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

**KHOA ĐÀO TẠO CHẤT LƯỢNG CAO**

**NGÀNH CÔNG NGHỆ THÔNG TIN**

\_\_\_\_\_\_\_\_\_\_\_\_\_\_\_



**HỌC MÁY**

**ỨNG DỤNG THUẬT TOÁN MÁY HỌC ĐỂ PHÁT TRIỂN “RECOMMENDER SYSTEMS” TRONG WEBSITE THƯƠNG MẠI**

|  |  |  |
| --- | --- | --- |
| **Giảng viên hướng dẫn:** | TS. Vũ Quang Huy |  |
|  |  |  |
| **Thành viên:** | Nguyễn Thị Cẩm Thu | 15110137 |
|  | Nguyễn Thái Học | 15110052 |
|  | Lâm Thành Tài | 15110121 |

**Tp. Hồ Chí Minh, tháng 12 năm 2018**

**NHẬN XÉT CỦA GIẢNG VIÊN**

Ngày…Tháng…Năm 2018

Giáo viên

(Ký tên và ghi rõ họ,tên)

**LỜI CẢM ƠN**

Nhóm thực hiện đề tài xin chân thành cảm ơn Thầy Vũ Quang Huy giảng viên đã hướng dẫn thực hiện, cung cấp tài liệu tham khảo trong quá trình thực hiện đề tài. Mặc dù nhóm đã cố gắng hoàn chỉnh các yêu cầu cơ bản nhưng bài báo cáo còn rất nhiều thiếu sót nhóm chúng em rất mong nhận được những ý kiến cũng như bổ sung các vấn đề phục vụ cho việc xây dựng đề tài của nhóm hoàn chỉnh hơn từ Thầy.

Sau cùng, chúng em xin kính chúc Thầy dồi dào sức khỏe, niềm tin để tiếp tục thực hiện sứ mệnh cao đẹp của mình là truyền đạt kiến thức cho thế hệ mai sau.

Xin chân thành cảm ơn!

Nhóm thực hiện đề tài.

TP.HCM, tháng 12 năm 2018

Nhóm SV thực hiện

Nguyễn Thị Cẩm Thu

Nguyễn Thái Học

Lâm Thành Tài

**MỤC LỤC**

[CHƯƠNG 1. LẬP KẾ HOẠCH 6](#_Toc532901035)

[CHƯƠNG 2. INTRODUCTION 8](#_Toc532901036)

[2.1. Lý do chọn đề tài 8](#_Toc532901037)

[2.2. Lợi ích của đề tài 9](#_Toc532901038)

[CHƯƠNG 3. GOAL 10](#_Toc532901039)

[CHUONG 4. OBJECTIVES 11](#_Toc532901040)

[CHƯƠNG 5. THEORY 12](#_Toc532901041)

[5.1. Phương pháp 12](#_Toc532901042)

[5.1.1. Content – based filtering 12](#_Toc532901043)

[5.1.2. Collaborative filtering 16](#_Toc532901044)

[5.1.3. Hybrids filtering 21](#_Toc532901045)

[5.2. Chủ đề 21](#_Toc532901046)

[5.2.1. Thương mại điện tử 21](#_Toc532901047)

[CHƯƠNG 6. MODEL 23](#_Toc532901048)

[6.1. Flowchart 23](#_Toc532901049)

[6.2. Block Diagram 24](#_Toc532901050)

[CHƯƠNG 7. DEVELOP 25](#_Toc532901051)

[7.1. Dataset 25](#_Toc532901052)

[7.2. Xây dựng thuật toán 25](#_Toc532901053)

[7.2.1. Trích xuất dữ liệu 25](#_Toc532901054)

[7.2.2. Xây dựng feature vector 26](#_Toc532901055)

[7.2.3. Tìm mô hình cho mỗi user. 27](#_Toc532901056)

[7.2.4. Đánh giá mô hình 28](#_Toc532901057)

[CHƯƠNG 8. CONCLUSION 29](#_Toc532901058)

[8.1. Kết quả đạt được 29](#_Toc532901059)

[8.2. Hạn chế 29](#_Toc532901060)

[8.3. Hướng phát triển 29](#_Toc532901061)

[Tài liệu tham khảo 30](#_Toc532901062)

**MỤC LỤC BẢNG BIỂU**

[Bảng 2.1‑1: Kế hoạch công việc để thực hiện đề tài 7](#_Toc532896195)

**MỤC LỤC HÌNH ẢNH**

[Hình 5.1‑1: Recommender Systems 12](#_Toc528106169)

[Hình 6.1‑1: Flowchart 22](#_Toc528106170)

[Hình 6.2‑1: Block Diagram 23](#_Toc528106171)

# CHƯƠNG 1. LẬP KẾ HOẠCH

|  |  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- | --- |
| **STT** | **Giai đoạn** | **Công việc** | **Ngày bắt đầu** | **Ngày kết thúc (dự kiến)** | **Kéo dài (Duration)** | **Người thực hiện chính** | **Collaborators** |
| **1** | **1: Tìm hiểu thuật toán** | Tìm hiểu chung về thương mại điện tử | 04/10/2018 | 09/10/2018 | 6 | Nguyễn Thị Cẩm Thu |  |
| **2** | Tìm hiểu thuật toán Content-based systems: | 04/10/2018 | 09/10/2018 | 6 | Lâm Thành Tài | Nguyễn Thị Cẩm Thu (Tổng hợp) |
| *Giới thiệu.* |  |  |  |  |  |
| *Bản chất vấn đề.* |  |  |  |  |  |
| *Phương thức hoạt động.* |  |  |  |  |  |
| **3** | Tìm hiểu thuật toán Collaborative filtering: | 04/10/2018 | 09/10/2018 | 6 | Nguyễn Thái Học | Nguyễn Thị Cẩm Thu (Tổng hợp) |
| *Giới thiệu.* |  |  |  |  |  |
| *Bản chất vấn đề.* |  |  |  |  |  |
| *Phương thức hoạt động.* |  |  |  |  |  |
| **4** | **2: Code** | Code thuật toán Content-based systems **(\*)** | 11/10/2018 | 10/11/2018 | 31 |  |  |
| **5** | Code thuật toán Collaborative filtering | 11/10/2018 | 10/11/2018 | 31 | Nguyễn Thị Cẩm Thu | Nguyễn Thái Học |
|  | **Code web:** |  |  |  |  |  |
| **6** | Font end | 11/10/2018 | 10/11/2018 | 31 | Nguyễn Thị Cẩm Thu | Lâm Thành Tài |
| **7** | Back end | 11/10/2018 | 10/11/2018 | 31 | Lâm Thành Tài | Nguyễn Thái Học |
| **8** | **3: Kết hợp** | Kết hợp web và thuật toán | 26/10/2018 | 15/11/2018 | 8 |  |  |
| **9** | **4: Testing** | Kiểm tra kết quả thuật toán | 03/11/2018 | 15/11/2018 | 4 |  |  |
| **10** | **5: Document** | Slide, Report | 04/10/2018 | 15/11/2018 | 43 | Nguyễn Thị Cẩm Thu |  |

Bảng ‑: Kế hoạch công việc để thực hiện đề tài

# CHƯƠNG 2. INTRODUCTION

## Lý do chọn đề tài

Recommender Systems được cho là phổ biến trong xã hội công nghệ phát triển ngày nay. Mục đích của Recommender Systems chủ yếu là để tìm ra hoặc gợi ý ra những items mà người dùng muốn mua hoặc quan tâm đến. Các trang web phổ biến đã áp dụng Recommender Systems như Facebook, Amazon, Youtube, … và còn nhiều các website thương mại điện tử khác cũng đã và đang dần dần áp dụng Recommender Systems vào hệ thống.

Việc đưa ra các gợi ý cho người dùng thông qua một số lượng lớn các dữ liệu đã được cho trước nhằm nắm bắt các sở thích của người dùng cũng như của nhóm các người dùng khác nhau là cốt lõi của Recommender Systems. Có rất nhiều phương pháp để có thể tạo ra một hệ thống Recommender Systems hoàn chỉnh, ví dụ như Content – based filtering, Collaborative filtering và Hybrid filtering. Từ đó, có rất nhiều hàm thuật toán có thể được áp dụng trên dữ liệu để dự đoán sở thích của người dùng như user – based, item – based và model – based, là các hàm được sử dụng để đánh giá về sở thích của người dùng. Tùy theo khối lượng dữ liệu vào khác nhau mà chúng ta có thể sử dụng các phương pháp và các hàm thuật toán khác nhau để áp dụng vào đó.

Dự đoán sở thích người dùng sẽ thông qua việc đánh giá sản phẩm của họ trên những website hoặc cách mà họ quan tâm đối với sản phẩm đó qua các hình thức như tìm kiếm, để lại bình luận hay đánh dấu “yêu thích” sản phẩm. Dựa vào những yếu tố trên nên nhóm đã quyết định chọn đề tài này làm đồ àn môn học Machine Learning cũng như Đồ án 3 nhằm mục đích tiếp thu thêm các kiến thức về Recommender Systems và củng cố thêm các kiến thức về mặt thuật tán cũng như cách tạo ra một trang web có áp dụng Recommender Systems.

## Lợi ích của đề tài

Recommender Systems được cho là mang tầm quan trọng trong cuộc sống ngày nay của chúng ta vì trong thời đại công nghiệp hóa – hiện đại hóa đất nước thì khối lượng dữ liệu lưu trữ ngày một nhiều hơn và phức tạp hơn thì Recommender Systems có thể giúp chúng ta trong việc giới hạn phạm vi của dữ liệu bằng cách đánh giá hoặc đưa gợi ý cho người dùng nhằm tiết kiệm thời gian cho người dùng nhiều hơn và xử lý được nhiều khối dữ liệu phức tạp hơn.

Việc ứng dụng Recommender Systems vào các website – hiện đang phổ biến và phát triển mạnh mẽ nên có thể sẽ đem lại cho nhóm chúng em nhiều kiến thức mới hơn và cái nhìn khác hơn về Machine Learning. Ngoài ra thì từ việc chọn Recommender Systems làm đề tài môn học Machine Learning sẽ giúp nhóm hiểu được thêm vè việc áp dung thuật toán vào các trang web hiện tại, giúp nhóm học được nhiều về các kiến thức trong môn học này.

# CHƯƠNG 3. GOAL

Xây dựng một hệ thống có thể thu thập thông tin về việc phân tích dữ liệu của người dùng sau đó gợi ý cho người dùng dựa trên những đánh giá của họ về các dữ liệu trước đó nhằm gợi ý đúng các sản phẩm cũng như các mặt hàng mà họ đã quan tâm.

Việc áp dụng Recommender Systems sẽ làm cho hệ thống cá nhân hóa và tác động mạnh hơn đến quan điểm của người dùng. Recommender Systems sẽ tạo ra được nhiều lòng tin và lợi ích hơn khi được áp dụng vào các trang web, nếu thành công thì có thể phục vụ cho người dùng mọi lúc, mọi nơi.

# CHUONG 4. OBJECTIVES

* Về phía Web Programming:
  + Học các kiến thức về Spring Boot, RESTFul API.
  + Tìm hiểu về HTML5 và Boostrap 3.3.7.
  + Xây dựng website.
* Về phía Machine Learning:
  + Tìm hiểu các loại thuật toán của Recommender Systems.
  + Áp dụng thuật toán Collaborative filtering hoặc Content – based filtering vào web app.
  + Phân tích các dữ liệu từ việc dự đoán đánh giá của người dùng đối với từng loại sản phẩm khác nhau.
  + Tìm hiểu về cách áp dụng các thuật toán vào web app và thử nghiệm web.

# CHƯƠNG 5. THEORY

## Phương pháp

### Content – based filtering

#### Giới thiệu Recommender Systems

Dựa theo các hiện tượng thực tế ngày nay thì thuật toán trên được áp dụng cho rất nhiều vấn đền ngoài xã hội nói chung và ngành công nghệ thông tin nói riêng. Hầu hết các trang web ứng dụng mua sắm, xem phim, đọc sách báo hay các web về thương mại điện tử đều áp dụng thuật toán này nhằm mục đích gợi ý cho người dùng về các mặt hàng, các sản phẩm, các loại phim mà hệ thống tự đề ra dựa theo những hoạt động mà họ đã thực hiện trước đó dựa theo mỗi sở thích của các khách hàng khác nhau.

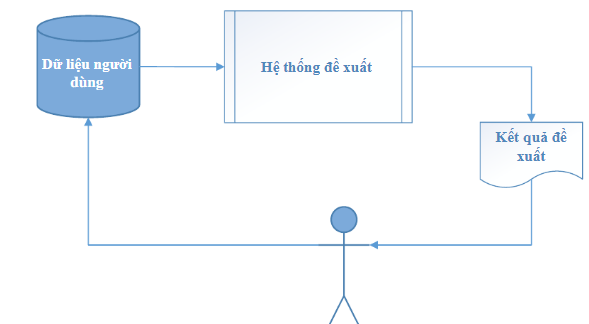
Các thuật toán trên được gọi chung là “Recommender Systems” – tạm dịch là “Hệ thống gợi ý”. Có hai thực thể chính trong “Recommender Systems” (RS) là Users và Items:

* Users: ở đây là người dùng hay còn gọi là khách hàng (lấy ví dụ điển hình như các tài khoản trên các trang web bán hàng, xem phim hay các kênh trên youtube).
* Items: ở đây là các sản phẩm (lấy ví dụ như là một bộ phim, một quyển sách, một bài hát hay một video clip)

Recommender Systems là thuật toán có nhiệm vụ dự đoán mức độ quan tâm của người dùng đối với một sản phẩm nào đó, thông qua đó có thể đưa ra các gợi ý phù hợp với từng người dùng.

Thông thường thì các cửa hàng truyền thống sẽ bị ảnh hưởng bởi “Long Tail” - ở đây tức là việc tạo ra doanh thu trên tổng số các sản phẩm hiện có trong cửa hàng đó, nhưng đôi khi một số nhỏ sản phẩm có thể tạo ra lượng doanh thu lớn còn số lượng lớn các sản phầm còn lại trong cửa hàng có thể sẽ chỉ tạo ra một lượng doanh thu nhỏ cho cửa hàng. “Long Tail” ở đây muốn nhắc đến đó là vấn đề về sản phẩm kém phổ biến hơn, bán chậm hơn.

Vì thế mà các cửa hàng online được ra đời và giải quyết được vấn đề đó, do có thể chứa được khối lượng lớn các thông tin nên việc trưng bày toàn bộ sản phẩm lên một trang web có thể là một điều không quá khó. Việc bán hàng trên các cửa hàng online có thể sẽ tiện lợi hơn, chi phí thấp hơn nhưng doanh thu sẽ cao hơn, mang lại cho người dùng sự tiện lợi và giải quyết được vấn đề nhu cầu cho mỗi người dùng.



Hình ‑: Recommender Systems

#### Khái niệm Content – based filtering

Content-based Systems là thuật toán nhằm đánh giá các đặc tính của một items được gợi ý. Lấy ví dụ về một sản phẩm như quần áo, có thể có nhiều khách hàng sẽ thích áo tay dài vậy nên dựa vào đó mà chúng ta có thể gợi ý các sản phẩm khác trong cửa hàng của chúng ta mà cùng tính chất với sản phẩm mà khách hàng đang muốn hướng đến.

Thuật toán này đòi hỏi phải có sự sắp xếp các sản phẩm vào từng nhóm hoặc phải đi tìm đặc trưng của các loại sản phẩm khác nhau. Tuy nhiên thì một vài sản phẩm có thể sẽ không biết thuộc nhóm nào hoặc việc tìm ra đặc trưng của sản phẩm đó đôi khi là bất khả thi.

#### Bản chất của thuật toán

Về bản chất thì chúng ta có thể coi thuật toán Content-based Systems như một người môi giới. Đơn giản vì thuật toán có thể dự đoán được sở thích của người dùng và tìm kiếm các tiềm năng có chung sở thích đó để có thể gợi ý cho người dùng.

**Ví dụ:**

* Bạn thích xem phim hành động, mà bạn thì hoàn toàn không thể nào biết hết được tất cả các phim hành động khác nhau, vì thế mà bạn sẽ cần một hệ thống gợi ý cho bạn việc đó, nó có thể giúp bạn lọc ra các bộ phim hành động để cuốn hút bạn dựa theo sở thích của bạn trước đó.

Nói cách khác, khi bạn mua một sản phẩm nào đó thì sẽ có hệ thống gợi ý cho bạn các sản phẩm tương ứng và tính chất tương tự với sản phẩm mà bạn đã mua.

#### Các thành phần đặc trưng trong thuật toán

**Utility Matrix**

Do có hai thực thể chính trong thuật toán là users và items nên mỗi users sẽ có một mức độ quan tâm đối với mỗi items khác nhau. Các items được users quan tâm có thể sẽ được đánh dấu lại và ứng với mỗi items thì sẽ có các mức độ đánh giá khác nhau bởi các users khác nhau. Ma trận Utility Matrix được tạo nên thông qua sự đánh giá đó.

Ma trận Utility Matrix là một ma trận có chứa tất cả các dữ liệu đã biết và các dữ liệu chưa dự đoán được. Thông thường thì lượng items được đánh giá có thể sẽ là một phần nhỏ trong tổng số các items và lượng items chưa được đánh giá có thể sẽ chiếm phần lớn hơn, do mỗi users có thể chỉ đánh giá một lượng nhỏ các items và thậm chí là có những user không đánh giá bất kỳ items nào. Vấn đề trên sẽ tạo cho ma trận Utility Matrix của chúng ta chứa một lượng lớn các dữ liệu chưa được dự đoán và các dữ liệu đã được dự đoán có thể chỉ chiếm một lượng nhỏ trong số đó.

Ma trận Utility Matrix có thể sẽ giúp thuật toán hoạt động tốt hơn nếu như số ô dữ liệu đã được dự đoán có được nhiều hơn. Vì thế mà các trang web bán hàng luôn tích cực gửi lời mời đánh giá sản phẩm cho bên phía họ để họ có thể nhận biết được chất lượng sản phẩm của cửa hàng bên mình và cũng như biết được sở thích của người dùng mà từ đó có thể đưa ra các chiến lược quảng bá sản phẩm tốt hơn, đúng đối tượng hơn.

Ma trận Utility Matrix được cho là điều không thể thiếu để có thể xây dựng gợi ý cho người dùng, có nhiều cách để có thể xây dựng ma trận trên đó là dựa trên sự đánh giá sản phẩm của một user và dựa trên hành vi của một user.

**Item Profiles**

Do Content – based Systems là thuật toán dựa trên nội dung của mỗi items nên việc xây dựng một “profile” cho một item là điều cần thiết – feature vector.

Từ ma trận Utility Matrix thì chúng ta có thể đơn giản hóa bài toán bằng cách xây dựng một feature vector hai chiều.

Nếu Utility Matrix có cấu trúc là một feature vector hai chiều thì chúng ta sẽ làm bài toán đi theo hướng Regression hoặc Classification tùy theo biến đánh giá của mỗi users ứng với mỗi items và tùy theo ứng dụng mà chọn thuật toán.

**Loss Function**

Đặt số user là N, số item là M, Utility Matrix là **Y**, thành phần ở cột thứ m và hàng thứ n của ma trận **Y** là mức độ quan tâm của một user thứ n đối với một item thứ m. Vì thế mà ma trận **Y** sẽ bị khuyết rất nhiều chỗ trống – những dữ liệu chưa được dự đoán hoặc chưa được đánh giá bởi người dùng. Gọi **R** là ma trận gồm 1 và 0 (rating or not rating), cụ thể hơn thì nếu r­ij­ = 1 tức là item thứ i rated bởi user thứ j, và rij = 0 trong trường hợp item thứ i not rated bởi user thứ j.

Mức độ quan tâm của một user đối với một item có thể tính bằng một hàm tuyến tính:

ymn = xm + bn

Trong đó:

* xm là một vector hàng.
* là một vector cột.
* bn là bias.
* ymn là thành phần ở hàng thứ m và cột thứ n của ma trận **Y**.

Xét 1 user thứ n bất kì và training set được cho là tập hợp các thành phần đã được điền của yn , ta có thể xây dựng hàm mất mát như sau:

Trong đó:

* là một regularization term với λ là một tham số dương.
* là số lượng các item mà user thứ n đã rated, là tổng các phần tử trên cột thứ n của ma trận **R**.

Do biểu thức hàm mất mát chỉ phụ thuộc vào các item được đánh giá nên ta có thể rút ngắn nó lại bằng cách:

* Đặt là vetor con của , hay có thể hiểu là các thành phần chưa được đánh giá ở cột thứ n trong ma trận **Y**.
* Đặt là ma trận con của ma trận đặc trưng X, là ma trận chứa các thành phần dữ liệu đã được đánh giá bởi user thứ n.

Ta sẽ có biểu thức viết gọn của hàm mất mát như sau:

Trong đó:

* là vector cột gồm các thành phần là 1.
* sẽ được xác đinh nếu như user thứ n đánh giá ít nhất một item.

#### Phương thức hoạt động

Như đã tìm hiểu các thành phần cũng như bản chất của thuật toán trên nên việc xác định phương thức hoạt động là điều khả thi.

Thuật toán Content – based Systems thực chất là một thuật toán dùng để gợi ý các items cho các users thông qua nội dung, đặc tính và thông tin của item đó.

Thuật toán thiết lập nên một ma trận bao gồm các thành phần đã được đánh giá và chưa được đánh giá của các item bởi các user và áp dụng thuật toán để có thể hoàn thiện các thành phần bị khuyết trong ma trận đó (các dữ liệu chưa được đánh giá).

Cách tạo ra ma trận sẽ là bước khởi đầu cho việc gợi ý items cho user nhằm mục đích quảng bá đúng người và đúng sản phẩm họ cẩn thiết. Ngoài ra thì thuật toán có thể dựa vào số lần truy cập hoặc tìm hiểu thông tin về một item của một user để có thể dựa vào đó mà gợi ý cho user những item có tính chất tương ứng.

### Collaborative filtering

#### Giới thiệu Collaborative filtering

Collaborative Filtering là một phương pháp gợi ý item với ý tưởng chính dựa trên hành vi của các user tương tự khác lên cùng một item. Việc suy ra này được thực hiện dựa trên ma trận similarity để đo độ giống nhau giữa các user.

Để tính ma trận similarity, trước tiên ta cần chuẩn hóa dữ liệu. Phương pháp phổ biến nhất là mean offset, tức trừ các ratings đi giá trị trung bình mà một user đưa ra cho các item.

Một hướng tiếp cận khác là thay vì đi tìm các user gần giống với một user (user – user Collaborative Filtering), người ta đi tìm các item gần với một item cho trước (item – item Collaborative Filtering). Trên thực tế, item – item Collaborative Filtering thường cho kết quả chính sác hơn so với user – user Collaborative Filtering.

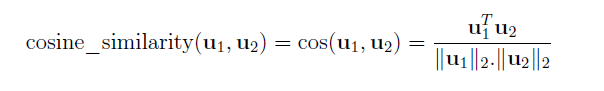
#### Khái niệm về Neighborhood-based collaborative filtering

Neighborhood-based collaborative filtering 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 của các user này tới các item khác mà hệ thống đã biết.

Việc xác định mức độ quan tâm của mỗi user tới một item dựa trên mức độ quan tâm của user tương tự tới item đó còn được gọi là *user-user collaborative filtering*. Có một hướng tiếp cận khác được cho là làm việc hiệu quả hơn là *item-item collaborative filtering*. Trong hướng tiếp cận này, thay vì xác định sự giống nhau giữa các user, hệ thống sẽ xác định sự giống nhau giữa các item. Từ đó, hệ thống gợi ý những item gần giống với những item mà user đó có mức độ quan tâm cao.

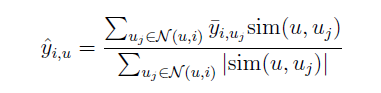
#### Thuật toán

* **user-user collaborative filtering**
  + Hàm xác định độ giống nhau:



Trong đó: U1,2 là các vector tương ứng với user 1 và user 2.

* + Điền các giá trị khuyết trong ma trận utility:



Trong đó: là tập hợp user gần giống nhất, tức có­­­­­­­­­ similarity cao nhất của đã đánh giá .

* **item-item collaborative filtering**

Có một các tiếp cận khác, thay vì tìm sự giống nhau giữa các user, ta có thể tìm sự giống nhau giữa các item. Từ đó nếu một user thích một item thì hệ thống nên gợi ý các item tương tự với user đó.

Quy trình dự đoán các đánh giá bị khuyết cũng tương tự như trong user-user Collaborative Filtering, chỉ khác là bây giờ ta cần tính độ giống nhau giữa các hàng.

Về mặt tính toán, item-item Collaborative Filtering có thể nhận được từ user-user Collaborative Filtering bằng cách chuyển vị ma trận utility, và coi như item đang đánh giá ngược user. Sau khi tính ra kết quả cuối cùng ta lại chuyển vị một lần nữa để thu được kết quả.

#### Giới thiệu Matrix Factorization (Matrix Decomposition)

Nhắc lại rằng trong Content – based Recommender Systems, mỗi item được mô tả bằng một vector x được gọi là item profile. Trong phương pháp này, ta cần tìm một vector hệ số w tương ứng với mỗi user sao cho rating đã biết mà user đó cho item xấp xỉ với:

Với cách làm trên, Utility Matrix Y, giả sử đã được điền hết, sẽ xấp xỉ với:

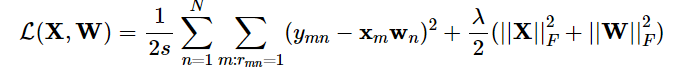
Với M, N lần lượt là số items và số users.

#### Xây dựng và tối ưu hàm mất mát của Matrix Factorization

* **Hàm mất mát:**

Tương tự như trong Content-based recommendation system, việc xây dựng hàm mất mát cũng được dựa trên các thành phần đã được điền của Utility Matrix Y, có khác một chút là không có thành phần bias và biến tối ưu là cả X và W.

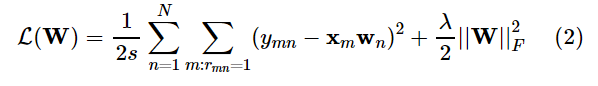
Việc xây dựng hàm mất mát cho Matrix Factorization là tương đối dễ hiểu:



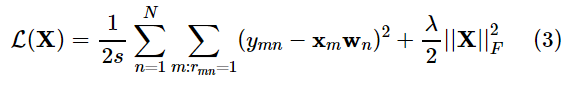
Trong đó rmn = 1 nếu item thứ m đã được đánh giá bởi user thứ n, là căn bậc hai của tổng bình phương tất cả các phần tử của ma trận s là toàn bộ số ratings đã có Thành phần thứ hai trong hàm mất mát phía trên là regularization.

* **Tối ưu hàm mất mát:**

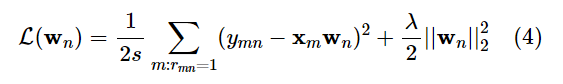
Khi cố định X, việc tối ưu W chính là bài toán tối ưu trong Content-based Recommendation Systems:



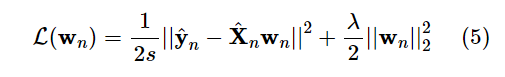
Khi cố định W, việc tối ưu X được đưa về tối ưu hàm:



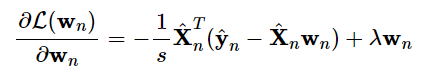
Chúng ta có thể thấy rằng bài toán (2) có thể được tách thành N bài toán nhỏ, mỗi bài toán ứng với việc đi tối ưu một cột của ma trận W:



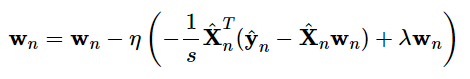
Vì biểu thức trong dấu ∑ chỉ phụ thuộc vào các items đã được rated bởi user đang xét, ta có thể đơn giản nó bằng cách đặt  là ma trận được tạo bởi các hàng tương ứng với các items đã được rated đó, và  là các ratings tương ứng. Khi đó:



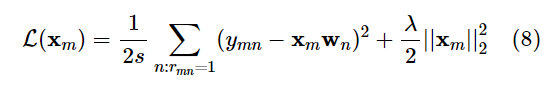
và đạo hàm của nó:



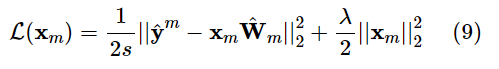
Công thức cập nhật cho mỗi cột của W là:



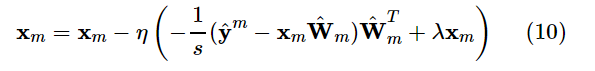
Tương tự như thế, mỗi cột của X, tức vector cho mỗi item, sẽ được tìm bằng cách tối ưu:



Đặt  là ma trận được tạo bằng các cột của W ứng với các users đã đánh giá item đó và  là vector ratings tương ứng. (8) trở thành:



Tương tự như trên, công thức cập nhật cho mồi hàng của X sẽ có dạng:



#### Giới thiệu Singular Value Decomposition (SVD)

* **Truncated SVD:**

Singular Value Decomposition (SVD) là một trong những phương pháp Matrix Factorization của Đại số tuyến tính.

Một ma trận Am×n bất kỳ đều có thể phân tích thành dạng:



Trong đó, U,V là các ma trận trực giao, Σ là ma trận đường chéo không vuông với các phần tử trên đường chéo σ1≥σ2≥⋯≥σr≥0=0=⋯=0và r là rank của ma trận A.

Viết lại biểu thức (5)(5) dưới dạng tổng của các ma trận rank 1:

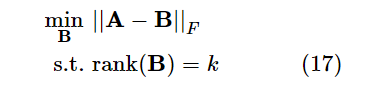


Chú ý rằng trong ma trận Σ, các giá trị trên đường chéo là không âm và giảm dần σ1≥σ2≥…,≥σr≥0=0=⋯=0. Thông thường, chỉ một lượng nhỏ các σiσimang giá trị lớn, các giá trị còn lại thường nhỏ và gần 0. Khi đó ta có thể xấp xỉ ma trận A bằng tổng của k<r ma trận có rank 1:



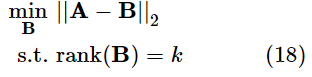
* **Best Rank k Approximation:**

Người ta chứng minh được rằng Ak chính là nghiệm của bài toán tối ưu:

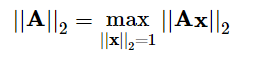


và như đã chứng minh ở trên .

Nếu sử dụng norm 2 của ma trận thay vì Frobenius norm để đo sai số, AkAk cũng là nghiệm của bài toán tối ưu:



và sai số:. Định nghĩa của norm 2 của một ma trận là:



Đây là lý do căn bậc hai của tổng bình phương của các phần tử của một ma trận không được gọi là norm 2 như đối với vector.

Nếu bạn muốn biết thêm:



tức norm 2 của một ma trận chính là singular value lớn nhất của ma trận đó.

**Frobenius norm và norm 2 là hai norms được sử dụng nhiều nhất trong ma trận. Như vậy, xét trên cả hai norm này, Truncated SVD đều cho xấp xỉ tốt nhất. Vì vậy Truncated SVD còn được gọi là Best low-rank Approximation.**

### Hybrids filtering

Những phương pháp khác nhau đều có những điểm mạnh cũng như điểm yếu của nó. Để tận dụng những điểm mạnh và hạn chế điểm yếu của những tiếp cận khác nhau, nhiều nghiên cứu đã tập trung phát triển các Recommender Systems dựa trên việc kết hợp Content – based filtering và Collaborative filtering.

Netflix là một ví dụ tốt về việc sử dụng hybrid recommender systems. Trang web đưa ra các đề xuất bằng cách so sánh thói quen xem và tìm kiếm của những người dùng tượng tự (collaborative filtering) cũng như bằng cách cung cấp phim có đặc điểm gần với những bộ phim mà người dùng đánh giá cao (content – based filtering).

## Chủ đề

### Thương mại điện tử

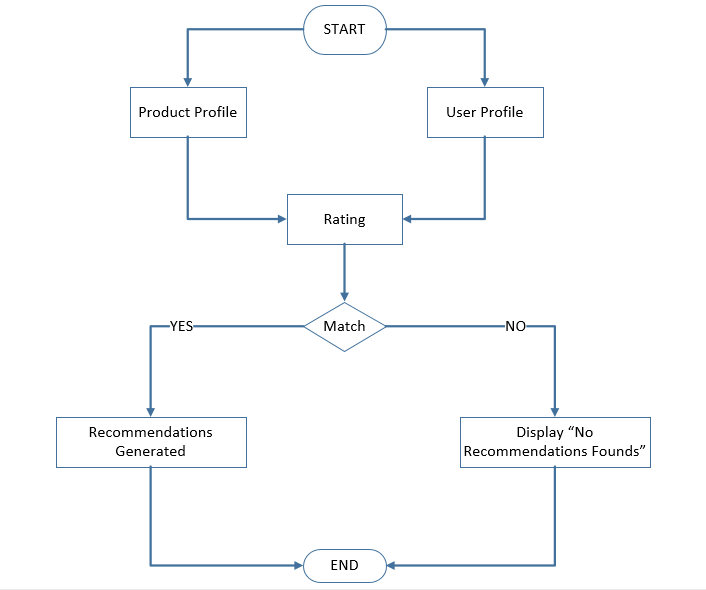
Thương mại điện tử là hình thức bán hàng hóa và dịch vụ trên các hệ thống điện tử như Internet và các mạng máy tính. Một số khái niệm được định nghĩa bởi các tổ chức uy tín thế giới như sau:

* Theo Tổ chức Thương mại thế giới (WTO), “Thương mại điện tử bao gồm việc sản xuất, quảng cáo, bán hàng và phân phối sản phẩm được mua bán và thanh toán trên mạng Internet, nhưng được giao nhận một cách hữu hình, cả các sản phẩm giao nhận cũng như những thông tin số hóa thông qua mạng Internet”.
* Theo Ủy ban Thương mại điện tử của Tổ chức Hợp tác kinh tế châu Á – Thái Bình Dương (APEC) định nghĩa: “Thương mại điện tử liên quan đến các giao dịch thương mại trao đổi hàng hóa và dịch vụ giữa các nhóm (cá nhân) mang tính điện tử chủ yếu thông qua các hệ thống có nền tảng dựa trên Internet”. Các kỹ thuật thông tin liên lac có thể là email, EDI, Internet và Extranet có thể được dùng để hỗ trợ thương mại điện tử.
* Theo Ủy ban châu Âu: “Thương mại điện tử có teher được định nghĩa chung là sự mua bán, trao đổi hàng hóa hay dịch vụ giữa các doanh nghiệp, gia đình, cá nhân hay tổ chức tư nhân bằng các giao dịch điện tử thông qua mạng Internet hay các mạng máy tính trung gian (thông tin liên lạc trực tuyến). Thuật ngữ bao gồm việc đặt hàng và dịch thông qua mạng máy tính, nhưng thanh toán và quá trình vận chuyển hàng hay dịch vụ cuối cùng có thể thực hiện trực tuyến hoặc bằng phương pháp thủ công”.

Tóm lại, thương mại điện tử chỉ xảy ra trong môi trường kinh doanh mạng Internet và các phương tiện điện tử giữa các nhóm (cá nhân) với nhau thông qua các công cụ, kỹ thuật và công nghệ điện tử.

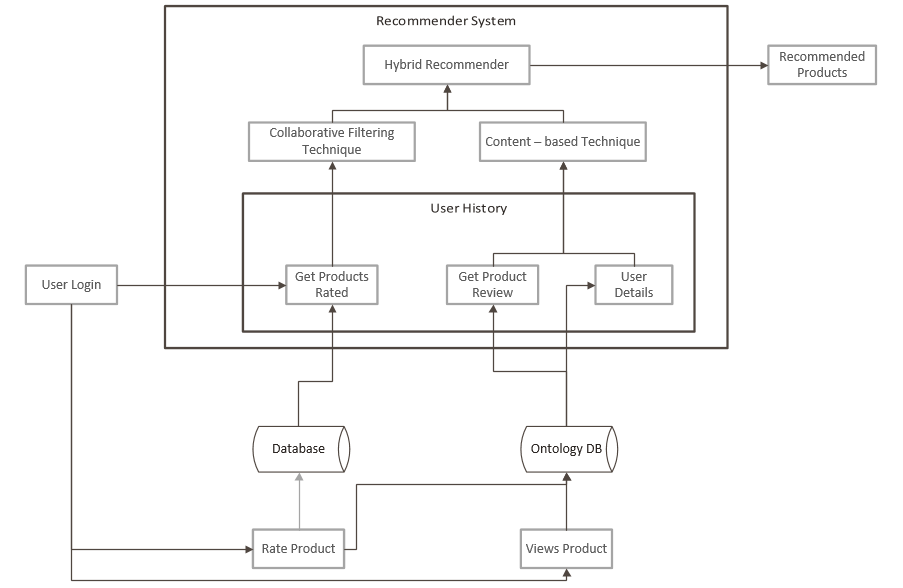
# CHƯƠNG 6. MODEL

## Flowchart



Hình ‑: Flowchart

## Block Diagram



Hình ‑: Block Diagram

# CHƯƠNG 7. DEVELOP

## Dataset

Đề tài sử dụng bộ cơ sở dữ liệu MoviLens 100k được công bố năm 1998 bởi GroupLens. Bộ cơ sở dữ liệu bao gồm 100 000 ratings từ 948 users cho 1682 bộ phim. Bộ cơ sỏ dữ liệu này gồm có các bộ dữ liệu nhỏ sau:

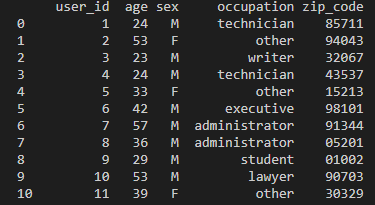
* U.data: Chứa toàn bộ rating của 943 users cho 1682 bộ phim. Mỗi user đánh giá ít nhất 20 bộ phim.
* Ua.base, ua.test, ub.base, ub.test: Là hai cách chia toàn bộ sữ liệu ra thành hai tập con, một cho training, một cho test.
* U.user: Chứa các thông tin về user bao gồm: id, tuổi, giới tính, nghề nghiệp, zipcode.
* U.genre: Chứa tên 19 thê 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, Weatern.
* U.item: Thông tin về mỗi bộ phim gồm cacsc thongot in sau: id của phim, tên phim, ngày phát hành, link trên imdb và các số nhị phân 0 và 1 thẻ hiện cho các thể loại phim mà bộ phim đó thuộc vào.

## Xây dựng thuật toán

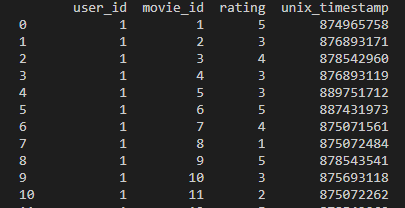
### Trích xuất dữ liệu

Từ bộ cơ sở dữ liệu MovieLens 100k chúng ta trích xuất các dữ liệu mà chúng ta cần sử dụng. Chúng ta sẽ lấy các dữ liệu sau:

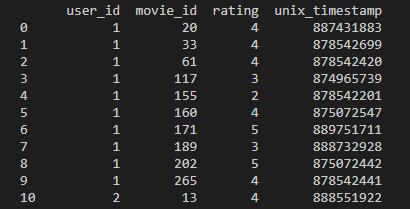
* U.user



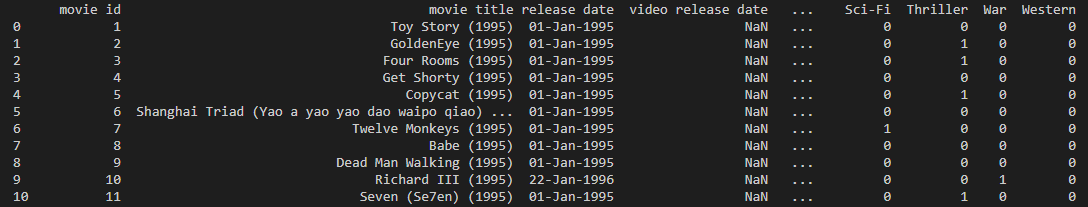
* Ua.base



* Ua.test



* U.item



Để có thể trích xuất các thông tin của các file dữ liệu trong bộ dữ liệu ta sử dụng thư viện “pandas” để trích xuất. Cài đặt “pandas” bằng cách “pip install pandas”.

Sau khi có được các thông tin của user, item chúng ta xây dựng feature vector cho mỗi item.

### Xây dựng feature vector

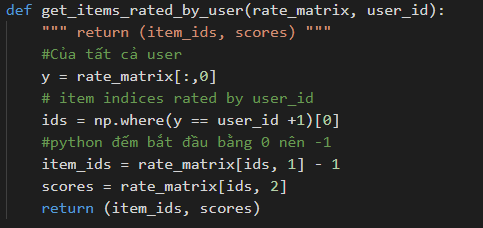
Chúng ta xây dựng feture vector dựa vào thể loại của các bộ phim, vì thế ta quan tâm đến mười chín gái trị nhị phân của mỗi item. Sau khi đã chuyển các thể loại item về dạng ma trận và trích mười chín giá trị thể loại, ta bắt đầu xây dựng feature vector cho mỗi item dựa trên ma trận thể loại phía trên và feature TF-IDF.

Để sử dụng TF-IDF chúng ta sử dụng thư viện “sklearn” để thực hiện.



Sau khi đã có feature vector của một bộ phim ta cần đi tìm những bộ phim mà user đó đã đánh giá và giá trị các rating đó.

Sau khi đã xây dựng thành công feature vector cho item, chúng ta sẽ đi tìm mô hình cho mỗi user .



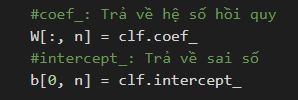
### Tìm mô hình cho mỗi user.

Bây giờ ta đi tìm các hệ số của Ridge Regression cho mỗi user.

Sau khi tính được các hệ số của Ridge là hệ số hồi quy và sai số. Chúng ta sẽ tiến hành tìm phương trình hồi quy tuyến tính:



Sau đó trả về hệ số hồi quy và trả về sai số:



Ratings cho mỗi items được dự đoán bằng cách tính:



Sau khi có đầy đủ các giá trị, ta sẽ thử vơi user có id là 100 thì ta sẽ nhận được kết quả như sau:



Sau khi đã tính được giá trị rating của item mà user đó đánh giá. Ta sẽ đánh giá lại mô hình chúng ta đã xây dựng.

### Đánh giá mô hình

Để có thể đánh giá mô hình, chúng ta sử dụng Root Mean Squared Error (RMSE) tuacws là căn bậc hai của trung bình cộng bình phương của lỗi. Khi đó, lỗi được tính bằng hiệu của true rating và predicted rating.



Như kết quả ta thấy, sai số vào khoảng 0.91 với tập training và 1.27 với tập test. Lý do vì chúng ta đã đơn giản hóa mô hình đi rất nhiều nhưng nó là phương pháp đơn giản nhất trong việc xây dựng hệ thống Recommendation System. Nó xây dựng mô hình cho mỗi user mà không phụ thuộc vào các users khác

# CHƯƠNG 8. CONCLUSION

## Kết quả đạt được

* Tìm hiểu được các loại thuật toán cho mô hình Recommendation System.
* Tìm và hiểu rõ phương pháp Content-base.
* Xây dựng thành công thuật toán để xây dựng Recommendation từ bộ cơ sở dữ liệu MovieLens 100k.

## Hạn chế

* Do sử dụng ngôn ngữ lập trình mới so với các thành viên trong nhóm nên kết quả đạt được của đề tài chưa được tốt như sự mong đợi là xây đựng thành ứng dụng.

## Hướng phát triển

* Tìm hiểu hệ thống Recommendation với ngôn ngũ mới phù hợp hơn với hình hình của nhóm và mục đích ứng dụng nó vào các đề tài rộng hơn sau này.

# Tài liệu tham khảo

1. MachineLearning cơ bản: <https://machinelearningcoban.com/>
2. StackOverflow: <https://stackoverflow.com/>
3. Viblo: <https://viblo.asia/>
4. Python: <https://www.python.org/>