

**BỘ GIÁO DỤC VÀ ĐÀO TẠO
TRƯỜNG ĐẠI HỌC NGUYỄN TẤT THÀNH
KHOA CÔNG NGHỆ THÔNG TIN**



**CUỘC THI KHOA HỌC DỮ LIỆU
KHOA CÔNG NGHỆ THÔNG TIN NĂM 2024**

**BÁO CÁO XÂY DỰNG ỨNG DỤNG KHOA HỌC DỮ LIỆU
HỆ THỐNG ĐỀ XUẤT PHIM**

Tên nhóm/đội: ERROR 404

Danh sách Sinh viên thực hiện:

Nguyễn Gia Chi Bảo	MSSV: 2200001314
Nguyễn Thành Tâm	MSSV: 2200000922
Lê Văn Đức	MSSV: 2200000234
Lê Thị Hạnh Uyên	MSSV: 2200005635
Phan Lâm Nhật Khang	MSSV: 2200008424

Tp HCM, tháng 8 năm 2024

LỜI MỞ ĐẦU

Hệ thống đề xuất phim ngày càng trở nên quan trọng trong việc nâng cao trải nghiệm người dùng trên các trang web xem phim. Tuy nhiên, việc tìm kiếm và khám phá các bộ phim mới một cách thủ công có thể mất nhiều thời gian và không hiệu quả. Do đó, việc sử dụng các phần mềm tích hợp hệ thống đề xuất phim dựa trên các yếu tố như thể loại, nội dung, diễn viên, và chủ đề để gợi ý các bộ phim liên quan là vô cùng cần thiết.

Hệ thống đề xuất phim phân tích các yếu tố của bộ phim mà người dùng chọn hoặc tìm kiếm, từ đó tự động gợi ý các bộ phim tương tự hoặc có liên quan mà họ có thể yêu thích. Điều này giúp người dùng dễ dàng khám phá thêm các bộ phim mới một cách nhanh chóng và hiệu quả, đồng thời làm phong phú thêm lựa chọn của họ.

Báo cáo này trình bày về hệ thống "Đề xuất phim nâng cao trải nghiệm người dùng trên trang web xem phim". Trong báo cáo này, chúng em sẽ trình bày chi tiết về quá trình phát triển và triển khai của phần mềm, bao gồm cách thức hoạt động, phương pháp và công nghệ được sử dụng, cũng như kết quả thực nghiệm và đánh giá hiệu suất của hệ thống.

Chúng em hy vọng rằng phần mềm này sẽ không chỉ là một công cụ hữu ích cho các trang web xem phim, mà còn là một đóng góp tích cực vào việc cải thiện trải nghiệm người dùng và tăng thời gian tương tác trên trang web.

Xin cảm ơn thầy cô đã quan tâm và đồng hành cùng chúng em trong hành trình này.

MỤC LỤC

CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI	1
1.1. Trình bày vấn đề của đề tài	1
1.2. Cách giải quyết của đề tài	1
1.3. Đối tượng và phạm vi nghiên cứu:	2
CHƯƠNG 2: CƠ SỞ LÝ THUYẾT.....	3
2.1. Sự hình thành và Phát triển của Machine Learning.....	3
2.1.1. Các phương pháp Machine Learning	4
2.1.2. Các thuật toán phổ biến của Machine Learning	6
2.1.3. Ứng dụng thực tế của Machine Learning	6
2.1.4. Nguồn dữ liệu và chuẩn bị dữ liệu	8
2.1.5. Sự khác biệt giữa Machine Learning với Deep Learning.	9
2.1.6. Tổng kết.....	10
2.2. Tính toán và ứng dụng Cosine Similarity	10
2.2.1. Tính toán Cosine Similarity	10
2.2.2. Ứng dụng Cosine Similarity trong hệ thống đề xuất phim.....	13
2.3. Recommendation System là gì?.....	14
2.3.1. Tại sao lại chọn Recommendation Systems	14
2.3.2. Các loại Recommendation Systems	15
CHƯƠNG 3: ỨNG DỤNG TRIỂN KHAI HỆ THỐNG ĐỀ XUẤT PHIM.....	18
3.1. Ngôn ngữ và công cụ	18
3.1.1. Ngôn ngữ	18
3.1.2. Công cụ.....	18
3.2. Cài đặt các thư viện cần thiết.....	19
3.3. Xử lý dữ liệu trên Google Colaboratory	19

3.3.1. Cấu trúc dữ liệu và tiền xử lý	19
3.3.2. Tiền xử lý dữ liệu	20
3.3.3. Tạo thẻ cho từng phim.....	20
3.3.4. Xây dựng mô hình đề xuất	21
3.3.5. Hàm gợi ý phim.....	22
3.3.6. Lưu trữ mô hình và dữ liệu.....	22
3.4. Thiết kế hệ thống và demo các bước chạy chương trình	22
3.4.1. Cấu trúc thư mục	22
3.4.2. Giao diện Code.....	23
3.4.3. Giao diện Web Application.....	24
CHƯƠNG 4: TỔNG KẾT	26
4.1. Kết quả đạt được	26
4.2. Ưu điểm	26
4.3. Điểm hạn chế	26
4.4. Hướng phát triển của đề tài.....	27
DANH MỤC TÀI LIỆU THAM KHẢO	28

DANH MỤC HÌNH ẢNH

Hình 2.1 : Machine Learning.....	3
Hình 2.2: Reinforcement Learning.....	5
Hình 2.3: Các phương pháp Machine Learning	5
Hình 2.4: Machine Learning Roadmap	8
Hình 2.5: Sơ đồ vecto	10
Hình 2.6: Recommendation System	14
Hình 2.7: Content-based và Collaborative Filtering.....	15
Hình 3.1: Giao diện Pycharm	18
Hình 3.2: Cấu trúc thư mục	22
Hình 3.3: Giao diện code.....	23
Hình 3.4: Giao diện nhập tên phim	24
Hình 3.5: Giao diện gợi ý phim.....	24
Hình 3.6: Giao diện kết quả tìm được	25

CHƯƠNG 1: GIỚI THIỆU ĐỀ TÀI

1.1. Trình bày vấn đề của đề tài

Trong thời đại số hóa hiện nay, các trang web xem phim trực tuyến đã trở thành một phần không thể thiếu trong đời sống giải trí của người dùng. Tuy nhiên, việc tìm kiếm và lựa chọn phim phù hợp với sở thích cá nhân lại trở nên thách thức do số lượng phim ngày càng gia tăng. Người dùng thường phải mất nhiều thời gian để khám phá các bộ phim mới mà họ có thể yêu thích. Điều này không chỉ gây ra sự phiền toái mà còn làm giảm trải nghiệm người dùng trên các trang web này. Vì vậy, một hệ thống đề xuất phim thông minh là rất cần thiết để giúp người dùng dễ dàng tìm thấy các bộ phim phù hợp với sở thích của mình.

1.2. Cách giải quyết của đề tài

Với sự phát triển mạnh mẽ của các thuật toán trí tuệ nhân tạo và máy học, đề tài lần này mà nhóm chúng em lựa chọn là phát triển một hệ thống đề xuất phim thông minh nhằm nâng cao trải nghiệm người dùng trên các trang web xem phim. Hệ thống sẽ sử dụng các thuật toán phân tích dữ liệu để tìm hiểu sở thích và hành vi của người dùng dựa trên các yếu tố như thể loại, nội dung, diễn viên và chủ đề của phim mà họ đã xem hoặc tìm kiếm. Từ đó, hệ thống sẽ tự động gợi ý các bộ phim liên quan mà người dùng có thể yêu thích.

Quá trình phát triển và triển khai hệ thống không chỉ giúp chúng em củng cố và vận dụng các kiến thức đã học từ giảng viên trong môi trường đại học mà còn là cơ hội để các thành viên trong nhóm rèn luyện, trao đổi, và bổ sung kiến thức cho nhau. Thông qua việc hợp tác và thảo luận với các thầy cô hướng dẫn, nhóm sẽ tiếp nhận những kinh nghiệm và kỹ năng quý báu từ những người đi trước và bạn bè xung quanh. Việc thực hiện đề tài này cũng là một phần quan trọng trong quá trình đánh giá năng lực của sinh viên và hoàn thành chương trình học. Nhóm chúng em mong muốn xây dựng được một hệ thống đề xuất phim hiệu quả, giúp người dùng dễ dàng khám phá và lựa chọn phim, từ đó nâng cao trải nghiệm và tương tác trên các trang web xem phim.

1.3. Đối tượng và phạm vi nghiên cứu:

- **Xây dựng một hệ thống đề xuất phim thông minh:** Hệ thống sẽ phân tích các yếu tố như thể loại, nội dung, diễn viên, và chủ đề của các bộ phim mà người dùng chọn hoặc tìm kiếm, từ đó gợi ý các bộ phim liên quan mà họ có thể yêu thích. Điều này giúp nâng cao trải nghiệm người dùng trên các trang web xem phim.

- **Python:** Ngôn ngữ lập trình chính được sử dụng để phát triển hệ thống, với các thư viện hỗ trợ như Pandas, Numpy, và Scikit-learn cho việc phân tích và xử lý dữ liệu.

- **Google Colaboratory:** Dùng làm môi trường để phát triển, thử nghiệm và huấn luyện các mô hình máy học và học sâu, nhờ vào khả năng tính toán mạnh mẽ và tích hợp dễ dàng với các thư viện phổ biến.

- **API của các trang web xem phim:** Sử dụng API để thu thập dữ liệu về các bộ phim, bao gồm thông tin về thể loại, nội dung, diễn viên, và chủ đề, cũng như dữ liệu về hành vi xem phim của người dùng.

- **Thuật toán và mô hình học máy:** Nghiên cứu về các thuật toán và mô hình học máy phù hợp để đề xuất phim, bao gồm cả sự hiểu biết về ưu điểm, nhược điểm, và cách áp dụng chúng trong bài toán cụ thể này.

- **Cây quyết định và Random Forest:** Được sử dụng rộng rãi trong việc đề xuất phim vì khả năng hiểu được mối quan hệ phi tuyến tính và xử lý tốt các biến số phức tạp.

- **Học sâu (Deep Learning):** Các mô hình như mạng nơ-ron hồi quy (RNN), mạng nơ-ron hồi quy dài hạn ngắn (LSTM), hoặc mạng nơ-ron tích chập (CNN) có thể được áp dụng để đề xuất phim với độ chính xác cao hơn.

CHƯƠNG 2: CƠ SỞ LÝ THUYẾT

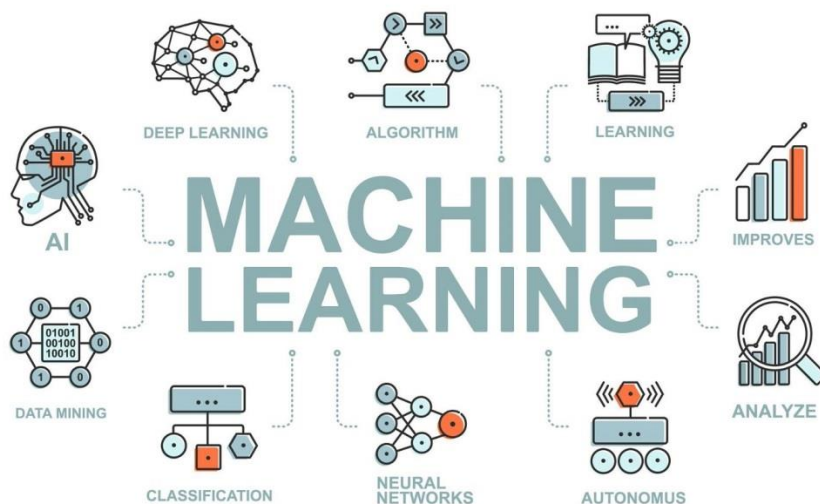
2.1. Sự hình thành và Phát triển của Machine Learning

Machine learning là một nhánh của trí tuệ nhân tạo (AI) và khoa học máy tính, tập trung vào việc sử dụng dữ liệu và thuật toán để bắt chước hành động của con người, dần dần cải thiện độ chính xác của nó.

Machine learning còn là một thành phần quan trọng của lĩnh vực khoa học dữ liệu đang phát triển. Thông qua việc sử dụng các phương pháp thống kê, các thuật toán được đào tạo để đưa ra các phân loại hoặc dự đoán và khám phá những thông tin chi tiết từ chính các dự án khai thác dữ liệu.

Thông qua các thông tin chi tiết có được để thúc đẩy việc đưa ra quyết định đối với các ứng dụng và doanh nghiệp, tác động mạnh đến các chỉ số tăng trưởng. Khi dữ liệu lớn tiếp tục nhu cầu mở rộng và phát triển đòi hỏi nhu cầu tuyển dụng các nhà khoa học dữ liệu sẽ tăng lên. Họ sẽ được yêu cầu giúp xác định các câu hỏi kinh doanh có liên quan nhất và dữ liệu để trả lời chúng.

Bài toán của machine learning thường được chia làm hai loại là dự đoán (prediction) và phân loại (classification). Các bài toán dự đoán thường là giá nhà, giá xe, v.v, còn các bài toán phân loại thường là nhận diện chữ viết tay, đồ vật, v.v.



Hình 2.1 : Machine Learning

2.1.1. Các phương pháp Machine Learning

- **Supervised Machine Learning:** Supervised learning còn được gọi là máy học có giám sát, được định nghĩa bằng cách sử dụng các tập dữ liệu được gắn nhãn để huấn luyện các thuật toán phân loại dữ liệu hoặc dự đoán kết quả một cách chính xác.

Khi dữ liệu đầu vào được đưa vào mô hình, mô hình sẽ điều chỉnh trọng lượng của nó cho đến khi nó được lắp một cách thích hợp. Điều này xảy ra như một phần của quá trình xác nhận chéo để đảm bảo rằng mô hình tránh trạng bị quá nhiều hoặc trạng bị thiếu thông tin. Supervised Machine Learning giúp các tổ chức giải quyết nhiều vấn đề trong thế giới thực trên quy mô lớn, chẳng hạn như phân loại thư rác trong một thư mục riêng biệt từ hộp thư đến của bạn.

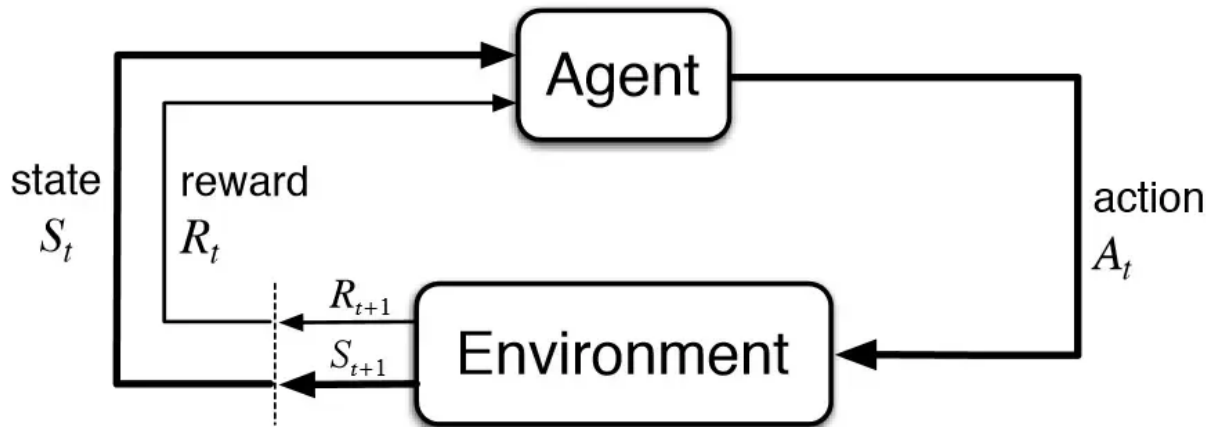
Một số phương pháp được sử dụng trong Supervised Machine Learning bao gồm: logistic regression, neural networks, linear regression, naive bayes, random forest, và support vector machine (SVM).

- **Unsupervised Machine Learning:** Unsupervised Machine Learning là phương pháp sử dụng các thuật toán máy học để phân tích và phân cụm các tập dữ liệu không được gắn nhãn. Không cần sự can thiệp của con người, các thuật toán này có thể phát hiện ra các mẫu hoặc nhóm dữ liệu ẩn. Khả năng phát hiện ra những điểm tương đồng và khác biệt trong thông tin của phương pháp này khiến nó trở nên lý tưởng cho việc phân tích dữ liệu khám phá, chiến lược bán chéo (cross-sell), phân khúc khách hàng cũng như nhận dạng hình ảnh và mẫu. Unsupervised machine learning cũng được sử dụng để giảm số lượng các tính năng trong một mô hình thông qua quá trình giảm kích thước. Phân tích thành phần chính (PCA) và phân tích giá trị đơn lẻ (SVD) là hai cách tiếp cận phổ biến cho việc này.

Các thuật toán khác được sử dụng trong học tập không giám sát bao gồm: k-means clustering, neural networks, và probabilistic clustering methods.

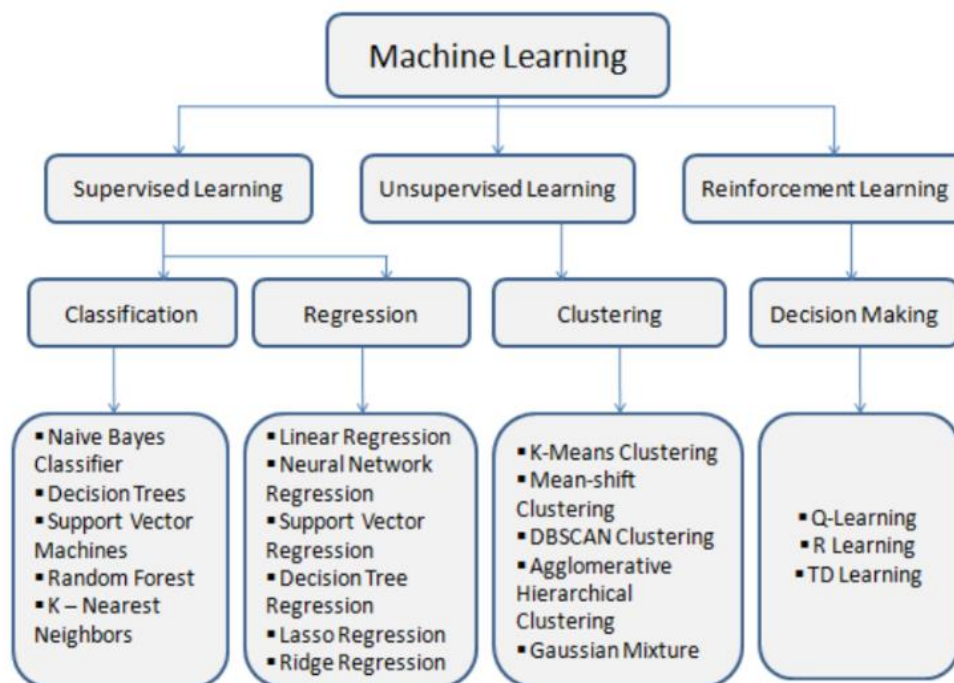
- **Semi-Supervised Learning:** Semi-supervised learning cung cấp một phương pháp hiệu quả giữa học tập có giám sát và không giám sát. Trong quá trình đào tạo, nó sử dụng một tập dữ liệu có nhãn nhỏ hơn để hướng dẫn phân loại và trích xuất tính năng từ một tập dữ liệu lớn hơn, không được gắn nhãn. Phương pháp Semi-supervised learning có thể giải quyết vấn đề không có đủ dữ liệu được gắn nhãn cho thuật toán học có giám sát. Nó cũng hữu ích nếu quá tốn kém để gắn nhãn đủ dữ liệu.

• **Reinforcement Learning:** là mô hình học máy cho phép một tác nhân (agent) học hỏi trong một môi trường tương tác bằng cách dùng phương pháp thử và lỗi thông qua việc sử dụng phản hồi từ các hành động và kinh nghiệm của chính nó.



Hình 2.2: Reinforcement Learning

Trong RL, tác nhân không được huấn luyện bằng dữ liệu có nhãn mà thay vào đó, nó học bằng cách liên tục thực hiện các hành động và nhận phản hồi dưới dạng phần thưởng hoặc hình phạt (rewards or punishments). Mục đích là tối đa hóa tổng số phần thưởng mà tác nhân thu được.



Hình 2.3: Các phương pháp Machine Learning

2.1.2. Các thuật toán phổ biến của Machine Learning.

- **Neural networks:** Mô phỏng cách thức hoạt động của bộ não con người, với một số lượng khổng lồ các nút xử lý được liên kết. Neural networks là thuật toán được dùng trong việc nhận dạng các mẫu và đóng một vai trò quan trọng trong các ứng dụng bao gồm dịch ngôn ngữ tự nhiên, nhận dạng hình ảnh, nhận dạng giọng nói và tạo hình ảnh.

- **Linear Regression:** Thuật toán này được sử dụng để dự đoán các giá trị số, dựa trên mối quan hệ tuyến tính giữa các giá trị khác nhau.

- **Logistic Regression:** Thuật toán giúp đưa ra dự đoán cho các biến phản hồi phân loại, chẳng hạn như câu trả lời “có/không” cho các câu hỏi. Nó có thể được sử dụng cho các ứng dụng như phân loại thư rác và kiểm soát chất lượng trên dây chuyền sản xuất.

- **Clustering:** Các thuật toán phân cụm có thể xác định các mẫu trong dữ liệu để nó có thể được nhóm lại. Máy tính có thể giúp các nhà khoa học dữ liệu bằng cách xác định sự khác biệt giữa các mục dữ liệu mà con người đã bỏ qua.

- **Decision Trees:** Là thuật toán được sử dụng để dự đoán giá trị số (hồi quy) và phân loại dữ liệu. Decision trees sử dụng một chuỗi phân nhánh của các quyết định được liên kết có thể được biểu diễn bằng sơ đồ cây. Một trong những ưu điểm của decision trees là chúng dễ xác thực và kiểm tra, không giống thuật toán Neural networks.

- **Random Forests:** Trong một khu rừng ngẫu nhiên, thuật toán máy học dự đoán một giá trị hoặc danh mục bằng cách kết hợp các kết quả từ một số cây quyết định.

2.1.3. Ứng dụng thực tế của Machine Learning

Một số ứng dụng thực tế của Learning Machine như:

- **Speech recognition:** Dùng để nhận dạng giọng nói tự động (ASR), nhận dạng giọng nói máy tính hoặc chuyển giọng nói thành văn bản. Đây là một khả năng sử dụng xử lý ngôn ngữ tự nhiên (NLP) để dịch giọng nói của con người sang định dạng viết.

- **Customer service:** Chatbots trực tuyến đang thay thế các tác nhân con người trong hành trình của khách hàng, thay đổi cách chúng ta nghĩ về sự tương tác của khách hàng trên website và nền tảng xã hội.

- **Computer vision:** Công nghệ AI này cho phép máy tính lấy thông tin có ý nghĩa từ video, hình ảnh kỹ thuật số và các đầu vào trực quan khác, sau đó thực thi hành động thích hợp. Recommendation engines: Sử dụng dữ liệu hành vi tiêu dùng trong quá khứ, các thuật toán AI

learning có thể giúp khám phá các xu hướng dữ liệu có thể được sử dụng để phát triển các chiến lược cross-sell hiệu quả hơn.

- **Automated stock trading:** Được thiết kế để tối ưu hóa danh mục đầu tư chứng khoán, các nền tảng giao dịch tần suất cao do AI điều khiển để hàng triệu giao dịch mỗi ngày mà không cần đến sự can thiệp của con người.

- **Fraud detection:** Các ngân hàng và các tổ chức tài chính có thể sử dụng máy học để phát hiện các giao dịch đáng ngờ.

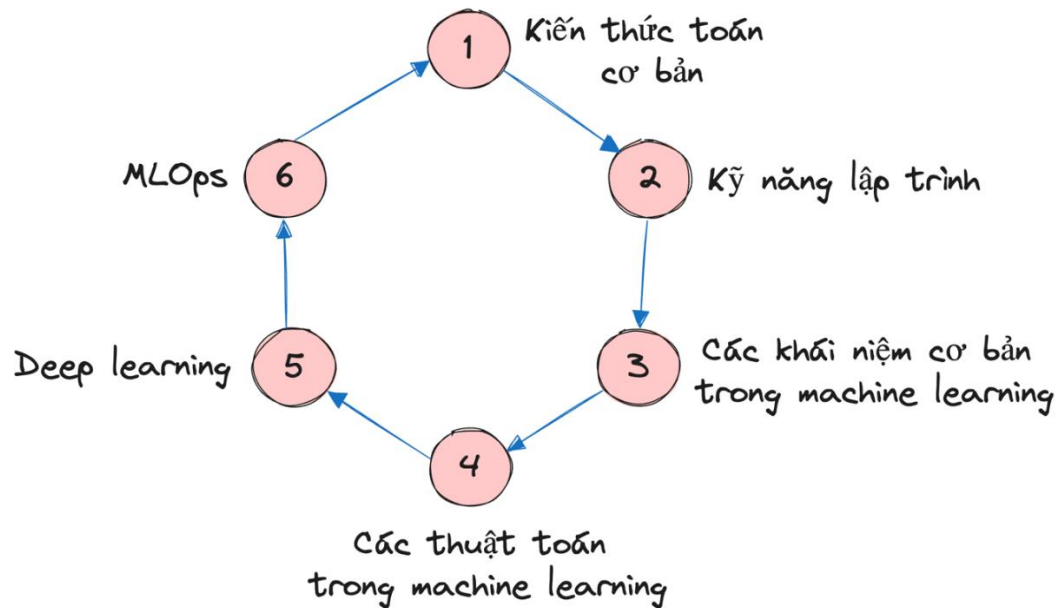
Và còn rất rất nhiều lĩnh vực mà Machine Learning có thể được áp dụng, machine learning tỏ ra cực kỳ hiệu quả, hơn hẳn con người trong cụ thể các lĩnh vực mà chúng được áp dụng.

Ví dụ đơn giản như dự báo thời tiết, người ta sẽ dùng các phép tính và những quan sát, ghi nhận về thời tiết trong quá khứ để dự báo về thời tiết của những ngày kế tiếp. Tuy nhiên sẽ thế nào nếu như có cực kỳ nhiều quan sát được thực hiện, có thể lên đến hàng triệu, hàng tỉ quan sát, lúc đó con người không thể nào thực hiện được việc tính toán trên dữ liệu lớn như vậy. Hơn nữa, việc tính toán với dữ liệu lớn như vậy có thể gặp sai sót và dẫn đến kết quả dự đoán bị sai.

Khi này, việc áp dụng Machine Learning vào để cho máy tính học các quan sát được ghi nhận trong quá khứ, chúng có thể dự đoán được thời tiết trong tương lai với độ chính xác cao hơn rất nhiều so với con người dự đoán.

Chính vì sự phổ biến và hiệu quả của Machine Learning, việc bạn biết và học về Machine Learning chắc chắn là một lợi thế lớn trong thời đại công nghệ 4.0 như ngày nay.

Machine learning Roadmap



Hình 2.4: Machine Learning Roadmap

2.1.4. Nguồn dữ liệu và chuẩn bị dữ liệu

- Phân tích dữ liệu:
 - Kết hợp dữ liệu từ nhiều nguồn và hợp lý hóa nó thành một tập dữ liệu.
 - Trực quan hóa dữ liệu để tìm kiếm xu hướng.
 - Sử dụng các công cụ và ngôn ngữ tập trung vào dữ liệu để tìm các mẫu trong dữ liệu.
 - Xác định các tính năng trong dữ liệu của bạn. Các tính năng bao gồm tập hợp con các thuộc tính dữ liệu mà bạn sử dụng trong mô hình của mình.
 - Làm sạch dữ liệu để tìm bất kỳ giá trị bất thường nào gây ra bởi lỗi trong quá trình nhập hoặc đo lường dữ liệu.
- Xử lý trước dữ liệu:
 - Chuẩn hóa dữ liệu số thành một tỷ lệ chung.
 - Áp dụng các quy tắc định dạng cho dữ liệu.
 - Giảm dư thừa dữ liệu thông qua đơn giản hóa.
 - Biểu diễn văn bản dưới dạng số.
 - Gán các giá trị chính cho các cá thể dữ liệu.

2.1.5. Sự khác biệt giữa Machine Learning với Deep Learning.

Machine Learning	Deep Learning
Machine Learning đòi hỏi nhiều hơn về sự can thiệp của con người để có được kết quả chính xác.	Deep Learning là tinh vi hơn để thiết lập nhưng đòi hỏi sự can thiệp của con người ít hơn nhiều sau đó.
Các chương trình machine learning có thuật toán ít phức tạp hơn deep learning và thường có thể chạy dễ dàng trên máy tính thông thường.	Deep learning thì đòi hỏi công cụ, tài nguyên và phần cứng phải mạnh hơn rất nhiều thì mới hoạt động được.
Hệ thống Machine Learning có thể được đưa lên và chạy nhanh, nhưng sức mạnh của chúng có thể bị hạn chế.	Các hệ thống Deep Learning đòi hỏi nhiều nỗ lực hơn để thiết lập nhưng có thể tạo ra kết quả ngay lập tức.
Machine Learning thường đòi hỏi dữ liệu có tổ chức và sử dụng các kỹ thuật cổ điển như hồi quy tuyến tính.	Deep Learning sử dụng các mạng lưới thần kinh và được thiết kế để xử lý một lượng lớn dữ liệu không có cấu trúc.
Machine learning đã được sử dụng ở những nơi như hộp thư đến email, ngân hàng và văn phòng bác sĩ của bạn.	Công nghệ Deep Learning cho phép các thuật toán ngày càng phức tạp và tự trị, chẳng hạn như ô tô tự lái hoặc robot phẫu thuật.

2.1.6. Tổng kết

- Machine learning là một nhánh của trí tuệ nhân tạo, nơi các máy tính được lập trình để học hỏi từ dữ liệu và cải thiện hiệu suất của mình một cách tự động mà không cần can thiệp trực tiếp từ con người.

- Có 4 mô hình machine learning phổ biến là supervised learning, unsupervised learning, semi-supervised learning và reinforcement learning.

- Để học machine learning hiệu quả, bạn nên bắt đầu từ kiến thức toán cơ bản, kỹ năng lập trình, các khái niệm cơ bản trong machine learning, các thuật toán trong machine learning, deep learning và MLOps.

2.2. Tính toán và ứng dụng Cosine Similarity

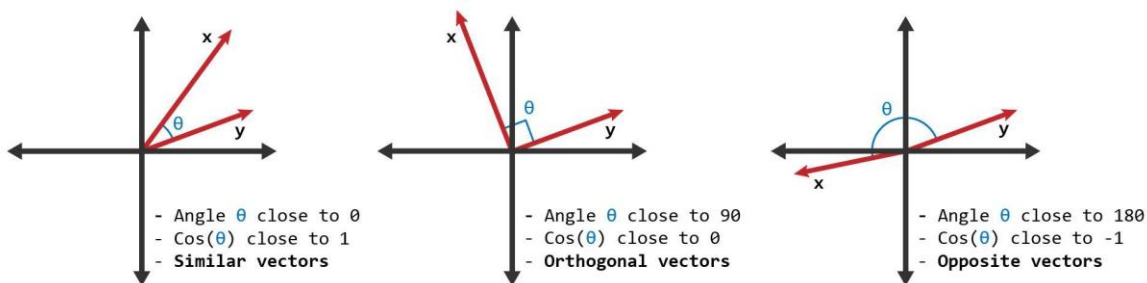
2.2.1. Tính toán Cosine Similarity

Cosine Similarity là một phương pháp đo lường độ tương tự giữa hai vector. Trong hệ thống đề xuất phim, Cosine Similarity được sử dụng để đo lường mức độ tương đồng giữa các phim dựa trên các đặc điểm như thể loại, diễn viên, và đạo diễn.

Cosine Similarity đo lường sự giống nhau về hướng giữa hai vector, bỏ qua sự khác biệt về độ lớn của chúng. Giá trị cosine của góc giữa hai vector cho biết mức độ tương tự giữa chúng:

- Khi góc giữa hai vector là 0, Cosine Similarity là 1 (tương tự hoàn toàn).
- Khi góc giữa hai vector là 90 độ, Cosine Similarity là 0 (không tương tự).
- Khi góc giữa hai vector là 180 độ, Cosine Similarity là -1 (hoàn toàn ngược hướng).

Đây là đồ họa hiển thị hai vector có điểm tương đồng gần 1, gần 0 và gần -1.



Hình 2.5: Sơ đồ vectơ

Độ tương tự cosine giữa hai vector A và B được tính bằng công thức:

$$\text{similarity}(A, B) = \cos(\theta) = \frac{A \cdot B}{\|A\| \|B\|}$$

where

- θ is the angle between the vectors,
- $A \cdot B$ is dot product between A and B and calculated as
 $A \cdot B = A^T B = \sum_{i=1}^n A_i B_i = A_1 B_1 + A_2 B_2 + \dots + A_n B_n,$
- $\|A\|$ represents the L2 norm or magnitude of the vector which is calculated as
 $\|A\| = \sqrt{A_1^2 + A_2^2 + \dots + A_n^2}.$

Ví dụ về tính độ tương tự cosine trong Python:

```
import numpy as np

def cosine_similarity(x, y):

    # Ensure length of x and y are the same
    if len(x) != len(y) :
        | return None

    # Compute the dot product between x and y
    dot_product = np.dot(x, y)

    # Compute the L2 norms (magnitudes) of x and y
    magnitude_x = np.sqrt(np.sum(x**2))
    magnitude_y = np.sqrt(np.sum(y**2))

    # Compute the cosine similarity
    cosine_similarity = dot_product / (magnitude_x * magnitude_y)

    return cosine_similarity
```


Cosine Similarity sẽ được sử dụng để tìm ra sự tương tự giữa hai tài liệu sau:

```
corpus = [ 'data science is one of the most important fields of science',  
           | 'this is one of the best data science courses',  
           | 'data scientists analyze data' ]
```

Sử dụng *sklearn*, chúng ta sẽ vector hóa các tài liệu:

```
from sklearn.feature_extraction.text import CountVectorizer  
  
# Create a matrix to represent the corpus  
X = CountVectorizer().fit_transform(corpus).toarray()  
  
print(X)  
  
[[0 0 0 1 1 1 1 1 2 1 2 0 1 0]  
 [0 1 1 1 0 0 1 0 1 1 1 0 1 1]  
 [1 0 0 2 0 0 0 0 0 0 0 0 1 0 0]]
```

Với các vector trên, bây giờ chúng ta có thể tính Cosine Similarity giữa các tài liệu kho
văn bản:

```
cos_sim_1_2 = cosine_similarity(X[0, :], X[1, :])  
cos_sim_1_3 = cosine_similarity(X[0, :], X[2, :])  
cos_sim_2_3 = cosine_similarity(X[1, :], X[2, :])  
  
print('Cosine Similarity between: ')  
print('\tDocument 1 and Document 2: ', cos_sim_1_2)  
print('\tDocument 1 and Document 3: ', cos_sim_1_3)  
print('\tDocument 2 and Document 3: ', cos_sim_2_3)  
  
Cosine Similarity between:  
Document 1 and Document 2: 0.6885303726590962  
Document 1 and Document 3: 0.21081851067789195  
Document 2 and Document 3: 0.2721655269759087
```

Đây là một ví dụ về việc sử dụng *sklearn* chức năng:

```
from sklearn.metrics.pairwise import cosine_similarity

cos_sim_1_2 = cosine_similarity([X[0, :], X[1, :]])

print('Cosine Similarity between Document 1 and Document 2 is \n',cos_sim_1_2 )

Cosine Similarity between Document 1 and Document 2 is
[[1.          0.68853037]
 [0.68853037 1.          ]]
```

2.2.2. Ứng dụng Cosine Similarity trong hệ thống đề xuất phim

Cosine Similarity là một phương pháp phổ biến trong các hệ thống đề xuất phim, đặc biệt là trong các hệ thống đề xuất dựa trên nội dung (Content-Based Recommendation Systems), hệ thống sẽ so sánh các đặc điểm của phim và đưa ra gợi ý cho người dùng. Ví dụ, nếu người dùng đã xem và đánh giá cao một phim hành động với diễn viên cụ thể, hệ thống sẽ gợi ý các phim tương tự dựa trên đặc điểm đó. Dưới đây là các bước và cách thức ứng dụng Cosine Similarity trong hệ thống đề xuất phim:

Bước 1: Tạo Vector Đặc Trưng cho mỗi phim

Mỗi bộ phim được biểu diễn dưới dạng một vector đặc trưng. Các đặc trưng này có thể bao gồm:

- Thể loại phim (ví dụ: hành động, hài, lãng mạn)
- Đạo diễn
- Diễn viên chính
- Từ khóa mô tả nội dung phim
- Tóm tắt nội dung

Bước 2: Tính Toán Cosine Similarity

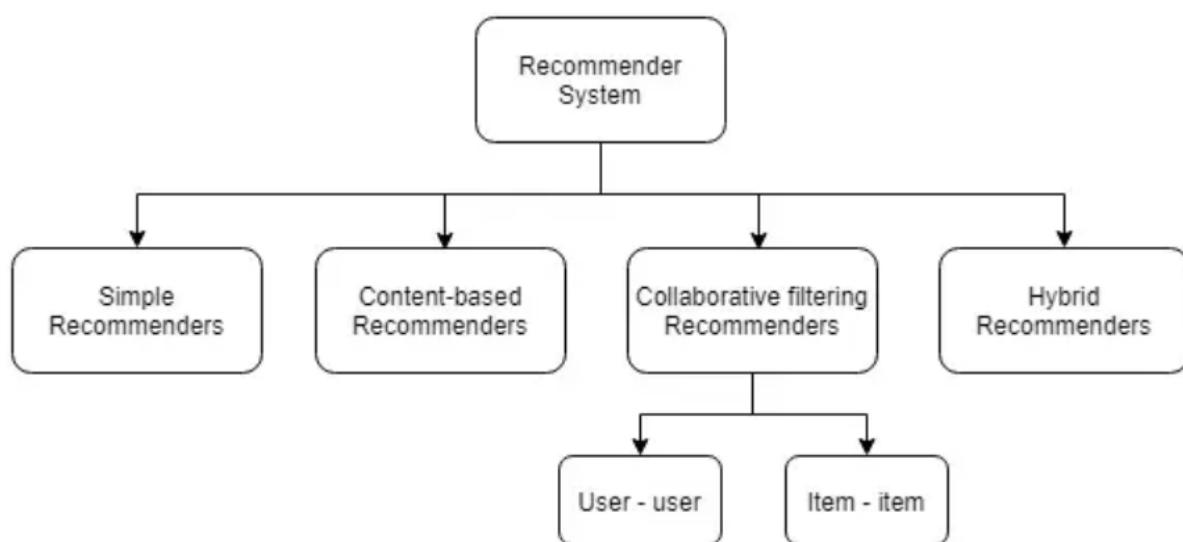
Độ tương tự Cosine đo lường sự giống nhau giữa hai vector bằng cách tính cosin của góc giữa chúng. Công thức tính toán Cosine Similarity giữa hai vector A và B đã được trình bày như trên phần **3.4.1.**

Bước 3: Đề Xuất Phim Dựa Trên Cosine Similarity

Dựa trên Cosine Similarity đã tính toán, hệ thống sẽ tìm ra các bộ phim có độ tương tự cao với phim mà người dùng đã xem hoặc yêu thích. Các phim có Cosine Similarity cao sẽ được đề xuất cho người dùng, giúp họ khám phá thêm các bộ phim mới có nội dung tương đồng.

2.3. Recommendation System là gì?

Recommendation System là một chương trình lọc có mục tiêu chính là dự đoán "xếp hạng" hoặc "sở thích" của người dùng đối với một mục hoặc sản phẩm cụ thể theo miền. Trong trường hợp của chúng tôi, mục cụ thể theo miền này là một bộ phim. Do đó, trọng tâm chính của hệ thống đề xuất của chúng tôi là lọc và dự đoán chỉ những bộ phim mà người dùng thích, với một số dữ liệu về người dùng.



Hình 2.6: Recommendation System

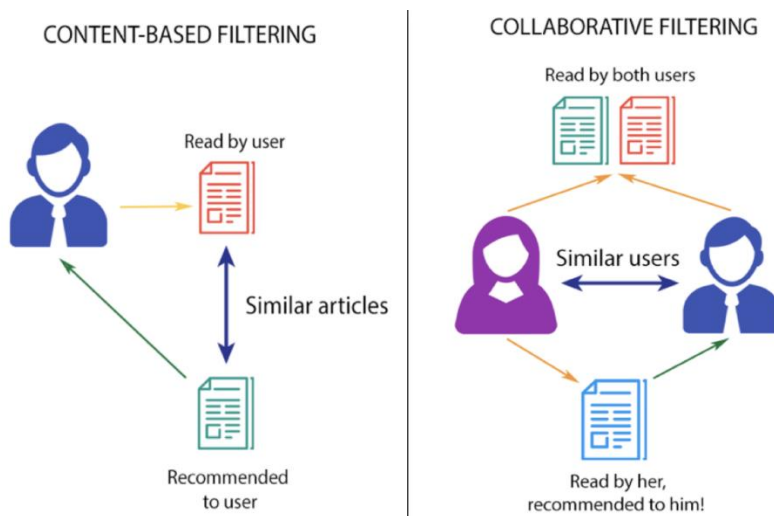
2.3.1. Tại sao lại chọn Recommendation Systems

- Hệ thống đề xuất cung cấp các gợi ý được cá nhân hóa dựa trên sở thích và xếp hạng của người dùng, đảm bảo rằng họ khám phá ra nội dung và sản phẩm có liên quan và thú vị với họ.
- Bằng cách cung cấp các đề xuất phù hợp, người dùng có nhiều khả năng tương tác với nền tảng hơn, tăng sự hài lòng và giữ chân người dùng.

- Các nền tảng thương mại điện tử như Netflix sử dụng công cụ đề xuất để quảng bá sản phẩm, giúp tăng doanh số và doanh thu khi người dùng khám phá và mua các mặt hàng mà họ có thể chưa từng cân nhắc.
- Trong bối cảnh kỹ thuật số ngày nay, hệ thống đề xuất giúp người dùng điều hướng giữa vô vàn nội dung có sẵn, giúp họ dễ dàng tìm thấy bộ phim hoặc loạt phim cụ thể mà họ muốn hoặc bằng bất kỳ ngôn ngữ nào họ muốn, như tiếng Anh hoặc tiếng Hàn.
- Thuật toán đề xuất giúp người dùng tiếp cận nội dung mới và đa dạng, mở rộng tầm nhìn và giới thiệu cho họ những mục mà họ có thể đã bỏ qua.
- Dựa trên hành vi và sở thích trong quá khứ, hệ thống đề xuất giúp người dùng đưa ra quyết định sáng suốt về những lựa chọn phức tạp và chủ quan như phim ảnh, âm nhạc hoặc sách.

2.3.2. Các loại Recommendation Systems

Có 2 loại Recommendation Systems-Content-based filtering và Collaborative Filtering. Chúng ta hãy thảo luận về chúng ngay bây giờ.



Hình 2.7: Content-based và Collaborative Filtering



Content-based Filtering

Là một chiến lược đề xuất gợi ý các mục tương tự như những mục mà người dùng đã thích trước đó. Nó tính toán mức độ tương đồng (thường sử dụng độ tương đồng cosin) giữa sở thích của người dùng và các thuộc tính mục, chẳng hạn như diễn viên chính, đạo diễn và thể loại. Ví dụ, nếu người dùng thích 'The Prestige', hệ thống sẽ đề xuất các bộ phim có Christian Bale, thể loại 'Thriller' hoặc phim của Christopher Nolan.

Tuy nhiên, lọc dựa trên nội dung có nhược điểm. Nó hạn chế việc tiếp xúc với các sản phẩm khác nhau, ngăn người dùng khám phá nhiều mặt hàng khác nhau. Điều này có thể cản trở việc mở rộng kinh doanh vì người dùng có thể không thử các loại sản phẩm mới.



Collaborative Filtering

Collaborative filtering là một chiến lược đề xuất xem xét hành vi của người dùng và so sánh với những người dùng khác trong cơ sở dữ liệu. Nó sử dụng lịch sử của tất cả người dùng để tác động đến thuật toán đề xuất. Không giống như hệ thống đề xuất dựa trên nội dung, bộ đề xuất lọc cộng tác dựa vào tương tác của nhiều người dùng với các mục để tạo ra các đề xuất. Nó không chỉ phụ thuộc vào dữ liệu của một người dùng để lập mô hình. Có nhiều cách tiếp cận khác nhau để triển khai lọc cộng tác, nhưng khái niệm cơ bản là ảnh hưởng chung của nhiều người dùng đến kết quả đề xuất.

➤ Có 2 loại thuật toán lọc cộng tác:

Collaborative Filtering dựa trên người dùng nhằm mục đích đề xuất các mục cho người dùng (A) dựa trên sở thích của những người dùng tương tự trong cơ sở dữ liệu. Nó bao gồm việc tạo ra một ma trận các mục được đánh giá/thích/nhấp bởi mỗi người dùng, tính toán điểm tương đồng giữa những người dùng, sau đó đề xuất các mục mà người dùng A chưa gặp nhưng những người dùng tương tự đã thích.

Ví dụ, nếu người dùng A thích những bộ phim cụ thể như Batman Begins, Justice League và The Avengers và người dùng B thích Thor, họ có chung sở thích về thể loại siêu anh hùng. Do đó, rất có thể người dùng A sẽ thích một bộ phim nổi tiếng như Thor và người dùng B sẽ thích The Avengers, một bộ phim có xếp hạng phim cao nhất.

➤ Nhược Điểm

Lọc cộng tác dựa trên người dùng có một số nhược điểm:

- **Sở thích người dùng thất thường:** Sở thích người dùng có thể thay đổi theo thời gian, khiến các mẫu tương đồng ban đầu giữa những người dùng trở nên lỗi thời. Điều này có thể dẫn đến các đề xuất không chính xác khi sở thích của người dùng thay đổi và họ bắt đầu xem phim mới.

- **Ma trận lớn:** Vì số lượng người dùng thường lớn hơn nhiều so với số lượng mục, nên việc duy trì ma trận lớn trở nên khó khăn và tốn nhiều tài nguyên. Cần phải tính toán lại thường xuyên để dữ liệu luôn được cập nhật.
- **Dễ bị tấn công Shilling:** Tấn công Shilling liên quan đến việc tạo hồ sơ người dùng giả với các mẫu sở thích thiên vị để thao túng hệ thống đề xuất. Lọc cộng tác dựa trên người dùng dễ bị tấn công như vậy, có khả năng dẫn đến các đề xuất thiên vị và bị thao túng.

Item-based Collaborative Filtering

Lọc cộng tác dựa trên mục tập trung vào việc tìm kiếm những bộ phim tương tự thay vì những người dùng tương tự để đề xuất cho người dùng 'A' dựa trên sở thích trước đây của họ. Nó xác định các cặp phim được cùng người dùng đánh giá/thích, đo lường mức độ tương đồng của chúng trên tất cả người dùng đã đánh giá cả hai, sau đó đề xuất những bộ phim tương tự dựa trên điểm tương đồng.

Ví dụ, khi so sánh phim 'A' và 'B', chúng tôi phân tích xếp hạng từ những người dùng đã xếp hạng cả hai phim. Nếu những xếp hạng này cho thấy sự tương đồng cao, điều đó chỉ ra rằng 'A' và 'B' là những bộ phim tương tự nhau. Do đó, nếu ai đó thích 'A', họ nên được giới thiệu 'B', và ngược lại.

➤ Ưu Điểm

Lọc bao gồm các sở thích phim ổn định, vì các bộ phim phổ biến không làm thay đổi sở thích của mọi người. Việc duy trì và tính toán ma trận dễ tiếp cận hơn vì thường có ít mục hơn người dùng. Các cuộc tấn công shilling cũng khó khăn hơn vì các mục không thể bị làm giả, khiến cách tiếp cận này trở nên mạnh mẽ hơn.

CHƯƠNG 3: ỨNG DỤNG TRIỂN KHAI HỆ THỐNG ĐỀ XUẤT PHIM

3.1. Ngôn ngữ và công cụ

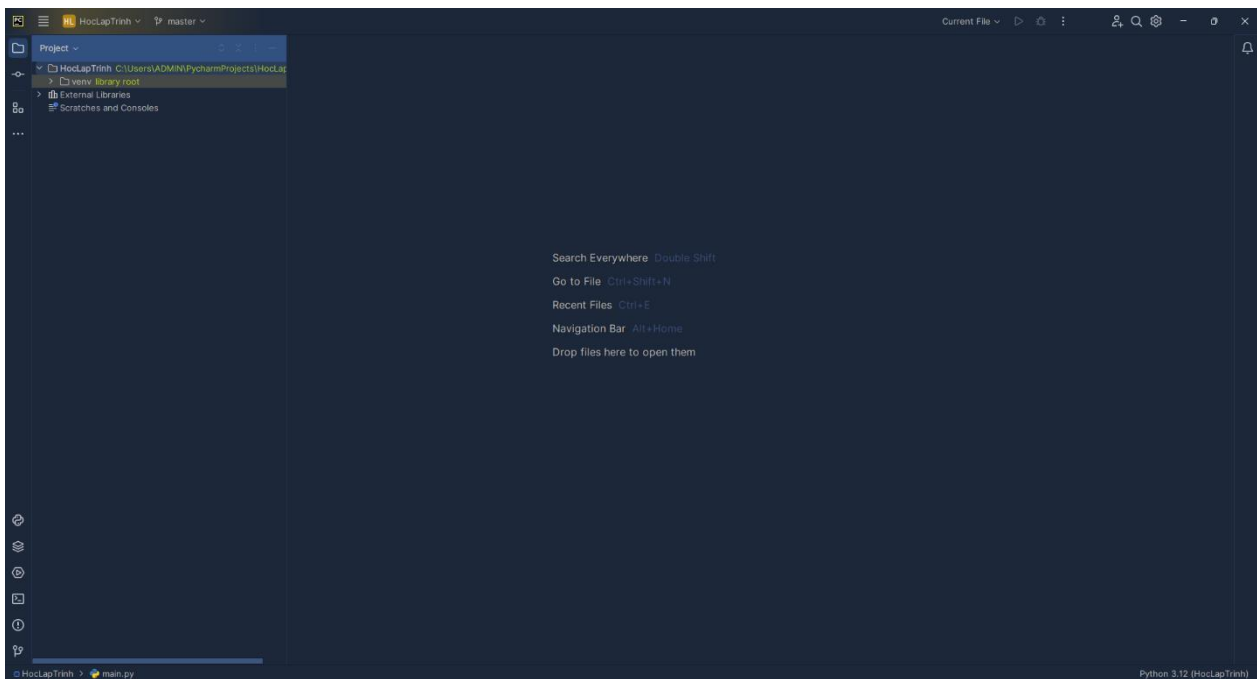
3.1.1. Ngôn ngữ

Sử dụng Python là ngôn ngữ chính để xây dựng hệ thống đề xuất phim, nhờ vào sự linh hoạt và sức mạnh của nó trong việc xử lý dữ liệu và phát triển các mô hình máy học.

3.1.2. Công cụ

- PyCharm là một môi trường phát triển tích hợp (IDE) mạnh mẽ và linh hoạt, được thiết kế đặc biệt cho việc phát triển ứng dụng Python. Được phát triển bởi JetBrains, PyCharm cung cấp một loạt các tính năng và công cụ hỗ trợ toàn diện để giúp nhà phát triển Python tăng năng suất và hiệu suất trong quá trình lập trình. Tạo môi trường ảo để phân tách các dự án và quản lý các thư viện phụ thuộc. Điều này giúp tránh xung đột giữa các phiên bản thư viện và làm cho dự án của bạn dễ quản lý hơn.

- Google Colaboratory: Một nền tảng trực tuyến cho phép chạy mã Python trên trình duyệt với khả năng sử dụng tài nguyên máy tính mạnh mẽ, đặc biệt hữu ích cho việc huấn luyện



Hình 3.1: Giao diện Pycharm

3.2. Cài đặt các thư viện cần thiết

- Sử dụng trình quản lý gói PIP để cài đặt các thư viện cần thiết như pandas, numpy, scikit-learn và các thư viện khác.

```
Terminal Local x + v
PS C:\Users\Xuan Vinh\Downloads\movie-recommendation-main\movie-recommendation-main> pip install pandas numpy scikit-learn
Defaulting to user installation because normal site-packages is not writeable
Requirement already satisfied: pandas in c:\users\xuan vinh\appdata\local\packages\pythonsoftwarefoundation.python.3.12_qbz5n2kfra8p0\localcache\local-packages\python312\site-packages (2.2.2)
Requirement already satisfied: numpy in c:\users\xuan vinh\appdata\local\packages\pythonsoftwarefoundation.python.3.12_qbz5n2kfra8p0\localcache\local-packages\python312\site-packages (1.26.4)
Collecting scikit-learn
  Downloading scikit_learn-1.5.1-cp312-cp312-win_amd64.whl.metadata (12 kB)
Requirement already satisfied: python-dateutil>=2.8.2 in c:\users\xuan vinh\appdata\local\packages\pythonsoftwarefoundation.python.3.12_qbz5n2kfra8p0\localcache\local-packages\python312\si
```

- Cài đặt thư viện xử lý ngôn ngữ tự nhiên:

```
Terminal Local x + v
PS C:\Users\Xuan Vinh\Downloads\movie-recommendation-main\movie-recommendation-main> pip install nltk
Defaulting to user installation because normal site-packages is not writeable
Collecting nltk
  Downloading nltk-3.8.1-py3-none-any.whl.metadata (2.8 kB)
Requirement already satisfied: click in c:\users\xuan vinh\appdata\local\packages\pythonsoftwarefoundation.python.3.12_qbz5n2kfra8p0\localcache\local-packages\python312\site-packages (from nltk) (8.1.7)
Requirement already satisfied: joblib in c:\users\xuan vinh\appdata\local\packages\pythonsoftwarefoundation.python.3.12_qbz5n2kfra8p0\localcache\local-packages\python312\site-packages (from nltk) (1.4.2)
Collecting regex>=2021.8.3 (from nltk)
  Downloading regex-2024.7.24-cp312-cp312-win_amd64.whl.metadata (41 kB)
```

3.3. Xử lý dữ liệu trên Google Colaboratory

3.3.1. Cấu trúc dữ liệu và tiền xử lý

- Đọc dữ liệu phim và diễn viên từ các tệp CSV:

```
import numpy as np
import pandas as pd

[ ] movies = pd.read_csv('data/tmdb_5000_movies.csv')
credits = pd.read_csv('data/tmdb_5000_credits.csv')
```

- Kết hợp 2 bộ dữ liệu trên tiêu đề phim:

```
[ ] movies = movies.merge(credits,on='title')
```

- Giữ lại các cột quan trọng cho hệ thống đề xuất:

```
# Keeping important columns for recommendation
movies = movies[['movie_id','title','overview','genres','keywords','cast','crew']]
```


3.3.2. Tiền xử lý dữ liệu

- Xử lý các cột genres, keywords, cast và crew để chuyển đổi chúng từ định dạng chuỗi sang danh sách:

```
[ ] import ast #for converting str to list

def convert(text):
    L = []
    for i in ast.literal_eval(text):
        L.append(i['name'])
    return L

[ ] movies['genres'] = movies['genres'].apply(convert)
```

```
▶ movies['keywords'] = movies['keywords'].apply(convert)
movies.head()
```

- Giữ lại ba diễn viên hàng đầu từ cột cast:

```
[ ] # Here i am just keeping top 3 cast

def convert_cast(text):
    L = []
    counter = 0
    for i in ast.literal_eval(text):
        if counter < 3:
            L.append(i['name'])
            counter+=1
    return L

[ ] movies['cast'] = movies['cast'].apply(convert_cast)
movies.head()
```

3.3.3. Tạo thẻ cho từng phim

- Kết hợp các cột đã xử lý để tạo ra một thẻ duy nhất cho từng phim:

```
[ ] # Keeping important columns for recommendation
movies = movies[['movie_id', 'title', 'overview', 'genres', 'keywords', 'cast', 'crew']]
```

- Chuyển đổi các danh sách trong thẻ thành chuỗi:

```
[ ] # dropping those extra columns
new_df = movies[['movie_id', 'title', 'tags']]
```

```
[ ] # Converting list to str
new_df['tags'] = new_df['tags'].apply(lambda x: " ".join(x))
new_df.head()
```

3.3.4. Xây dựng mô hình đề xuất

- Chuyển đổi các thẻ thành dạng văn bản và thực hiện stemming:

```
[ ] import nltk
    from nltk.stem import PorterStemmer

[ ] ps = PorterStemmer()

[ ] def stems(text):
    T = []

    for i in text.split():
        T.append(ps.stem(i))

    return " ".join(T)

[ ] new_df['tags'] = new_df['tags'].apply(stems)
```

- Tạo ma trận đếm từ và tính toán độ tương đồng cosine giữa các phim:

```
[ ] from sklearn.feature_extraction.text import CountVectorizer
    cv = CountVectorizer(max_features=5000, stop_words='english')
```

```
[ ] from sklearn.metrics.pairwise import cosine_similarity
```

```
[ ] vector = cv.fit_transform(new_df['tags']).toarray()
```

```
[ ] similarity = cosine_similarity(vector)
```

3.3.5. Hàm gợi ý phim

Viết hàm để gợi ý phim dựa trên độ tương đồng:

```
[ ] def recommend(movie):  
    index = new_df[new_df['title'] == movie].index[0]  
    distances = sorted(list(enumerate(similarity[index])),reverse=True,key = lambda x: x[1])  
    for i in distances[1:6]:  
        print(new_df.iloc[i[0]].title)
```

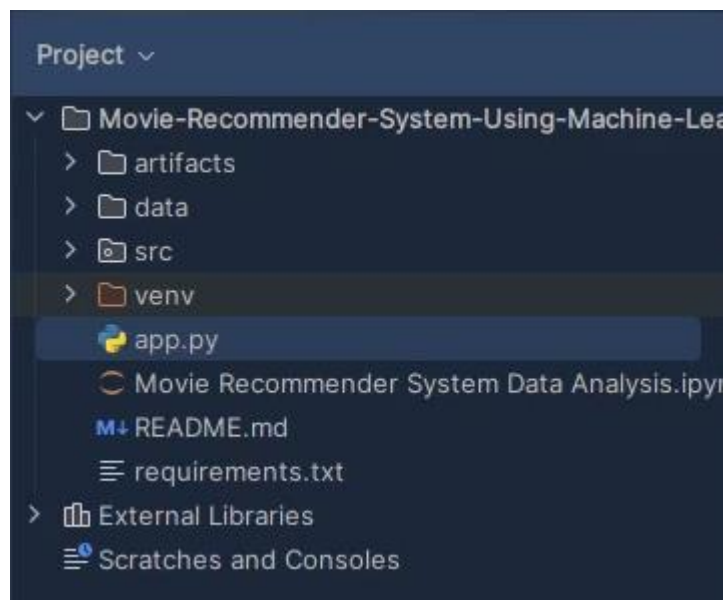
3.3.6. Lưu trữ mô hình và dữ liệu

Sử dụng pickle để lưu trữ mô hình và dữ liệu đã xử lý:

```
[ ] import pickle  
  
[ ] pickle.dump(new_df,open('artifacts/movie_list.pkl','wb'))  
    pickle.dump(similarity,open('artifacts/similarity.pkl','wb'))
```

3.4. Thiết kế hệ thống và demo các bước chạy chương trình

3.4.1. Cấu trúc thư mục



Hình 3.2: Cấu trúc thư mục

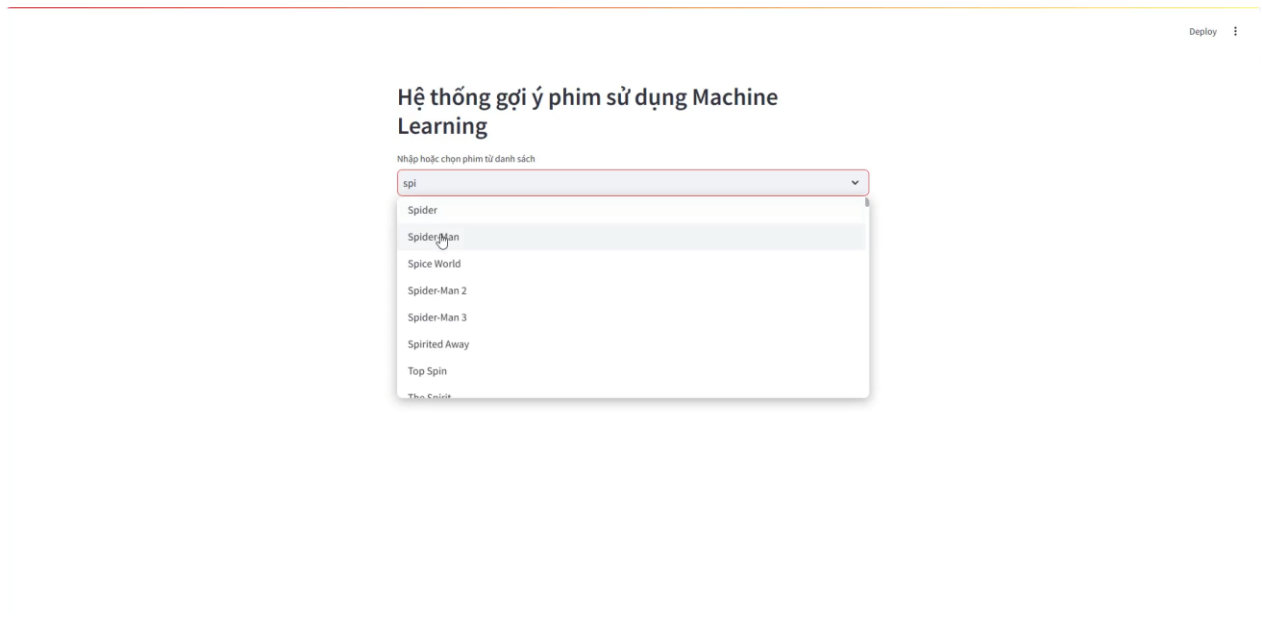
3.4.2. Giao diện Code

```
2 import streamlit as st
3 import requests
4 import pandas as pd
5
6 !usage
7 def fetch_poster(movie_id):
8     url = f"https://api.themoviedb.org/3/movie/{movie_id}?api_key=8265bd1679663a7ea12ac168da84d2e8&language=en-US"
9     data = requests.get(url).json()
10    poster_path = data.get('poster_path')
11    if poster_path:
12        return "https://image.tmdb.org/t/p/w500/" + poster_path
13    return None
14
15 !usage
16 def recommend(movie):
17     index = movies[movies['title'] == movie].index[0]
18     distances = sorted(list(enumerate(similarity[index])), reverse=True, key=lambda x: x[1])
19     recommended_movie_names = []
20     recommended_movie_posters = []
21     for i in distances[1:6]:
22         # Fetch the movie poster
23         movie_id = movies.iloc[i[0]].movie_id
24         poster = fetch_poster(movie_id)
25         if poster:
```

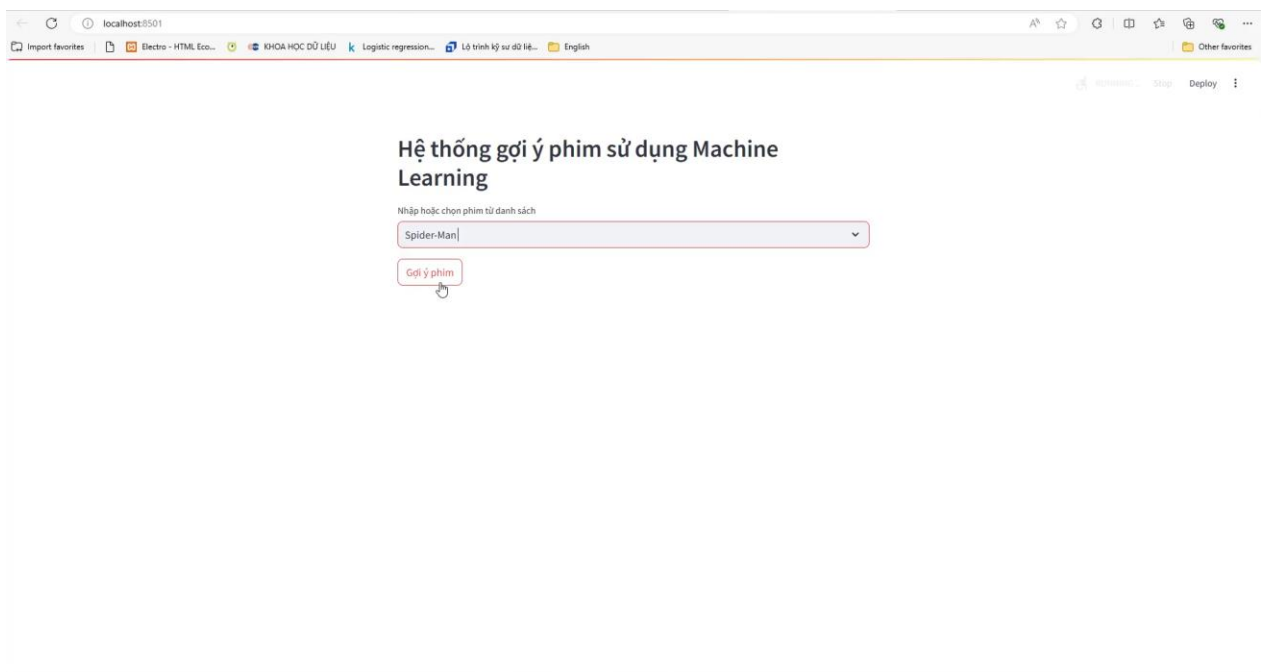
```
26 recommended_movie_posters.append(poster)
27 recommended_movie_names.append(movies.iloc[i[0]].title)
28
29 return recommended_movie_names, recommended_movie_posters
30
31 st.header('Hệ thống gợi ý phim sử dụng Machine Learning')
32
33 # Load data
34 with open('artifacts/movie_list.pkl', 'rb') as f:
35     movies = pickle.load(f)
36
37 with open('artifacts/similarity.pkl', 'rb') as f:
38     similarity = pickle.load(f)
39
40 movie_list = movies['title'].values
41 selected_movie = st.selectbox("Nhập hoặc chọn phim từ danh sách", movie_list)
42
43 if st.button('Gợi ý phim'):
44     recommended_movie_names, recommended_movie_posters = recommend(selected_movie)
45     cols = st.columns(5)
46     for i, col in enumerate(cols):
47         if i < len(recommended_movie_names):
48             with col:
49                 st.text(recommended_movie_names[i])
50                 st.image(recommended_movie_posters[i])
```

Hình 3.3: Giao diện code

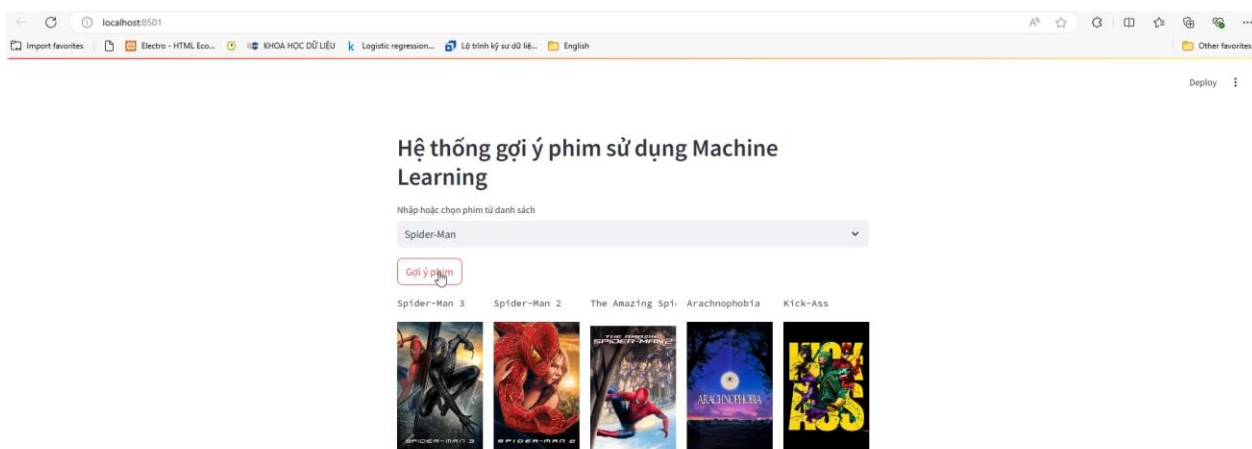
3.4.3. Giao diện Web Application



Hình 3.4: Giao diện nhập tên phim



Hình 3.5: Giao diện gợi ý phim



Hình 3.6: Giao diện kết quả tìm được

CHƯƠNG 4: TỔNG KẾT

4.1. Kết quả đạt được

- Nhóm đã xây dựng được hệ thống đề xuất phim thông minh với khả năng phân tích dữ liệu về sở thích và hành vi của người dùng để gợi ý các bộ phim liên quan một cách chính xác. Hệ thống này hoạt động với tốc độ nhanh và độ chính xác cao, giúp người dùng dễ dàng khám phá các bộ phim mới.

- Tối ưu hóa mô hình và hệ thống để có thể hoạt động hiệu quả với các dữ liệu đầu vào khác nhau từ các trang web xem phim. Hệ thống có khả năng xử lý thông tin từ nhiều nguồn khác nhau và đưa ra các gợi ý phim phù hợp.

- Sử dụng môi trường Google Colaboratory để huấn luyện và kiểm tra các mô hình máy học và học sâu. Nhờ vào GPU miễn phí của Google Colaboratory, hệ thống đã được kiểm tra thành công với dữ liệu lớn và đa dạng.

- Tạo ra một hệ thống dễ dàng sử dụng cho người dùng ở mọi lứa tuổi. Giao diện và cách thức hoạt động của hệ thống được thiết kế đơn giản và trực quan, phù hợp với nhu cầu của người dùng.

4.2. Ưu điểm

- Hệ thống đề xuất chính xác: Hệ thống có khả năng phân tích và hiểu rõ sở thích của người dùng, từ đó gợi ý các bộ phim liên quan một cách chính xác và hiệu quả.

- Tốc độ xử lý và gửi gợi ý nhanh: Hệ thống hoạt động với tốc độ nhanh, giúp người dùng nhận được các gợi ý phim ngay lập tức sau khi thực hiện tìm kiếm hoặc chọn bộ phim.

- Dễ dàng sử dụng cho mọi người: Giao diện của hệ thống được thiết kế đơn giản và dễ hiểu, cho phép người dùng dễ dàng sử dụng và tương tác với hệ thống.

4.3. Điểm hạn chế

- Hệ thống phụ thuộc vào dữ liệu nhập vào rõ ràng nhất định: Hệ thống có thể gặp khó khăn trong việc phân tích và gợi ý chính xác nếu dữ liệu đầu vào không đầy đủ hoặc không rõ ràng. Các bộ phim có thông tin mơ hồ hoặc không đủ chi tiết có thể làm giảm độ chính xác của gợi ý.

- Tính khả thi của các thuật toán: Một số thuật toán gợi ý phim có thể cần tinh chỉnh thêm để cải thiện độ chính xác, đặc biệt là khi xử lý dữ liệu đa dạng và không đồng nhất.

- Tích hợp với các nền tảng khác: Hiện tại, hệ thống chủ yếu tích hợp với các trang web xem phim, và có thể không hoàn toàn tương thích với các nền tảng khác hoặc các dịch vụ stream phim khác.

4.4. Hướng phát triển của đề tài

- Ứng dụng mở rộng: Hướng phát triển tiếp theo có thể bao gồm việc mở rộng hệ thống để áp dụng cho các lĩnh vực khác như đề xuất nội dung cho các nền tảng học tập trực tuyến, hay các dịch vụ phát trực tiếp khác.

- Tích hợp với các nền tảng khác: Để tiếp cận được nhiều người dùng hơn, nhóm có thể xem xét việc tích hợp hệ thống với các nền tảng phổ biến khác như Zalo, Facebook, hoặc Email, nhằm mở rộng khả năng gợi ý và cảnh báo cho người dùng.

- Cải thiện thuật toán và dữ liệu: Tiếp tục nghiên cứu và phát triển các thuật toán gợi ý phim để cải thiện độ chính xác và hiệu quả. Cập nhật và mở rộng dữ liệu đầu vào để bao phủ nhiều thể loại và nguồn thông tin hơn.

Nhóm chúng em hy vọng rằng những cải tiến và phát triển trong tương lai sẽ giúp hệ thống đề xuất phim ngày càng hoàn thiện và hữu ích hơn cho người dùng.

Mặc dù đã rất nỗ lực, song với kiến thức và thời gian còn hạn chế nên không thể tránh khỏi những thiếu sót. Nhóm rất mong nhận được sự cảm thông, nhận xét, đánh giá từ các thầy cô để bài của nhóm được hoàn thiện hơn nữa. Chúng em xin chân thành cảm ơn.

DANH MỤC TÀI LIỆU THAM KHẢO

- [1] <https://200lab.io/blog/tong-quan-machine-learning> (200lab.io)
- [2] [Khoa học dữ liệu \(phamdinhkhanh github.io\)](#)
- [3] [Residual Block Explained Papers With Code](#)
- [4] [Object detection: Bounding box regression with Keras, TensorFlow, and Deep Learning - PyImageSearch](#)
- [5] [Hiểu về độ đo IoU trong nhận diện thực thể Trí tuệ nhân tạo \(trituenhantao.10\)](#)
- [6] [Non Maximum Suppression: Theory and Implementation in PyTorch \(learnopencv.com\)](#)
- [7] <https://github.com/Baoishere/movie-recommendation> (Baoishere)
- [8] [Google Colab](#)
- [9] <https://www.analyticsvidhya.com/blog/2020/11/create-your-own-movie-movie-recommendation-system/>