TRƯỜNG ĐẠI HỌC GIAO THÔNG VẬN TẢI **CỘNG HÒA XÃ HỘI CHỦ NGHIÃ VIỆT NAM**

**PHÂN HIỆU TẠI THÀNH PHỐ HỒ CHÍ MINH** Độc lập – Tự do – Hạnh phúc

# NHIỆM VỤ THIẾT KẾ TỐT NGHIỆP

BỘ MÔN: **CÔNG NGHỆ THÔNG TIN**

-------\*\*\*-------

**Mã sinh viên:** 5451074063 **Họ tên SV:**  Phạm Công Quân

**Khóa:** 54 **Lớp:** CQ.54.CNTT

1. **Tên đề tài.**

**NGHIÊN CỨU THUẬT TOÁN COLLABORATIVE FILTERING VÀ ỨNG DỤNG THUẬT TOÁN VÀO GỢI Ý BÀI NHẠC**.

1. **Mục đích, yêu cầu.**

* **Mục đích:**

Xây dựng hệ thống khuyến nghị sản phẩm nhằm giúp cho cửa hàng có thể bán được nhiều sản phẩm hơn làm tăng thu nhập cho doanh nghiệp và cửa hàng, giúp cho họ có thể bán được những sản phẩm trong kho mà không cần phải trưng bày ra nhiều, bên cạnh đó nó cũng giúp cho người dùng có thể dễ dàng lựa chọn được các sản phẩm mà mình ưa thích mà không cần phải tự mình tìm kiếm.

* **Yêu cầu:**

+ Nghiên cứu thuật toán trong hệ thống khuyến nghị cụ thể là thuật toán collaborative filtering.

+ Đánh giá thuật toán trên dữ liệu Movielens - 100k.

+Áp dụng thuật toán vào trong gợi ý bài nhạc.

**Nội dung và phạm vi đề tài.**

* **Nội dung:**

+ Tổng quan về ngôn ngữ Java và frame work Spring.

+ Giới thiệu về apache mahout.

+ Tổng quan về machine learning.

+ Tổng quan về hệ thống khuyến nghị.

+ Phân tích và đánh giá thuật toán.

+ Áp dụng thuật toán vào gợi ý bài nhạc dựa trên thư viện mahout.

* **Phạm vi đề tài:**

Nghiên cứu thuật toán collaborative filtering và áp dụng thuật toán này vào gợi ý bài nhạc bằng việc sử dụng thư viện mahout.

1. **Công nghệ, công cụ và ngôn ngữ lập trình.**

* **Công nghệ sử dụng:** Spring framework, JPA, Mahout library, Bootstrap, Movielens-100k.
* **Công cụ:** Spring STS Tool, Tomcat, Mysql workbench.
* **Ngôn ngữ lập trình:** JAVA.
* **Cơ sở dữ liệu:** MySQL.

1. **Các kết quả chính dự kiến sẽ đạt được và ứng dụng.**

+ Quyển báo cáo đề tài tốt nghiệp.

+ Hiểu được thuật toán collaborative filtering.

+ Áp dụng được thuật toán này vào gợi ý bài nhạc.

1. **Giáo viên và cán bộ hướng dẫn.**

Họ tên: TRẦN PHONG NHÃ

Đơn vị công tác: Bộ môn Công nghệ Thông tin – Phân hiệu Trường Đại học Giao thông Vận tải.

Điện thoại: 0906761014 Email: tpnha@utc2.edu.vn

|  |  |
| --- | --- |
| **Ngày ...... tháng ...... năm 2018**  **BM Công nghệ Thông tin** | **Đã giao nhiệm vụ TKTN**  **Giảng viên hướng dẫn** |
| **Trần Quang Hải Bằng** | **Trần Phong Nhã** |

Đã nhận nhiệm vụ TKTN

Sinh viên: Phạm Công Quân Ký tên:

Điện thoại: 01659372062 Email: quanpham1110@gmail.com

**LỜI CẢM ƠN**

Lời đầu tiên cho phép em gửi tới Quý Thầy Cô của **Bộ môn Công nghệ Thông tin – Phân hiệu Trường Đại học Giao thông Vận tải** lời chúc sức khỏe và lòng biết ơn sâu sắc nhất, cảm ơn Quý Thầy Cô trong suốt quá trình em học tập và rèn luyện tại trường đã cùng với những tâm huyết, tri thức và hiểu biết của mình để truyền đạt, giảng dạy và chỉ bảo tận tâm cho em những kiến thức quý báu về chuyên môn cũng như về đời sống. Đặc biệt em xin gửi lời cảm ơn đến thầy **Trần Phong Nhã**, là người Thầy đã tận tâm hướng dẫn, chỉ bảo và góp ý cho em qua những buổi nói chuyện, thảo luận về đồ án tốt nghiệp với đề tài “**NGHIÊN CỨU THUẬT TOÁN COLLABORATIVE FILTERING VÀ ỨNG DỤNG THUẬT TOÁN VÀO GỢI Ý BÀI NHẠC**”. Tuy đề tài không được lớn nhưng nếu không được sự hướng dẫn và chỉ bảo tận tình của thầy thì em nghĩ đồ án tốt nghiệp của em khó có thể hoàn thiện được.

Mặc dù em đã nổ lực để hoàn thành đồ án một cách tốt nhất, nhưng chắc chắn sẽ có những thiếu sót không thể tránh khỏi. Em kính mong nhận được sự cảm thông và những ý kiến đóng góp chân thành từ phía Quý Thầy Cô.

Lời sau cùng, em cũng không biết nói gì hơn ngoài kính chúc Quý Thầy Cô trong **Bộ môn Công nghệ Thông tin** và đặc biệt là Thầy **Trần Phong Nhã** thật dồi dào sức khỏe và gặt hái được nhiều thành công hơn nữa trong cuộc sống cũng như trong sự nghiệp giảng dạy của mình, để tiếp tục thực hiện sứ mệnh cao quý của mình là truyền đạt kiến thức cho thế hệ mai sau.

Em xin chân thành cảm ơn!

*Tp. Hồ Chí Minh, ngày 01 tháng 06 năm 2018*

Sinh viên thực hiện

**Phạm Công Quân**

**NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN**

|  |
| --- |
| ***Tp. Hồ Chí Minh, ngày ….… tháng ….… năm ….…***  **Giảng viên hướng dẫn**  **Trần Phong Nhã** |

**MỤC LỤC**

[NHIỆM VỤ THIẾT KẾ TỐT NGHIỆP.........................................................................i](#_Toc516526703)

[LỜI CẢM ƠN iii](#_Toc516526704)

[NHẬN XÉT CỦA GIẢNG VIÊN HƯỚNG DẪN iv](#_Toc516526705)

[DANH MỤC CHỮ VIẾT TẮT viii](#_Toc516526706)

[DANH MỤC BẢNG BIỂU ix](#_Toc516526707)

[DANH MỤC HÌNH VẼ x](#_Toc516526708)

[CHƯƠNG 1. MỞ ĐẦU 1](#_Toc516526709)

[1. Lý do chọn đề tài. 1](#_Toc516526710)

[2. Mục đích nghiên cứu. 1](#_Toc516526713)

[3. Đối tượng và phạm vi nghiên cứu. 2](#_Toc516526716)

[4. Phương pháp nghiên cứu. 2](#_Toc516526718)

[5. Ý nghĩa khoa học và thực tiễn. 2](#_Toc516526720)

[6. Các ứng dụng trong thực tiễn. 2](#_Toc516526722)

[6.1. Gợi ý bài hát. 2](#_Toc516526723)

[6.2. Gợi ý phim. 3](#_Toc516526726)

[6.3. Gợi ý du lịch. 4](#_Toc516526727)

[6.4. Các yếu tố ảnh hưởng. 4](#_Toc516526728)

[7. Cấu trúc báo cáo đồ án tốt nghiệp. 5](#_Toc516526729)

[CHƯƠNG 2. CƠ SỞ LÝ THUYẾT 6](#_Toc516526735)

[1. Tổng quan về ngôn ngữ Java.[1] 6](#_Toc516526736)

[2. Spring Framework. [2] 7](#_Toc516526737)

[2.1. Giới thiệu về Spring Framework. 7](#_Toc516526738)

[2.2. Sự ra đời Spring Framework. 8](#_Toc516526739)

[2.3. Kiến trúc của Spring framework. 9](#_Toc516526740)

[2.4. Các lợi ích của Spring framework. 10](#_Toc516526741)

[3. JPA Framework (Java Persistence API). 10](#_Toc516526742)

[3.1. Giới thiệu về JPA Framework. 10](#_Toc516526743)

[3.2. Ưu điểm khi dùng JPA Framework. 11](#_Toc516526745)

[3.3. Sự phát triển của JPA. 11](#_Toc516526746)

[3.4. Kiến trúc JPA Framework. 11](#_Toc516526747)

[4. Giới thiệu về Apache-Mahout.[3] 13](#_Toc516526749)

[4.1. Giới thiệu về Mahout. 13](#_Toc516526750)

[4.2. Kiến trúc Apache Mahout. 14](#_Toc516526751)

[5. Học máy (Machine learning-ML). [4] 15](#_Toc516526752)

[5.1. Giới thiệu về Machine learning. 15](#_Toc516526753)

[5.2. Định nghĩa về Machine learning. 16](#_Toc516526757)

[6. Hệ thống khuyến nghị (Recommendation System). [5] 18](#_Toc516526769)

[6.1. Giới thiệu về hệ thống khuyến nghị. 18](#_Toc516526770)

[6.2. Các khái niệm chính. 20](#_Toc516526771)

[6.3. Thông tin phản hồi từ người dùng và hai dạng bài toán chính trong RS. 22](#_Toc516526772)

[7. Các kỹ thuật chính trong RS. 22](#_Toc516526773)

[8. Giới thiệu ma trận utility.[5] 23](#_Toc516526774)

[9. Kết luận chương 2. 25](#_Toc516526777)

[CHƯƠNG 3. PHÂN TÍCH 26](#_Toc516526778)

[1. Sơ lược về Collaborative filtering. [6] 26](#_Toc516526779)

[2. Công thức tính độ tương tự (Similarity). 28](#_Toc516526781)

[3. Các bước tính toán. [6] 29](#_Toc516526782)

[4. Công thức tính độ tương tự của cosin. 31](#_Toc516526783)

[5. Công thức tính độ tương tự Pearson. 35](#_Toc516526784)

[6. Đánh giá thuật toán. 38](#_Toc516526785)

[6.1. Dữ liệu thực nghiệm.[5] 38](#_Toc516526786)

[6.2. Độ đo. 39](#_Toc516526787)

[7. Ưu và khuyết điểm của Collaborative filtering ( lọc cộng tác ).[6] 40](#_Toc516526788)

[7.1. Ưu điểm. 40](#_Toc516526789)

[7.2. Khuyết điểm. 41](#_Toc516526790)

[7.3. Cách khắc phục 41](#_Toc516526791)

[8. Kĩ thuật nâng cao-Kĩ thuật phân rã ma trận.[7] 42](#_Toc516526792)

[9. Kết luận chương 3. 45](#_Toc516526793)

[CHƯƠNG 4. ỨNG DỤNG THUẬT TOÁN VÀO GỢI Ý BÀI NHẠC 46](#_Toc516526794)

[1. Sơ đồ use-case tổng quát. 46](#_Toc516526795)

[2. Sơ đồ lớp UML. 47](#_Toc516526802)

[3. Sơ đồ cơ sở dữ liệu. 48](#_Toc516526804)

[4. Lưu đồ mô tả chức năng. 48](#_Toc516526805)

[4.1. Lưu đồ đăng ký. 49](#_Toc516526806)

[4.2. Lưu đồ đánh giá sản phẩm. 50](#_Toc516526807)

[4.3. Lưu đồ gợi ý sản phẩm cho user. 51](#_Toc516526808)

[5. Giao diện trang web. 52](#_Toc516526809)

[5.1. Giao diện trang home. 52](#_Toc516526810)

[5.2. Giao diện xem các bài hát theo thể loại nhạc. 54](#_Toc516526816)

[5.3. Giao diện trang xem chi tiết sản phẩm. 54](#_Toc516526819)

[5.4. Check-out sản phẩm. 55](#_Toc516526822)

[5.5. Giao diện quản lý music store. 56](#_Toc516526828)

[5.6. Tạo mới Album. 56](#_Toc516526829)

[5.7. Chỉnh sửa Album. 57](#_Toc516526830)

[5.8. Quản lý User. 57](#_Toc516526831)

[KẾT LUẬN VÀ KIẾN NGHỊ 59](#_Toc516526832)

[1. Kết quả đạt được. 59](#_Toc516526833)

[2. Tồn tại.............................................. 59](#_Toc516526841)

[3. Hướng phát triển. 59](#_Toc516526846)

[TÀI LIỆU THAM KHẢO 60](#_Toc516526850)

**DANH MỤC CHỮ VIẾT TẮT**

|  |  |  |  |
| --- | --- | --- | --- |
| **STT** | **Mô tả** | **Ý nghĩa** | **Ghi chú** |
| 1 | SQL | Structured Query Language |  |
| 2 | MVC | Model – View - Controller |  |
| 3 | JVM | Java Virtual Machine |  |
| 4 | JSE | Java Standard Edition |  |
| 5 | JDBC | Java Database Connectivity |  |
| 6 | EJB | Enterprise Java Beans |  |
| 7 | JMS | Java Message Service |  |
| 8 | ORM | Object-Relational Mapping |  |
| 9 | API | Application Programming Interface |  |
| 10 | CSDL | Cơ Sở Dữ Liệu |  |
| 11 | XML | Extensible Markup Language |  |
| 12 | CF | Colabornative filtering |  |
| 13 | RS | Recommender system |  |
| 14 | RMSE | Root Mean Squared Error |  |
| 15 | UML | Unified Modeling Language |  |
| 16 | ML | Machine learning |  |
| 17 | KNN | K Nearest neighborhood |  |
| 18 | POJO | Plain Old Java Object. |  |

**DANH MỤC BẢNG BIỂU**

[Bảng 1: Mô tả thuật toán collaborative filtering. 30](#_Toc515890214)

[Bảng 2: Giá trị đánh giá của user lên một item. 32](#_Toc515890215)

[Bảng 3: Độ tương quan của user-user theo công thức cosin. 32](#_Toc515890216)

[Bảng 4: Độ tương quan của item-item theo công thức cosin. 33](#_Toc515890217)

[Bảng 5: Dự đoán đánh giá của user u lên item i theo user-user similarity. 34](#_Toc515890218)

[Bảng 6: Dự đoán đánh giá của user u lên item i theo item-item similarity. 34](#_Toc515890219)

[Bảng 7: Độ tương quan của user-user theo công thức Pearson. 36](#_Toc515890220)

[Bảng 8: Độ tương quan của item-item theo công thức Pearson. 37](#_Toc515890221)

[Bảng 9: Dự đoán đánh giá của user u lên item i theo user-user similarity. 37](#_Toc515890222)

[Bảng 10: Dự đoán đánh giá của user u lên item i theo item-item similarity. 38](#_Toc515890223)

[Bảng 11: Kết quả tính RMSE với K từ 2-30 39](#_Toc515890224)

[Bảng 12: Kết quả tính RMSE với K từ 100-390 40](#_Toc515890225)

[Bảng 13: Độ đo của Cosin và Pearson Similarity 40](#_Toc515890226)

[Bảng 14: Giải thuật phân rã ma trận. 44](#_Toc515890227)

**DANH MỤC HÌNH VẼ**

[Hình 1: Gợi ý bài hát của Zing.mp3.vn. 2](#_Toc515890237)

[Hình 2: Gợi ý phim của trang phimmoi.net. 3](#_Toc515890238)

[Hình 3: Gợi ý du lịch của trang dulichfun.com. 4](#_Toc515890239)

[Hình 4: Mô hình Spring. 9](#_Toc515890240)

[Hình 5: Kiến trúc JPA tổng quát. 12](#_Toc515890241)

[Hình 6: Kiến trúc Apache Mahout. 14](#_Toc515890242)

[Hình 7: Tổng quan về Machine Learning. 15](#_Toc515890243)

[Hình 8: Hệ thống khuyến nghị sản phẩm. 19](#_Toc515890244)

[Hình 9: Hệ thống gợi ý bài nhạc của Mp3.zing.vn. 20](#_Toc515890245)

[Hình 10: Hệ thống gợi ý sản phẩm của Amazon. 20](#_Toc515890246)

[Hình 11: Ma trận biểu diễn dữ liệu trong RS (user-item-rating matrix) 21](#_Toc515890247)

[Hình 12: Công thức đo độ lỗi RMSE. 22](#_Toc515890248)

[Hình 13: Ma trận mô tả user-item. 26](#_Toc515890249)

[Hình 14: Mô tả công thức tính độ tương quan. 28](#_Toc515890250)

[Hình 15: Mô hình hệ thống lọc cộng tác dựa trên sản phẩm. 31](#_Toc515890251)

[Hình 16: Công thức tính Cosin theo user-user. 31](#_Toc515890252)

[Hình 17: Công thức tính Cosin theo item-item. 31](#_Toc515890253)

[Hình 18: Công thức dự đoán số sao của user u lên item i. 33](#_Toc515890254)

[Hình 19: Công thức dự đoán đánh giá theo Pearson similarity (item-item). 33](#_Toc515890255)

[Hình 20: Công thức tính độ tương quan Pearson theo user-user.[10] 35](#_Toc515890256)

[Hình 21: Mô tả chia ma trận utility 42](#_Toc515890257)

[Hình 22:Sơ đồ use-case tổng quát. 46](#_Toc515890258)

[Hình 23: Sơ đồ lớp UML 47](#_Toc515890259)

[Hình 24: Sơ đồ cơ sở dữ liệu 48](#_Toc515890260)

[Hình 25: Lưu đồ đăng ký thành viên. 49](#_Toc515890261)

[Hình 26: Lưu đồ đánh giá sản phẩm. 50](#_Toc515890262)

[Hình 27: Lưu đồ gợi ý sản phẩm cho user theo độ tương quan của user. 51](#_Toc515890263)

[Hình 28: Lưu đồ gợi ý sản phẩm cho user theo độ tương quan của item. 52](#_Toc515890264)

[Hình 29: Giao diện trang home. 53](#_Toc515890265)

[Hình 30: Giao diện các bài hát thuộc thể loại. 54](#_Toc515890266)

[Hình 31: Giao diện trang chi tiết bài hát. 54](#_Toc515890267)

[Hình 32: Giao diện trang check-out sản phẩm. 55](#_Toc515890268)

[Hình 33: Giao diện trang đăng nhập. 55](#_Toc515890269)

[Hình 34: Giao diện trang đăng ký. 55](#_Toc515890270)

[Hình 35: Quản lý Album. 56](#_Toc515890271)

[Hình 36: Giao diện trang tạo mới Album. 56](#_Toc515890272)

[Hình 37: Giao diện trang chỉnh sửa Album. 57](#_Toc515890273)

[Hình 38: Giao diện trang quản lý user. 57](#_Toc515890274)

[Hình 39: Giao diện trang chỉnh sửa User. 58](#_Toc515890275)

# 

# CHƯƠNG 1. MỞ ĐẦU

1. **Lý do chọn đề tài.**

Về lý do chọn đề tài thì đối với em có rất nhiều lý do, nhưng tác động mạnh nhất vẫn là sự phát triển của Internet, bởi vì Internet đã mở ra cho con người rất nhiều cơ hội mới và nhiều thách thức, con người có cơ hội tiếp cận với thông tin gần như vô hạn nó nâng tầm hiểu biết của con người chúng ta càng đi xa hơn bởi vì lượng thông tin trên internet tăng lên hàng giây, bởi vậy việc tìm được những thông tin cần thiết và đáng tin cậy không phải lúc nào cũng dễ dàng.

Bên cạnh sự phát triển mạnh mẽ của Internet đó thì có một sự phát triển luôn cần sự gắn kết của Internet đó là thương mại điện tử, Internet càng ngày càng phát triển và nó cũng kéo theo sự phát triển của thương mại điện tử, hầu hết mọi thứ có giá trị đều được con người trao đổi và rao bán trên này một cách thuận tiện, với sự thuận tiện này khách hàng có thể trao đổi mua hàng hay đặt hàng ở bất kỳ đâu mà họ muốn, nhưng bên cạnh đó có một sự khó khăn của việc mua bán hàng hóa này là, làm sao khách hàng có thể tìm kiếm được món hàng họ thích một cách nhanh chóng, dù họ tìm được những sản phẩm như họ mong muốn nhưng nếu họ muốn tìm một sản phẩm như sản phẩm họ đang có thì phải làm thế nào, chính vì điều này hệ thống khuyến nghị ra đời đã giúp ích cho rất nhiều người và doanh nghiệp đặc biệt là các ông lớn như Amazon, Youtube, ebay... khi áp dụng hệ thống gợi ý này vào đã giúp họ tăng doanh thu một cách nhanh chóng, và đây cũng là lý do để tôi chọn đề tài để giúp tôi có thể nhìn nhận và hiểu sâu hơn về hệ thống khuyến nghị. Và cụ thể là tôi đã chọn đề tài “**NGHIÊN CỨU THUẬT TOÁN COLLABORATIVE FILTERING VÀ ỨNG DỤNG THUẬT TOÁN VÀO GỢI Ý BÀI NHẠC**”.

1. **Mục đích nghiên cứu.**

Mục đích của em đặt ra là để tìm hiểu tổng quan về hệ thống khuyến nghị, các thuật toán được sử dụng trong hệ thống tư vấn này , ứng dụng nó vào trong thực tiễn.

Mục tiêu cụ thể trong đề tài nghiên cứu này là tìm hiểu về lọc cộng tác (collaborative filtering) dựa trên sự tương quan của người dùng hoặc sản phẩm, tìm hiểu các thuật toán tính độ tương quan này và áp dụng nó vào trong gợi ý bài nhạc.

1. **Đối tượng và phạm vi nghiên cứu.**

Nghiên cứu các phương pháp tính độ tương tự và dự đoán các sản phẩm (item) sẽ giới thiệu cho người dùng, đánh giá dữ liệu dựa trên tập dữ liệu có sẵn là movies-len 100k.

1. **Phương pháp nghiên cứu.**

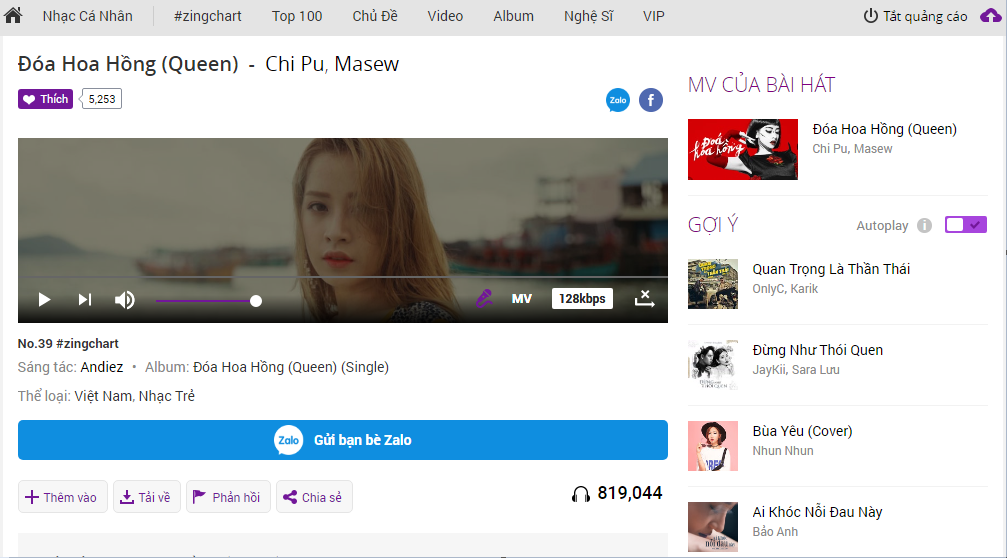
Nghiên cứu lý thuyết và phân tích các quá trình thực hiện để đánh giá và rút ra kết luận từ kết quả đạt được.

1. **Ý nghĩa khoa học và thực tiễn.**

Hệ thống khuyến nghị là nhằm mục đích để gợi ý các sản phẩm tới người dùng. Nó rất quan trọng và phổ biến trong các hệ thống bán hàng online hiện nay. Người ta có thể sử dụng mọi cách để làm tăng tính thương mại hóa bằng phương pháp gợi ý hay tư vấn. Phương pháp collaborative filtering chỉ là một phần nhỏ trong thế giới online nhưng nó đóng một vai trò hết sức quan trọng, từ khi nó ra đời nó đã khắc phục được các yếu tố gợi ý sản phẩm đến người dùng.

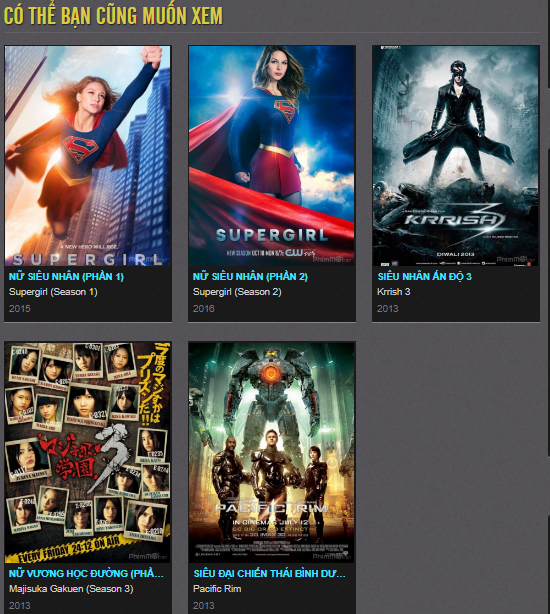
1. **Các ứng dụng trong thực tiễn.**
   1. ***Gợi ý bài hát*.**

Trang mp3.zing.vn là một trang nghe nhạc phổ biến của cộng đồng Việt Nam , nó đã áp dụng gợi ý này vào trong gợi ý các bài hát.



Hình 1: Gợi ý bài hát của Zing.mp3.vn.

* 1. ***Gợi ý phim.***



Hình 2: Gợi ý phim của trang phimmoi.net.

* 1. ***Gợi ý du lịch.***



Hình 3: Gợi ý du lịch của trang dulichfun.com.

* 1. ***Các yếu tố ảnh hưởng.***

**- Ảnh hưởng của dữ liệu đầu vào:** Trong nhiều hệ thống tư vấn, số những đánh giá thu được thường rất nhỏ so với số những đánh giá cần có cho dự đoán. Sự thành công của hệ thống tư vấn lọc cộng tác phụ thuộc vào giá trị của đại đa số những người dùng chính. Chẳng hạn, trong hệ thống tư vấn điện ảnh, có rất nhiều bộ phim đã được đánh giá chỉ bởi một vài người và những bộ phim này khả năng được tư vấn là rất ít, thậm chí ngay cả khi trong số đó có những người dùng đưa ra đánh giá rất cao về chúng. Cũng như vậy, đối với những người dùng mà thị hiếu của họ khác thường so với một số đông người khác thì sẽ không có người dùng nào được tư vấn về những thị hiếu giống họ, dẫn đến việc tư vấn nghèo nàn. Một phương pháp vượt qua tính thưa thớt trong đánh giá là sử dụng thông tin cá nhân của người dùng khi tính toán sự tương đồng giữa những người dùng. Hai người dùng được xem là giống nhau không khi được đánh giá có sở thích về các bộ phim là giống nhau mà chúng còn phải thuộc cùng một đối tượng. Chẳng hạn, sử dụng giới tính, tuổi, địa chỉ khu vực, nền giáo dục và thông tin công việc của những người dùng trong việc tư vấn nhà hàng. Sự mở rộng của những kỹ thuật lọc cộng tác thường được gọi là “lọc demographic” .

Một vấn đề trước khi xây dựng ma trận đánh giá, với những sản phẩm ít được đánh giá, hoặc những người dùng ít đánh giá sản phẩm, những người dùng và sản phẩm này sẽ không hữu ích trong quá trình tư vấn. Vấn đề cần chọn lọc ra những sản phẩm và người dùng để tham gia trong quá trình tư vấn. Rõ ràng những sản phẩm mới hoặc người dùng mới không thể tham gia trong quá trình dự đoán, hoặc những sản phẩm hay người dùng có đánh giá ít hơn 1 ngưỡng nào đó cũng được loại ra.

**- Ảnh hưởng của thuật toán tính độ tương tự:** Để xây dựng ma trận tương tự, các giá trị được tính toán theo một công thức được đề xuất ở trên hoặc một công thức khác. Các công thức tính toán khác nhau sẽ cho ra các ma trận đánh giá khác nhau, dẫn đến kết quả tư vấn không đồng nhất. Nhiều chuyên gia đánh giá, với thuật toán có sự tham gia của đánh giá trung bình sẽ có kết quả tốt hơn. Chương 3 sẽ thí nghiệm kết quả tư vấn với các thuật toán tính độ tương tự và đưa ra nhận xét về độ chính xác của kết quả tư vấn khi áp dụng các độ tương tự khác nhau.

**- Ảnh hưởng của số lượng lân cận tham gia vào dự đoán:** Thông qua quá trình tính ma trận tương tự, hệ thống tư vấn sử dụng những sản phẩm lân cận nhất với sản phẩm đang xét để đưa vào dự đoán, số lượng lân cận ảnh hưởng đến chất lượng của kết quả tư vấn. Chọn số lượng lân cận có thể dựa vào số lượng lân cận của sản phẩm hoặc dựa vào giá trị của độ đo tương tự. Ví dụ: Trong hệ thống tư vấn phim với số bộ phim là 1000 bộ phim, khi cần tư vấn 1 bộ phim, có thể sẽ xét trên 100 bộ phim lân cận với bộ phim đang xét, hoặc là xét với những bộ phim lân cận có độ tương tự lớn 0.75. Số lượng lân cận tham gia vào dự đoán ảnh hưởng đến quá trình đưa ra dự đoán, nhiều chuyên gia lựa chọn những sản phẩm có độ tương tự 0.75 để tham gia vào dự đoán.

**7. Cấu trúc báo cáo đồ án tốt nghiệp.**

**Chương 1: Mở đầu.**

**Chương 2: Cơ sở lý thuyết.**

**Chương 3: Phân tích.**

**Chương 4: Ứng dụng thuật toán vào gợi ý bài nhạc.**

**Kết luận và kiến nghị.**

**CHƯƠNG 2. CƠ SỞ LÝ THUYẾT**

## **1. Tổng quan về ngôn ngữ Java.**[1]

- Ngôn ngữ lập trình Java ban đầu được phát triển bởi Sun Microsystems do James Gosling khởi xướng và phát hành vào năm 1995 (Java 1.0 [J2SE]). Tính đến thời điểm này phiên bản mới nhất của Java Standard Edition (JSE) là 10. Với ưu thế về đa nền tảng (multi platform) Java càng lúc càng được ứng dụng rộng rãi trên nhiều thiết bị từ máy tính đến mobile và nhiều thiết bị phần cứng khác...

- Java là ngôn ngữ lập trình hướng đối tượng nên nó cũng có 4 đặc điểm chung của các ngôn ngữ hướng đối tượng:

* Tính trừu tượng (Abstraction): là tiến trình xác định và nhóm các thuộc tính, các hành động liên quan đến một thực thể đặc thù, xét trong mối tương quan với ứng dụng đang phát triển.
* Tính đa hình (Polymorphism): cho phép một phương thức có các tác động khác nhau trên nhiều loại đối tượng khác nhau. Với tính đa hình, nếu cùng một phương thức ứng dụng cho các đối tượng thuộc các lớp khác nhau thì nó đưa đến những kết quả khác nhau. Bản chất của sự việc chính là phương thức này bao gồm cùng một số lượng các tham số.
* Tính kế thừa (Inheritance): Điều này cho phép các đối tượng chia sẻ hay mở rộng các đặc tính sẵn có mà không phải tiến hành định nghĩa lại.
* Tính đóng gói (Encapsulation): là tiến trình che giấu việc thực thi những chi tiết của một đối tượng đối với người sử dụng đối tượng ấy.

- Bên cạnh đó Java còn có một số đặc tính khác:

* Độc lập nền (Write Once, Run Anywhere): Không giống như nhiều ngôn ngữ lập trình khác như C và C ++, khi Java được biên dịch, nó không được biên dịch sang mã máy cụ thể, mà thay vào đó là mã byte code chạy trên máy ảo Java (JVM). Điều này đồng nghĩa với việc bất cứ thiết bị nào có cài đặt JVM sẽ có thể thực thi được các chương trình Java.
* Đơn giản: học Java thật sự dễ hơn nhiều so với C/C++, nếu bạn đã quen với các ngôn ngữ lập trình hướng đối tượng thì việc học Java sẽ dễ dàng hơn. Java trở nên đơn giản hơn so với C/C++ do đã loại bỏ tính đa kế thừa và phép toán con trỏ từ C/C++.
* Bảo mật: Java hỗ trợ bảo mật rất tốt bởi các thuật toán mã hóa như mã hóa một chiều (one way hashing) hoặc mã hóa công cộng (public key)...
* Đa luồng: Với tính năng đa luồng Java có thể viết chương trình có thể thực thi nhiều task cùng một lúc. Tính năng này thường được xử dụng rất nhiều trong lập trình game.
* Hiệu suất cao: nhờ vào trình thu gom rác (garbage collection), giải phóng bộ nhớ đối với các đối tượng không được dùng đến.
* Linh hoạt: Java được xem là linh hoạt hơn C/C ++ vì nó được thiết kế để thích ứng với nhiều môi trường phát triển.
* Một số framework được phát triển dựa trên ngôn ngữ Java là Spring, Struts, Zkoss framework, bên cạnh đó còn có một API (Application Programming Interface) của Java để giúp ta dễ dàng kết nối với cơ sở dữ liệu là JPA (Java Persistence API ) với trong đề tài này em chọn Spring framework và JPA để thuận tiện cho việc chạy thử chương trình.

**2. Spring Framework.** [2]

***2.1. Giới thiệu về Spring Framework.***

- Spring là một Framework phát triển các ứng dụng Java được sử dụng bởi hàng triệu lập trình viên. Nó giúp tạo các ứng dụng có hiệu năng cao, dễ kiểm thử, sử dụng lại code…

- Spring nhẹ và trong suốt (nhẹ: kích thước nhỏ, version cơ bản chỉ khoảng 2MB; trong suốt: hoạt động một cách trong suốt với lập trình viên)

- Spring là một mã nguồn mở, được phát triển, chia sẻ và có cộng đồng người dùng rất lớn.

- Spring Framework được xây dựng dựa trên 2 nguyên tắc chính là : Dependency Injection và Aspect Oriented Programming.

- Những tính năng cốt lõi của Spring có thể được sử dụng để phát triển Java Desktop, ứng dụng mobile, Java Web. Mục tiêu chính của Spring là giúp phát triển các ứng dụng J2EE(Java 2 Platform Enterprise Edition) một cách dễ dàng hơn dựa trên mô hình sử dụng POJO (Plain Old Java Object).

***2.2. Sự ra đời Spring Framework*.**

- Có thể nói rằng Spring Framework ra đời là nhờ sự thành công của EJB (Enterprise Java Beans). EJB là một kiến trúc phát triển dành cho các ứng dụng doanh nghiệp đòi hỏi tính mạnh mẽ và khả năng mở rộng cao. EJB là một thành phần quan trọng của nền tảng Java EE. EJB được tạo ra bởi hai gã khổng lồ công nghệ là IBM và Sun vào những năm cuối thập niên 90. Dưới sự phát triển của hai tập đoàn này, EJB đã trở nên nặng nề hơn. Tiêu chuẩn EJB cũng nhanh chóng được thông qua. Sau tất cả, EJB được ca ngợi là giải pháp tốt nhất để xây dựng các ứng dụng phân tán dành cho doanh nghiệp.Nhưng rồi những chỉ trích về chi tiết kỹ thuật của EJB đã tăng nhanh còn hơn sự nổi tiếng của nó. EJB là một framework lớn với độ phức tạp cao. Các nhà phát triền khi làm việc với EJB đã nhận ra rằng nó không hoạt động như những gì nó đã đề ra. EJB chỉ tuyệt vời trên bàn giấy, nhưng ra thực tế đó lại là một chuyện khác.

EJB vẫn được tiếp tục phát triển, dễ sử dụng hơn nhiều so với buổi ban đầu. Nhưng những chỉ trích vẫn còn đó, nỗi thất vọng trong cộng đồng Java ngày càng tăng lên, đòi hỏi cần có một giải pháp thay thế.

Vào năm 2002, Spring Framework phát hành phiên bản đầu tiên bởi Rod Johnson. Việc xây dựng các ứng dụng doanh nghiệp trở nên đơn giản và dễ dàng hơn. Rod Johnson đã cho xuất bản một cuốn sách được phổ biến rộng rãi có nhan đề "J2EE Development without EJB". Điều này đã cho Spring Framework rất nhiều động lực để thay thế EJB.

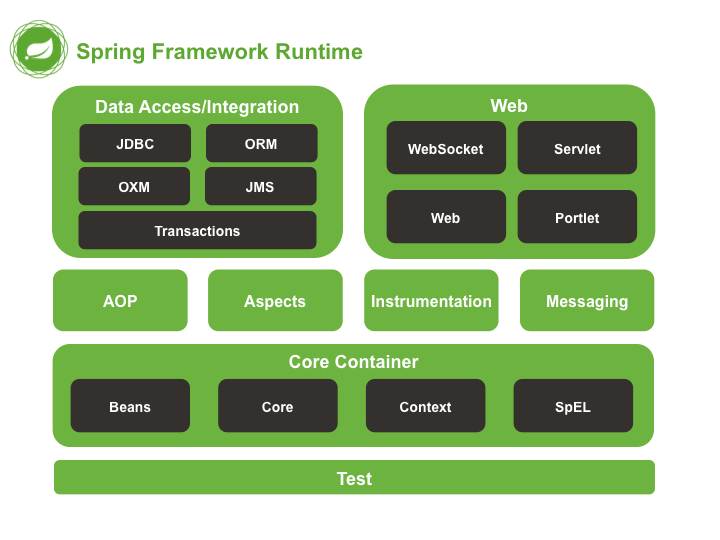
Ngày nay, Spring đã trở thành framework mã nguồn mở phổ biến nhất để xây dựng các ứng dụng doanh nghiệp. Cách tiếp cận thực tế ban đầu của Rod Johnson tiếp tục được phát triển và hướng tới một bộ công cụ hoàn chỉnh dành cho xây dựng các ứng dụng doanh nghiệp. Theo một số nguồn, trên 50% các ứng dụng web Java hiện nay đang sử dụng Spring.

Để ngăn chặn sự phức tạp trong phát triển các ứng dụng, Spring Framework thường dựa trên các quan điểm như sau:

* Đơn giản hóa công việc phát triển thông qua việc sử dụng các đối tượng Java đơn giản hay còn được gọi là POJO (Plain Old Java Object)
* Nới lỏng ràng buộc giữa các thành phần thông qua việc sử dụng Dependency Injection và viết các interface
* Tiếp cận lập trình khai báo bằng cách sử dụng các quy tắc (convention) và các khía cạnh (aspect) chung
* Giảm thiểu các mã nghi thức và soạn sẵn (boilerplate) thông qua việc sử dụng các khuôn mẫu (template).

## ***2.3. Kiến trúc của Spring framework*.**

**-** Spring được chia làm nhiều module khác nhau, tùy theo mục đích phát triển ứng dụng mà ta dùng 1 trong các module đó. Dưới đây là kiến trúc tổng thể của Spring Framework.



Hình 4: Mô hình Spring.

* **Test:** Tầng này cung cấp khả năng hỗ trợ kiểm thử với JUnit và TestNG.
* **Spring Core Container:**

Bao gồm các module Spring core, beans, context và expression languate (EL):

* Spring core, bean cung cấp tính năng IOC và Dependency Injection.
* Spring Context hỗ trợ đa ngôn ngữ (internationalization), các tính năng Java EE như EJB.
* Expression Language được mở rộng từ Expresion Language trong JSP(Java Server Page). Nó cung cấp hỗ trợ việc setting/getting giá trị, các method cải tiến cho phép truy cập collections, index, các toán tử logic…

**- AOP , Aspects and Instrumentation:**

Những module này hỗ trợ cài đặt lập trình hướng khía cạnh (Aspect Oriented Programming), hỗ trợ tích hợp với AspectJ.

**- Data Access / Integration**.

Nhóm này bao gồm JDBC (Java Database Connectivity), ORM (Object Relational Mapping), OXM (Object/XML Mapping), JMS (Java Message Service) và module Transaction. Những module này cung cấp khả năng giao tiếp với database.

**- Web.**

Hay còn gọi là Spring MVC (Model-View-Controller), nhóm này gồm Web, Web-Servlet… hỗ trợ việc tạo ứng dụng web.

## ***2.4. Các lợi ích của Spring framework.***

* Spring cho phép lập trình viên sử dụng POJOs. Việc sử dụng POJOs giúp bạn không phải làm việc với EJB, ứng dụng, các luồng chạy, cấu hình… đơn giản hơn rất nhiều.
* Spring được tổ chức theo kiểu mô đun. Số lượng các gói và các lớp khá nhiều, nhưng bạn chỉ cần quan tâm đến những gì bạn cần và không cần quan tâm đến phần còn lại.
* Spring hỗ trợ sử dụng khá nhiều công nghệ như ORM Framework, các logging framework, J2EE, các thư viện tạo lịch trình.
* Module Web của Spring được thiết kế theo mô hình MVC nên nó cung cấp đầy đủ các tính năng giúp thay thế các web framework khác như Struts.

**3. JPA Framework (Java Persistence API).**

***3.1. Giới thiệu về JPA Framework.***

- JPA là tập hợp các lớp các hàm xử lý các thao tác đến database như thêm mới ,cập nhật, xóa… (CRUD=Insert/Construct, Retrieve, Update, Delete). JPA giúp cho lập trình viên ít sử dụng SQL hơn mà chủ yếu làm việc với các đối tượng của java class. Tất cả những việc đọc từ cơ sở dữ liệu (database) hay lưu sự thay đổi vào database đã được JPA thực hiện ngầm trong quá trình chạy/runtime. JPA giúp chúng ta giải quyết vấn đề tương tác database với câu lệnh SQL.

***3.2. Ưu điểm khi dùng JPA Framework.***

- Mã nguồn mở và nhẹ.

- Thao tác nhanh chóng.

- Viết ít code hơn

- Performance tốt

- Độc lập về database

- Không phải làm việc nhiều với SQL

- Tạo bảng tự động: JPA Framework cung cấp các thiết bị để tạo ra các bảng CSDL tự động. Như vậy lập trình viên không cần phải tự tạo ra các bảng CSDL bằng tay.

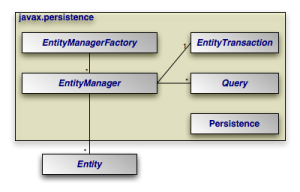
***3.3. Sự phát triển của JPA.***

* Các phiên bản trước đây của EJB(2.1 trở về trước) từ tầng DAO (Data Access Object) hay tầng tương tác với database được thực hiện bởi EntityBean.
* EJB3.0 giới thiệu JPA1.0 đầu tiên trong đặc tả của Java EE 5, năm 2006.
* JPA2.0 cho Java EE 6, năm 2009.
* JPA2.1 cho Java EE 7, năm 2013.

JPA chính là đặc tả được đưa ra bởi SUN nên cần phải có các bước thực thi đặc tả này hay còn gọi là JPA provider.

* Hibernate.
* Eclipselink.
* Toplink.
* Spring Data JPA.

***3.4. Kiến trúc JPA Framework.***

[](http://s2code.net/wp-content/uploads/2015/09/JPA-archetecture.png)

Hình 5: Kiến trúc JPA tổng quát.

* **EntityManagerFactory**: Được cấu hình để kết nối database và được dùng để tạo ra các EntityManager.
* **EntityManager**: Interface này tương tự như Session trong Hibernate. EntityManager thực hiện các chức năng thêm mới xóa cập nhật tìm kiếm, các Entity. Các Entity này tương ứng với các bảng trong Database. Entity Manager quản lý các Entity trong Persistence Context. Entity Manager có thể là  “Container-managed Entity Manager” hoặc “Application-manager Entity Manager”. Nói ngắn gọn thì Application-manager Entity Manager là  Entity Manager được dùng cho các ứng dụng dạng Java (ví dụ ứng dụng console, swing), các Container-managed Entity Manager là  Entity Manager được dùng cho nền tảng Java EE, cụ thể là JSP/Servlet, EJB….
* **EntityTransaction**: các quan hệ một một với EntityManager. Mỗi hoạt động (CRUD) của EntityManager đều dưới sự kiểm soát của EntityTransaction.
* **Persistence**: chứa hàm static để lấy về đối tượng EntityManagerFactory
* **Persistence Context**: chứa các đối tượng của Entity mà mỗi đối tượng được phân biệt với nhau bởi **@Id.**Trong ngữ cảnh của Persistence Context, thì các đối tượng Entity và  vòng đời của đối tượng được quản lý bởi 1 Entity Manager nhất định nào đó.
* **Persistence** **Unit**: dùng để định nghĩa các kiểu Entity,cái mà được quản lí bởi EntityManager trong 1 ứng dụng. Mỗi 1 Persistence Unit tương ứng với 1 nguồn dữ liệu (datasource).
* **Query:**các JPA providers sẽ thực thi interface này để truy vấn các đối tượng Entity theo 1 tiêu chuẩn nào đó.
* **Entity:** là  java class hay POJO (Plain Old Java Object) được ánh xạ/mapping đến bảng trong Database. Một Entity được quản lí trong 1 Persistence Context, trong đó các Entity được phân biệt với nhau theo **Id.** Việc cấu hình Entity có thể dùng **Annotation** hoặc dùng file **XML.**
* Nếu không có @Entity thì JPA không hiểu và không thể thực mapping class java đến bảng trong Database tương ứng.

@Id: Mọi @Entity đều phải có 1 primary key hay @Id để JPA phân biệt các đối tượng của Entity là khác nhau. Nếu class trên không cấu hình gì thêm thì nó sẽ mặc định ánh xạ/mapping đến 1 bảng.Vậy cần phải cấu hình đầy đủ nhằm tuân theo tên cột, tên bảng của Database.

## **4.** **Giới thiệu về Apache-Mahout.**[3]

### ***4.1. Giới thiệu về Mahout*.**

Apache Mahout là một dự án mã nguồn mở mới của Apache Software Foundation (ASF-Quỹ phần mềm Apache) với mục tiêu chính là tạo các thuật toán học máy có khả năng mở rộng, các thuật toán này là miễn phí sử dụng theo giấy phép Apache. Mahout bao gồm các việc thực hiện để phân cụm, phân loại, collaborative filtering và lập trình tiến hóa. Hơn nữa, nó khéo léo sử dụng thư viện Apache Hadoop để cho phép Mahout mở rộng hiệu quả trong đám mây này.

Dự án Mahout được bắt đầu bởi một số người tham gia vào cộng đồng Apache Lucene (tìm kiếm mã nguồn mở) với một sự quan tâm tích cực trong việc học máy và mong muốn về tính thực hiện mạnh mẽ, có đầy đủ các tài liệu cần thiết, có khả năng mở rộng của các thuật toán học máy phổ biến cho việc phân cụm và phân loại. Thuật toán này ban đầu được gọi là "Map-Reduce for Machine Learning on Multicore" (Map-Reduce cho học máy theo đa lõi) nhưng sau phát triển để trình bày các cách tiếp cận học máy rộng hơn. Mahout cũng nhằm mục đích:

* Xây dựng và hỗ trợ một cộng đồng những người dùng và những người đóng góp sao cho mã này vượt trên bất kỳ tác động nào của người đóng góp cụ thể, bất kỳ công ty, hoặc quỹ tài trợ đại học. Tập trung vào trường hợp sử dụng thực tế, thế giới thực, đối lập với nghiên cứu hay các kĩ thuật mới.
* Cung cấp các tài liệu và ví dụ chất lượng.

Đặc điểm chính của Apache Mahout:

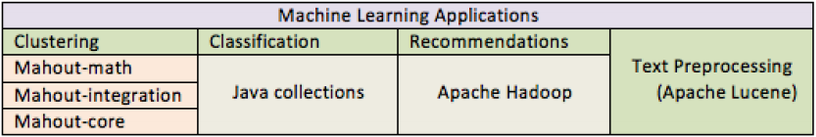
* Đây là một dự án của nền tảng phần mềm Apache.
* Là một thư viện học máy khả năng mở rộng:
  + Thi hành MapReduce theo quy mô đường thẳng với dữ liệu
  + Thuật toán tuần tự nhanh ( quá trình thi hành không phụ thuộc vào kích thước của tập dữ liệu )
* Chủ yếu là các thuật toán phân nhóm, phân loại, và khuyến nghị (lọc cộng tác).
* Thuật toán học máy có thể được thi hành tuần tự.
* Chạy trên nền tảng Hadoop cho việc mở rộng.
* Nó là một thư viện Java (không có giao diện người dùng).
* Là một thư viện dùng chung.

Mặc dù tương đối mới trong thư viện mã nguồn mở, Mahout đã có một số lượng lớn các chức năng, đặc biệt liên quan đến việc phân cụm và lọc cộng tác (Collaborative filtering-CF). Các đặc tính chính của Mahout là:

* Taste CF. Taste là một dự án mã nguồn mở cho CF được khởi đầu bởi Sean Owen trên SourceForge và được tặng cho Mahout vào năm 2008.
* Một số việc thực hiện phân cụm của Map-Reduce có sẵn, gồm k-Means, fuzzy k-Means, Canopy, Dirichlet và Mean-Shift.
* Thực hiện phân loại Naive Bayes phân tán và Naive Bayes phụ.
* Phân phổi các hàm chức năng phù hợp cho công việc lập trình.
* Thư viện ma trận và vectơ.

### ***4.2.Kiến trúc Apache Mahout*.**

Bảng dưới đây thể hiện các thiết kế cao cấp của một Mahout:



Hình 6: Kiến trúc Apache Mahout.

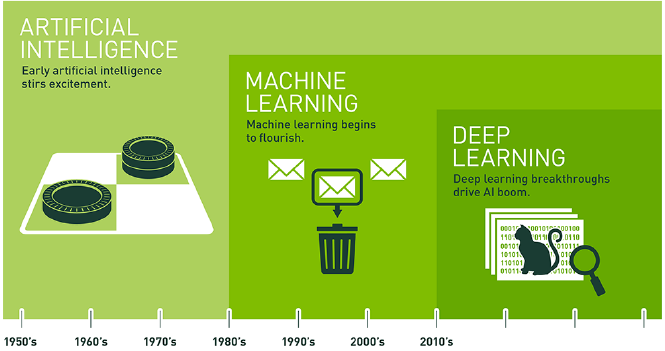
- Ứng dụng học máy truy cập API, cung cấp hỗ trợ triển khai kỹ thuật như:phân cụm (Clustering), phân loại(Classification), và khuyến nghị(Recommendation).

- Text Preprocessing: Ứng dụng đòi hỏi bộ tiền xử lý để nhập văn bản, nó có thể đạt được với Apache Lucene. Apache Hadoop cung cấp cách xử lý dữ liệu và lưu trữ để cho phép khả năng mở rộng.

* 1. **Học máy (Machine learning-ML).** [4]
  2. ***Giới thiệu về Machine learning*.**

Machine learning là một tập con của trí tuệ nhân tạo nó là một lĩnh vực nhỏ trong khoa học máy tính có khả năng tự học hỏi dựa trên dữ liệu được đưa vào mà không cần phải được lập trình cụ thể.

Những năm gần đây, khi mà khả năng tính toán của các máy tính được nâng lên một tầm cao mới và lượng dữ liệu khổng lồ được thu thập bởi các hãng công nghệ lớn, Machine Learning đã tiến thêm một bước dài và một lĩnh vực mới được ra đời gọi là Deep Learning. Deep Learning đã giúp máy tính thực thi những việc tưởng chừng như không thể vào 10 năm trước: phân loại cả ngàn vật thể khác nhau trong các bức ảnh, tự tạo chú thích cho ảnh, bắt chước giọng nói và chữ viết của con người, giao tiếp với con người, hay thậm chí cả sáng tác văn hay âm nhạc.Deep learning là một tập con của machine learning , Machine learning là một tập con của trí tuệ nhân tạo. Dưới đây là một hình minh họa cho điểu này.



Hình 7: Tổng quan về Machine Learning.

* 1. ***Định nghĩa về Machine learning.***

Machine learning là thuật toán có khả năng học tập từ dữ liệu, có nghĩa là chương trình máy tính sẽ học từ kinh nghiệm E để hoàn thành nhiệm vụ T, với kết quả được đo bằng phép đánh giá P, nếu kết quả của nó khi thực hiện nhiệm vụ T khi được đánh giá bởi P và cải thiện theo kinh nghiệm E.

* + - Nhiệm vụ T trong ML thường được mô tả thông qua việc một hệ thống ML xử lý một điểm dữ liệu như thế nào. Ví dụ trong bài toán phân loại ảnh thì mỗi ảnh là một điểm dữ liệu, trong bài toán phân nhóm khách hàng thì mỗi khách hàng là một điểm dữ liệu, trong bài toán xác định một tin nhắn có phải là rác hay không thì mỗi tin nhắn là một điểm dữ liệu. Mỗi điểm dữ liệu bao gồm nhiều đặc trưng khác nhau, mỗi đặc trưng được biểu diễn dưới dạng một con số.

Chúng ta thường biểu diễn một điểm dữ liệu như một vector x € Rd trong đó mỗi phần tử xi là một đặc trưng. Ví dụ trong một bức ảnh mỗi giá trị của một điểm ảnh có thể được coi là một đặc trưng, vector chứa toàn bộ giá trị các pixel của ảnh có thể coi là một vector đặc trưng.

* Phép đánh giá P : để kiểm tra hiệu năng của một thuật toán machine learning chúng ta cần phải thiết kế các phép đánh giá có thể đo đạc được kết quả. Thông thường khi thực hiện một thuật toán machine learning dữ liệu sẽ được chia thành hai phần riêng biệt : tập huấn luyện (training set) và tập kiểm thử (test set). Tập huấn luyện sẽ được dùng để tìm các tham số mô hình, tập kiểm thử được dùng để đánh giá năng lực của mô hình tìm được. Có một điểm cần lưu ý rằng khi tìm các tham số mô hình ta chỉ được dùng các thông tin trong tập huấn luyện. Việc đánh giá có thể được áp dụng lên cả hai tập hợp. Muốn mô hình thực hiện tốt trên tập kiểm thử thì nó trước hết phải hoạt động tốt trên tập huấn luyện.

Các thuật toán thực tế liên tục được cập nhật dựa trên dữ liệu mới được thêm vào, thuật toán này được gọi là online learning hoặc online training. Phần dữ liệu mới này ban đầu không được hệ thống sử dụng để xây dựng mô hình, nhưng về sau có thể được mô hình sử dụng để cải tiến. Ngược với online learning là offline learning, ở đó hệ thống xây dựng mô hình một lần dựa trên một tập chính là tập huấn luyện. Các điểm dữ liệu không được dùng trong quá trình xây dựng hệ thống được coi là tập kiểm thử.

* Kinh nghiệm E: Việc huấn luyện các mô hình machine learning có thể coi là việc cho chúng trải nghiệm trên các tập dữ liệu (dataset) chính là training set. Các tập dữ liệu khác nhau sẽ cho các mô hình trải nghiệm khác nhau chất lượng của các tập dữ liệu này cũng ảnh hưởng tới hiệu năng của mô hình.

Dựa trên tính chất của các tập dữ liệu, các thuật toán machine learning có thể phân loại thành hai nhóm chính là học có giám sát (supervised learning) và học không giám sát (unsupervised learning).

Supervised learning là thuật toán dự đoán đầu ra của một hay nhiều dữ liệu mới dựa trên các cặp (đầu vào, đầu ra) đã biết từ trước. Supervised learning là nhóm phổ biến nhất trong các thuật toán machine learning.

Ngược lại trong unsupervised learning chúng ta không biết được kết quả đầu ra mà chỉ biết các vector đặc trưng của dữ liệu đầu vào . Các thuật toán unsupervised learning sẽ dựa vào cấu trúc của dữ liệu để thực hiện một công việc nào đó ví dụ như phân nhóm hoặc giảm số chiều dữ liệu, tổng quát lại thì unsupervised learning chỉ có dữ liệu đầu vào X mà không có đầu ra tương ứng.

Ranh giới giữa unsupervised learning và supervised learning đôi khi là không rõ ràng, thông thường người ta thường coi các bài phân lớp (classification) và hồi quy (regression) là supervised learning các bài phân cụm (clustering) hay ước lượng phân phối (density estimation) là unsupervised learning.

* Hàm mất mát và tham số mô hình: Mỗi mô hình machine learning được mô tả bởi các tham số mô hình. Công việc của machine learning là đi tìm các tham số mô hình phù hợp với mỗi bài toán. Việc đi tìm các tham số mô hình có liên quan mật thiết đến các phép đánh giá. Mục đích của chúng ta là đi tìm các tham số mô hình sao cho các phép đánh giá cho kết quả tốt nhất. Trong bài toán phân lớp kết quả tốt có thể được hiểu là ít điểm dữ liệu bị phân lớp sai nhất, còn trong bài toán hồi quy kết quả tốt là khi sự sai lệch giữa đầu ra dự đoán và đầu ra thực sự là ít nhất. Quan hệ giữa một phép đánh giá và các tham số mô hình thường được mô tả thông qua một hàm số được gọi là hàm mất mát (lost function hay cost function). Hàm mất mát này thường có giá trị nhỏ khi phép đánh giá cho kết quả tốt và ngược lại. Việc đi tìm các tham số mô hình sao cho phép đánh giá trả về kết quả tốt tương đương với việc tối thiểu hàm mất mát. Như vậy việc xây dựng một mô hình machine learning chính là việc đi giải một bài toán tối ưu, quá trình đó đươc coi là quá trình học của machine learning.

1. **Hệ thống khuyến nghị (Recommendation System).** [5]

***6.1. Giới thiệu về hệ thống khuyến nghị.***

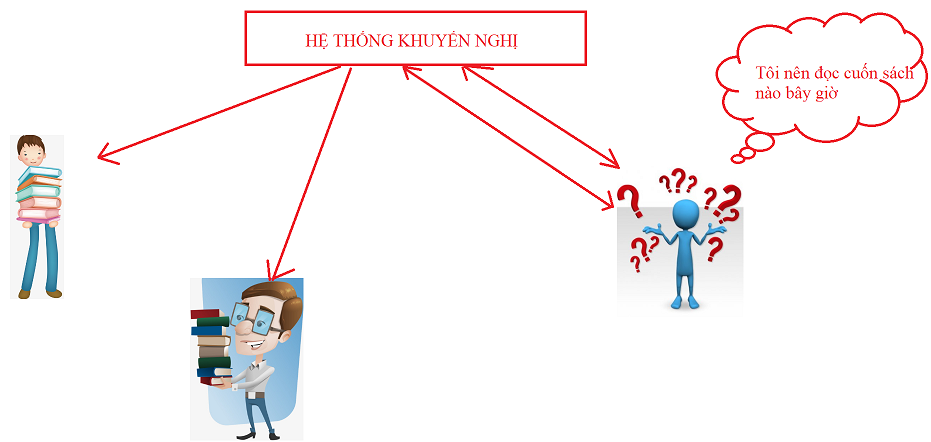
Trong thực tế có lẽ chúng ta đã gặp những hiện tượng sau đây nhiều lần. Youtube tự động chạy các clip liên quan đến clip bạn đang xem hoặc tự động gợi ý những clip mà có thể bạn sẽ thích hoặc khi bạn mua một món hàng trên Amazon ,hệ thống sẽ tự động gợi ý những sản phẩm có liên quan đến những sản phẩm đó hoặc nó có thể biết bạn thích món hàng nào dựa trên lịch sử mua hàng của bạn. Facebook gợi ý kết bạn, Netflix tự động gợi ý phim cho người dùng và còn rất nhiều ví dụ khác mà hệ thống có khả năng tự động gợi ý cho người dùng những sản phẩm mà họ có thể thích, bằng cách quảng cáo những đối tượng đó làm cho hiệu quả của việc maketing cũng sẽ tăng lên. Đằng sau các hệ thống này có một tên gọi chung đó là hệ thống khuyến nghị.

Hệ thống khuyến nghị (Recommendation system) là một mảng khá rộng của machine learning nhưng nó mới chỉ thực sự bùng nổ vào khoảng 10-15 năm gần đây. Có hai thực thể chính trong một recommendation system là người dùng (user) và sản phẩm (item) . Mục đích chính của recommendation system là dự đoán mức độ quan tâm của một người dùng tới một sản phẩm nào đó qua đó ta tìm cách gợi ý để đưa ra những sản phẩm phù hợp. Ví dụ khi chúng ta đi so sánh sự khác nhau căn bản giữa các cửa hàng thực và cửa hàng điện tử và xét trên khía cạnh lựa chọn sản phẩm để quảng bá, làm cách nào để chúng ta có thể quảng bá đúng sản phẩm tới khách hàng mà khách hàng không cần phải chạm tay vào từng sản phẩm để thử nghiệm, trong thực tế thì điều này rất khó và khi làm thương mại thì những sản phẩm bán chạy nhất chỉ chiếm một phần nhỏ của tổng số lượng sản phẩm.

Các cửa hàng thực thường có hai khu vực, một là khu trưng bày hai là kho. Nguyên tắc dễ thấy để đạt doanh thu cao là trưng bày ra các sản phẩm phổ biến nhất ở những nơi dễ nhìn thấy và cất những sản phẩm ít phổ biến ở trong kho, cách làm này có một hạn chế rõ rệt : những sản phẩm được trưng ra mang tính phổ biến chứ chưa chắc đã phù hợp với một khách hàng cụ thể. Một cửa hàng có thể có món hàng một khách hàng tìm kiếm nhưng có thể không bán được vì khách hàng không thể nhìn thấy sản phẩm đó trên giá, việc này dẫn đến khách hàng không tiếp cận được sản phẩm ngay cả khi chúng được trưng ra. Ngoài ra vì không gian có hạn cửa hàng không thể nào trưng ra tất cả các sản phẩm mà mỗi loại chỉ đưa ra một số lượng nhỏ sản phẩm. Ở đây phần lớn doanh thu (80%) đến từ phần nhỏ số sản phẩm phổ biến nhất(20%), nếu sắp xếp các sản phẩm của cửa hàng theo doanh số từ cao đến thấp ta sẽ nhận thấy có thể phần nhỏ các sản phẩm tạo ra phần lớn doanh số và một danh sách dài phía sau chỉ tạo ra một lượng đóng góp nhỏ.

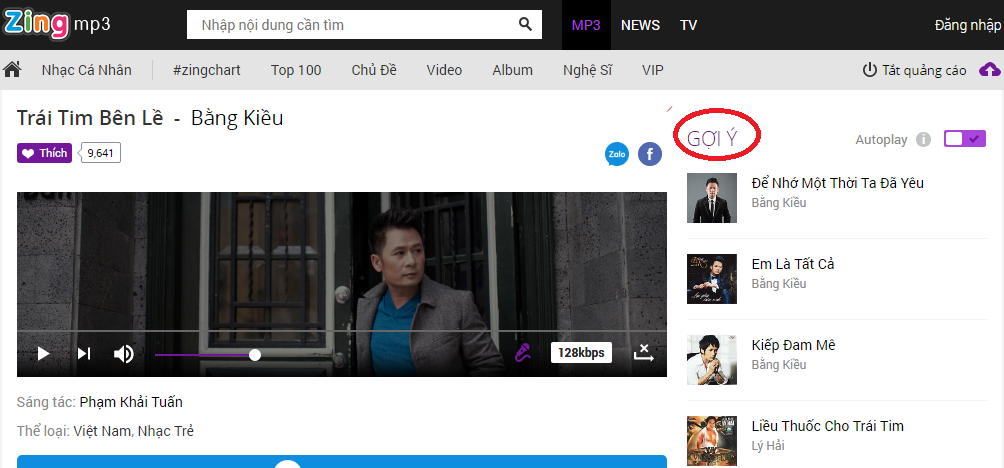
Với các cửa hàng điện tử nhược điểm trên hoàn toàn có thể tránh được vì không gian trưng bày của các cửa hàng điện tử gần như là vô tận, mọi sản phẩm đều có thể được trưng ra. Hơn nữa việc sắp xếp online là linh hoạt, tiện lợi với chi phí chuyển đổi gần như bằng 0 khiến việc mang sản phẩm tới khách hàng trở nên thuận tiện hơn. Doanh thu vì thế có khả năng được tăng lên.

Hệ thống gợi ý được ứng dụng rất thành công trong dự đoán sở thích/thói quen của người dùng dựa vào sở thích/thói quen của họ trong quá khứ. RS đang được ứng dụng trong rất nhiều lĩnh vực khác nhau như thương mại điện tử (hỗ trợ bán hàng trực tuyến), giải trí (gợi ý phim ảnh, bài hát,..), giáo dục đào tạo (gợi ý nguồn tài nguyên học tập, nghiên cứu,..). Chính vì khả năng ứng dụng rộng rãi của nó, RS mở ra nhiều tiềm năng trong nghiên cứu cũng như trong xây dựng các hệ thống thực tế, đặc biệt là các hệ hỗ trợ người dùng ra quyết định.



Hình 8: Hệ thống khuyến nghị sản phẩm.

Ngoài lĩnh vực thương mại điện tử như đã thấy thì hiện tại RS cũng được ứng dụng khá thành công trong nhiều lĩnh vực khác như trong giải trí: gợi ý phim ảnh (ví dụ, hệ thống của Netflix - www.netflix.com), gợi ý các video clip (ví dụ, hệ thống của YouTube - www.youtube.com); trong giáo dục và đào tạo (gợi ý nguồn tài nguyên học tập như sách, bài báo, địa chỉ web,… cho người học); trong các hệ thống trợ giảng thông minh ( như dự đoán năng lực học tập của người học ).



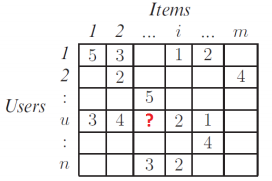
Hình 9: Hệ thống gợi ý bài nhạc của Mp3.zing.vn.



Hình 10: Hệ thống gợi ý sản phẩm của Amazon.

**6.2. Các khái niệm chính.**

Trong RS, thông thường người ta quan tâm đến ba thông tin chính là người dùng (user), mục tin (item, item có thể là sản phẩm, bộ phim, bài hát, bài báo,.. tùy hệ thống), và phản hồi (feedback) của người dùng trên mục tin đó (thường là các xếp hạng/đánh giá – rating biểu diễn mức độ thích/quan tâm của họ). Các thông tin này được biểu diễn thông qua một ma trận như trong Hình 11. Ở đó, mỗi dòng là một user, mỗi cột là một item, và mỗi ô là một giá trị phản hồi (ví dụ, xếp hạng) biểu diễn “mức độ thích” của user trên item tương ứng. Các ô có giá trị là những item mà các user đã xếp hạng trong quá khứ. Những ô trống là những item chưa được xếp hạng (điều đáng lưu ý là mỗi user chỉ xếp hạng cho một vài item trong quá khứ, do vậy có rất nhiều ô trống trong ma trận này – còn gọi là ma trận thưa – sparse matrix).



Hình 11: Ma trận biểu diễn dữ liệu trong RS (user-item-rating matrix)

Nhiệm vụ chính của RS là dựa vào các ô đã có giá trị trong ma trận trên (dữ liệu thu được từ quá khứ), thông qua mô hình đã được xây dựng, RS dự đoán các ô còn trống (của user hiện hành), sau đó sắp xếp kết quả dự đoán (ví dụ, từ cao xuống thấp) và chọn ra Top-N items theo thứ tự, từ đó gợi ý chúng cho người dùng.

Trong hệ thống gợi ý ta có:

• Gọi U là một tập hợp n người dùng (user), |U| = n, và u là một người dùng cụ thể nào đó (u ∈ U).

• Gọi I là một tập hợp m mục thông tin (item), |I| = m, và i là một mục thông tin cụ thể nào đó (i ∈ I).

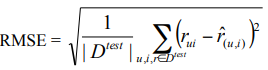
• Gọi R là một tập hợp các giá trị phản hồi của người dùng (các phản hồi này thường là ‘sở thích’ (preference) của người dùng) và rui ∈ R (R ⊂ ℜ) là xếp hạng của người dùng u trên mục thông tin i.

• Gọi Dtrain ⊆ U × I × R là tập dữ liệu huấn luyện.

• Gọi Dtest ⊆ U × I × R là tập dữ liệu kiểm thử.

• Gọi r: U × I → R. (u, i) ↦ rui

Mục tiêu của RS là tìm một hàm rˆ : U × I → ℜ sao cho một hàm ξ(r,rˆ) thỏa mãn một điều kiện nào đó. Ví dụ, nếu ξ là một hàm ước lượng độ chính xác thì nó cần phải được tối đa hóa, còn nếu ξ là hàm để đo độ lỗi như Root Mean Squared Error thì nó cần phải được tối tiểu.



Hình 12: Công thức đo độ lỗi RMSE.

**6.3. Thông tin phản hồi từ người dùng và hai dạng bài toán chính trong RS.**

Trong RS, giá trị phản hồi rui của mỗi người dùng trên mục tin sẽ được ghi nhận lại để làm cơ sở cho việc dự đoán các giá trị kế tiếp. Tùy theo hệ thống mà giá trị này sẽ có ý nghĩa khác nhau, ví dụ nó có thể dùng để đo độ “phù hợp” hay “mức độ thích” (thường là các đánh giá trên các sản phẩm) trong các hệ thống thương mại điện tử hay “năng lực/kết quả thực hiện” của người dùng trong các hệ thống e-learning.

Giá trị rui có thể được xác định một cách tường minh như thông qua việc đánh giá/xếp hạng (ví dụ, rating từ 1 sao đến 5 sao; hay like (1) và dislike (0),…) mà u đã bình chọn cho i; hoặc rui có thể được xác định một cách không tường minh thông qua số lần click chuột, thời gian mà u đã duyệt/xem i,…

Có 2 dạng bài toán chính trong RS là dự đoán xếp hạng (rating prediction) của các hệ thống có phản hồi tường minh như đã trình bày ở trên và dự đoán mục thông tin (item prediction/recommendation) là việc xác định xác suất mà người dùng thích mục tin tương ứng.

1. **Các kỹ thuật chính trong RS.**

Hiện tại, trong RS có rất nhiều giải thuật được đề xuất, tuy nhiên có thể gom chúng vào trong các nhóm chính :

• Nhóm giải thuật lọc cộng tác (Collaborative Filtering): trong nhóm này, các giải thuật chủ yếu sử dụng:

* Phương pháp láng giềng (Neighborhood-based, còn gọi là Memory based), trong đó hoặc là dựa trên dữ liệu quá khứ của người dùng “tương tự - similarity” (user-based ), hoặc là dựa trên dữ liệu quá khứ của những item “tương tự” (item-based)
* Dựa trên mô hình (Model-based): Nhóm này liên quan đến việc xây dựng các mô hình dự đoán dựa trên dữ liệu thu thập được trong quá khứ. Như mô hình Bayesian, các mô hình nhân tố tiềm ẩn: trong đó kỹ thuật phân rã ma trận (matrix factorization) là một điển hình.

• Nhóm giải thuật lọc trên nội dung (Content-based Filtering): Gợi ý các item dựa vào hồ sơ của người dùng hoặc dựa vào nội dung/thuộc tính của những item tương tự như item mà người dùng đã chọn trong quá khứ.

• Nhóm lai ghép: Kết hợp cả 2 cách trên.

• Nhóm kỹ thuật không cá nhân hóa (non-personalization)

Một trong những khuyết điểm của phương pháp lọc theo nội dung là khó khăn trong việc thu thập thông tin, trong khi phần lớn các mô hình dựa trên lọc cộng tác chỉ cần 3 thông tin (user id, item id, feedback) là có thể hoạt động tốt. Do vậy khuynh hướng hiện nay đa phần các nhà nghiên cứu thiên về nhóm lọc cộng tác.

**8. Giới thiệu ma trận utility.**[5]

**-** Như đã phân tích ở trên có hai thực thể chính trong hệ thống khuyến nghị (Recommendation system-RS) là user và item. Mỗi user sẽ có mức độ quan tâm tới từng item khác nhau. Mức độ quan tâm này nếu đã biết trước được gán cho một giá trị ứng với mỗi cặp user-item. Thông tin về mức độ quan tâm của một user tới một item có thể được thu thập thông qua một hệ thống đánh giá (review và rating). Hoặc có thể dựa trên việc user đã click vào thông tin của item trên website hoặc có thể dựa trên việc thời gian và số lần một user xem thông tin của một item. Trong đề tài này em sẽ dựa trên hệ thống rating.

- Với một hệ thống rating, mức độ quan tâm của một user tới một item được đo bằng giá trị của user đó đánh giá cho item đó, chẳng hạn số sao cho mỗi sản phẩm. Tập hợp tất cả các rating, bao gồm cả những giá trị chưa biết cần được dự đoán tạo nên một ma trận gọi là ma trận utility. Công việc của một hệ thống khuyến nghị là dự đoán giá trị của những user chưa đánh giá từ đó đưa ra gợi ý cho user, vì vậy bài toán khuyến nghị sản phẩm được coi là bài toán hoàn thiện ma trận.

- Thông thường thì có rất nhiều user và item trong hệ thống và mỗi user thường rate một số lượng rất nhỏ các item thậm chí có những user không rate một item nào , rõ ràng càng rating nhiều thì độ chính xác của hệ thống cần được cải thiện, vì vậy các hệ thống luôn khuyến khích user bày tỏ sự quan tâm của họ tới các item thông qua việc đánh giá các item đó. Việc đánh giá các item vì thế không những giúp các user khác biết được chất lượng của item đó mà còn giúp cho hệ thống biết được sở thích của user, qua đó có chiến lược quảng bá hợp lý.

Không có ma trận Utility, gần như không thể gợi ý được sản phẩm tới ngừời dùng, ngoài cách luôn luôn gợi ý các sản phẩm phổ biến nhất. Vì vậy, trong các hệ thống khuyến nghị việc xây dựng ma trận Utility là rất quan trọng. Tuy nhiên, việc xây dựng ma trận này thường có gặp nhiều khó khăn. Có hai hướng tiếp cận phổ biến để xác định giá trị rating cho mỗi cặp user-item trong ma trận Utility:

* Nhờ người dùng đánh giá (rate) sản phẩm. Ví dụ như Amazon luôn nhờ người dùng rate các sản phẩm của họ bằng cách gửi các email nhắc nhở nhiều lần. Rất nhiều hệ thống khác cũng làm việc tương tự. Tuy nhiên, cách tiếp cận này có một vài hạn chế, vì thường thì người dùng ít khi rate sản phẩm. Và nếu có, đó có thể là những đánh giá thiên lệch bởi những người sẵn sàng rate.
* Hướng tiếp cận thứ hai là dựa trên hành vi của users. Rõ ràng, nếu một người dùng mua một sản phẩm trên Amazon, xem một clip trên Youtube (có thể là nhiều lần), hay đọc một bài báo, thì có thể khẳng định rằng ngừời dùng đó thích sản phẩm đó. Facebook cũng dựa trên việc bạn like những nội dung nào để hiển thị tin tức của bạn những nội dung liên quan. Bạn càng đam mê facebook, facebook càng được hưởng lợi, thế nên nó luôn mang tới bạn những thông tin mà khả năng cao là bạn muốn đọc. Thường thì với cách này, ta chỉ xây dựng được một ma trận với các thành phần là 1 và 0, với 1 thể hiện người dùng thích sản phẩm, 0 thể hiện chưa có thông tin. Trong trường hợp này, 0 không có nghĩa là thấp hơn 1, nó chỉ có nghĩa là ngừời dùng chưa cung cấp thông tin. Chúng ta cũng có thể xây dựng ma trận với các giá trị cao hơn 1 thông qua thời gian hoặc số lượt mà người dùng xem một sản phẩm nào đó. Đôi khi, nút dislike cũng mang lại những lợi ích nhất định cho hệ thống, lúc này có thể gán giá trị tương ứng bằng -1 chẳng hạn.

**9.** **Kết luận chương 2.**

Trong phần này em đã trình bày sơ lược qua về các khái niệm của machine learning và tổng quan về hệ thống khuyến nghị cách xây dựng ma trận utility để tính toán dữ liệu trong thực tế, trong phần tiếp theo em sẽ đi phân tích sâu hơn về thuật toán trong hệ thống khuyến nghị.

# CHƯƠNG 3. PHÂN TÍCH

## **1. Sơ lược về Collaborative filtering.** [6]

**-** Trong cuộc sống hằng ngày, mọi người thường tin vào những lời giới thiệu từ những người khác thông qua lời nói, thư từ văn bản, các nguồn tin thu được trên các phương tiện thông tin đại chúng. Về bản chất, collabrative filtering cũng chính là hình thức tư vấn như trên, tuy nhiên việc này được thực hiện tự động bởi các máy tính. Các nhà phát triển hệ thống gợi ý đầu tiên vào năm 1992 (D. Goldberg et al., 1992) đã đưa ra thuật ngữ “collabrative filtering” (lọc cộng tác). Giả thuyết của collabrative filtering là: “Nếu người dùng u và u' đánh giá cho n sản phẩm tương tự nhau, hoặc có hành vi tương tự nhau (như: xem, mua, nghe…) thì họ sẽ có các đánh giá tương tự nhau đối với các sản phẩm khác”.

Trong hệ thống gợi ý, người ta thường biểu diễn các đánh giá của người dùng cho các sản phẩm qua ma trận gồm một tập người dùng U và tập sản phẩm I như Hình 13.



Hình 13: Ma trận mô tả user-item.

Như đã nói ở trước thì người dùng được ký hiệu là u và các sản phẩm được ký hiệu là i. Ký hiệu rui để chỉ mức độ thích của người dùng u cho một sản phẩm i nào đó, chẳng hạn giá trị này trong khoảng từ 1 (không thích) đến 5 (thích nhất) đối với dữ liệu bán hàng hoặc nghe nhạc, ui là một hàm dùng để dự đoán đánh giá của người dùng u cho sản phẩm i (có thể xem như hàm tiện ích). Theo mô hình nearest-neighborhood, việc dự đoán đánh giá của một người dùng trên một mục tin (item) được dựa trên các đánh giá của những người dùng “lân cận” trên mục tin đó. Vì vậy, một độ đo tương tự (similarity) giữa những người dùng cần được định nghĩa trước khi một tập những người dùng láng giềng gần nhất được chọn ra.

Hệ tư vấn dựa trên collaborative filtering (lọc cộng tác) sẽ dự đoán các giá trị đánh giá của những sản phẩm cho những người dùng cụ thể dựa trên trọng số ban đầu của sản phẩm được cung cấp bởi người dùng khác. Giống như trước, hàm ui của sản phẩm i cho người dùng u được đánh giá dựa trên những hàm (uj, i) được gán bởi sản phẩm i và người dùng uj Є U, đây là những người mà có cùng sở thích và thị hiếu giống u. Chẳng hạn, trong ứng dụng về tư vấn sách, để tư vấn những sách mới cho người dùng u, hệ tư vấn lọc cộng tác sẽ tìm những người tương tự với người dùng u; nghĩa là những người dùng khác nhau nhưng có cùng thị hiếu về sách (trọng số về các sách giống nhau là như nhau). Sau đó, chỉ những quyển sách có độ quan tâm nhất sẽ được tư vấn cho người dùng.

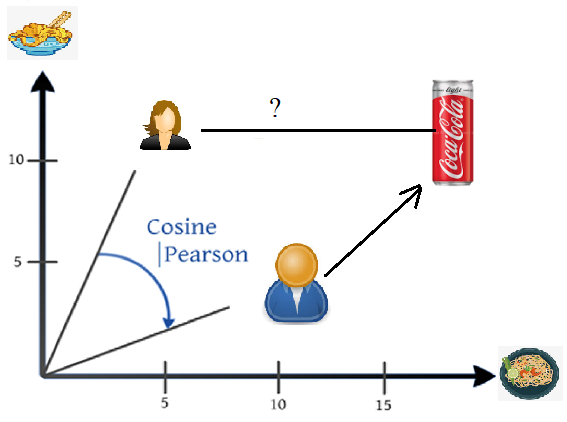
Mở rộng của vấn đề này là tìm tập N sản phẩm mà rất có thể người dùng sẽ quan tâm – công việc chính trong các hệ tư vấn hiện nay. Giả sử rằng chúng ta có một hệ thống dự đoán các đánh giá của người dùng lên các sản phẩm mới, và dựa vào đó ta có thể lấy ra N sản phẩm được dự đoán có trọng số cao nhất. Cuối cùng, hai vấn đề quan trọng nhất đó là làm thế nào để tìm được tập K người dùng tương tự với người dùng u nhất dựa trên thị hiếu về sản phẩm hay làm thế nào để tìm ra tập K sản phẩm tương tự với sản phẩm i nhất để có thể đưa ra các dự đoán trọng số đánh giá của người dùng và sản phẩm tương ứng.

Phương pháp lọc cộng tác có đặc trưng cơ bản là nó thường sử dụng toàn bộ dữ liệu đã có để dự đoán đánh giá của một người dùng nào đó về sản phẩm mới. Nhờ lợi thế là nó có khả năng đưa trực tiếp dữ liệu mới vào bảng dữ liệu, do đó nó đạt được khá nhiều thành công khi được áp dụng vào các ứng dụng thực tế. Cũng do đó mà các kỹ thuật này thường đưa ra các dự đoán chính xác hơn trong các hệ trực tuyến.

**2. Công thức tính độ tương tự (Similarity).**

Thông thường, có hai cách tiếp cận của lọc cộng tác theo mô hình K láng giềng: hệ dựa trên người dùng (User) – tức dự đoán dựa trên sự tương tự giữa các người dùng và hệ dựa trên sản phẩm (Item) – dự đoán dựa trên sự tương tự giữa các sản phẩm. Hệ dựa trên người dùng xác định sự tương quan giữa hai người dùng thông qua việc so sánh các đánh giá của họ trên cùng sản phẩm, sau đó dự đoán đánh giá sản phẩm i bởi người dùng u, hay chính là đánh giá trung bình của những người dùng tương tự với người dùng u.

Độ tương tự giữa người dùng u và người dùng u' có thể được tính theo Pearson (L. Herlocker et al., 1999), Cosin, khoảng cách Euclidean nhưng phân tính thực nghiệm cho thấy rằng đối với hệ dựa trên người dùng thì tính độ tương tự theo Pearson sẽ tốt hơn so với một vài cách khác như công thức cosin. Dưới đây sẽ trình bày một vài công thức tính độ tương quan nhưng khi chạy thực nghiệm chương trình thì em chỉ tập trung vào công thức Pearson similarity.



Hình 14: Mô tả công thức tính độ tương quan.

* 1. **Các bước tính toán.** [6]

Cho danh sách m người dùng U={u1,u2,..um} và một danh sách gồm n sản phẩm

I = {i1,i2, ... in}.

rui là đánh giá sản phẩm i của người dùng u và r nhận giá trị từ 0 đến 5. Với r=0 khi sản phẩm i chưa được đánh giá bởi người dùng u. Mỗi người dùng u có danh sách sản phẩm Iu mà người dùng đã đánh giá về những sản phẩm đó.

Yêu cầu : Dự đoán giá trị số Pui dự đoán khả năng sản phẩm i € Iu cho người dùng u.

Tư vấn danh sách Ip gồm n sản phẩm mà người dùng u thích nhất.

Các bước trong quá trình gợi ý theo phương pháp collaborative filtering.

Bước 1: Tiền xử lý dữ liệu: Dữ liệu được thu thập từ những đánh giá sản phẩm của người dùng.

Bước 2: Xây dựng ma trận đánh giá. Hàng là người dùng cột là sản phẩm (tùy theo định nghĩa của mỗi người)

Bước 3: Tính độ tương tự của hai sản phẩm.

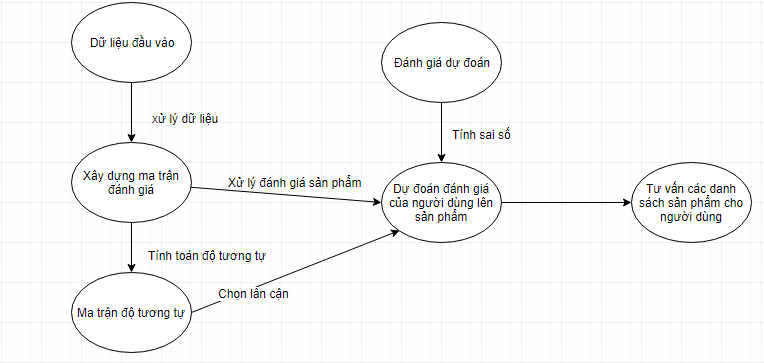
Bước 4: Tính dự đoán.

Input đầu vào: Số sao mà user u đánh giá lên item i.

Output đầu ra: Số sao mà user u chưa đánh giá cho item i và số sản phẩm cần gợi ý.

|  |
| --- |
| Các bước tính toán của giải thuật |
| 1: function USER-CF(u, r, Dtrain)  2: for u=1 to N do  3: for u’=1 to N’ do  4: Tính Sim\_uu'  5: end for  5: end for  5: Sort Sim\_uu' // sắp xếp giảm dần độ tương tự  6: for k=1 to K do  7: Ku ← k // Các người dùng k gần nhất của u hoặc có thể tính theo ngưỡng  8: end for  9: for i = 1 to M do  10: Tính ui  11: end for  12: end function |
| Chú thích : u: đánh giá trung bình của người dùng u trên tất cả các item  r: đánh giá của người dùng trên tập huấn luyện  K: người dùng k gần nhất  N, N’, M: người dùng thứ n và thứ m  Dtrain: tập dữ liệu huấn luyện |

Bảng 1: Mô tả thuật toán collaborative filtering.



Hình 15: Mô hình hệ thống lọc cộng tác dựa trên sản phẩm.

* 1. **Công thức tính độ tương tự của cosin.**

Công thức tính độ tương tự theo Cosin:

Simcosin(u, u’) =

Hình 16: Công thức tính Cosin theo user-user.

Simcosin(i, i’) =

Hình 17: Công thức tính Cosin theo item-item.

Trong đó :

rui và ru’i là đánh giá của người dùng u và u’ hoặc (item i và i’) trên item i tương ứng.

Iuu’ là tập các item được đánh giá bởi cả người dùng u và người dùng u’.

Uii’ là tập các người dùng có đánh giá trên cả hai mục tin i và i’.

Xét một ví dụ có 4 người dùng đánh giá cho 5 sản phẩm với mức độ đánh giá (rating) từ 1 đến 5. Các dấu ‘?’ là giá trị còn thiếu mà ta cần phải dự đoán.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Người | Sản phẩm | | | | |
|  | Item 1 | Item 2 | Item 3 | Item 4 | Item 5 |
| User 1 | 5 | 3 | 4 | 4 | ? |
| User 2 | 3 | 1 | 2 | 3 | 3 |
| User 3 | 4 | 3 | 4 | 3 | 5 |
| User 4 | 3 | 3 | 1 | 5 | 4 |
| User 5 | 1 | 5 | 5 | 2 | 1 |

Bảng 2: Giá trị đánh giá của user lên một item.

Áp dụng công thức Cosin ta tính độ tương tự của hai user như sau:

Gọi user 1 = u1, user 2 = u2

Cos(u1, u2) = =0.97532

Tương tự với các user còn lại ta tính được bảng sau.

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Người | Sản phẩm | | | | | CosinSimilarity |
|  | Item 1 | Item 2 | Item 3 | Item 4 | Item 5 | User1-user |
| User 1 | 5 | 3 | 4 | 4 | ? | 1 |
| User 2 | 3 | 1 | 2 | 3 | 3 | 0.97532 |
| User 3 | 4 | 3 | 4 | 3 | 5 | 0.99224 |
| User 4 | 3 | 3 | 1 | 5 | 4 | 0.89072 |
| User 5 | 1 | 5 | 5 | 2 | 1 | 0.79668 |

Bảng 3: Độ tương quan của user-user theo công thức cosin.

Tương tự với độ tương quan của user-user ta cũng có thể tính toán độ tương quan của các item-item như sau:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Người | Sản phẩm | | | | |
|  | Item 1 | Item 2 | Item 3 | Item 4 | Item 5 |
| User 1 | 5 | 3 | 4 | 4 | ? |
| User 2 | 3 | 1 | 2 | 3 | 3 |
| User 3 | 4 | 3 | 4 | 3 | 5 |
| User 4 | 3 | 3 | 1 | 5 | 4 |
| User 5 | 1 | 5 | 5 | 2 | 1 |
| Cosin(i5-i) | 0.9941 | 0.7388 | 0.7226 | 0.9395 | 1 |

Bảng 4: Độ tương quan của item-item theo công thức cosin.

Sau khi tính toán độ tương tự giữa các người dùng hay giữa các item, chúng ta có thể dự đoán đánh giá của người dùng u trên sản phẩm i với công thức:

Pred(u, i) =

**Hình 18:** Công thức dự đoán số sao của user u lên item i.

Với là giá trị đánh giá của user u lên item i.

L(u): là tập các sản phẩm lân cận của người dùng u

Tương tự với phương pháp Item-kNN, xếp hạng (đánh giá/ feedback) của người dùng u trên sản phẩm i được dự đoán qua công thức:

Pred(u, i) =

Hình 19: Công thức dự đoán đánh giá theo Pearson similarity (item-item).

Áp dụng công thức trên ta có:

15 = =3.3507

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Người | Sản phẩm | | | | |
|  | Item 1 | Item 2 | Item 3 | Item 4 | Item 5 |
| User 1 | 5 | 3 | 4 | 4 | 3.3507 |
| User 2 | 3 | 1 | 2 | 3 | 3 |
| User 3 | 4 | 3 | 4 | 3 | 5 |
| User 4 | 3 | 3 | 1 | 5 | 4 |
| User 5 | 1 | 5 | 5 | 2 | 1 |

Bảng 5: Dự đoán đánh giá của user u lên item i theo user-user similarity.

Đối với item-item sẽ là:

15 = = 4.0751

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Người | Sản phẩm | | | | |
|  | Item 1 | Item 2 | Item 3 | Item 4 | Item 5 |
| User 1 | 5 | 3 | 4 | 4 | 4.0751 |
| User 2 | 3 | 1 | 2 | 3 | 3 |
| User 3 | 4 | 3 | 4 | 3 | 5 |
| User 4 | 3 | 3 | 1 | 5 | 4 |
| User 5 | 1 | 5 | 5 | 2 | 1 |

Bảng 6: Dự đoán đánh giá của user u lên item i theo item-item similarity.

* 1. **Công thức tính độ tương tự Pearson.**

Cũng giống với công thức tính độ tương tự của Cosin, công thức Pearson được tính với công thức như sau:

Công thức tổng quát:

Sim(u, u’) =

Công thức mở rộng trong Mahout:

Sim(u, u’) =

Hình 20: Công thức tính độ tương quan Pearson theo user-user.[10]

Trong đó:

, là tập các người dùng có đánh giá trên cả hai sản phẩm i và i’.

là tập các sản phẩm được đánh giá bởi cả người dùng u và người dùng u'.

là giá trị đánh giá trung bình của tất cả các người dùng u trên sản phẩm i.

là giá trị đánh giá trung bình của tất cả các người dùng u’ trên sản phẩm i.

Sau khi tính toán độ tương tự giữa các người dùng hay giữa các sản phẩm, chúng ta có thể dự đoán đánh giá của người dùng u trên sản phẩm i theo công thức trong Hình 18.

Về cơ bản công thức trong hình 20 cũng dựa theo công thức Cosin nhưng có tính bổ trợ thêm giá trị trung bình của người dùng hoặc sản phẩm (item), việc bổ trợ thêm giá trị trung bình này trong thư viện mahout nhằm đảm sự cân bằng nếu người dùng dễ tính hoặc khó tính đánh giá một sản phẩm (đối với những người dễ tính họ có thể đánh giá 5 sao cho một sản phẩm, nhưng ngược lại đối với người dùng khó tính họ có thể đánh giá sản phẩm này có thể 2 hoặc 3 sao)

Áp dụng vào ví dụ bên trên ta có:

Sum(u1, u2) = 5\*3+3\*1+4\*2+4\*3=38.

1 = (5+3+4+4)/4=4, 2 = (3+1+2+3)/4=2.25

Sum(u1) = (5+3+4+4)=16, Sum(u2) = (3+1+2+3)=9,

Sum(u12) = 52 + 32 + 42 + 42 =66, Sum(u22) = 32 + 12 + 22 + 32 = 23

Sim(u1,u2) = = 0.8528.

Tương tự giữa user 1 với các user còn lại ta sẽ tính được similarity theo Pearson như bên dưới:

|  |  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- | --- |
| Người | Sản phẩm | | | | | PearsonSimilarity |
|  | Item 1 | Item 2 | Item 3 | Item 4 | Item 5 | User-user |
| User 1 | 5 | 3 | 4 | 4 | ? | 1 |
| User 2 | 3 | 1 | 2 | 3 | 3 | 0.8528 |
| User 3 | 4 | 3 | 4 | 3 | 5 | 0.7071 |
| User 4 | 3 | 3 | 1 | 5 | 4 | 0.0 |
| User 5 | 1 | 5 | 5 | 2 | 1 | -0.7921 |

Bảng 7: Độ tương quan của user-user theo công thức Pearson.

Tương tự với độ tương quan của người dùng ta cũng có thể tính toán độ tương quan của các item-item theo công thức Pearson như sau:

Sum(i5, i1) = 3\*3+4\*5+3\*4+1\*1=42.

1 = (3+4+3+1)/4=2.75, 5 = (3+5+4+1)/4=3.25

Sum(i1) = (3+4+3+1)=11, Sum(i5) = (3+5+4+1)=13,

Sum(i12) = 32 + 42 + 32 + 12 =35, Sum(i52) = 32 + 52 + 42 + 12 = 51.

Sim(i1,i1) = = 0.96945.

Tương tự như trên ta sẽ tính được độ tương tự của item-item:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Người | Sản phẩm | | | | |
|  | Item 1 | Item 2 | Item 3 | Item 4 | Item 5 |
| User 1 | 5 | 3 | 4 | 4 | ? |
| User 2 | 3 | 1 | 2 | 3 | 3 |
| User 3 | 4 | 3 | 4 | 3 | 5 |
| User 4 | 3 | 3 | 1 | 5 | 4 |
| User 5 | 1 | 5 | 5 | 2 | 1 |
| Pearson(i5,i) | 0.96945 | -0.4781 | -0.4276 | 0.5816 | 1 |

Bảng 8: Độ tương quan của item-item theo công thức Pearson.

Sau khi tính toán độ tương tự giữa các người dùng hay giữa các sản phẩm, chúng ta có thể dự đoán đánh giá của người dùng u trên sản phẩm i như sau:

Rating (1,5) = = 3.9065 (user-user)

Ta được kết quả như sau:

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Người | Sản phẩm | | | | |
|  | Item 1 | Item 2 | Item 3 | Item 4 | Item 5 |
| User 1 | 5 | 3 | 4 | 4 | 3.9065 |
| User 2 | 3 | 1 | 2 | 3 | 3 |
| User 3 | 4 | 3 | 4 | 3 | 5 |
| User 4 | 3 | 3 | 1 | 5 | 4 |
| User 5 | 1 | 5 | 5 | 2 | 1 |

Bảng 9: Dự đoán đánh giá của user u lên item i theo user-user similarity.

Rating (1,5) = = 6.2046 (item-item)

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Người | Sản phẩm | | | | |
|  | Item 1 | Item 2 | Item 3 | Item 4 | Item 5 |
| User 1 | 5 | 3 | 4 | 4 | 5 |
| User 2 | 3 | 1 | 2 | 3 | 3 |
| User 3 | 4 | 3 | 4 | 3 | 5 |
| User 4 | 3 | 3 | 1 | 5 | 4 |
| User 5 | 1 | 5 | 5 | 2 | 1 |

Bảng 10: Dự đoán đánh giá của user u lên item i theo item-item similarity.

1. **Đánh giá thuật toán.**
   1. ***Dữ liệu thực nghiệm.****[5]*

Với việc đánh giá này, tôi sẽ sử dụng tập dữ liệu MovieLens 100K (đây là dữ liệu chuẩn dùng để đánh giá giải thuật lọc cộng tác)

[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) đánh giá (ratings) từ 943 người dùng (users) cho 1682 bộ phim. Ngoài ra, chúng ta cũng có thể tìm thấy các bộ cơ sở dữ liệu tương tự với khoảng 1M, 10M, 20M ratings.

Trong [bộ cơ sở dữ liệu MovieLens 100k](https://grouplens.org/datasets/movielens/100k/) này sẽ có rất nhiều các file nhỏ như 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 cần sử dụng ở đây.
* **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.
* **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.
* **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. Trong này, 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.
  1. ***Độ đo.***

Để đánh giá hiệu quả của thuật toán chúng ta cần sử dụng công thức Root Mean Squared Error (RMSE) :

RMSE = 2

Với Dtest : Số lượng các cặp đánh giá giao giữa người dùng u và u’ hoặc số lượng các cặp sản phẩm i và i’ trên tập kiểm tra.

rui : dự đoán của người dùng u trên item i trên tập kiểm tra.

ui: dự đoán của người dùng u trên item i.

Dưới đây là kết quả đánh giá dựa trên tập dữ liệu của movilens-100k sử dụng tập dữ liệu là **ub.base** việc đánh giá này nhằm kiểm định kết quả đánh giá của chúng ta với các giá trị thực nhưng ở đây tôi sẽ sử dụng lại dữ liệu trong tập **ub.base** để đánh giá thuật toán.

Bảng 11: Kết quả tính RMSE với K từ 2-30

|  |  |  |
| --- | --- | --- |
| Tham số K | RMSE với Cosin | RMSE với Pearson |
| 2 | 1.249844710851007 | 1.2754938650252967 |
| 10 | 1.2096312779354008 | 1.1874008265483782 |
| 20 | 1.1822267646911335 | 1.1475808403818566 |
| 30 | 1.1694813898497562 | 1.1383470564033484 |
| 40 | 1.1493500061296122 | 1.127845258340805 |
| 50 | 1.1367208291338056 | 1.1116822751291577 |

Trong bảng kết quả trên, với K lớn thì lỗi RMSE cho kết quả thấp (không đáng kể) . Từ tập dữ liệu trên tiếp tục chọn K=100-400 để kiểm thử kết quả thực nghiệm như sau:

Bảng 12: Kết quả tính RMSE với K từ 100-390

|  |  |  |
| --- | --- | --- |
| Tham số K | RMSE với Cosin | RMSE với Pearson |
| 100 | 1.085864106044843 | 1.0672557829858877 |
| 200 | 1.047492924676348 | 1.0327148555677892 |
| 300 | 1.0258476282724776 | 1.0272725536472702 |
| 400 | 1.0171486860535335 | 1.032775086858951 |

Qua hai kết quả đánh giá trên ta thấy về mặt tổng quan thì kết quả dự đoán của Pearson tốt hơn Cosin vì nó cho kết quả lỗi nhỏ hơn và khi K càng lớn thì độ lỗi sẽ giảm dần nhưng vẫn không thay đổi nhiều vì vậy với lỗi RMSE càng nhỏ chúng ta sẽ gợi ý sản phẩm đến cho người dùng sẽ càng chính xác hơn.

Xét ví dụ bên trên ta sẽ có độ đo RMSE theo công thức Cosin và Pearson Similarity.

Bảng 13: Độ đo của Cosin và Pearson Similarity

|  |  |  |
| --- | --- | --- |
|  | Cosin | Pearson |
| User-user | 2.1636011565627085 | 1.0717968940734863 |
| Item-item | 1.493375939289739 | 1.0717968940734863 |

Nhìn vào bảng trên ta thấy công thức tính độ tương quan theo công thức Pearson sẽ cho kết quả tốt hơn về cả User và item, nhưng bên cạnh đó thì độ tương quan theo item-item vẫn cho kết quả là tối ưu hơn.

1. **Ưu và khuyết điểm của Collaborative filtering ( lọc cộng tác ).**[6]
   1. ***Ưu điểm.***

Xét một cách tổng quát thì thuật toán này không yêu cầu quá nặng vào việc tính toán và do đó có thể đưa ra những tư vấn có độ chính xác cao và nhanh chóng cho một số lượng lớn người dùng. Bởi vì hệ này không yêu cầu mô tả nội dung một cách tường minh mà chỉ dựa vào sự đánh giá của người dùng để ước lượng xem có bao nhiêu người dùng đồng ý và không đồng ý. Đó cũng chính là lý do tại sao mà hệ này có thể đưa ra những gợi ý bất ngờ cho người dùng.

* 1. ***Khuyết điểm.***

Có một số khuyết điểm trong collaborative filtering nhưng ở đây tôi chỉ đưa ra 3 nhược điểm cơ bản sau:

- Vấn đề về người dùng mới: Khi có một người dùng mới chưa cung cấp bất cứ một đánh giá nào hoặc một tài nguyên mới chưa nhận được bất cứ đánh giá nào từ người dùng. Hệ thống thiếu dữ liệu để cho ra các tư vấn phù hợp, vì vậy cần đề xuất khai thác dữ liệu từ người dùng hoặc có thể sử dụng tư vấn lai..

- Vấn đề về item mới: Trong lọc thông tin các sản phẩm thường xuyên được bổ sung cập nhật vào hệ thống, khi xuất hiện một sản phẩm mới, tất cả đánh giá của người dùng cho sản phẩm này đều là rỗng. Do đó lọc cộng tác sẽ không thể tư vấn sản phẩm cho bất kỳ người dùng nào trong hệ thống.

- Vấn đề dữ liệu thưa: Kết quả dự đoán của lọc cộng tác phụ thuộc chủ yếu dựa vào số đánh giá có trước của người dùng đối với các sản phẩm. Tuy nhiên đối với các hệ thống thực tế, số lượng người dùng và sản phẩm là rất lớn, số những người đánh giá biết trước thường rất nhỏ so với số lượng các đánh giá cần được dự đoán.

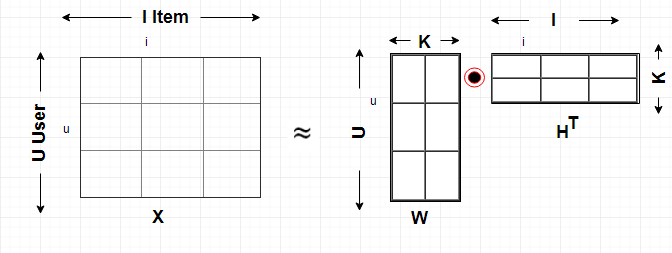
* 1. ***Cách khắc phục***

Để khắc phục vấn đề trên, chúng ta có thể kết hợp cả lọc cộng tác (collaborative filtering) và lọc dựa trên một số thuộc tính của người dùng cung cấp để dự đoán sở thích của khách hàng mới.

Ví dụ như thông qua việc đăng ký thông tin tài khoản của khách hàng, chúng ta sẽ thu thập thêm một số thông tin của người dùng mới bằng cách tạo một biểu mẫu (form) cho người dùng nhập một số thông tin cần thiết như: tính cách, nghề nghiệp, thu nhập hoặc một số thông tin về sở thích... Việc ghi nhận lại các thông tin này của người dùng sẽ giúp hệ thống tư vấn tốt hơn cho người dùng mới này. Đối với sản phẩm mới nhập vào, chúng được hiển thị đầu tiên trên trang web hoặc có biểu tượng 'New' để nhận biết đây là sản phẩm mới của hệ thống. Ngoài ra, khi hiển thị chi tiết mỗi sản phẩm, trang web có một không gian để hiển thị các sản phẩm tương tự với sản phẩm mà người dùng đang xem dựa vào một số thuộc tính tương tự. Vì vậy, những sản phẩm mới nhập cũng có thể được gợi ý cho người dùng.

1. **Kĩ thuật nâng cao-Kĩ thuật phân rã ma trận.[7]**

Kỹ thuật phân rã ma trận là việc chia một ma trận lớn X thành hai ma trận có kích thước nhỏ hơn W và H, sao cho ta có thể xây dựng lại X từ hai ma trận nhỏ hơn này càng chính xác càng tốt, nghĩa là X ~ WHT , được minh họa như trong Hình 21. Trong đó, W ∈ ℜ|U|xK là một ma trận mà mỗi dòng u là một véc tơ bao gồm K nhân tố tiềm ẩn mô tả người dùng u; và H ∈ ℜ|I|xK là một ma trận mà mỗi dòng i là một véc tơ bao gồm K nhân tố tiềm ẩn mô tả cho item i ( K<<|U|, K<<|I|).



Hình 21: Mô tả chia ma trận utility

Gọi wuk và hik là các phần tử tương ứng của hai ma trận W và H, khi đó xếp hạng của người dùng u trên mục tin i được dự đoán bởi công thức:

Như vậy vấn đề then chốt của kỹ thuật MF là làm sao để tìm được giá trị của hai tham số W và H. Hai tham số này có được bằng cách tối ưu hóa hàm mục tiêu. Hàm mục tiêu thông dụng nhất trong RS tương tự như hàm lỗi RMSE, biểu diễn như sau:

Một trong những kỹ thuật có thể dùng để tối ưu hóa hàm mục tiêu là dùng SGD (Stochastic Gradient Descent). Để tối ưu hóa hàm mục tiêu, trước tiên ta khởi tạo các giá trị ngẫu nhiên cho W và H, sau đó từng bước cập nhật giá trị của chúng cho đến khi hàm mục tiêu hội tụ về giá trị nhỏ nhất . Để làm được điều đó, ta cần phải biết là nên tăng hay nên giảm các giá trị của W và H qua mỗi lần cập nhật, do vậy cần phải tìm đạo hàm tương ứng của chúng:

Sau khi tìm đạo hàm, các phần tử của W và H sẽ được cập nhật với giá trị của đạo hàm, qua công thức:

Trong đó β là tốc độ học (0 < β < 1). Quá trình cập nhật sẽ được thực hiện đến khi nào hàm mục tiêu đạt được giá trị nhỏ nhất. Để ngăn chặn sự quá khớp (overfitting), người ta thay đổi hàm mục tiêu để thêm vào một đại lượng gọi là chính tắc hóa (regularization) để điều khiển độ lớn của các giá trị trong W và H. Hàm mục tiêu bây giờ trở thành:

Trong đó λ là hệ số chính tắc hóa (0 ≤ λ ≤ 1) .

Với hàm mục tiêu mới này, giá trị của wuk và hik được cập nhật qua công thức:

Giải thuật trong bảng 14 mô tả chi tiết quá trình huấn luyện mô hình MF bằng kỹ thuật SGD. Trước hết, giá trị của W và H được khởi tạo một cách ngẫu nhiên, ví dụ, lấy từ phân phối chuẩn N (0, σ2 ) với trung bình là 0 và độ lệch là σ2 = 0.01, như ở dòng 2-3. Ở mỗi lần lặp, ta chọn ngẫu nhiên một dòng (u, i, rui) từ tập dữ liệu huấn luyện, kế tiếp là tính giá trị dự đoán cho u và i này, như ở các dòng 5-9. Sau đó cập nhật các giá trị của W and H tương ứng của u và i như ở dòng 11-14.

Dưới đây là giải thuật phân rã ma trận sử dụng SGD. Với Số lần lặp (Iter); số nhân tố tiềm ẩn K; tốc độ học β; và regularization λ.

Bảng 14: Giải thuật phân rã ma trận.

|  |
| --- |
| Thuật toán |
| 1: procedure MF-SGD(Dtrain, Iter, K, β, λ)  // W[|U|][K] và H[|I|][K] là 2 ma trận nhân tố tiềm ẩn cần tìm  2: W := N(0,σ2) // Khởi tạo ngẫu nhiên theo phân phối chuẩn  3: H := N(0,σ2) // Khởi tạo ngẫu nhiên theo phân phối chuẩn  4: for (iter:=1; iter <= Iter \* |Dtrain|; iter++)  5: Chọn ngẫu nhiên một dòng (u, i, rui) từ Dtrain  6: ui := 0  7: for (k:=1; k<=K; k++)  8: ui := ui + W[u][k] \* H[i][k]  9: end for  10: eui := rui – ui  11: for (k:=1; k<=K; k++)  12: W[u][k] := W[u][k] + β \* (eui \* H[i][k] - λ \* W[u][k])  13: H[i][k] := H[i][k] + β \* (eui \* W[u][k] - λ \* H[i][k])  14: end for  15: Break nếu đã convergence  16: end for  17: return {W, H}  18: end procedure |

Sau quá trình tối ưu (huấn luyện), ta có được hai tham số W và H, khi đó, xếp hạng của người dùng u cho mục thông tin i được dự đoán qua công thức bên dưới. Như vậy ta đã tìm được đánh giá của user u lên item i.

1. **Kết luận chương 3.**

Trong chương này em đã phân tích rõ về các bước thực hiện của hệ thống khuyến nghị và kĩ thuật nâng cao của nó. Tuy có rất nhiều kĩ thuật nâng cao nhưng em chỉ trình bày một kĩ thuật, trong chương sau em sẽ áp dụng thuật toán collaborative filtering vào trong thực tế bằng việc sử dụng thư viện mahout.

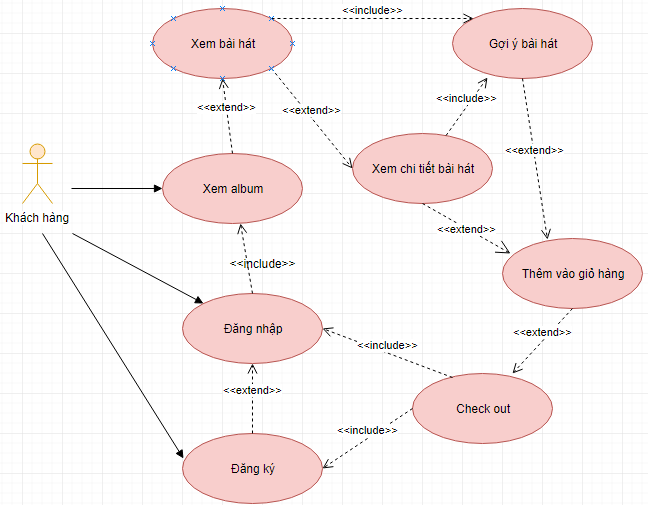
**CHƯƠNG 4. ỨNG DỤNG THUẬT TOÁN VÀO GỢI Ý BÀI NHẠC**

* + 1. **Sơ đồ use-case tổng quát.**

Để minh họa cho các thuật toán ở chương 3 trong chương này em sẽ xây dựng một trang web nhạc và áp dụng thuật toán đó vào trong gợi ý bài nhạc, để làm được điều này em đã sử dụng một số framework sau để xây dựng trang web:

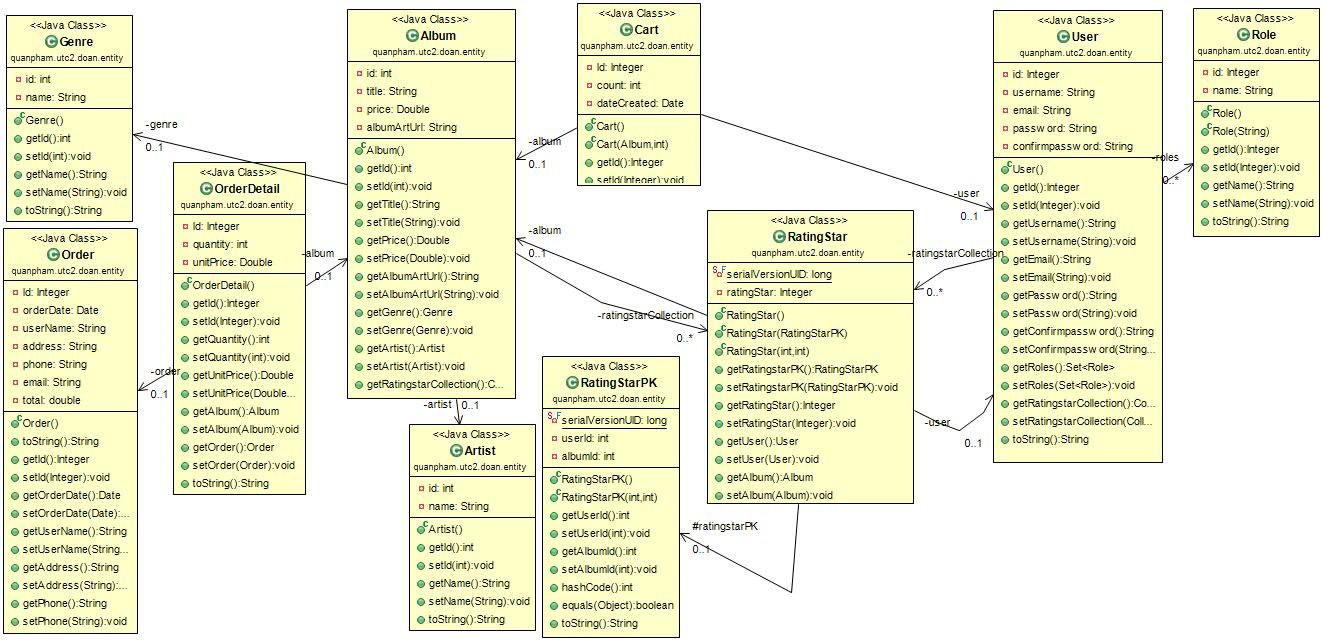
* Spring MVC.
* JPA
* Apache mahout

Về chi tiết các framework và thư viện này em đã trình bày ở chương 1 nên em sẽ không nêu lại ở đây. Dưới đây là sơ đồ use-case tổng quát của hệ thống.



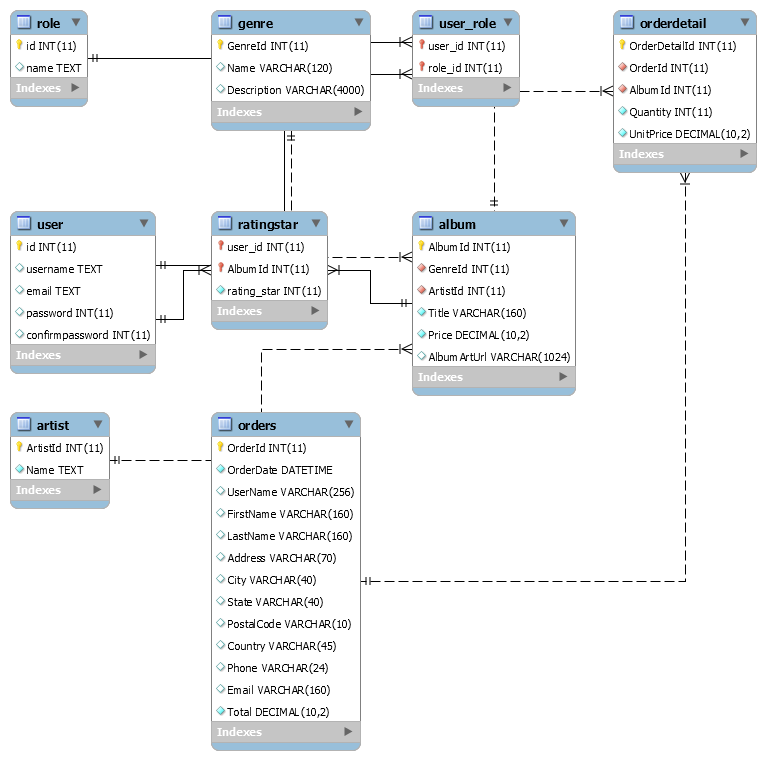
Hình 22:Sơ đồ use-case tổng quát.

## **Sơ đồ lớp UML.**

****

Hình 23: Sơ đồ lớp UML

## **Sơ đồ cơ sở dữ liệu.**

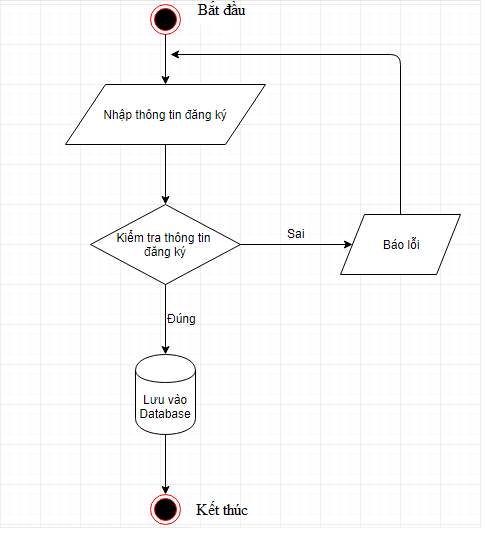


Hình 24: Sơ đồ cơ sở dữ liệu

**4. Lưu đồ mô tả chức năng.**

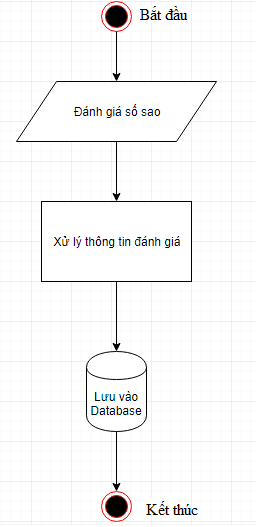
***4.1. Lưu đồ đăng ký.***

Đối với quá trình đăng ký thành viên ta gồm các bước sau:



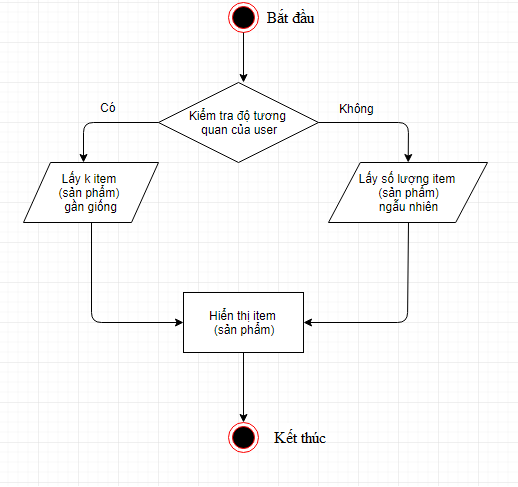
Hình 25: Lưu đồ đăng ký thành viên.

### ***4.2. Lưu đồ đánh giá sản phẩm.***

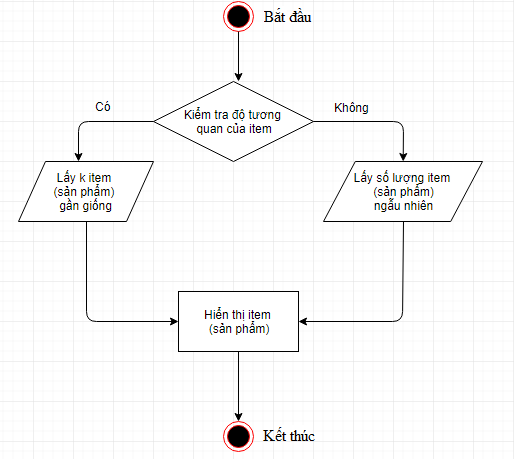


Hình 26: Lưu đồ đánh giá sản phẩm.

***4.3.*** ***Lưu đồ gợi ý sản phẩm cho user.***



Hình 27: Lưu đồ gợi ý sản phẩm cho user theo độ tương quan của user.

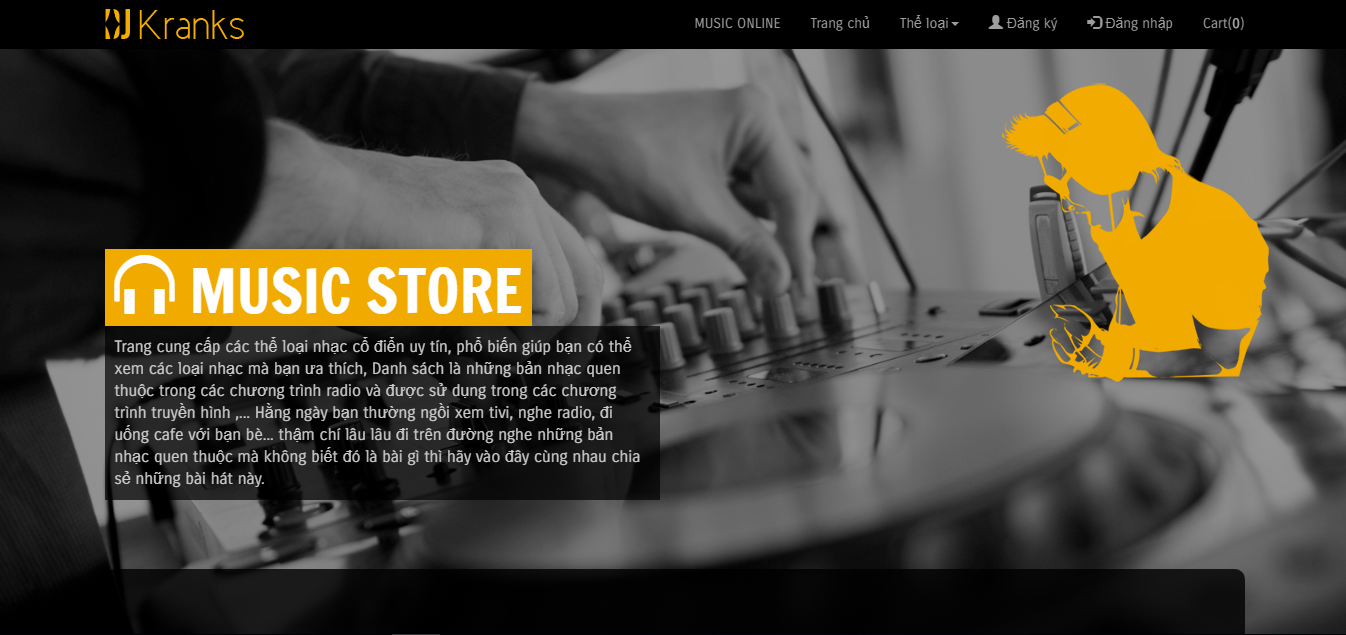


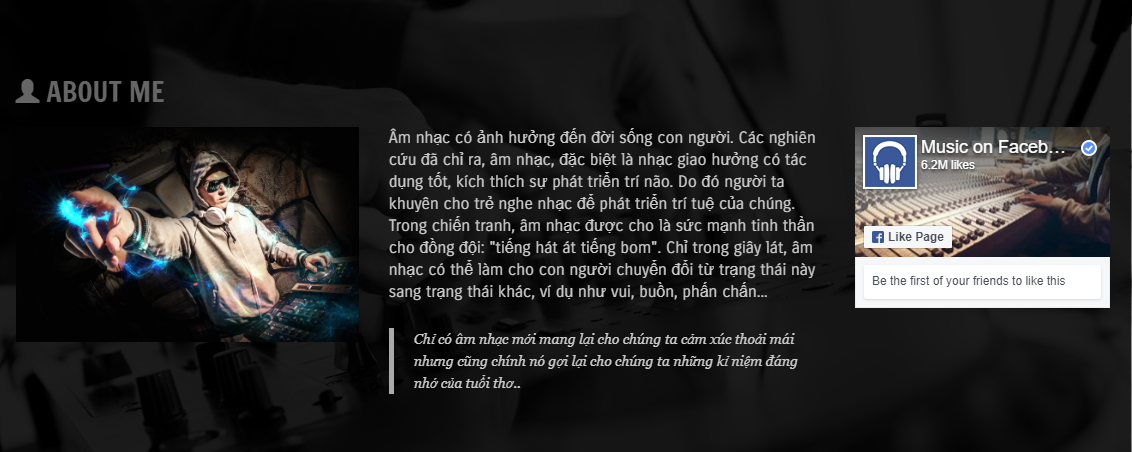
Hình 28: Lưu đồ gợi ý sản phẩm cho user theo độ tương quan của item.

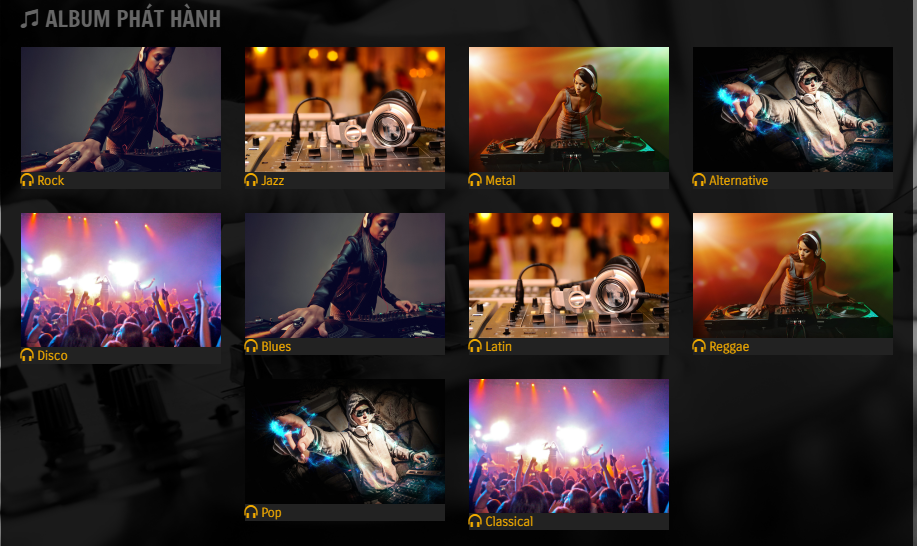
## **5. Giao diện trang web.**

***5.1. Giao diện trang home.***

Dưới đây là giao diện của trang chủ, khi người đùng chưa đăng nhập nó sẽ gợi ý bài hát ngẫu nhiên, nhưng khi người dùng đã đăng nhập nó sẽ tính sự tương quan của người dùng và đưa ra các gợi ý phù hợp.



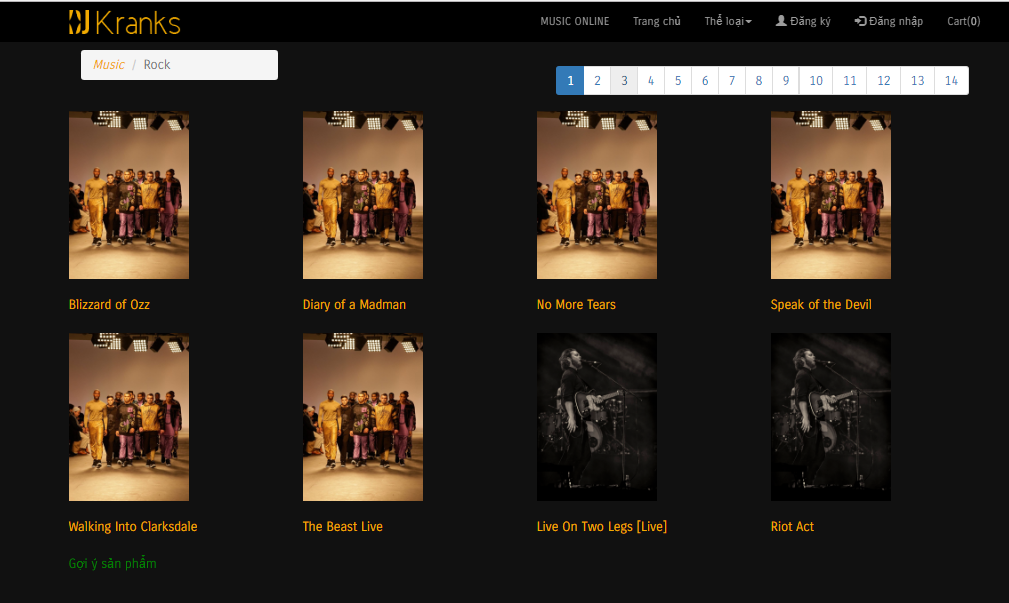




Hình 29: Giao diện trang home.

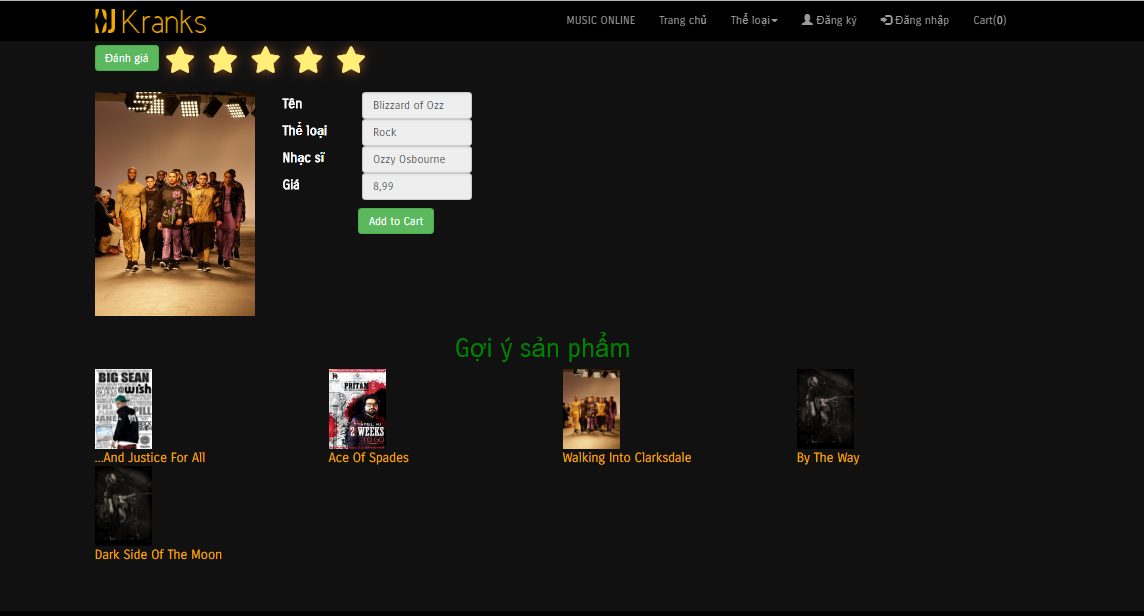
***5.2. Giao diện xem các bài hát theo thể loại nhạc.***

Người dùng có thể xem các thể loại nhạc mà họ muốn, và trong này nó sẽ gợi ý các bài liên quan cho người dùng nhưng trong trường hợp này gợi ý sản phẩm sẽ không có sản phẩm nào bởi vì người dùng chưa đăng nhập.



Hình 30: Giao diện các bài hát thuộc thể loại.

***5.3. Giao diện trang xem chi tiết sản phẩm.***

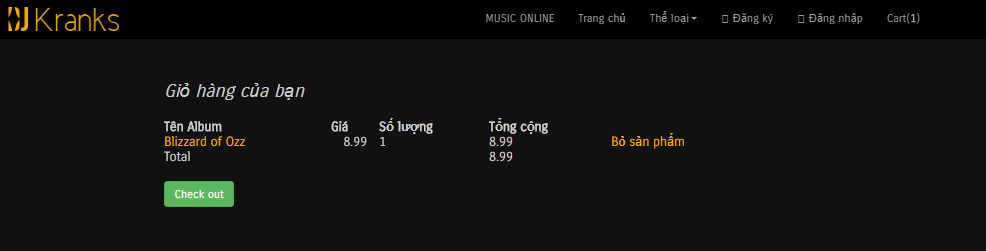


Hình 31: Giao diện trang chi tiết bài hát.

Gợi ý bài nhạc ở trên sẽ tính theo sự tương quan của item item và nó sẽ hiển thị các sản phẩm liên quan đến bài nhạc mà ta đang chọn.

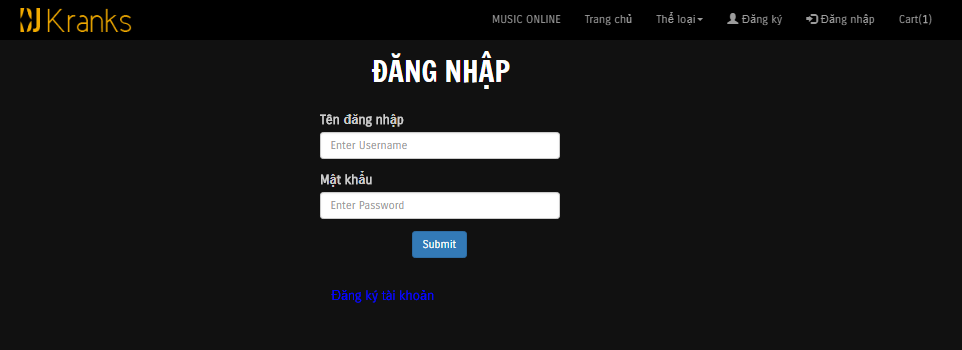
***5.4. Check-out sản phẩm.***

Người dùng có thể thêm sản phẩm vào giỏ hàng và có thể check-out nó.

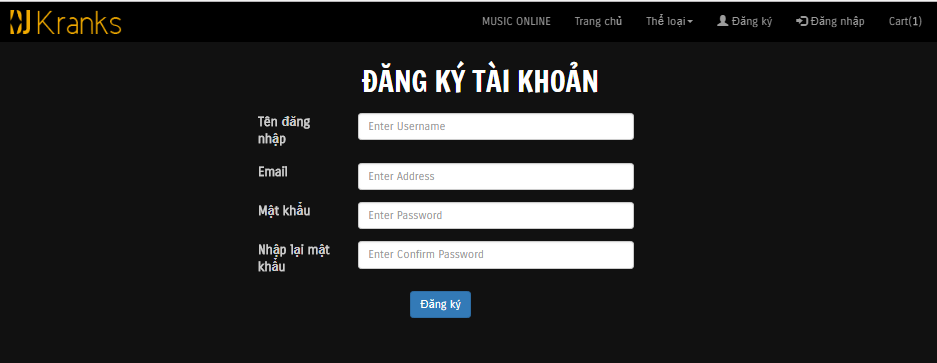


Hình 32: Giao diện trang check-out sản phẩm.

Nếu người dùng chọn Check-out nó sẽ kiểm tra xem người dùng đã đăng nhập chưa, nếu chưa đăng nhập yêu cầu chúng ta phải đăng nhập hoặc nếu chưa có tài khoản yêu cầu chúng ta phải đăng ký tài khoản.

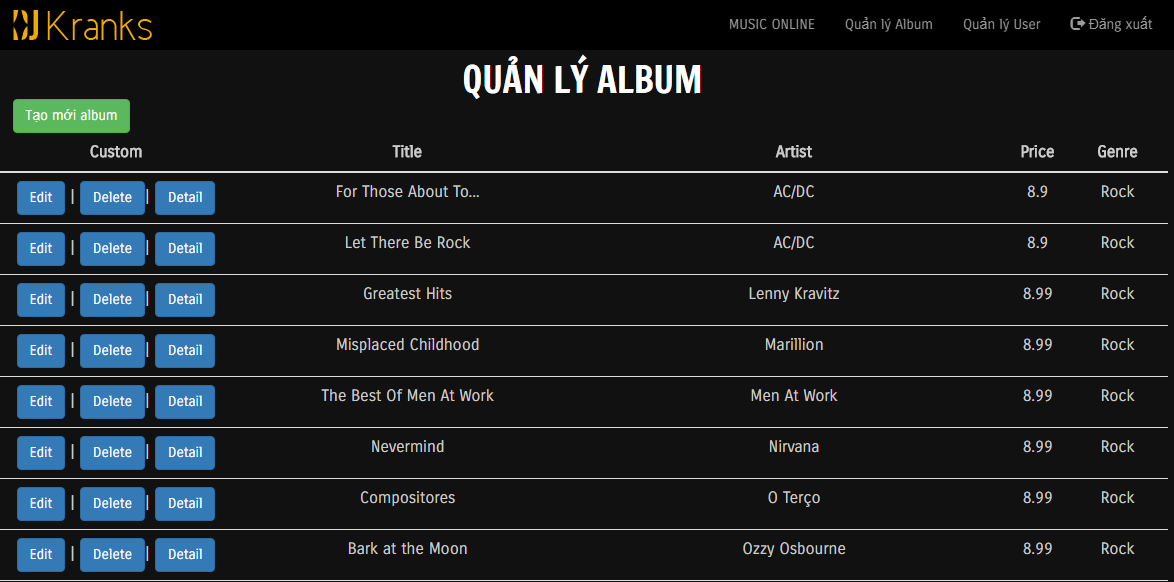


Hình 33: Giao diện trang đăng nhập.



Hình 34: Giao diện trang đăng ký.

***5.5. Giao diện quản lý music store.***

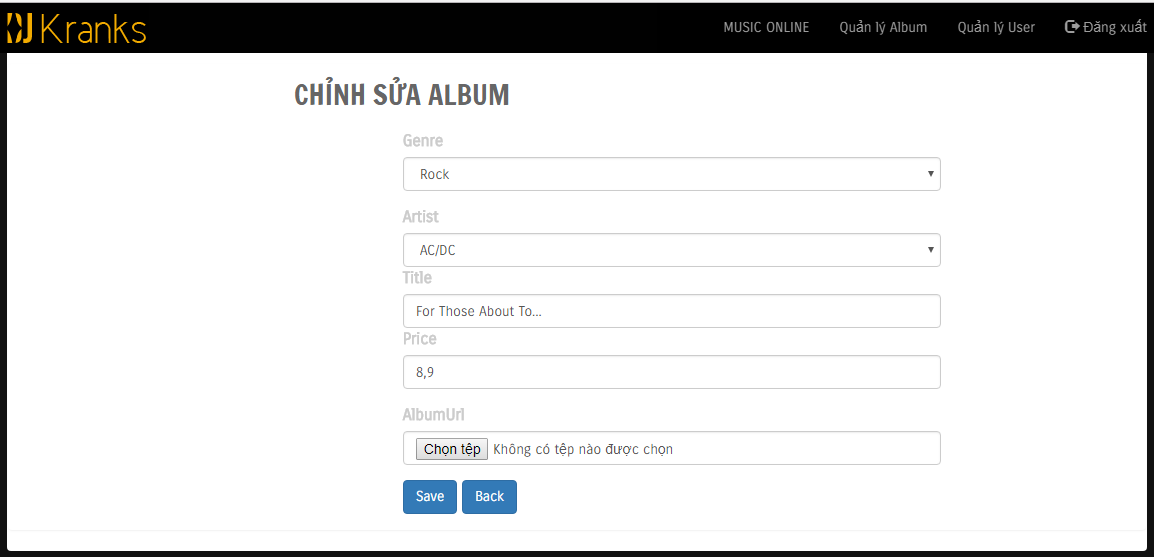
 Hình 35**: Quản lý Album.**

### ***5.6. Tạo mới Album.***



Hình 36: Giao diện trang tạo mới Album.

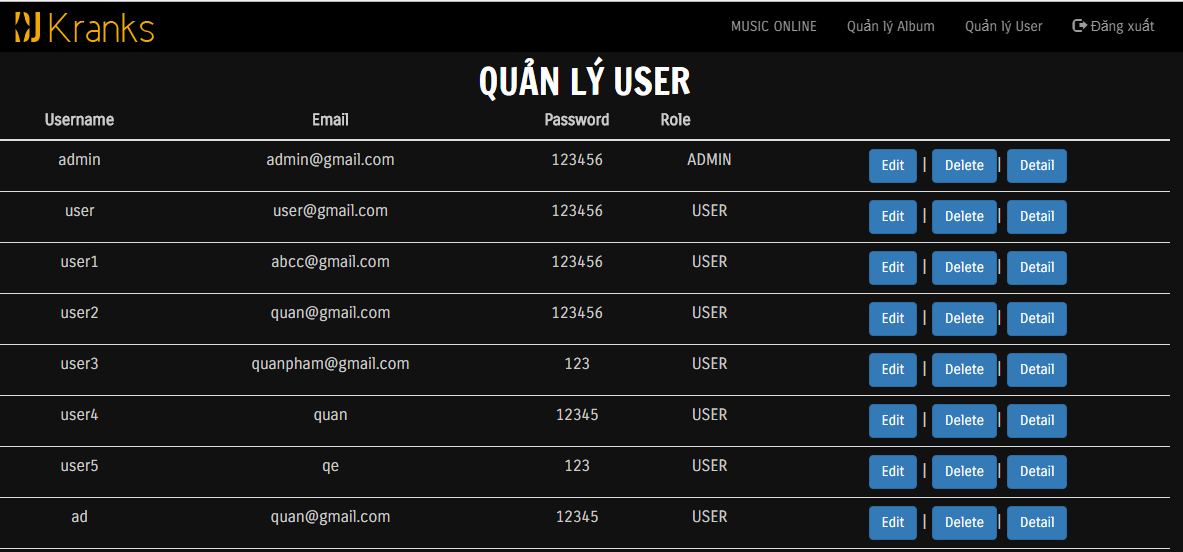
### ***5.7. Chỉnh sửa Album.***



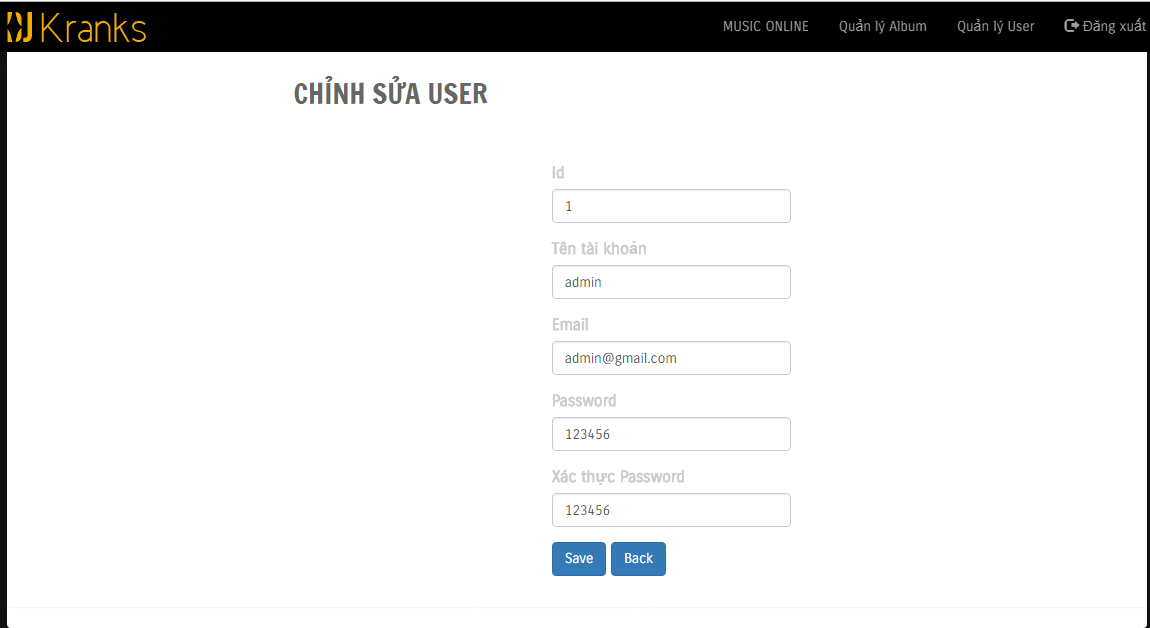
Hình 37: Giao diện trang chỉnh sửa Album.

***5.8. Quản lý User.***

Ngoài ra còn một số chức năng như quản lý về user



**Hình 38:** Giao diện trang quản lý user.



Hình 39: Giao diện trang chỉnh sửa User.

**KẾT LUẬN VÀ KIẾN NGHỊ**

* 1. **Kết quả đạt được****.**

Trong suốt khoảng thời gian làm đồ án, em đã cố gắng làm hết khả năng của em và vì thời gian làm đồ án có hạn nên kết quả đạt được không nhiều nhưng em cũng học được khá nhiều điều mới mẻ và những việc em đã đạt được như sau:

* Tìm hiểu được hệ thống khuyến nghị, và biết được các phương pháp gợi ý sản phẩm cho người dùng cụ thể là gợi ý theo người dùng hoặc sản phẩm.
* Nắm bắt được thuật toán collaborative filtering.
* Áp dụng được collaborative filtering vào trong gợi ý bài nhạc.
* Xây dựng được trang web gợi ý nhạc đơn giản.
* Tìm hiểu được một vài thông tin về thư viện mahout trong hệ thống khuyến nghị.
* Biết thêm được framework Spring và JPA.
  1. **Tồn tại.**

Trong quá trình làm đồ án hiển nhiên sẽ có những chức năng chưa hoàn thành được cùng với đó là thời gian có hạn nên sẽ có những chức năng em chưa hoàn thiện:

* Chưa áp dụng được các kĩ thuật nâng cao vào trong thực tế như kĩ thuật phân rã ma trận để giải quyết vấn đề ma trận thưa.
* Chưa kết hợp được nhiều phương pháp lại với nhau để cho ra gợi ý tốt hơn như kết hợp giữ content-based và Collaborative filtering.
* Chưa tận dụng được các thông tin người dùng để phân tích.
  1. **Hướng phát triển.**
* Phát triển trang web thành một web thương mại điện tử
* Áp dụng thêm được các kĩ thuật nâng cao.
* Khai thác được dữ liệu từ người dùng.

# TÀI LIỆU THAM KHẢO

[1] <http://hoclaptrinhweb.org/lap-trinh/hoc-java/224-bai-1-tong-quan-ve-java.html>

Thời gian truy cập: (8h - 12/5/2018)

[2] <https://kipalog.com/posts/Gioi-thieu-Spring-Framework>

Thời gian truy cập: (10h - 12/5/2018)

[3] <https://viblo.asia/p/apache-mahout-va-ung-dung-cua-no-qzaGzLbLkyO>

Thời gian truy cập: (14h30 - 13/5/2018)

[4] <https://machinelearningcoban.com/2016/12/26/introduce/>

Thời gian truy cập: (14h30 - 2/6/2018)

[5] <https://machinelearningcoban.com/2017/05/24/collaborativefiltering/>

Thời gian truy cập: (21h30 - 2/6/2018)

BÁO

[6] Hệ thống gợi ý sản phẩm trong bán hàng trực tuyến của Nguyễn Thái Nghệ và Nguyễn Hùng Dũng

[7]<https://www.researchgate.net/publication/310059523_He_thong_goi_y_Ky_thuat_va_ung_dung>.

Thời gian truy cập: (21h30 - 3/6/2018)

SÁCH

[8] Machine learning cơ bản của Vũ Hữu Tiệp

[9] Recommender System (The Textbook) của Charu C. Aggarwal

Link:<http://pzs.dstu.dp.ua/DataMining/recom/bibl/1aggarwal_c_c_recommender_systems_the_textbook.pdf>.

Thời gian truy cập: (10h30 - 3/6/2018)

[10] Mahout in Action 2012.