# C:\Users\BALAMURUGAN\Desktop\sowndharya_project\annauniversity.jpgC:\Users\BALAMURUGAN\Desktop\sowndharya_project\kct-logo.jpgHỆ THỐNG GỢI Ý TRÒ CHƠI TRỰC TUYẾN SỬ DỤNG KỸ THUẬT LỌC NỘI DUNG VÀ LỌC CỘNG TÁC.

**BÁO CÁO DỰ ÁN**

***Nộp bởi***

# AKASH S (17BCS206) MOHANAGANPATHY V (17BCS217) SAMPATH KUMAR M (17BCS219)

***Để hoàn thành một phần yêu cầu cấp bằng.***

*của*

# CỬ NHÂN KỸ THUẬT

trong

# Khoa học máy tính và Kỹ thuật máy tính

**Đại học KUMARAGURU CÔNG NGHỆ COIMBATORE - 641049**

Một cơ sở tự trị trực thuộc Đại học Anna, Chennai

**Tháng 5 năm 2021**

# C:\Users\BALAMURUGAN\Desktop\sowndharya_project\kct.pngC:\Users\BALAMURUGAN\Desktop\sowndharya_project\annauniversity.jpgTRƯỜNG CAO ĐẲNG CÔNG NGHỆ KUMARAGURU, THÀNH PHỐ COIMBATORE, MÃ BƯU ĐIỆN 641 049

Một cơ sở tự trị trực thuộc Đại học Anna, Chennai

# CHỨNG NHẬN XÁC THỰC

Chứng nhậnđónàybáo cáo dự án "VIDEO TRỰC TUYẾN"  **ĐỀ XUẤT TRÒ CHƠIHỆ THỐNGSỬ DỤNGNỘI DUNGVÀ**

**KỸ THUẬT LỌC CỘNG TÁC là công trình chính thức của AKASH.S(17BCS206)MohanaganapathyV(17BCS217)và**

**SAMPATH KUMAR M (17BCS219) đã thực hiện công việc dự án dưới sự giám sát của tôi.**

A picture containing text, tool  Description automatically generated

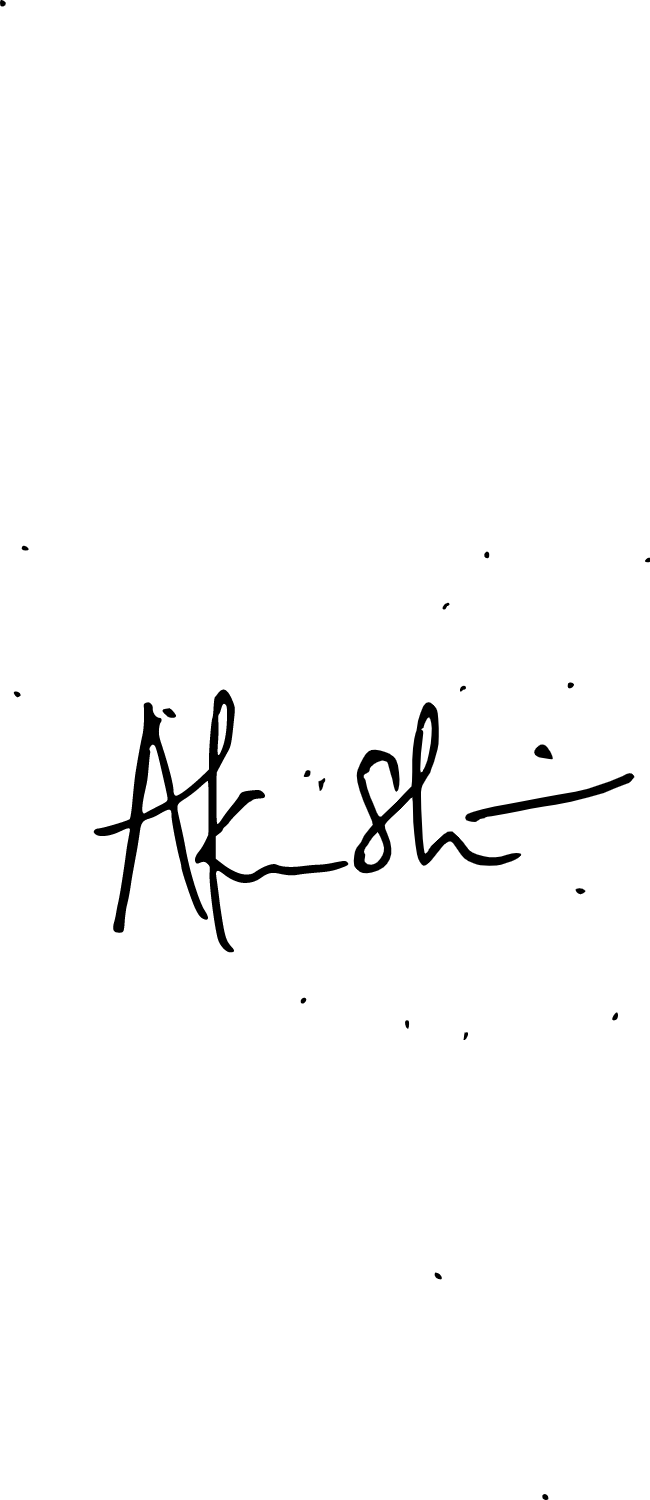
|  |  |
| --- | --- |
| **CHỮ KÝ** | **CHỮ KÝ** |
| **Tiến sĩ Devaki. P, Tiến sĩ.** | **Bharathi Priya C, Giảng viên Trợ lý** |
| **TRƯỞNG KHOA** | **Người giám sát** |
| Khoa Công nghệ Thông tin và Kỹ thuật máy tính | Bộ phậncủaMáy tínhKhoa họcvà Kỹ thuật, |
| Trường Cao đẳng Kỹ thuật Kumaraguru | Trường Cao đẳng Kỹ thuật Kumaraguru |
| Coimbatore – 641049 | Coimbatore – 641049 |

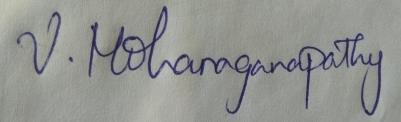
Các thí sinh có Số Đăng Ký Đại Học 17BCS206, 17BCS217, 17BCS219 đã được kiểm tra trong kỳ thi Thuyết Trình Dự Án diễn ra vào …………………

Giám khảo nội bộGiám khảo bên ngoài

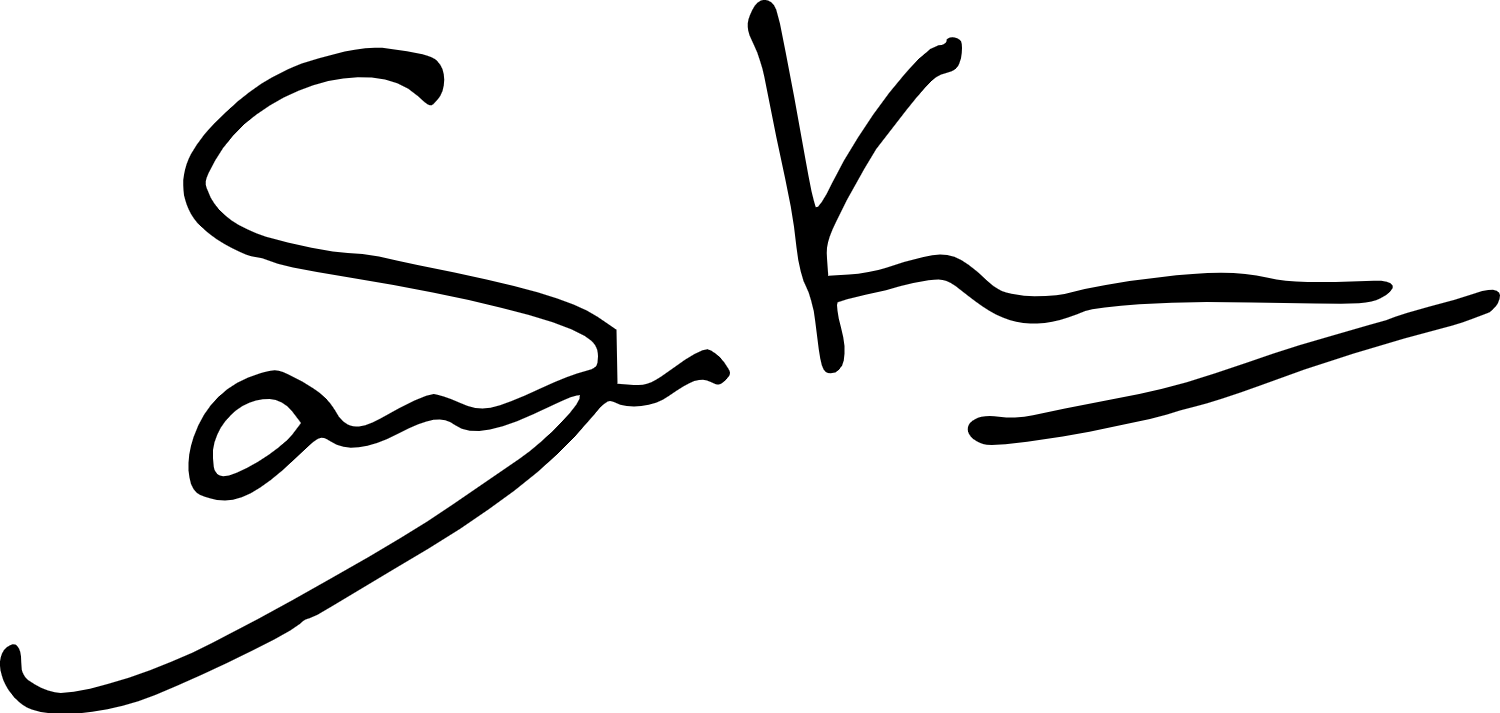
# Tuyên bố

Chúng tôi khẳng định rằng dự án có tiêu đề “HỆ THỐNG GỢI Ý TRÒ CHƠI TRỰC TUYẾN SỬ DỤNG NỘI DUNG VÀ”.

**KỸ THUẬT LỌC CỘNG TÁC được nộp để hoàn thành một phần yêu cầu cấp bằng Kỹ sư Công nghệ Thông tin và Kỹ thuật Máy tính là công trình nghiên cứu độc lập của chúng tôi. Nó chưa từng là một phần của bất kỳ dự án nào khác được nộp để xin cấp bằng hoặc chứng chỉ, cả ở trường này hay bất kỳ trường đại học nào khác.**



# AKASH S (17BCS206) MOHANAGANAPATHY V (17BCS217)

**SAMPATH KUMAR M (17BCS219)**

Tôi xác nhận rằng tuyên bố được nêu trên bởi các ứng viên là đúng sự thật.

A picture containing text, tool  Description automatically generated

## Bharathi Priya C

Giảng viên chính

Khoa Công nghệ Thông tin và Kỹ thuật, Trường Cao đẳng Kỹ thuật Kumaraguru

Coimbatore – 641049.

# LỜI CẢM ƠN

Chúng tôi xin bày tỏ lòng biết ơn sâu sắc đến ban lãnh đạo Trường Cao đẳng Kỹ thuật Kumaraguru vì đã cung cấp cho chúng tôi cơ sở hạ tầng cần thiết, giúp chúng tôi hoàn thành dự án một cách thành công.

Chúng tôi xin cảm ơn Hiệp hội Động kinh Hoa Kỳ đã cho phép chúng tôi sử dụng cơ sở dữ liệu của họ cho dự án của chúng tôi.

Chúng tôi xin gửi lời cảm ơn đến Hiệu trưởng, Tiến sĩ D Saravanan, vì đã cung cấp cho chúng tôi các cơ sở vật chất cần thiết để thực hiện dự án.

Chúng tôi xin ghi nhận Tiến sĩ P Devaki, Giáo sư và Trưởng Khoa Khoa học Máy tính và Kỹ thuật, vì sự hỗ trợ và khuyến khích của bà trong suốt dự án này.

Chúng tôi xin cảm ơn điều phối viên dự án của chúng tôi, Tiến sĩ L Latha, Giáo sư, Khoa Khoa học Máy tính và Kỹ thuật, và hướng dẫn viên Ms. Bharathi Priya C, Giảng viên, Khoa Khoa học Máy tính và Kỹ thuật, vì những nỗ lực liên tục, sự hướng dẫn và thời gian quý báu của họ.

Chúng tôi xin chân thành và nhiệt liệt cảm ơn các thành viên của Khoa Khoa học Máy tính và Kỹ thuật thuộc Trường Cao đẳng Công nghệ Kumaraguru vì những lời chúc tốt đẹp, sự giúp đỡ kịp thời và hỗ trợ mà họ đã dành cho chúng tôi trong suốt quá trình thực hiện dự án. Chúng tôi rất biết ơn gia đình và bạn bè đã ủng hộ chúng tôi.

AKASH S MOHANAGANAPATHY V SAMPATH KUMAR M

# MỤC LỤC

|  |  |  |  |
| --- | --- | --- | --- |
| **CHƯƠNG SỐ** |  | **TIÊU ĐỀ** | **Số trang** |
|  | **TÓM TẮT** | | 7 |
| 1 | **GIỚI THIỆU** | | 8 |
|  | 1.1 | NGHIÊN CỨU KHÁI NIỆM | 8 |
|  | 1.2 | MỤC TIÊU | 9 |
|  | 1.3 | PHẠM VI CỦA DỰ ÁN | 9 |
| 2 | **TỔNG QUAN TÀI LIỆU** | | 10 |
| 3 | **ĐỊNH NGHĨA VẤN ĐỀ** | | 14 |
| 4 | **HỆ THỐNG ĐƯỢC ĐỀ XUẤT** | | 14 |
|  | 4.1 | TẬP DỮ LIỆU | 15 |
|  | 4.2 | PHƯƠNG PHÁP NGHIÊN CỨU | 15 |
|  | 4.3 | SƠ ĐỒ LUỒNG | 16 |
|  | 4.4 | TRIỂN KHAI | 17 |
|  |  | 4.4.1 THU THẬP DỮ LIỆU | 17 |
|  |  | 4.4.2 TIỀN XỬ LÝ DỮ LIỆU | 17 |
|  |  | 4.4.3 TRỰC QUAN HÓA DỮ LIỆU | 19 |
|  |  | 4.4.4 TẬP DỮ LIỆU ĐÀO TẠO & KIỂM TRA | 22 |
|  |  | 4.4.5 ĐÁNH GIÁ & PHÂN TÍCH | 28 |
| 5 | **CÁC YÊU CẦU HỆ THỐNG** | | 29 |
| 6 | **KẾT QUẢ** | | 29 |
| 7 | **KẾT LUẬN** | | 30 |

|  |  |  |
| --- | --- | --- |
| 8 | **PHỤ LỤC** | 31 |
| 9 | **TÀI LIỆU THAM KHẢO** | 36 |

**TÓM TẮT**

Khu vui chơi ảo là một trong những lĩnh vực phát triển nhanh nhất trong những năm gần đây. Trong trường hợp trò chơi điện tử, việc sản xuất nhiều tựa game nổi tiếng nhất tương đương với sản xuất phim. Doanh thu từ trò chơi điện tử đạt hàng triệu, và tuy nhiên, có rất ít công trình nghiên cứu về việc tư vấn trò chơi điện tử. Trò chơi điện tử và game thủ tạo ra một lượng lớn dữ liệu, vì mọi hành động của họ trong một trò chơi đều được ghi lại. Sự đa dạng hóa này đã làm tăng đáng kể số lượng người dùng tham gia vào các cộng đồng trực tuyến trong lĩnh vực giải trí này, và do đó, số lượng và loại trò chơi có sẵn cũng tăng lên. Bối cảnh quá tải thông tin này hỗ trợ sự phát triển của các hệ thống gợi ý có thể tận dụng dữ liệu mà các nền tảng trò chơi điện tử thu thập, từ đó theo kịp xu hướng các trò chơi mới ra mắt hàng năm. Trong bài báo này, một phương pháp mới để gợi ý trò chơi điện tử được trình bày, thông qua việc sử dụng lọc hợp tác. Để cải thiện các gợi ý, một phương pháp mới để ước lượng xếp hạng ngầm được đề xuất, xem xét thời gian chơi và đưa ra gợi ý tốt hơn. Hệ thống gợi ý được đề xuất cải thiện kết quả của các phương pháp khác đã được trình bày trong lĩnh vực này.

# GIỚI THIỆU

* 1. **NGHIÊN CỨU KHÁI NIỆM**

Mục tiêu là cung cấp gợi ý cho một nhóm người dùng về các sản phẩm có thể thú vị đối với họ. Thông tin có sẵn trên web đang tăng lên đều đặn và nội dung được cung cấp bởi các trang web lớn như Amazon, Netflix, YouTube và Steam cũng vậy. Nội dung có thể là bất cứ thứ gì, chẳng hạn như sách, phim, trò chơi và tài liệu khoa học, và đôi khi số lượng nội dung này có thể gây choáng ngợp. Việc tìm kiếm kết quả tốt không chỉ khó khăn do khối lượng thông tin khổng lồ này, mà còn do sự thiếu hiểu biết của người dùng hoặc đơn giản là sự hiếm có của nội dung. Việc lọc thông tin dường như là một nhu cầu hợp lý cho người dùng của những nền tảng này nhằm truy xuất nội dung cá nhân hóa và đây là nơi các hệ thống gợi ý được sử dụng để hỗ trợ. Các hệ thống gợi ý được sử dụng để tự động đáp ứng mong muốn nhận được chỉ nội dung phù hợp và hiện nay rất phổ biến. Thực tế, tất cả các nền tảng đã đề cập ở trên đều triển khai một loại hệ thống gợi ý nào đó nhằm phục vụ nhu cầu của người dùng bằng cách cung cấp nội dung cá nhân hóa với mục tiêu cải thiện trải nghiệm người dùng, từ đó nâng cao lòng trung thành của khách hàng, bán được nhiều và đa dạng sản phẩm hơn, và cuối cùng là cải thiện doanh thu của các công ty. Chơi trò chơi điện tử là một hoạt động thường xuyên, theo nghĩa là nó tương tự hơn với việc nghe nhạc hơn là xem phim. Một trò chơi ưa thích được chơi nhiều lần, nhưng người dùng cũng muốn khám phá các trò chơi mới. Điều này đại diện cho một thách thức kép cho ngành công nghiệp: nhu cầu về các trò chơi điện tử khuyến khích người dùng quay lại, cũng như giúp người dùng tìm ra các trò chơi mới mà họ sẽ tiêu thụ nhiều như những trò chơi đã thích. Mối quan hệ giữa các trò chơi điện tử tương tự khiến cho việc lập luận rằng có thể đối mặt với thách thức thứ hai là hợp lý. Việc phát triển một thuật toán giải quyết các thách thức đã đề cập có thể mang lại lợi ích lớn cho ngành công nghiệp, cộng đồng người dùng và thậm chí cho các nhà phát triển trò chơi, bằng cách dự đoán những gì người dùng muốn nhất cũng như quảng bá các sản phẩm mới.

# MỤC TIÊU

Mục đích của hệ thống gợi ý này là cung cấp hướng dẫn cho một nhóm khách hàng về các sản phẩm có thể thú vị đối với họ. Các hệ thống gợi ý thường được quan sát trong thương mại điện tử (nơi mà khách hàng mua các sản phẩm như sách, quần áo hoặc trò chơi điện tử trực tuyến), thường được áp dụng thông qua các phương pháp lọc hợp tác. Hệ thống này bằng cách so sánh các sản phẩm tương tự hoặc khách hàng tương tự dựa trên đánh giá của người dùng. Nếu khách hàng thích các sản phẩm giống nhau, họ có khả năng tương tự, và nếu các sản phẩm được đánh giá cao bởi cùng một khách hàng, những sản phẩm đó cũng có khả năng tương tự.

# PHẠM VI CỦA DỰ ÁN

Phạm vi của dự án này là tạo ra một hệ thống gợi ý có thể phân tích, dự đoán và gợi ý trò chơi điện tử bằng cách sử dụng lọc hợp tác để giúp các cá nhân như game thủ xác định các trò chơi tương tự với sở thích và mối quan tâm của họ. Mục tiêu chính khác là đạt được độ chính xác tốt nhất khi so sánh các thuật toán khác nhau.

# TỔNG QUAN TÀI LIỆU

1. Cheuque, Germán, José Guzmán, và Denis Parra. “Hệ thống gợi ý cho các nền tảng trò chơi điện tử trực tuyến: Trường hợp của STEAM.” Tài liệu Hội nghị Thế giới về Web 2019. 2019.

Trong mô hình được đề xuất bởi Cheuque, Germán, José Guzmán, và Denis Parra, các tác giả đã đề xuất thử nghiệm các mô hình gợi ý lần lượt trên các máy phân tích (FM), mạng nơ-ron sâu (DeepNN) và một mô hình được phát triển từ cả hai (DeepFM). Các tập dữ liệu được thu thập từ lịch sử mua hàng của người dùng trên nền tảng Steam (nền tảng trò chơi điện tử). Các mô hình hệ thống gợi ý như phương pháp bình phương tối thiểu luân phiên (ALS), mô hình phân tích ma trận, máy phân tích (FM), DeepFM, DeepNN đã được thực hiện, và sau đó phân tích cảm xúc cũng được thực hiện. Kết quả thí nghiệm cho thấy DeepNN là thuật toán hoạt động tốt nhất.

1. Stefan Heinz, Yvonne Lau và Daniel Epstein, “Metarecommendr: Hệ thống gợi ý cho trò chơi điện tử, phim và chương trình truyền hình”, tháng 4 năm 2017.

Trong mô hình được đề xuất bởi Stefan Heinz, Yvonne Lau và Daniel Epstein, mục tiêu chính của tạp chí này là cung cấp những gợi ý tốt nhất để phù hợp với sở thích của người dùng. Tạp chí giải quyết các vấn đề bằng cách tạo ra một ứng dụng nhằm cải thiện việc tìm kiếm gợi ý tiếp theo của người dùng. Cách chính để đạt được mục tiêu là thông qua việc thực hiện một số quy trình như Thu thập Dữ liệu, Tiền xử lý Dữ liệu, Phân tích Dữ liệu Khám Phá và Kiến trúc Mô hình như lọc dựa trên nội dung (các mô hình được huấn luyện riêng biệt trên các đánh giá của nhà phê bình và tóm tắt), lọc hợp tác (Phân rã Giá trị Đặc trưng - SVD) và các thuật toán như hệ số tương quan Pearson và phân tích cảm xúc để hoàn thiện. Nhiều mô hình học máy đã được thử nghiệm bao gồm Hồi quy Logistic, Naive Bayes, SVM và các loại mạng nơ-ron khác nhau. Cuối cùng, kết quả tốt nhất đạt được với Mạng Nơ-ron Tích chập (CNN) với độ chính xác 88 phần trăm trong kết quả.

1. He, Xiangnan, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, và Tat-Seng Chua. Neural collaborative filtering. Trong Kỷ yếu của hội nghị quốc tế lần thứ 26 về web toàn cầu, trang 173-182. 2017.

Các tác giả của tạp chí này đã nỗ lực phát triển các kỹ thuật dựa trên mạng nơ-ron để giải quyết các vấn đề trong hệ thống gợi ý. Để giải quyết các vấn đề, một thuật toán mới được trình bày được gọi là lọc hợp tác dựa trên mạng nơ-ron (NCF). Các thí nghiệm đã được thực hiện trên Phân rã Ma trận, Mạng Nơ-ron Đa lớp, Phân rã Ma trận Tổng quát và sự kết hợp của GMF và MLP. Kết quả cho thấy sau các thí nghiệm rộng rãi, NCF cho thấy sự cải thiện đáng kể so với các phương pháp khác.

1. Pérez-Marcos, J., Martin-Gomez, L., Jiménez-Bravo, D. M., López, V. F., & Moreno-García, M. N. (2020). Hệ thống lai cho gợi ý trò chơi điện tử dựa trên đánh giá ngầm và mạng xã hội. Tạp chí Trí tuệ Nhân tạo và Tính toán Con người, 11(11), 4525-4535.

Các tác giả của tạp chí này đã đề xuất một hệ thống gợi ý lai kết hợp cả lọc dựa trên nội dung và lọc hợp tác. Đây là một phương pháp mới để ước lượng các đánh giá ngầm bằng cách xem xét thời gian chơi (giờ chơi của người dùng). Bộ dữ liệu được sử dụng từ nền tảng Kaggle được biết đến với dữ liệu trò chơi điện tử Steam để thực hiện các đánh giá. Sai số trung bình tuyệt đối chuẩn hóa (NMAE) và Sai số căn bậc hai trung bình chuẩn hóa (NRMSE) đã được thực hiện với sự trợ giúp của thuật toán KNN dựa trên mục và thuật toán SVD++. Kết quả cho thấy thuật toán KNN dựa trên mục hoạt động tốt nhất cho phương pháp NMAE và thuật toán SVD++ cho kết quả tốt nhất cho phương pháp NRMSE.

1. Bertens, P., Guitart, A., Chen, P. P., & Perianez, A. (2018, tháng 8). Hệ thống gợi ý mục dựa trên máy học cho trò chơi điện tử. Trong Hội nghị IEEE về Trí tuệ Tính toán và Trò chơi (CIG) 2018 (trang 1-4). IEEE.

Các tác giả của tạp chí này đã đề xuất một cách để cải thiện hệ thống gợi ý bằng cách so sánh hai thuật toán quan trọng. Một mô hình dựa trên Tập hợp (cây cực kỳ ngẫu nhiên) và một thuật toán mạng nơ-ron sâu được so sánh và đánh giá để cải thiện hệ thống gợi ý. Bộ dữ liệu cho việc đánh giá này được sử dụng từ Age of Ishtaria, được phát triển bởi Silicon Studio. Việc đánh giá mô hình được thực hiện,

và kết quả cho thấy cây cực kỳ ngẫu nhiên cho độ chính xác tốt hơn trong việc dự đoán so với mạng nơ-ron sâu.

1. Hệ thống Gợi ý trong Lĩnh vực Trò chơi Điện tử: So sánh Nhiều Thuật toán, Thang Đánh giá và Phản hồi Ngầm/ Rõ ràng Sử dụng Nền tảng Steam, Graz, tháng 9 năm 2016.

Tác giả của tạp chí này đã đề xuất một phương pháp bằng cách so sánh hiệu suất của một số thuật toán đã chọn. Phản hồi ngầm đã được chuyển đổi thành đánh giá rõ ràng và đã được so sánh với các đánh giá gốc do người dùng cung cấp từ nền tảng Steam. Các quy trình Khám Phá Tri thức như Lựa chọn Dữ liệu, Tiền xử lý Dữ liệu, Chuyển đổi Dữ liệu, Khai thác Dữ liệu, Diễn giải và Đánh giá cũng đã được thực hiện. Các thuật toán như Cơ sở, Lọc dựa trên nội dung, Lọc hợp tác và Lai đã được thử nghiệm trên năm bộ dữ liệu khác nhau. Cuối cùng, Sai số căn bậc hai trung bình chuẩn hóa (NRMSE) được tính toán và trình bày cho mỗi thuật toán.

1. Yang, L., Hong, T., & Gopalakrishnan, A. Một Khung cho Hệ thống Gợi ý Dựa trên Lý thuyết Trò chơi trong Mạng Xã hội. Trong Hội nghị Quốc tế lần thứ 10 về Kiến thức và Công nghệ Thông minh (KST) 2018 (trang 95-100). IEEE.

Các tác giả của tạp chí này đã đề xuất một hệ thống gợi ý dựa trên lý thuyết trò chơi sử dụng đánh giá người dùng-mục để gợi ý. Lý thuyết Trò chơi và Simrank (Tương tự dựa trên đi bộ ngẫu nhiên) được sử dụng làm các thuật toán cốt lõi để xây dựng hệ thống gợi ý. Các đánh giá người dùng-mục được phân tách thành các nhóm tương tự và các độ tương tự được tính toán bằng thuật toán Simrank. Việc đánh giá được thực hiện bằng cách sử dụng MAE (Sai số Trung bình Tuyệt đối) và RMSE (Sai số Căn bậc hai). Cuối cùng, kết quả thí nghiệm cho thấy mô hình được trình bày trong tạp chí này cung cấp kết quả chính xác dưới một số điều kiện khi so sánh với các phương pháp truyền thống.

1. Anwar, S. M., Shahzad, T., Sattar, Z., Khan, R., & Majid, M. (2017, tháng 1). Hệ thống gợi ý trò chơi sử dụng lọc hợp tác (GAMBIT). Trong Hội nghị Quốc tế lần thứ 14 về Khoa học và Công nghệ Ứng dụng Bhurban (IBCAST) (trang 328-332). IEEE.

Các tác giả của tạp chí này đã đề xuất một phương pháp sử dụng kỹ thuật lọc hợp tác dựa trên người dùng để gợi ý trò chơi cho người dùng dự kiến. Hệ thống sử dụng

các đánh giá cá nhân do người dùng cung cấp cùng với các đánh giá của trò chơi. Để cung cấp cả gợi ý chung và gợi ý cá nhân hóa, lọc dựa trên mục được thực hiện bằng cách sử dụng công thức tương quan Pearson và lọc dựa trên người dùng được thực hiện để tìm sự tương đồng giữa người dùng hoạt động và các game thủ khác, cũng được tính toán bằng công thức tương quan Pearson. Kết quả cũng được so sánh bằng cách sử dụng sai số trung bình tuyệt đối (MAE). Kết quả cho thấy hệ thống được đề xuất hoạt động hiệu quả.

1. Ahirwadkar, Bhakti & Deshmukh, Sachin. (2020). Deepautoencf: Một Autoencoder Denoising cho Hệ thống Gợi ý. Tạp chí Khoa học Máy tính và Kỹ thuật Ấn Độ. 11. 244-250. 10.21817/indjcse/2020/v11i3/201103199.

Các tác giả của tạp chí này đã cung cấp một nghiên cứu về các phương pháp mạng nơ-ron sâu (DNN) có thể được sử dụng cho hệ thống gợi ý. Bài báo cũng cung cấp hiệu suất của Bộ mã hóa khử nhiễu (DAE). Các phương pháp học sâu như Mạng nơ-ron tích chập (CNN), Mạng nơ-ron đa lớp (MLP), Bộ mã hóa (AE) và Mạng nơ-ron hồi tiếp (RNN) đã được thực hiện. Việc đánh giá được thực hiện trên tập dữ liệu MovieLens – 1M với và không có các đặc trưng bổ sung của người dùng và mục được cung cấp trong tập dữ liệu. Cuối cùng của quá trình đánh giá, bộ mã hóa khử nhiễu có thể hoạt động tốt hơn khi thêm nhiễu có thể giúp thuật toán dự đoán các đánh giá chưa biết.

1. Kim, JaeWon, JeongA Wi, SooJin Jang, và YoungBin Kim. “Đề xuất tuần tự trên các nền tảng trò chơi board-game.” Symmetry 12, số 2 (2020): 210.

Các tác giả của tạp chí này đã đề xuất một phương pháp xây dựng hệ thống gợi ý dựa trên học sâu (DNN) sử dụng dữ liệu người dùng quy mô lớn từ một cộng đồng trực tuyến liên quan đến trò chơi bàn. Bộ dữ liệu BGG được đề xuất có những đặc điểm đối xứng với các bộ dữ liệu khác và cho thấy khả năng áp dụng cho nhiều hệ thống gợi ý khác nhau. Các mô hình CNN và RNN như đơn vị hồi tiếp có cổng (GRU) và SASRec đã được thực hiện. Phân tích hiệu suất và các chỉ số đánh giá cũng đã được tiến hành. Kết quả cho thấy rằng bất kể kích thước của các tương tác khởi đầu lạnh, phương pháp dựa trên CNN vượt trội hơn các hệ thống gợi ý khác và nắm bắt được sở thích cá nhân hóa trong nền tảng trò chơi bàn.

# ĐỊNH NGHĨA VẤN ĐỀ

Trò chơi điện tử là một trong những lĩnh vực giải trí phát triển nhanh nhất, không chỉ vì trò chơi điện tử tạo ra một lượng doanh thu và dữ liệu khổng lồ. Doanh số bán trò chơi điện tử đạt hàng triệu, tuy nhiên, có rất ít nghiên cứu về hệ thống gợi ý trò chơi điện tử. Trong quá khứ, người dùng thường nhận được gợi ý cho các trò chơi từ nhiều nguồn và trang web khác nhau. Mặc dù phương pháp này đơn giản và thân thiện với người dùng, nhưng nó vẫn có một số hạn chế. Việc xây dựng một hệ thống gợi ý để đề xuất trò chơi điện tử cho người dùng sẽ mang lại những lợi ích lớn cho ngành công nghiệp game. Hệ thống gợi ý này sẽ đề xuất các trò chơi điện tử cho người dùng dựa trên sở thích và thói quen trước đó của họ, từ đó lấp đầy một khoảng trống trong lĩnh vực này.

# HỆ THỐNG ĐƯỢC ĐỀ XUẤT

Bài viết này trình bày một hệ thống gợi ý để gợi ý trò chơi điện tử. Đề xuất là xây dựng một hệ thống gợi ý, sử dụng phương pháp lọc cộng tác. Mục đích của phương pháp trong hệ thống, lọc cộng tác, là cung cấp các gợi ý từ các mục đã được lọc. Để thực hiện lọc cộng tác, chúng tôi thu thập đánh giá của từng người dùng, trong trường hợp trò chơi điện tử, có thể thu thập thông qua số giờ chơi. Công trình này so sánh với một số thuật toán khác để đạt được độ chính xác tốt nhất và kiểm tra thuật toán nào hoạt động hiệu quả hơn.

# DỮ LIỆU TẬP HỢP

Dữ liệu được sử dụng đã được công khai trên nền tảng Kaggle. Dữ liệu huấn luyện được chia theo tỷ lệ 70:30. 30% dữ liệu được tách ra và sử dụng cho việc huấn luyện. Một số phần từ tập dữ liệu như ID người dùng, Giờ, trò chơi, v.v. được sử dụng trong mô hình huấn luyện. Các phần dữ liệu này được giữ theo thứ tự, và số lượng cột đã được giảm bớt để làm cho tập dữ liệu trở nên hiệu quả hơn.

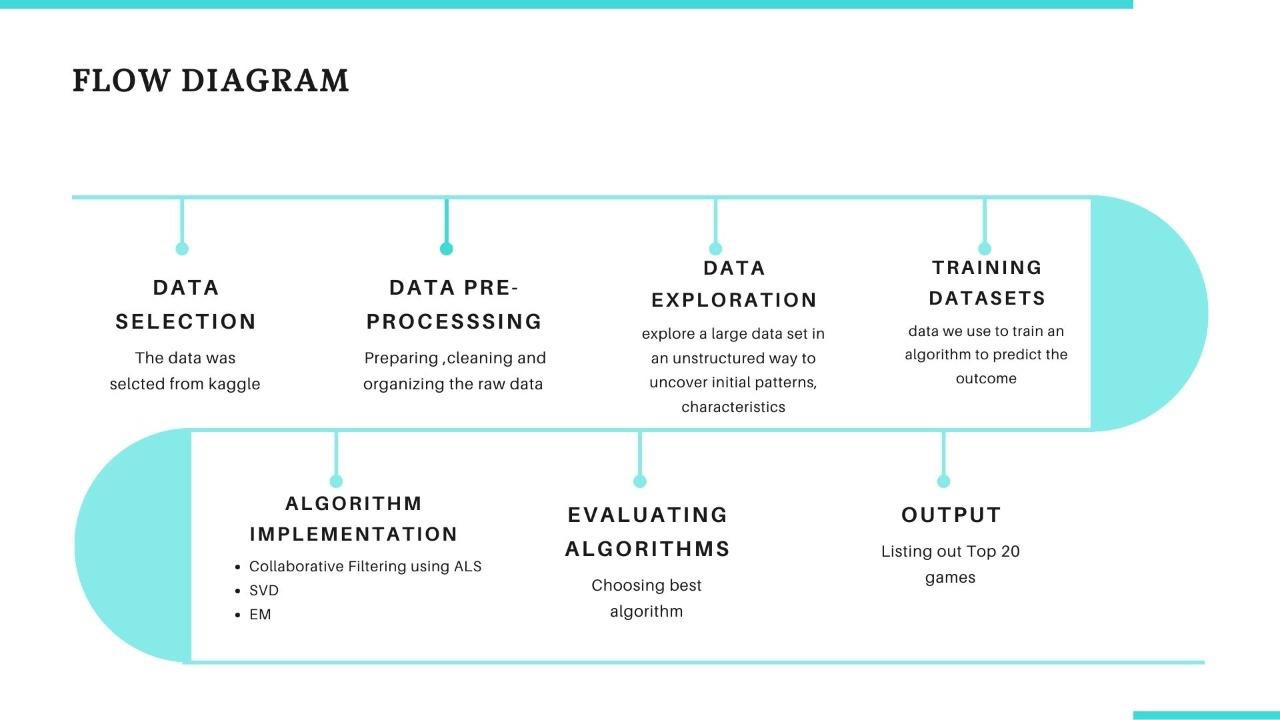
Tập dữ liệu đầu tiên là tập dữ liệu người dùng. Nó chứa mã người dùng, tiêu đề trò chơi, hành vi ('mua' hoặc 'chơi') và một giá trị liên quan đến hành vi. Mỗi hàng trong tập dữ liệu đại diện cho hành vi của một người dùng đối với một trò chơi, hoặc là 'chơi' hoặc là 'mua'. Nếu hành vi là 'chơi', giá trị liên quan đến nó tương ứng với số giờ chơi. Nếu hành vi là 'mua', giá trị liên quan đến nó là 1, có nghĩa là người dùng đã mua trò chơi. Trong trường hợp của tập dữ liệu người dùng này, giá trị liên quan đến 'mua' luôn là 1. Một phần của tập dữ liệu người dùng được hiển thị trong bảng dưới đây. Cần lưu ý rằng tập dữ liệu gốc không có tiêu đề, và những tiêu đề được hiển thị trong bảng dưới đây được thêm vào để tiện lợi dựa trên mô tả dữ liệu.

# PHƯƠNG PHÁP NGHIÊN CỨU

Trong dự án này, chúng tôi trình bày một Hệ thống Gợi ý dựa trên các trò chơi video trực tuyến. Hệ thống có khả năng trích xuất thông tin từ các tập dữ liệu đã chọn và được xử lý, làm sạch thông qua việc lọc để loại bỏ tiếng ồn không mong muốn, khoảng trống, loại bỏ các giá trị sai lệch và điền giá trị trung bình vào dữ liệu trống. Lọc hợp tác là một phương pháp của hệ thống gợi ý, được thực hiện để tìm và xác định các trò chơi video tương tự theo sở thích của người dùng. Lọc hợp tác lọc các đối tượng tương tự với sự trợ giúp của sở thích lịch sử của người dùng trên một tập hợp các mục. Hệ thống Gợi ý giúp xác định các đối tượng tương tự và gợi ý chúng cho người dùng. Một chuỗi các thuật toán được triển khai cùng với lọc hợp tác, nhằm giúp xác định và đánh giá độ chính xác.

# SƠ ĐỒ LUỒNG

Sơ đồ thể hiện các giai đoạn liên quan trong dự án.



Hình: Sơ đồ luồng

# TRIỂN KHAI

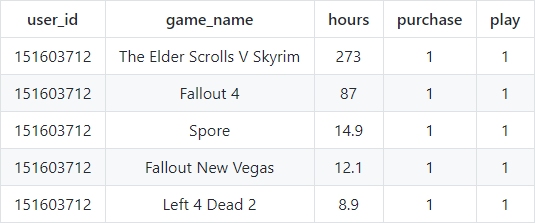
## Thu thập Dữ liệu

Đối với dự án này, hai tập dữ liệu khác nhau được sử dụng. Cả hai đều có sẵn miễn phí trên Kaggle và chứa dữ liệu được trích xuất từ Steam.

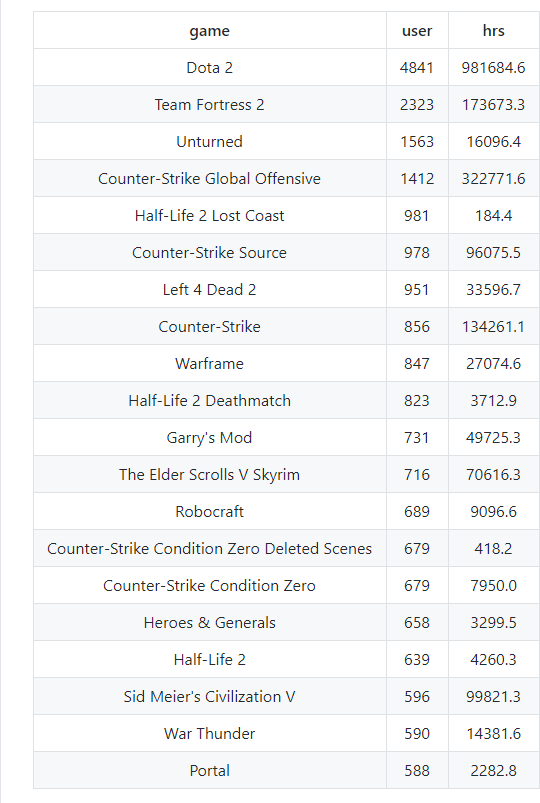
## Tiền xử lý dữ liệu

Tập dữ liệu người dùng chứa tổng cộng 200.000 hàng, bao gồm 5.155 trò chơi độc nhất và 12.393 người dùng độc nhất. Để thuận tiện, chúng tôi đã định dạng lại cấu trúc tập dữ liệu người dùng bằng cách tách thông tin được lưu trữ trong cột 'hành vi' thành hai cột: 'mua' và 'chơi'. Đối với mỗi hàng, cột 'chơi' có giá trị 1 nếu người dùng đã chơi trò chơi hoặc 0 nếu người dùng không có ghi chép về số giờ đã chơi.

Mỗi hàng trong tập dữ liệu người dùng đã được định dạng lại đại diện cho một tương tác duy nhất giữa người dùng và trò chơi. Một phần của tập dữ liệu người dùng đã được định dạng lại được hiển thị trong bảng dưới đây.



Sử dụng tập dữ liệu người dùng đã được định dạng lại, chúng tôi bắt đầu khám phá và phân tích dữ liệu được lưu trữ trong đó. Chúng tôi cố gắng đánh giá xem các trò chơi được mua nhiều nhất có tương ứng với các trò chơi được chơi nhiều nhất hay không. Đối với mỗi trò chơi, chúng tôi tính toán tổng số người dùng và tổng thời gian mà trò chơi đã được chơi bởi tất cả người dùng. Kết quả được hiển thị trong bảng dưới đây theo thứ tự giảm dần dựa trên số lượng người dùng, điều này áp dụng cho 20 trò chơi hàng đầu có nhiều người dùng nhất.

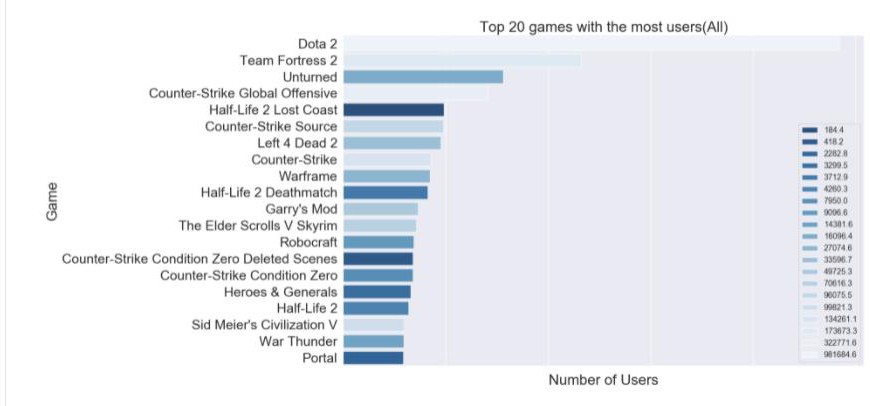


Chúng ta có thể thấy rằng đối với một số trò chơi, có mối quan hệ giữa số lượng người chơi nhiều nhất và số lượng mua nhiều nhất. Ví dụ, Dota 2 rõ ràng là trò chơi phổ biến nhất, nó có khả năng...

số lượng người dùng lớn nhất và tổng số giờ chơi cao nhất. Tuy nhiên, điều này không phải lúc nào cũng đúng, một ví dụ thú vị là 'Half-Life 2 Lost Coast' có số lượng người dùng cao (981 người dùng), nhưng tổng số giờ chơi lại khá thấp (184,4 giờ). Một lý do khả dĩ cho điều này có thể là trò chơi này được mua như một phần của gói trò chơi.

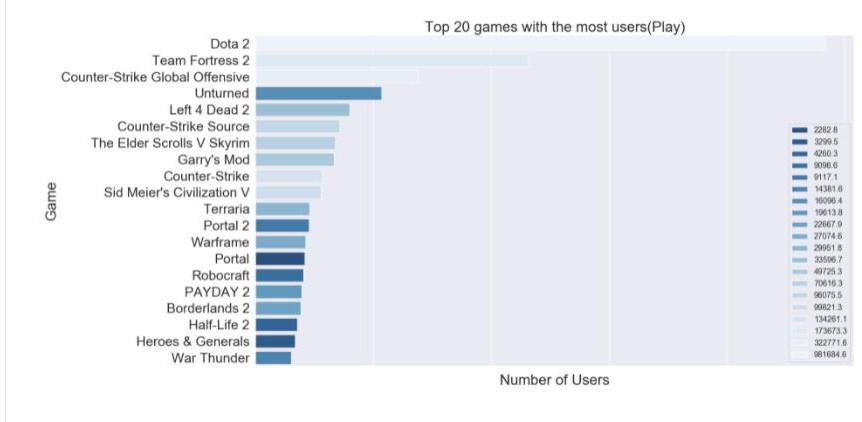
## Trực quan hóa dữ liệu:

Chúng tôi sử dụng biểu đồ histogram để trực quan hóa tốt hơn các kết quả được hiển thị trong bảng ở trên. Các tiêu đề trò chơi được sắp xếp theo thứ tự giảm dần dựa trên số lượng người dùng. Độ gradient màu đại diện cho tổng số giờ chơi, từ nhiều nhất đến ít nhất.



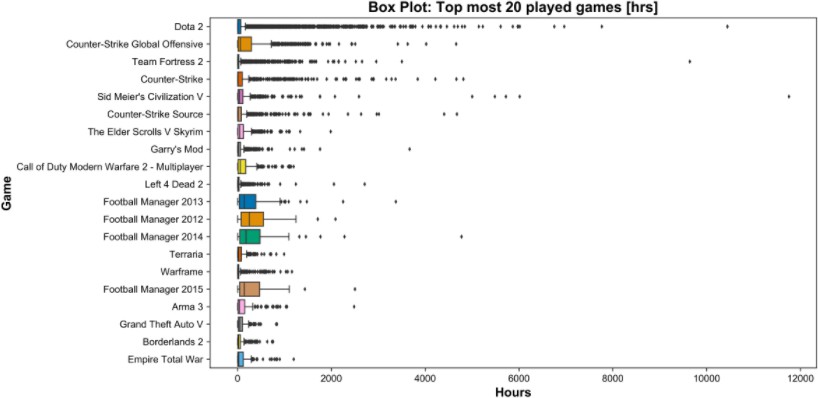
Chúng ta có thể thấy rõ rằng trong một số trường hợp không có mối quan hệ giữa tổng số người dùng và tổng số giờ chơi, có nghĩa là một số lượng người dùng cao không đại diện cho tổng số giờ chơi cao tương ứng.

Chúng tôi tái tạo cùng một loại biểu đồ, nhưng lần này chỉ xem xét những người dùng đã chơi các trò chơi. Do đó, đối với mỗi trò chơi, chúng tôi đã loại bỏ những người dùng đã mua nhưng không bao giờ chơi.



Khi so sánh biểu đồ mới này với biểu đồ trước đó, chúng tôi có thể thấy rằng một số trò chơi đã rớt khỏi top 20 trò chơi dựa trên số lượng người dùng. Ví dụ, 'Counter-Strike Condition Zero', đứng thứ 15 trong biểu đồ xem xét tất cả người dùng đã mua trò chơi, không xuất hiện trong top 20 trò chơi chỉ xem xét những người dùng đã chơi trò chơi. Một ví dụ ngược lại là 'Terraria', xuất hiện trong biểu đồ thứ hai với vị trí thứ 11 trong khi không được liệt kê trong biểu đồ đầu tiên. Như đã đề cập trước đó, một giải thích khả thi cho sự khác biệt này có thể là nhiều trò chơi đã được mua như một phần của gói trò chơi.

Để có cái nhìn tốt hơn về phân phối dữ liệu người dùng và thói quen chơi game của người dùng, một biểu đồ box plot được tạo ra cho 20 trò chơi được chơi nhiều nhất, dựa trên tổng số giờ chơi.



Như chúng ta có thể thấy, phân phối dữ liệu cho mỗi trò chơi được xem xét không đối xứng. Hơn nữa, 75% các điểm dữ liệu cho mỗi trò chơi nằm trong khoảng hàng trăm giờ, với một số trò chơi có các giá trị ngoại lệ rất lớn. Chúng ta có thể thấy ví dụ một người dùng đã chơi hơn 10.000 giờ "Dota 2". Một ví dụ thú vị khác, một người dùng đã chơi gần 12.000 giờ "Sid Meier's Civilization V".

## Huấn luyện và Kiểm tra Datasets:

Trước khi triển khai các thuật toán sẽ được sử dụng cho hệ thống gợi ý lọc cộng tác, tập dữ liệu huấn luyện và kiểm tra được tạo ra từ tập dữ liệu người dùng đã được định dạng lại.

Phiên bản tập dữ liệu người dùng đã được định dạng lại của chúng tôi có tổng cộng 128804 hàng, mỗi hàng có thông tin duy nhất về các tương tác giữa người dùng và trò chơi. Chúng tôi quyết định trích xuất 20% tất cả các tương tác giữa người dùng và trò chơi (25761 hàng) cho tập dữ liệu kiểm tra và giữ lại phần còn lại (103043 hàng) cho tập dữ liệu huấn luyện.

Tập dữ liệu huấn luyện được sử dụng để triển khai các mô hình gợi ý lọc cộng tác. Khi hoàn thành, các mô hình sẽ được sử dụng để tạo ra các gợi ý cho tất cả người dùng được liệt kê trong tập dữ liệu kiểm tra.

## Gợi ý lọc cộng tác với ALS:

Phần này mô tả một triển khai đơn giản của thuật toán gợi ý lọc cộng tác sử dụng phân tích ma trận với dữ liệu ngầm. Công việc được trình bày dựa trên "Gợi ý lọc cộng tác ALS ngầm" và các bài viết blog "Giới thiệu nhẹ nhàng về hệ thống gợi ý với phản hồi ngầm".

Lọc cộng tác không yêu cầu bất kỳ thông tin nào về các mục hoặc người dùng để cung cấp gợi ý. Nó chỉ sử dụng các tương tác giữa người dùng và các mục được thể hiện bằng đánh giá.

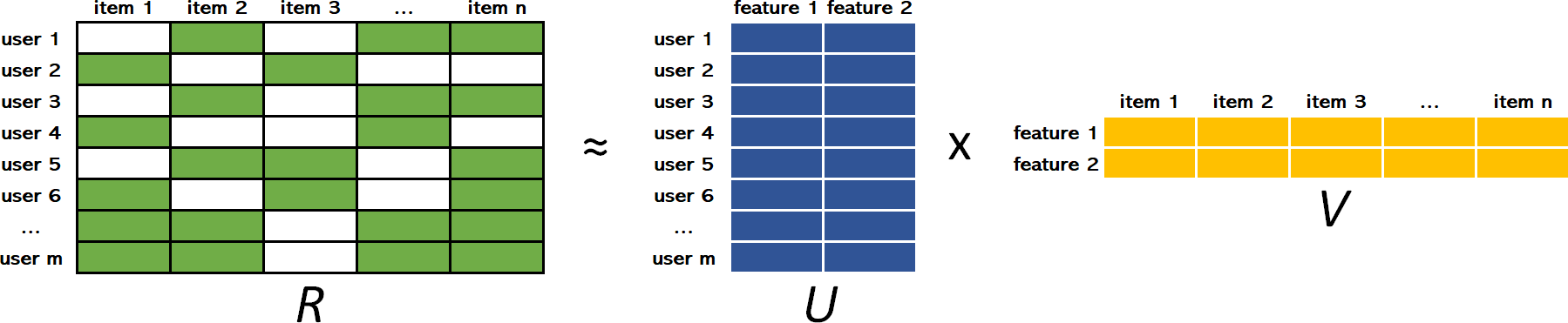
Dữ liệu được sử dụng cho hệ thống gợi ý này là dữ liệu người dùng Steam đã được định dạng lại. Dữ liệu không chứa rõ ràng đánh giá hoặc sở thích của người dùng đối với các trò chơi, mà thay vào đó được thể hiện ngầm qua số giờ người dùng đã chơi các trò chơi.

Mô hình Alternating Least Squares (ALS) được sử dụng để phù hợp với dữ liệu và tạo ra các gợi ý. Mô hình ALS đã được triển khai sẵn trong thư viện python implicit. cảm ơn [Ben Frederickson.](http://www.benfrederickson.com/fast-implicit-matrix-factorization/) Như được mô tả trong tài liệu của nó ở đây, thuật toán ALS có sẵn thông qua thư viện implicit là một Mô hình Gợi ý dựa trên các thuật toán được mô tả trong

bài báo Lọc cộng tác cho các tập dữ liệu phản hồi ngầm với các tối ưu hóa hiệu suất được mô tả trong Ứng dụng của Phương pháp Gradient Đồng nhất cho Lọc cộng tác phản hồi ngầm. Lợi thế của việc sử dụng thư viện python implicit so với việc triển khai thủ công thuật toán là tốc độ cần thiết để tạo ra gợi ý vì mô hình ALS trong thư viện implicit sử dụng Python cho phép phân tán mã giữa các luồng.

Thuật toán ALS sử dụng phân tích ma trận, về cơ bản là việc lấy một ma trận lớn và phân tích nó thành các ma trận nhỏ hơn mà tích của chúng bằng ma trận ban đầu. Đối với trường hợp hệ thống gợi ý lọc cộng tác với dữ liệu ngầm, phân tích ma trận về mặt toán học giảm ma trận ban đầu "tất cả người dùng so với tất cả các mục" thành các ma trận nhỏ hơn "tất cả người dùng so với một số đặc trưng" và "tất cả các mục so với một số đặc trưng". Các đặc trưng được đề cập được học từ dữ liệu và không nhất thiết đại diện cho bất kỳ siêu dữ liệu thực nào.

ALS là một quá trình tối ưu hóa lặp đi lặp lại cố gắng đạt được một đại diện phân tích gần hơn và gần hơn (U x V) của ma trận ban đầu R ở mỗi lần lặp.



Trong hình trên, R là ma trận ban đầu người dùng-mục chứa dữ liệu ngầm bên trong. U và V có trọng số đo lường cách mà mỗi người dùng-mục liên quan đến mỗi đặc trưng. Mục tiêu là tính toán trọng số của U và V sao cho R ≈ U x V. Thuật toán ALS (Alternating Least Squares) luân phiên tối ưu hóa U và cố định V và ngược lại cho đến khi hội tụ mà gần gũi nhất với R.

## Gợi ý lọc cộng tác với Tối đa hóa Kỳ vọng và Phân tích Giá trị Đơn:

**Thuật toán Tối đa hóa Kỳ vọng: Phân phối Giờ Chơi Trò Chơi:**

Thuật toán Tối đa hóa Kỳ vọng (EM) là một phương pháp cho ước lượng khả năng tối đa trong sự hiện diện của các biến tiềm ẩn. Đây là một phương pháp thích hợp để ước lượng các tham số của một phân phối dữ liệu nhất định.

Để xây dựng một hệ thống đánh giá (vì tập dữ liệu người dùng có dữ liệu ngầm), chúng tôi quyết định sử dụng các phân phối số giờ chơi cho mỗi trò chơi với thuật toán EM thay vì sử dụng các phần trăm.

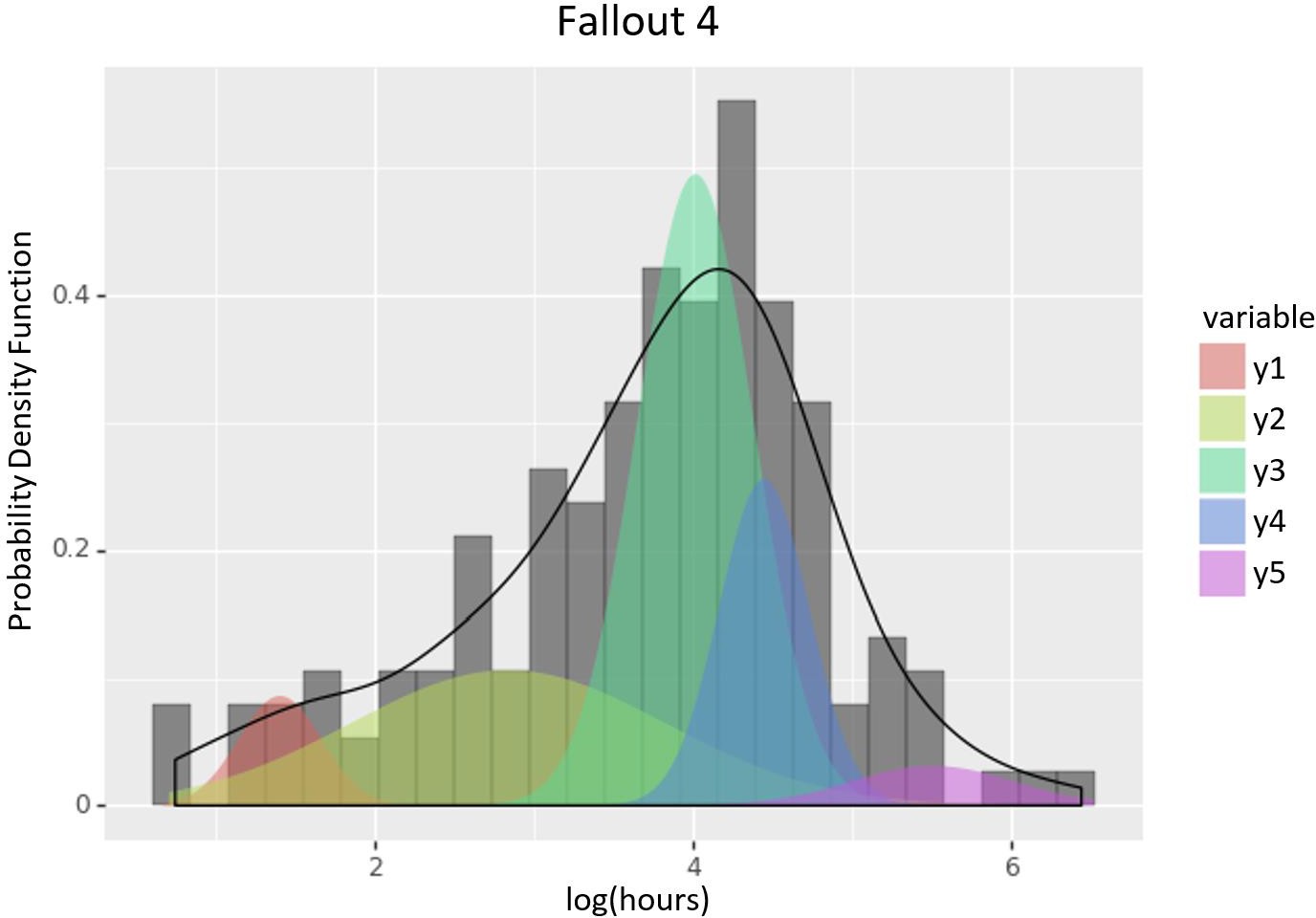
Chúng tôi tạo ra hệ thống đánh giá dựa trên phân phối số giờ chơi, cho mỗi trò chơi có sẵn trong tập dữ liệu người dùng. Chúng tôi sử dụng 5 nhóm (tương đương với hệ thống đánh giá 5 sao) để xác định một đánh giá mà người dùng sẽ đưa cho một trò chơi mà họ đã chơi dựa trên số giờ mà mỗi người đã chơi mỗi trò chơi so với những người khác.

Steam cho phép người dùng hoàn tiền cho các trò chơi mà họ đã chơi dưới 2 giờ. Chúng tôi đã quyết định xem xét yếu tố này cho hệ thống gợi ý của mình. Do đó, các tương tác giữa người dùng và mục có thời gian chơi dưới 2 giờ sẽ không được xem xét.

Mã dưới đây được sử dụng để vẽ đầu ra của thuật toán EM cho một trò chơi nhất định.



## Đầu ra:



Như chúng ta có thể thấy trong biểu đồ ở trên cho 'The Fallout 4', thuật toán EM thực hiện rất tốt việc tìm kiếm các nhóm (5) người có thói quen chơi game tương tự và có khả năng đánh giá một trò chơi theo cách tương tự. Chúng ta có thể thấy một số người dùng đã chơi trò chơi 'The Fallout 4' trong rất ít giờ. Có thể một số người dùng này đã mất hứng thú với trò chơi ngay sau khi bắt đầu chơi. Phân bố dày đặc hơn cho các nhóm 3 và.

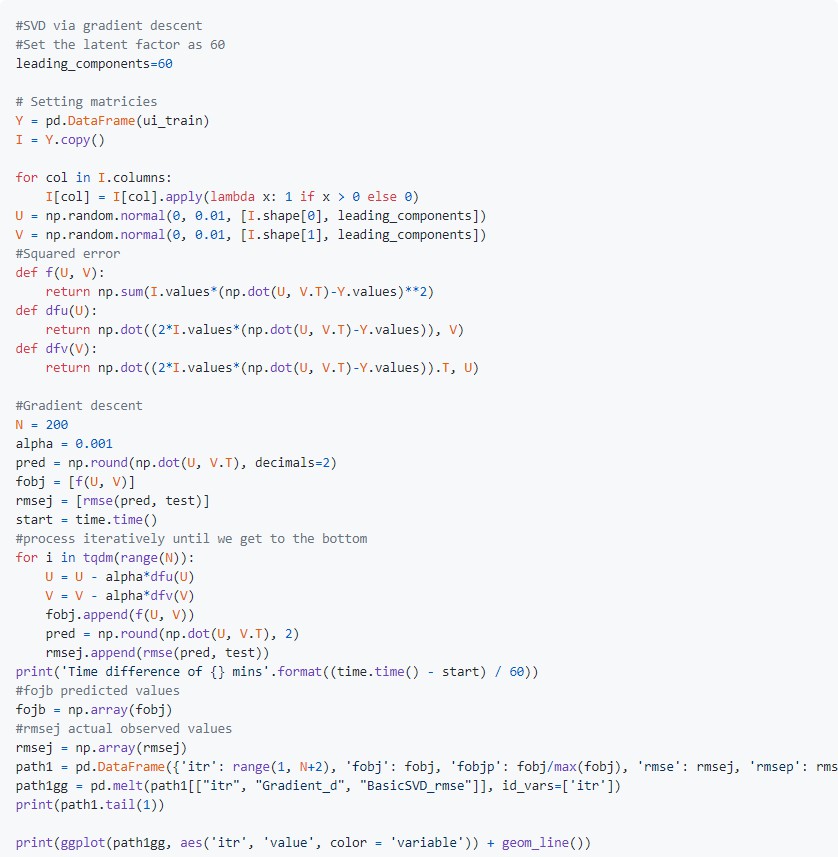
Điều này cho thấy rằng nhiều người dùng quan tâm đến trò chơi này. Do đó, trò chơi như vậy sẽ được đánh giá cao.

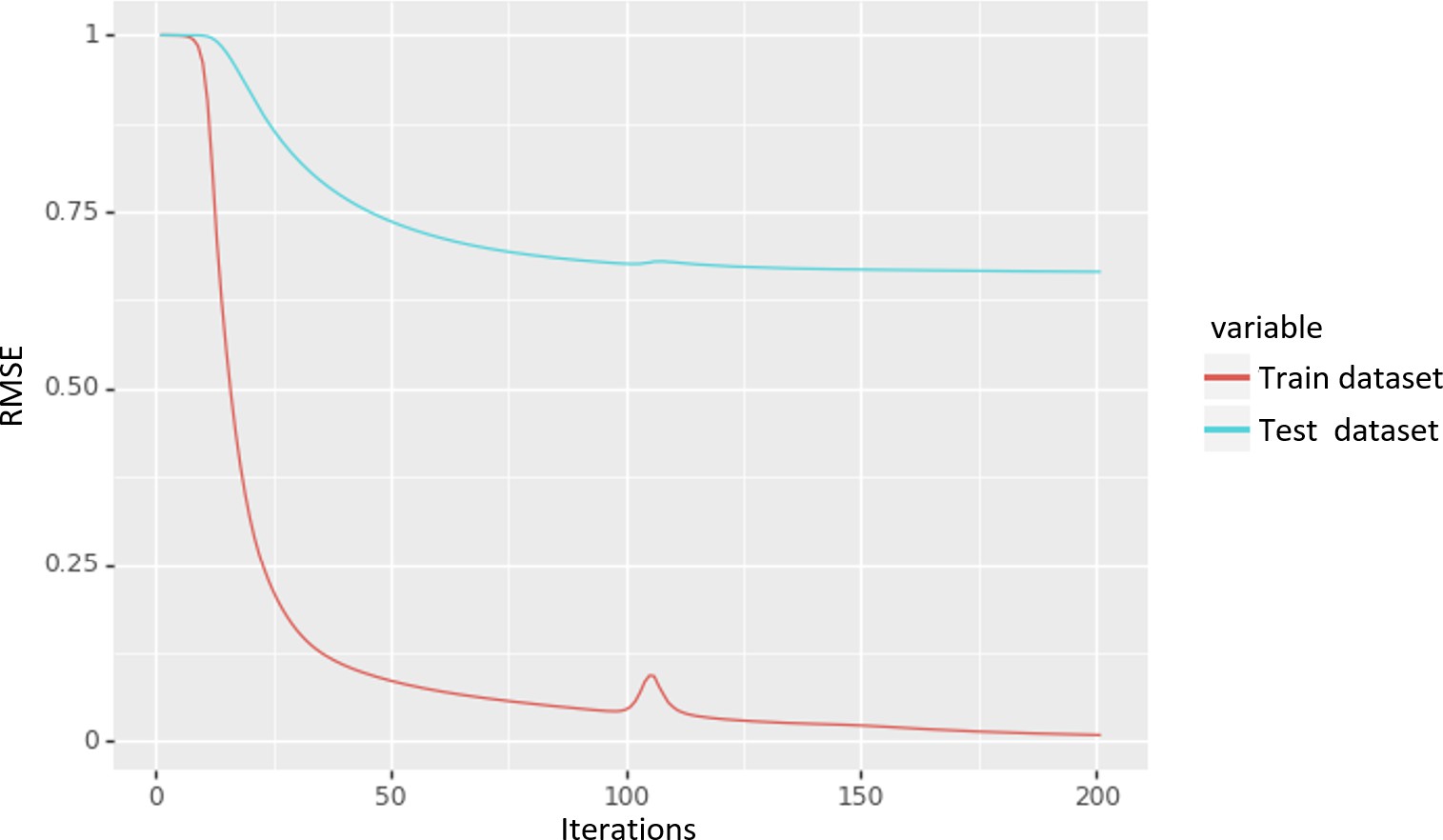
## Phân rã Giá trị Đơn lẻ thông qua Tối ưu hóa Độ dốc.

Theo tài liệu tham khảo của chúng tôi, một triển khai thuật toán SVD cơ bản không tạo ra các gợi ý đủ tốt dựa trên tập dữ liệu của chúng tôi. Do đó, chúng tôi quyết định triển khai thuật toán SVD thông qua phương pháp Giảm Độ Dốc, như đã được đề xuất trong đó.

Chúng tôi sử dụng thuật toán SVD để phân tích ma trận người dùng-mặt hàng thành các vector riêng và giá trị riêng (giống như những gì phân rã trị riêng thực hiện) và phương pháp gradient descent để xử lý dữ liệu thiếu thông qua việc dự đoán. Gradient descent là một phương pháp tối ưu hóa lồi mà chúng tôi sử dụng để tìm các ma trận U và V tối ưu đại diện cho ma trận người dùng-mặt hàng gốc, thay thế các giá trị thiếu bằng các giá trị mới được ước lượng thông qua việc sử dụng những người dùng và trò chơi tương tự.

Tương tự như những gì đã thực hiện trong tài liệu tham khảo của chúng tôi, chúng tôi thiết lập tốc độ học là 0.001 và số lần lặp là 200 trong khi theo dõi Sai số bình phương gốc (RMSE). Các ma trận U và V được khởi tạo với các giá trị ngẫu nhiên được rút ra từ phân phối chuẩn [0, 0.01]. Hàm được theo dõi đo lường RMSE giữa các giá trị thực và các giá trị dự đoán.





Đồ thị cho thấy rằng SVD thông qua phương pháp gradient descent hội tụ về giá trị không trên tập dữ liệu huấn luyện, trong khi RMSE cho tập dữ liệu huấn luyện của chúng tôi duy trì xung quanh giá trị khoảng 0.68.

Thú vị là, chúng ta thấy rằng sau khoảng 75 - 100 lần lặp, độ chính xác trên tập dữ liệu kiểm tra ngừng cải thiện (RMSE duy trì ở mức giá trị tương tự). Độ chính xác trên dữ liệu kiểm tra có thể được cải thiện bằng cách sử dụng nhiều thành phần dẫn đầu hơn, với sự đánh đổi là thời gian tính toán yêu cầu nhiều hơn. Do đó, đối với dữ liệu đã sử dụng, chúng ta có thể dừng tính toán sau 75 - 100 lần lặp vì độ chính xác trên tập dữ liệu kiểm tra không còn cải thiện nữa. Hành vi này nhất quán với các kết quả được trình bày trong tài liệu tham khảo của chúng tôi.

## 4.4.5. Đánh giá & Phân tích:

Để so sánh các thuật toán khác nhau được sử dụng để tạo ra các gợi ý, chúng tôi tạo một kịch bản tính toán, cho mỗi người dùng, tỷ lệ giữa số lượng trò chơi trong tập dữ liệu kiểm tra người dùng nằm trong 20 gợi ý hàng đầu so với tổng số trò chơi trong tập dữ liệu kiểm tra người dùng. Trung bình của tỷ lệ từ tất cả người dùng sau đó được tính toán. Tỷ lệ này có phần thấp vì mỗi khi không thể tạo ra gợi ý cho một người dùng nhất định, tỷ lệ được tính toán sẽ được đặt thành 0.

Ý tưởng tính toán tỷ lệ theo cách này được lấy cảm hứng từ chỉ số độ chính xác tại K được sử dụng trong bài báo nghiên cứu KDD: Mô hình tương tự dựa trên sự chú ý theo thời gian thực cho hệ thống gợi ý.



Đầu tiên, chúng tôi so sánh thuật toán gợi ý dựa trên nội dung bằng cách sử dụng các đầu vào khác nhau. Các đầu vào này có thể là một cột từ tập dữ liệu gốc hoặc sự kết hợp của các cột khác nhau. Chúng tôi có thể thấy trong bảng dưới đây tỷ lệ được tính toán bằng cách sử dụng các đầu vào khác nhau. Trong trường hợp của chúng tôi, kết quả tốt nhất tương ứng với việc sử dụng sự kết hợp của các cột: thể loại, nhà xuất bản và nhà phát triển. Chính việc triển khai thuật toán gợi ý dựa trên nội dung này được sử dụng trong quá trình so sánh với hai thuật toán gợi ý hợp tác khác.

Chúng tôi đã tính toán tỷ lệ cho cả hai hệ thống gợi ý dựa trên lọc hợp tác theo cách tương tự như đã mô tả trước đó. Kết quả của chúng với hệ thống gợi ý dựa trên nội dung được giữ lại được trình bày trong bảng dưới đây.

# CÁC YÊU CẦU HỆ THỐNG

Để phát triển mô hình, chúng tôi sử dụng VS Code, Jupyter Notebook và trình biên dịch Python. Một máy tính với dung lượng RAM từ 4-16 GB là cần thiết, với hệ điều hành Windows 8 hoặc.

Cần phiên bản 8.1 hoặc 10 và dung lượng lưu trữ 10 GB. Các trình duyệt web như Google Chrome, Mozilla Firefox hoặc bất kỳ trình duyệt nào khác là cần thiết. Các thư viện Python như scipy, matplotlib, cv2, random, tqdm, numpy, pandas và os đã được sử dụng.

# KẾT QUẢ

Ba thuật toán khác nhau đã được triển khai, bao gồm Thuật toán Bình phương tối thiểu luân phiên (ALS), Phân rã giá trị riêng (SVD) và Tối ưu hóa kỳ vọng (EM), với mục tiêu chính là sử dụng Thuật toán Bình phương tối thiểu luân phiên trong Lọc cộng tác nhằm đạt được độ chính xác tốt hơn. Cả ba thuật toán đều được áp dụng để gợi ý các trò chơi video tương tự dựa trên sở thích của người dùng. Các mô hình có khả năng phân loại gợi ý theo sở thích của người dùng một cách hiệu quả. Qua điều tra, Thuật toán Bình phương tối thiểu luân phiên cho kết quả tốt nhất vì nó cung cấp độ chính xác cao hơn hai thuật toán còn lại. Các kết quả thu được như sau:



Như chúng ta có thể thấy, hệ thống gợi ý hợp tác sử dụng ALS là tốt nhất. Hiệu suất của hệ thống gợi ý hợp tác với EM và SVD cùng với hệ thống gợi ý dựa trên nội dung đều kém xa so với nó.

# KẾT LUẬN

Cuối cùng, chúng tôi đã triển khai ba thuật toán hệ thống gợi ý khác nhau, một thuật toán dựa trên nội dung và hai thuật toán lọc cộng tác (một sử dụng thuật toán ALS và thuật toán còn lại sử dụng EM và SVD). Như đã trình bày trong phần Đánh giá & Phân tích, đối với dự án của chúng tôi, hệ thống gợi ý cộng tác với thuật toán ALS cung cấp những gợi ý tốt nhất dựa trên đánh giá của chúng tôi. Dựa trên kinh nghiệm của chúng tôi với dự án này, chúng tôi hiểu rõ hơn về cách thức hoạt động của hệ thống lọc cộng tác. Thực tế, nó không sử dụng bất kỳ thông tin nào về các mục mà hoàn toàn dựa vào các tương tác giữa người dùng và mục cũng như các phép toán ma trận để tạo ra các gợi ý. Chúng tôi đã phải tìm ra một phương pháp để xử lý tập dữ liệu của mình (chỉ tập dữ liệu người dùng cho hệ thống gợi ý cộng tác) vì nó chỉ chứa dữ liệu ngầm. Cả hai phương pháp được mô tả trong blog này đều xử lý tập dữ liệu ngầm theo những cách khác nhau để tạo ra các gợi ý. Ví dụ, để loại bỏ một số dữ liệu nhiễu, chúng tôi đã sử dụng phân tích giá trị riêng (SVD) như một kỹ thuật giảm chiều để dễ dàng xử lý tập dữ liệu lớn của chúng tôi. Mặt khác, chúng tôi nhận thấy rõ ràng rằng phương pháp dựa trên nội dung yêu cầu mô tả về các mục để tạo ra gợi ý. Chúng tôi đã gặp một số thách thức khi triển khai hệ thống gợi ý dựa trên nội dung vì nó sử dụng hai tập dữ liệu khác nhau. Khi dự án bắt đầu sử dụng hai tập dữ liệu (tập dữ liệu người dùng và tập dữ liệu trò chơi), chúng tôi đã kỳ vọng tìm thấy tất cả các trò chơi có sẵn trong tập dữ liệu người dùng trong tập dữ liệu trò chơi, vì cả hai đều đến từ Steam. Tuy nhiên, khi dự án tiến triển, chúng tôi nhận ra rằng điều này không đúng. Trong số 5152 tiêu đề trò chơi có sẵn trong tập dữ liệu người dùng, chúng tôi chỉ có thể tìm thấy 3036 tiêu đề trò chơi trong tập dữ liệu trò chơi. Điều này gây ra một vấn đề lớn cho hệ thống gợi ý dựa trên nội dung vì nó dựa vào giả định rằng tất cả các tiêu đề trò chơi có sẵn trong tập dữ liệu người dùng đều có thông tin trong tập dữ liệu trò chơi. Vì lý do này, không thể tạo ra gợi ý cho mọi trò chơi mà người dùng đã mua. Hơn nữa, điều này cũng không thể tạo ra gợi ý cho một số người dùng do vấn đề này. Điều này có thể ảnh hưởng đến hiệu suất của thuật toán dựa trên nội dung. Nếu chúng tôi phải thực hiện một dự án tương tự về hệ thống gợi ý dựa trên nội dung một lần nữa, điều quan trọng là phải đảm bảo rằng chúng tôi có thông tin về mọi mục khi chọn các tập dữ liệu.

# PHỤ LỤC

*nhập implicit nhập pathlib nhập pandas như pd*

*nhập scipy.sparse như sparse*

Lớp Gợi ý Hợp tác Ngầm:

*# Thuộc tính lớp.*

*Tên cột nội bộ*

người dùng quốc tế, mặt hàng quốc tế

user\_internal, item\_internal

biến\_ẩn = 'đo\_lường\_ngầm'

*Tên cột gốc*

user\_o, item\_o, impl\_o = Không, Không, Không

*Dữ liệu và bảng tra cứu.*

dữ liệu, tra\_cứu\_người\_dùng, tra\_cứu\_mục = Không, Không, Không

*Mô hình và ma trận thưa.*

mô hình, m\_user\_item, m\_item\_user = Không, Không, Không

def khởi\_tạo(self, đường\_dẫn\_dữ\_liệu=None): nếu đường\_dẫn\_dữ\_liệu không phải là None:

*tự.load\_dữ\_liệu(đường\_dẫn\_dữ\_liệu)*

*nếu self.data không phải là None:*

*tự.load\_model()*

def load\_data(self, data\_path): # Tải dữ liệu huấn luyện.

df\_data = pd.read\_csv(pathlib.Path(data\_path))

*Số cột.*

col\_user = df\_data.columns[0] # Tên của cột 'user'.

col\_item = df\_data.columns[1] # Tên của cột 'item'.

col\_impl = df\_data.columns[2] # Tên của cột 'đo lường ngầm'.

*Lấy tên nội bộ cho các cột dữ liệu.*

col\_user\_intl = self. user\_intl # Tên nội bộ cho 'người dùng'.

col\_item\_intl col\_impl\_intl

= self. item\_intl

= tự. thực hiện\_quốc\_tế

Tên nội bộ cho 'mục'.

*Tên nội bộ cho 'thước đo ngầm'.*

*Xác minh xem dữ liệu NA có tồn tại trong tập dữ liệu hay không. In ra 'Số lượng dữ liệu NA theo từng cột:' In ra df\_data.isna().sum(axis=0)*

*Chuyển đổi 'người dùng' và 'mặt hàng' thành ID số.*

df\_data[col\_user\_intl] = df\_data[col\_user].astype('category').cat.codes df\_data[col\_item\_intl] = df\_data[col\_item].astype('category').cat.codes

*Tạo bảng tra cứu cho 'user\_id - người dùng' và 'item\_id - mặt hàng'. lookup\_user = df\_data[[col\_user\_intl, col\_user]].drop\_duplicates() lookup\_user[col\_user\_intl] = lookup\_user[col\_user\_intl].astype(str) lookup\_user[col\_user] = lookup\_user[col\_user].astype(int).astype(str) lookup\_game = df\_data[[col\_item\_intl, col\_item]].drop\_duplicates() lookup\_game[col\_item\_intl] = lookup\_game[col\_item\_intl].astype(str)*

*Dọn dẹp dataframe với các cột: 'user\_id', 'item\_id' và 'thước đo ngầm'.*

in\_print(col\_impl) in\_print(col\_impl\_intl)

df\_data.rename(columns={col\_impl: col\_impl\_intl}, inplace=True) df\_data = df\_data[[col\_user\_intl, col\_item\_intl, col\_impl\_intl]]

*Xác minh tất cả dữ liệu 'biện pháp ngầm' được xem xét có giá trị lớn hơn không. in('\nThống kê cột \'{}\' :'.format(col\_impl)) in(df\_data[col\_impl\_intl].describe())*

*Gán kết quả cho các biến lớp. self.data = df\_data self.lookup\_users = lookup\_user self.lookup\_items = lookup\_game*

*self. user\_o self. item\_o self. impl\_o*

= col\_user

= col\_item

= col\_impl

def load\_model(self):

*Lấy tên nội bộ cho các cột dữ liệu.*

col\_user\_intl col\_item\_intl col\_impl\_intl

= self. user\_intl

= self. item\_intl

= tự. thực hiện\_quốc\_tế

*nếu self.data không phải là None:*

df\_data = self.data

*# Tạo ma trận thưa: item-user và user-item.*

sparse\_item\_user = sparse.csr\_matrix((df\_data[col\_impl\_intl].astype(float),

(df\_data[col\_item\_intl], df\_data[col\_user\_intl])))

sparse\_user\_item = sparse.csr\_matrix((df\_data[col\_impl\_intl].astype(float),

(df\_data[col\_user\_intl], df\_data[col\_item\_intl])))

*ma trận người dùng*

# Khởi tạo mô hình als và phù hợp với nó bằng cách sử dụng item-thưa.

model = implicit.als.AlternatingLeastSquares(factors=20,

*regularization=0.1, iterations=20)*

*lue.*

# Tính toán độ tin cậy bằng cách nhân với giá trị alpha đã định nghĩa

alpha\_val = 15

data\_conf = (sparse\_item\_user \* alpha\_val).astype('double')

*# Phù hợp dữ liệu với mô hình*

model.fit(data\_conf)

*# Gán kết quả cho các biến lớp. self. model = model*

*self. m\_item\_user = sparse\_item\_user*

*self. m\_user\_item = sparse\_user\_item*

*else:*

*self. model = None self. m\_item\_user = None self. m\_user\_item = None*

def similar\_items(self, items, n\_similar):

model = self. model lookup\_items = self.lookup\_items n\_similar = n\_similar + 1

*# Lấy tên cột.*

col\_item\_intl = self. item\_intl

col\_item\_o = self. item\_o

*# Sử dụng các phương thức thư viện implicit để lấy các mục tương tự.*

output = []

*for item in items:*

item\_id = lookup\_items[col\_item\_intl]. \ loc[lookup\_items[col\_item\_o] == str(item)]

*if item\_id.empty:*

item\_names = [-999] \* n\_similar else:

item\_id = item\_id.to\_string(index=False).strip() similar = model.similar\_items(int(item\_id), n\_similar)

item\_names = []

*for item\_id\_r, score in similar:*

item\_name = lookup\_items[col\_item\_o]. \ loc[lookup\_items[col\_item\_intl] == str(item\_id\_r)]. \ to\_string(index=False).strip()

item\_names.append(item\_name)

output.append(item\_names)

*# Tạo dataframe để lưu trữ kết quả các mục tương tự.*

col\_names = list(map(str, range(1, n\_similar)))

df\_similar = pd.DataFrame(output, columns=[col\_item\_o] + col\_names)

*return df\_similar*

def recommend(self, users, n\_recommendation):

model = self. model lookup\_users = self.lookup\_users lookup\_items = self.lookup\_items

*# Lấy tên cột.*

col\_user\_intl = self. user\_intl

col\_user\_o = self. user\_o

col\_item\_intl = self. item\_intl

col\_item\_o = self. item\_o

*# Sử dụng phương thức gợi ý của thư viện implicit.*

output = []

*for user in users:*

user\_id = lookup\_users[col\_user\_intl]. \ loc[lookup\_users[col\_user\_o] == str(user)]

*if user\_id.empty:*

item\_names = [-999] \* n\_recommendation else:

user\_id = user\_id.to\_string(index=False).strip() recommended = model.recommend(int(user\_id),

*self. m\_user\_item,*

*n\_recommendation, False)*

item\_names = []

*for item\_id, score in recommended: item\_name = lookup\_items[col\_item\_o]. \*

loc[lookup\_items[col\_item\_intl] == str(item\_id)]. \ to\_string(index=False).strip()

item\_names.append(item\_name) output.append([user, \*item\_names])

*# Tạo dataframe để lưu trữ các gợi ý.*

col\_names = list(map(str, range(1, n\_recommendation + 1))) df\_recommendation = pd.DataFrame(output,

*columns=[col\_user\_o] + col\_names)*

*return df\_recommendation*

*nếu tên == "main":*

*Lấy người dùng từ dữ liệu thử nghiệm để tạo ra các khuyến nghị.*

test\_location = r'../../data/model\_data/steam\_user\_test.csv' df\_test = pd.read\_csv(test\_location)

danh\_sách\_người\_dùng = df\_test['user\_id'].duy\_nhất()

*Tạo mô hình gợi ý hợp tác (ALS).*

train\_location = r'../../data/model\_data/steam\_user\_train.csv' f\_implicit = ImplicitCollaborativeRecommender(train\_location)

*df\_sim = f\_implicit.similar\_items(['Dota 2', 'xxxxx', 'Fallout 4', 'Left 4 Dead 2'], 20)*

df\_rec = f\_implicit.recommend(list\_users, 20) df\_rec.to\_csv(r'../../data/output\_data/Collaborative\_recommender\_als\_output.c

sv,

*index=False)*

# TÀI LIỆU THAM KHẢO

1. Cheuque, Germán, José Guzmán, và Denis Parra. Hệ thống gợi ý cho các nền tảng trò chơi trực tuyến: Trường hợp của STEAM. Tài liệu đồng hành của Hội nghị World Wide Web 2019. 2019.
2. Stefan Heinz, Yvonne Lau và Daniel Epstein, “Metarecommendr: Hệ thống gợi ý cho trò chơi điện tử, phim và chương trình truyền hình”, tháng 4 năm 2017.
3. He, Xiangnan, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, và Tat-Seng Chua. Neural collaborative filtering. Trong Kỷ yếu của hội nghị quốc tế lần thứ 26 về web toàn cầu, trang 173-182. 2017.
4. Pérez-Marcos, J., Martin-Gomez, L., Jiménez-Bravo, D. M., López, V. F., & Moreno-García, M. N. (2020). Hệ thống lai cho gợi ý trò chơi điện tử dựa trên đánh giá ngầm và mạng xã hội. Tạp chí Trí tuệ Nhân tạo và Tính toán Con người, 11(11), 4525-4535.
5. Bertens, P., Guitart, A., Chen, P. P., & Perianez, A. (2018, tháng 8). Hệ thống gợi ý mục dựa trên máy học cho trò chơi điện tử. Trong Hội nghị IEEE về Trí tuệ Tính toán và Trò chơi (CIG) 2018 (trang 1-4). IEEE.
6. Hệ thống Gợi ý trong Lĩnh vực Trò chơi Điện tử: So sánh Nhiều Thuật toán, Thang Đánh giá và Phản hồi Ngầm/ Rõ ràng Sử dụng Nền tảng Steam, Graz, tháng 9 năm 2016.
7. Yang, L., Hong, T., & Gopalakrishnan, A. Một Khung cho Hệ thống Gợi ý Dựa trên Lý thuyết Trò chơi trong Mạng Xã hội. Trong Hội nghị Quốc tế lần thứ 10 về Kiến thức và Công nghệ Thông minh (KST) 2018 (trang 95-100). IEEE.
8. Anwar, S. M., Shahzad, T., Sattar, Z., Khan, R., & Majid, M. (2017, tháng 1). Hệ thống gợi ý trò chơi sử dụng lọc hợp tác (GAMBIT). Trong Hội nghị Quốc tế lần thứ 14 về Khoa học và Công nghệ Ứng dụng Bhurban (IBCAST) (trang 328-332). IEEE.
9. Kim, JaeWon, JeongA Wi, SooJin Jang, và YoungBin Kim. “Khuyến nghị tuần tự trên các nền tảng trò chơi board.” Symmetry 12, số 2 (2020): 210.
10. Ahirwadkar, Bhakti & Deshmukh, Sachin. (2020). Deepautoencf: Một Autoencoder Denoising cho Hệ thống Gợi ý. Tạp chí Khoa học Máy tính và Kỹ thuật Ấn Độ. 11. 244-250. 10.21817/indjcse/2020/v11i3/201103199.
11. Song, Yading, Simon Dixon, và Marcus Pearce. A survey of music recommendation systems and future perspectives. Hội nghị Quốc tế lần thứ 9 về Mô hình hóa Âm nhạc Máy tính và Khôi phục. Tập 4. 2012.
12. Sánchez-Moreno, Diego, và cộng sự. A collaborative filtering method for music recommendation using playing coefficients for artists and users. Tạp chí Hệ thống Chuyên gia với Ứng dụng 66 (2016): 234-244.
13. Choi, Keunho, và cộng sự. A hybrid online-product recommendation system: Kết hợp lọc cộng tác dựa trên đánh giá ngầm và phân tích mẫu tuần tự. Nghiên cứu và ứng dụng thương mại điện tử 11.4 (2012): 309-317.
14. Wang, Hao, Naiyan Wang, và Dit-Yan Yeung. Học sâu hợp tác cho hệ thống gợi ý. Kỷ yếu hội nghị quốc tế lần thứ 21 của ACM SIGKDD về khám phá tri thức và khai thác dữ liệu. 2015.
15. Dong, Xin, và cộng sự. A hybrid collaborative filtering model with deep structure for recommender systems. Tài liệu Hội nghị AAAI về Trí tuệ Nhân tạo. Tập 31. Số 1. 2017.