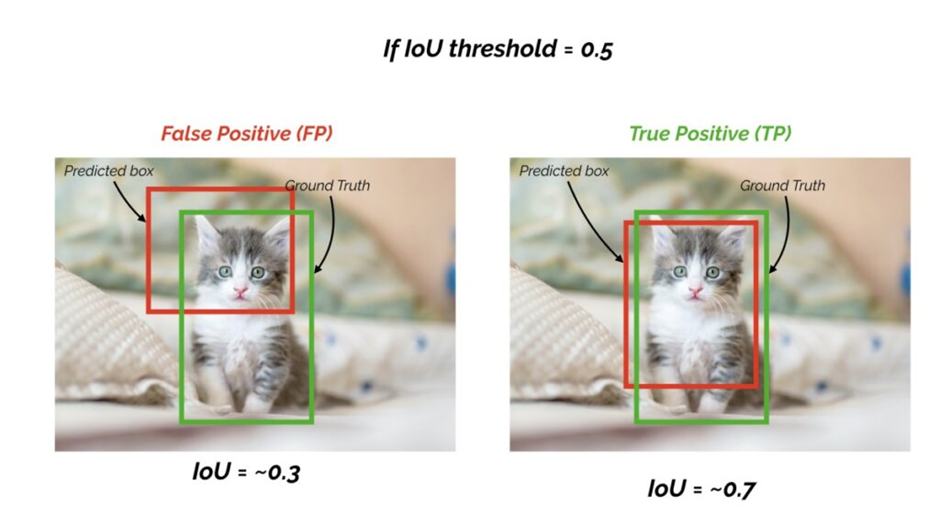
Trong 100 ảnh mèo dự đoán đúng 90 ảnh, còn 10 ảnh được dự đoán là không phải. Nếu ta coi cat là “positive” và non-cat là “negative”, thì 90 ảnh được dự đoán là cat, được gọi là True Positive, còn 10 ảnh được dự đoán non-cat kia được gọi là False Negative

Trong 1000 ảnh non-cat, dự đoán đúng được 940 ảnh là non-cat, được gọi là True Negative, còn 60 ảnh bị dự đoán nhầm sang cat được gọi là False Positive

Có thể tới đây nhiều người sẽ khá là lẫn lộn, “True”, “False” rồi “Positive”, “Negative”. Vậy để có một cách dễ nhớ, có một mánh nhỏ như sau

True/False ý chỉ những gì ta đã dự đoán là đúng hay chưa

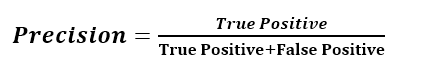
Positive/Negative chỉ những gì ta dự đoán (có hoặc không) Nói cách khách, nếu thấy chữ True tức là dự đoán là đúng (là cat hay non-cat, chỉ cần đúng), còn False thì ngược lại.



Precision

Như đã nói phía trên, sẽ có rất nhiều trường hợp thước đo Accuracy không phản ánh đúng hiệu quả của mô hình. Giả sử mô hình dự đoán tất cả 1100 ảnh là Non-cat, thì Accuracy vẫn đạt tới 1000/1100 = 90.9%, khá cao nhưng thực chất mô hình khá là tồi Vì vậy chúng ta cần một metric có thể khắc phục được những yếu điểm này. Precision là một trong những metrics có thể khắc phục được, công thức như sau:

Áp dụng vào bài toán Cat/Non-cat, Precision sẽ được tính như sau:

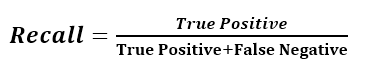


Precision(cat) = 90/(90+60) = 60% Precision(non-cat) = 940/(940+10) = 98.9%

Có thể thấy việc dự đoán Cat chưa thực sự tốt nhờ phép đó Precision này. Precision sẽ cho chúng ta biết thực sự có bao nhiêu dự đoán Positive là thật sự True

Recall

Recall cũng là một metric quan trọng, nó đo lường tỷ lệ dự báo chính xác các trường hợp positive trên toàn bộ các mẫu thuộc nhóm positive. Công thức của Recall như sau:



Áp dụng vào bài toán Cat/Non-cat, Precision sẽ được tính như sau:

Recall(cat) = 90/(90+10) = 90%

Recall(non-cat) = 940/(940+60) = 94%

Recall cao đồng nghĩa với việc True Positive Rate cao, tức là tỷ lệ bỏ sót các điểm thực sự là positive là thấp

**Chương 3. Đánh giá hệ thống gợi ý với cơ sở dữ liệu MovieLens 100k**

### **3.1 Cơ sở dữ liệu MovieLens 100k**

[Bộ cơ sở dữ liệu MovieLens 100k](https://grouplens.org/datasets/movielens/100k/) được công bố năm 1998 bởi [GroupLens](https://grouplens.org/). Bộ cơ sở dữ liệu này bao gồm 100,000 (100k) ratings từ 943 users cho 1682 bộ phim. Các bạn cũng có thể tìm thấy các bộ cơ sở dữ liệu tương tự với khoảng 1M, 10M, 20M ratings. Trong bài viết này, tôi sử dụng bộ cơ sở dữ liệu nhỏ nhất này nhằm mục đích minh hoạ.

Sau khi download và giải nén, chúng ta sẽ thu được rất nhiều các file nhỏ, chúng ta chỉ cần quan tâm các file sau:

* u.data: Chứa toàn bộ các ratings của 943 users cho 1682 movies. Mỗi user rate ít nhất 20 movies. Thông tin về thời gian rate cũng được cho nhưng chúng ta không sử dụng trong bài viết này.
* ua.base, ua.test, ub.base, ub.test: là hai cách chia toàn bộ dữ liệu ra thành hai tập con, một cho training, một cho test. Chúng ta sẽ thực hành trên ua.base và ua.test. Bạn đọc có thể thử với cách chia dữ liệu còn lại.
* u.user: Chứa thông tin về users, bao gồm: id, tuổi, giới tính, nghề nghiệp, zipcode (vùng miền), vì những thông tin này cũng có thể ảnh hưởng tới sở thích của các users. Tuy nhiên, trong bài viết này, chúng ta sẽ không sử dụng các thông tin này, trừ thông tin về id để xác định các user khác nhau.
* u.genre: Chứa tên của 19 thể loại phim. Các thể loại bao gồm: unknown, Action, Adventure, Animation, Children's, Comedy, Crime, Documentary, Drama, Fantasy, Film-Noir, Horror, Musical, Mystery, Romance, Sci-Fi, Thriller, War, Western,
* u.item: thông tin về mỗi bộ phim. Một vài dòng đầu tiên của file:
* 1|Toy Story (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Toy%20Story%20(1995)|0|0|0|1|1|1|0|0|0|0|0|0|0|0|0|0|0|0|0
* 2|GoldenEye (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?GoldenEye%20(1995)|0|1|1|0|0|0|0|0|0|0|0|0|0|0|0|0|1|0|0
* 3|Four Rooms (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Four%20Rooms%20(1995)|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|0|1|0|0
* 4|Get Shorty (1995)|01-Jan-1995||http://us.imdb.com/M/title-exact?Get%20Shorty%20(1995)|0|1|0|0|0|1|0|0|1|0|0|0|0|0|0|0|0|0|0

Trong mỗi dòng, chúng ta sẽ thấy id của phim, tên phim, ngày phát hành, link trên imdb, và các số nhị phân 0, 1 phía cuối để chỉ ra bộ phim thuộc các thể loại nào trong 19 thể loại đã cho trong u.genre. Một bộ phim có thể thuộc nhiều thể loại khác nhau. Thông tin về thể loại này sẽ được dùng để xây dựng item profiles.

Với cơ sở dữ liệu này, chúng ta sẽ sử dụng thư viện [pandas](https://pandas.pydata.org/) để trích xuất dữ liệu, có thể được cài đặt bằng pip install pandas.

**import** pandas **as** pd

*#Reading user file:*

u\_cols **=** ['user\_id', 'age', 'sex', 'occupation', 'zip\_code']

users **=** pd.read\_csv('ml-100k/u.user', sep**=**'|', names**=**u\_cols,

encoding**=**'latin-1')

n\_users **=** users.shape[0]

**print** 'Number of users:', n\_users

*# users.head() #uncomment this to see some few examples*

Number of users: 943

*#Reading ratings file:*

r\_cols **=** ['user\_id', 'movie\_id', 'rating', 'unix\_timestamp']

ratings\_base **=** pd.read\_csv('ml-100k/ua.base', sep**=**'\t', names**=**r\_cols, encoding**=**'latin-1')

ratings\_test **=** pd.read\_csv('ml-100k/ua.test', sep**=**'\t', names**=**r\_cols, encoding**=**'latin-1')

rate\_train **=** ratings\_base.as\_matrix()

rate\_test **=** ratings\_test.as\_matrix()

**print** 'Number of traing rates:', rate\_train.shape[0]

**print** 'Number of test rates:', rate\_test.shape[0]

Number of traing rates: 90570

Number of test rates: 9430

### **3.2. Xây dựng item profiles**

Công việc quan trọng trong content-based recommendation system là xây dựng profile cho mỗi item, tức feature vector cho mỗi item. Trước hết, chúng ta cần load toàn bộ thông tin về các items vào biến items:

*#Reading items file:*

i\_cols **=** ['movie id', 'movie title' ,'release date','video release date', 'IMDb URL', 'unknown', 'Action', 'Adventure',

'Animation', 'Children\'s', 'Comedy', 'Crime', 'Documentary', 'Drama', 'Fantasy',

'Film-Noir', 'Horror', 'Musical', 'Mystery', 'Romance', 'Sci-Fi', 'Thriller', 'War', 'Western']

items **=** pd.read\_csv('ml-100k/u.item', sep**=**'|', names**=**i\_cols,

encoding**=**'latin-1')

n\_items **=** items.shape[0]

**print** 'Number of items:', n\_items

Number of items: 1682

Vì ta đang dựa trên thể loại của phim để xây dựng profile, ta sẽ chỉ quan tâm tới 19 giá trị nhị phân ở cuối mỗi hàng:

X0 **=** items.as\_matrix()

X\_train\_counts **=** X0[:, **-**19:]

Tiếp theo, chúng ta sẽ xây dựng feature vector cho mỗi item dựa trên ma trận thể loại phim và feature [TF-IDF](https://stanford.edu/~rjweiss/public_html/IRiSS2013/text2/notebooks/tfidf.html). Tôi sẽ mô tả kỹ hơn về TF-IDF trong một bài viết khác. Tạm thời, chúng ta sử dụng thư viện sklearn.

*#tfidf*

**from** sklearn.feature\_extraction.text **import** TfidfTransformer

transformer **=** TfidfTransformer(smooth\_idf**=**True, norm **=**'l2')

tfidf **=** transformer.fit\_transform(X\_train\_counts.tolist()).toarray()

Sau bước này, mỗi hàng của tfidf tương ứng với feature vector của một bộ phim.

Tiếp theo, với mỗi user, chúng ta cần đi tìm những bộ phim nào mà user đó đã rated, và giá trị của các rating đó.

**import** numpy **as** np

**def** **get\_items\_rated\_by\_user**(rate\_matrix, user\_id):

"""

in each line of rate\_matrix, we have infor: user\_id, item\_id, rating (scores), time\_stamp

we care about the first three values

return (item\_ids, scores) rated by user user\_id

"""

y **=** rate\_matrix[:,0] *# all users*

*# item indices rated by user\_id*

*# we need to +1 to user\_id since in the rate\_matrix, id starts from 1*

*# while index in python starts from 0*

ids **=** np.where(y **==** user\_id **+**1)[0]

item\_ids **=** rate\_matrix[ids, 1] **-** 1 *# index starts from 0*

scores **=** rate\_matrix[ids, 2]

**return** (item\_ids, scores)

### **3.3. Tìm mô hình cho mỗi user**

Bây giờ, ta có thể đi tìm các hệ số của Ridge Regression cho mỗi user:

**from** sklearn.linear\_model **import** Ridge

**from** sklearn **import** linear\_model

d **=** tfidf.shape[1] *# data dimension*

W **=** np.zeros((d, n\_users))

b **=** np.zeros((1, n\_users))

**for** n **in** range(n\_users):

ids, scores **=** get\_items\_rated\_by\_user(rate\_train, n)

clf **=** Ridge(alpha**=**0.01, fit\_intercept **=** True)

Xhat **=** tfidf[ids, :]

clf.fit(Xhat, scores)

W[:, n] **=** clf.coef\_

b[0, n] **=** clf.intercept\_

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

*# predicted scores*

Yhat **=** tfidf.dot(W) **+** b

Dưới đây là một ví dụ với user có id là 10.

n **=** 10

np.set\_printoptions(precision**=**2) *# 2 digits after .*

ids, scores **=** get\_items\_rated\_by\_user(rate\_test, n)

Yhat[n, ids]

**print**('Rated movies ids :', ids )

**print**('True ratings :', scores)

**print**('Predicted ratings:', Yhat[ids, n])

Rated movies ids: [ 37 109 110 226 424 557 722 724 731 739]

True ratings: [3 3 4 3 4 3 5 3 3 4]

Predicted ratings: [3.18 3.13 3.42 3.09 3.35 5.2 4.01 3.35 3.42 3.72]

### **3.4. 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:

**def** **evaluate**(Yhat, rates, W, b):

se **=** 0

cnt **=** 0

**for** n **in** xrange(n\_users):

ids, scores\_truth **=** get\_items\_rated\_by\_user(rates, n)

scores\_pred **=** Yhat[ids, n]

e **=** scores\_truth **-** scores\_pred

se **+=** (e**\***e).sum(axis **=** 0)

cnt **+=** e.size

**return** sqrt(se**/**cnt)

**print** 'RMSE for training:', evaluate(Yhat, rate\_train, W, b)

**print** 'RMSE for test :', evaluate(Yhat, rate\_test, W, b)

RMSE for training: 0.908980456283

RMSE for test : 1.27032827004

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. Kết quả tốt hơn có thể được thấy trong bài tiếp theo: Collaborative Filtering.

**Tài liệu tham khảo**

1.Adomavicius, G. and A. Tuzhilin, 2005.

Toward the next generation of recommender systems: a survey of the stateof-the-art and possible extensions. IEEE Transactions on Knowledge And Data Engineering.

Link: <https://ieeexplore.ieee.org/document/1423975>

2. Adomavicius, G. And Y. Kwon, 2008.

Overcoming accuracy-diversity tradeoff in recommender systems: a variance-based approach. In Proceedings of the 18th Workshop on Information Technology and Systems, WITS 2008, Paris, France.

Link: <https://citeseerx.ist.psu.edu/viewdoc/download?doi=10.1.1.423.3952&rep=rep1&type=pdf>

3. Adomavicius, G. and Y. Kwon, 2010.

Improving recommendation diversity using ranking-based techniques. IEEE Transactions on Knowledge and Data Engineering.

Link: <https://ieeexplore.ieee.org/abstract/document/5680904>

4.Breese, J.S. and D. Heckerman, 1998.

Empirical analysis of predictive algorithms for collaborative filtering. Morgan Kaufmann, pp. 43–52.

Link: <https://arxiv.org/abs/1301.7363>

5.Herlocker J.L et al, 2004.

Evaluating collaborative filtering recommender systems. ACM Trans. Inf. Syst., vol. 22, no. 1, pp. 5–53.

Link: https://dl.acm.org/doi/10.1145/963770.963772