TRƯỜNG ĐẠI HỌC KỸ THUẬT CÔNG NGHIỆP

**KHOA ĐIỆN TỬ**

**BỘ MÔN: TIN HỌC CÔNG NGHIỆP**



**BÀI TẬP LỚN**

**KHOA HỌC DỮ LIỆU**

*Đề tài: Gợi ý Anime theo sở thích người xem*

Sinh viên: Vũ Ngọc Bình

Lớp: K56KMT

Giáo viên hướng dẫn: TS.Nguyễn Tuấn Linh

**THÁI NGUYÊN – 2023**

|  |  |
| --- | --- |
| **TRƯỜNG ĐHKTCN** | **CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM** |
| **KHOA ĐIỆN TỬ** | ***Độc lập - Tự do - Hạnh phúc*** |

**BÀI TẬP LỚN**

**BÀI TẬP LỚN : MÔN KHOA HỌC DỮ LIỆU**

BỘ MÔN: TIN HỌC CÔNG NGHIỆP

*Sinh viên: Vũ Ngọc Bình Lớp: K55KMT*

*MSSV: K205480106040 Ngành: Kỹ thuật máy tính*

*Ngày giao đề: 01/05/2024 Ngày hoàn thành: 04/06/2024*

1.Tên đề tài : Gợi ý Anime theo sở thích người xem

.

2. Yêu cầu: Đọc được dữ liệu tải về. Xây dựng được chương trình demo, nhập tên phim và thể loại từ bàn phím. Chương trình sẽ xuất ra một số tên phim gợi ý cho người xem.

|  |  |
| --- | --- |
| TỔ TRƯỞNG BỘ MÔN | GIÁO VIÊN HƯỚNG DẪN |
| *(Ký và ghi rõ họ tên)* | *(Ký và ghi rõ họ tên)* |

NHẬN XÉT CỦA GIÁO VIÊN HƯỚNG DẪN

*Thái Nguyên, ngày.….tháng…..năm 2023*

**GIÁO VIÊN HƯỚNG DẪN**

*(Ký ghi rõ họ tên)*

NHẬN XÉT CỦA GIÁO VIÊN CHẤM

*Thái Nguyên, ngày.….tháng…..năm 2023*

**GIÁO VIÊN CHẤM**

*(Ký ghi rõ họ tên)*

**CÁC HÌNH ẢNH SỬ DỤNG TRONG BÁO CÁO**

***Hình 1. Ví dụ về hệ thống gợi ý***

***Hình 2. Netflix – Dịch vụ xem phim trực tuyến trả phí***

***Hình 3. Anime – Những bộ phim hoạt hình sản xuất tại Nhật Bản***

***Hình 4. Bộ dữ liệu Anime Recommendation Database 2020***

***Hình 5. Bộ dữ liệu anime.csv sau khi tải về***

***Hình 6. Thuật toán Cosine Similarity***

***Hình 7. Logo Visual Studio Code và giao diện làm việc***

***Hình 8. Ngôn ngữ lập trình Python***

***Hình 9. Cấu trúc chia các ô riêng lẻ của Jupyter Notebook***

**MỤC LỤC**

[**LỜI MỞ ĐẦU** 6](#_Toc136350876)

[**CHƯƠNG I. GIỚI THIỆU CHUNG** 7](#_Toc136350877)

[**1.1.** **Hệ thống gợi ý phim** 7](#_Toc136350878)

[***1.1.1.******Hệ thống gợi ý là gì ?*** *7*](#_Toc136350879)

[***1.1.2.******Hệ thống gợi ý phim là gì ?*** *8*](#_Toc136350880)

[***1.1.3.******Ý nghĩa của hệ thống gợi ý phim*** *9*](#_Toc136350881)

[***1.1.4.******Thuật toán được sử dụng trong bài tập*** *10*](#_Toc136350882)

[**1.2.** **Anime** 10](#_Toc136350883)

[***1.2.1.******Anime là gì ?*** *10*](#_Toc136350884)

[***1.2.2.******Một số trang web xem Anime trực tuyến*** *11*](#_Toc136350885)

[**CHƯƠNG II. PHÂN TÍCH DỮ LIỆU** 12](#_Toc136350886)

[**2.1.** **Bộ dữ liệu sử dụng trong bài** 12](#_Toc136350887)

[***2.1.1.******Giới thiệu về bộ dữ liệu*** *12*](#_Toc136350888)

[***2.1.2. Các trường trong bộ dữ liệu*** *12*](#_Toc136350889)

[**2.2.** **Thuật toán tính toán khoảng cách cosine** 14](#_Toc136350890)

[***2.2.1.*** ***Khái niệm*** *14*](#_Toc136350891)

[***2.2.2. Ý nghĩa của thuật toán Cosine Similarity*** *15*](#_Toc136350892)

[**2.3.** **Thuật toán CountVectorizer** 16](#_Toc136350893)

[***2.3.1. Khái niệm*** *16*](#_Toc136350894)

[***2.3.2. Ví dụ về CountVectorizer*** *16*](#_Toc136350895)

[**CHƯƠNG III. THIẾT KẾ CHƯƠNG TRÌNH** 18](#_Toc136350896)

[**3.1. Ngôn ngữ lập trình và môi trường làm việc** 18](#_Toc136350897)

[***3.1.1. Visual Studio Code*** *18*](#_Toc136350898)

[***3.1.2. Python*** *19*](#_Toc136350899)

[***3.1.3. Jupyter Notebook*** *21*](#_Toc136350900)

[**3.2.** **Code chương trình** 22](#_Toc136350901)

[**CHƯƠNG IV. THỰC NGHIỆM VÀ KẾT LUẬN** 28](#_Toc136350902)

[**4.1. Chạy chương trình** 28](#_Toc136350903)

[**4.2. Kết quả đã đạt được và hạn chế** 29](#_Toc136350904)

[**4.3. Hướng phát triển của bài tập** 29](#_Toc136350905)

[**TÀI LIỆU THAM KHẢO** 30](#_Toc136350906)

# **LỜI MỞ ĐẦU**

Với sự phát triển của khoa học công nghệ, phim ảnh ngày nay là một hình thức giải trí phổ biến với mọi người. Phim có đa dạng mọi thể loại tùy vào sở thích người xem, đáp ứng được mọi nhu cầu xem dù là khó khăn nhất. Có người thích xem những phim hành động máu lửa, kích thích mạnh các giác quan người xem, lại có người thích những phim nhẹ nhàng hơn về tình cảm học đường, về tuổi mới lớn… Không chỉ là một hình thức giải trí, phim ảnh còn là cách chúng ta nhìn cuộc sống với một góc nhìn mới, rất thú vị và đầy tính nghệ thuật.

Bên cạnh những bộ phim được vào vai bởi người thật, người ta còn yêu thích những bộ phim hoạt hình được vẽ tỉ mỉ đến từng chi tiết. Phim hoạt hình hiện nay đã không còn chỉ dành cho trẻ em, mà thậm chí được người lớn vô cùng yêu thích. Được đầu tư vào cả hình ảnh và nội dung, đưa đến cho người xem trải nghiệm nghe nhìn vô cùng sống động mà khó phim người thật đóng nào có thể sánh bằng. Ngành công nghiệp phim hoạt hình là ngành công nghiệp trẻ, nhưng có tiềm năng rất lớn sánh ngang với các ngành công nghiệp tỷ đô khác. Nhật Bản đang là quốc gia phát triển rất mạnh về ngành này. Những bộ phim hoạt hình đến từ đất nước mặt trời mọc được rất nhiều người yêu thích vì phong cách nghệ thuật độc đá cùng với sự sáng tạo của các nhà làm phim đã tạo ra một tên gọi riêng cho những bộ phim này – Anime.

Mỗi người đều có một gu thưởng thức phim riêng. Vì thế nên luôn có mọi thể loại phim cho người xem tha hồ lựa chọn. Tuy có đầy đủ tất cả thể loại, nhưng đối với những người mới xem hoặc những người muốn thưởng thức phim theo đúng chủ đề họ muốn, đây vừa là ưu điểm cũng như nhược điểm. Có quá nhiều phim cũng như thể loại khiến cho người mới xem bị ngộp, không thể chọn ra bộ phim mình muốn xem, hoặc những người cần tìm bộ phim có thể loại hoặc chủ đề mong muốn cũng khó có thể tìm ra được “cây kim đáy bể” này. Vì vậy, một hệ thống giúp gợi ý phim theo sở thích xem là rất cần thiết.

Trong phạm vi bài báo cáo này em xin được trình bày quá trình thực hiện bài tập **“Gợi ý Anime theo sở thích người xem”** của môn học Khoa học dữ liệu. Cuối cùng, mặc dù đã cố gắng rất nhiều nhưng do thời gian có hạn, khả năng dịch và hiểu tài liệu chưa tốt nên nội dung bài tập này không thể tránh khỏi những thiếu sót, rất mong được sự chỉ bảo, góp ý của các thầy cô và các bạn.

# **CHƯƠNG I. GIỚI THIỆU CHUNG**

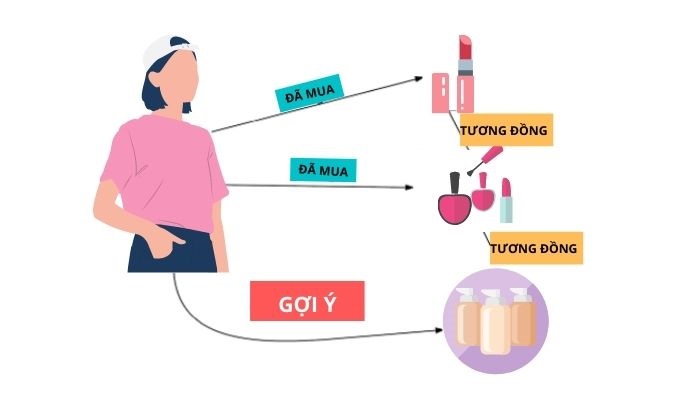
## **Hệ thống gợi ý phim**

### **Hệ thống gợi ý là gì ?**

Hệ thống gợi ý (Recommender systems hoặc Recommendation systems) là một công nghệ hoặc phần mềm được sử dụng để đề xuất các mục, nội dung hoặc hành động có thể quan tâm hoặc hữu ích cho người dùng. Nó được áp dụng rộng rãi trong nhiều lĩnh vực và ứng dụng khác nhau, từ các nền tảng mua sắm trực tuyến, dịch vụ streaming, tìm kiếm web, mạng xã hội cho đến hệ thống email và ngân hàng trực tuyến.

Hệ thống gợi ý thường dựa trên phân tích dữ liệu và sử dụng các thuật toán máy học hoặc trí tuệ nhân tạo để tìm hiểu về sở thích và hành vi của người dùng. Các thuật toán này phân tích dữ liệu lịch sử, đánh giá, lựa chọn và các yếu tố khác để đưa ra các đề xuất cá nhân hoặc tổng hợp thông tin liên quan.

Qua việc cung cấp các gợi ý cá nhân, hệ thống gợi ý có thể giúp người dùng khám phá thêm thông tin mới, sản phẩm hoặc nội dung mà họ có thể không biết đến. Nó cũng cải thiện trải nghiệm người dùng, giúp tiết kiệm thời gian và tạo ra sự tương tác tốt hơn giữa người dùng và hệ thống.



***Hình 1. Ví dụ về hệ thống gợi ý***

### **Hệ thống gợi ý phim là gì ?**

Hệ thống gợi ý phim là một dạng hệ thống gợi ý nhằm đề xuất các bộ phim có thể phù hợp với sở thích và sự quan tâm của người dùng. Hệ thống này dựa trên phân tích dữ liệu và sử dụng các thuật toán máy học hoặc trí tuệ nhân tạo để hiểu về lựa chọn, đánh giá và hành vi xem phim của người dùng. Trong dịch vụ xem phim trực tuyến, hệ thống gợi ý có thể đề xuất các bộ phim hoặc chương trình dựa trên sở thích hoặc các nội dung đã được người dùng truy cập.



***Hình 2. Netflix – Dịch vụ xem phim trực tuyến trả phí***

Hệ thống gợi ý phim có thể sử dụng nhiều phương pháp khác nhau để đưa ra các đề xuất phim cá nhân. Dưới đây là một số phương pháp phổ biến được sử dụng:

* **Dựa trên sở thích cá nhân:** Hệ thống theo dõi lịch sử xem phim của người dùng, đánh giá và đánh dấu các bộ phim mà họ đã xem hoặc yêu thích. Dựa trên thông tin này, hệ thống gợi ý các bộ phim có thể phù hợp với sở thích cá nhân.
* **Dựa trên nội dung:** Hệ thống phân tích nội dung của các bộ phim, bao gồm các yếu tố như thể loại, diễn viên, đạo diễn, câu chuyện, và hợp nhất thông tin này với sở thích cá nhân của người dùng. Dựa trên phân tích nội dung, hệ thống gợi ý các bộ phim có nội dung tương tự hoặc liên quan.
* **Dựa trên thông tin xã hội:** Hệ thống sử dụng thông tin xã hội như đánh giá, bình luận và chia sẻ từ người dùng khác để đưa ra các đề xuất phim. Nếu người dùng có sự tương tác hoặc chia sẻ liên quan đến một bộ phim cụ thể, hệ thống có thể gợi ý các bộ phim tương tự hoặc phổ biến trong cộng đồng.
* **Kết hợp các phương pháp:** Hệ thống gợi ý phim thường kết hợp nhiều phương pháp để cung cấp các đề xuất phim đa dạng và phù hợp với người dùng. Bằng cách kết hợp thông tin về sở thích cá nhân, nội dung phim và thông tin xã hội, hệ thống gợi ý có thể đưa ra các đề xuất tốt hơn.

### **Ý nghĩa của hệ thống gợi ý phim**

Hệ thống gợi ý phim có ý nghĩa quan trọng và nhiều lợi ích đối với người dùng, như sau:

* **Khám phá phim mới:** Một trong những lợi ích chính của hệ thống gợi ý phim là giúp người dùng khám phá và tiếp cận với các bộ phim mới mà họ có thể chưa biết đến trước đó. Thay vì phải tìm kiếm một cách thủ công, hệ thống gợi ý cung cấp danh sách các bộ phim có thể phù hợp với sở thích và lựa chọn của người dùng, giúp họ mở rộng phạm vi xem phim và khám phá những nội dung mới thú vị.
* **Tiết kiệm thời gian:** Thay vì phải tự mò mẫm và tìm kiếm một bộ phim để xem, hệ thống gợi ý phim giúp tiết kiệm thời gian cho người dùng. Các đề xuất phim đã được lọc và tùy chỉnh dựa trên sở thích cá nhân, giúp người dùng nhanh chóng tìm thấy các bộ phim có thể phù hợp với họ.
* **Trải nghiệm cá nhân hóa:** Hệ thống gợi ý phim có khả năng cá nhân hóa đề xuất dựa trên lịch sử xem phim, đánh giá và sở thích của người dùng. Điều này giúp cung cấp trải nghiệm xem phim cá nhân hóa và đáp ứng được mong đợi và yêu cầu riêng của từng người dùng.
* **Đa dạng hóa nội dung:** Hệ thống gợi ý phim đảm bảo rằng người dùng được tiếp cận với đa dạng nội dung phim. Bằng cách đề xuất các bộ phim từ các thể loại, đạo diễn, diễn viên và quốc gia khác nhau, hệ thống giúp mở rộng sự đa dạng và cung cấp nhiều lựa chọn cho người dùng.
* **Tăng cường trải nghiệm người dùng:** Hệ thống gợi ý phim cung cấp một trải nghiệm người dùng tốt hơn và nâng cao sự tương tác giữa người dùng và nền tảng xem phim. Bằng cách đưa ra các đề xuất phim chính xác và phù hợp, hệ thống giúp người dùng tìm kiếm, chọn lựa và xem phim một cách dễ dàng và thú vị hơn.

### **Thuật toán được sử dụng trong bài tập**

Thuật toán được sử dụng trong bài là **“Tính toán sự tương đồng giữa 2 vector dựa trên góc cosine”**. Khoảng cách cosine được tính bằng cách đo lường cosine của góc giữa hai vector trong không gian nhiều chiều.

Ta sẽ biến đổi và kết hợp các trường được sử dụng trong dữ liệu như tên phim, thể loại của phim, số tập phim,… thành một vector đặc trưng cho bộ phim đó. Khi ta nhập thông tin vào hệ thống, thông tin đó sẽ được biến đổi thành một vector, sau đó sẽ mang đi so sánh với các vector đã được biến đổi trong hệ thống để tính toán sự tương đồng. Cuối cùng hệ thống sẽ đề xuất ra những phim có sự tương đồng nhất với thông tin đã nhập.

## **Anime**

### **Anime là gì ?**

Anime là thuật ngữ dùng để chỉ phim hoạt hình có nguồn gốc từ Nhật Bản. Thuật ngữ "Anime" là viết tắt của từ "Animation" (hoạt hình) trong tiếng Anh và được sử dụng rộng rãi trên toàn thế giới để đề cập đến các loại phim hoạt hình Nhật Bản.

Anime không chỉ dành riêng cho trẻ em, mà cũng có các thể loại và nội dung đa dạng, phù hợp với mọi lứa tuổi và sở thích. Anime có thể bao gồm các thể loại như hành động, phiêu lưu, hài hước, tình cảm, kỳ ảo, khoa học viễn tưởng, viễn tưởng, đời thường, lịch sử, kinh dị và nhiều thể loại khác.



***Hình 3. Anime – Những bộ phim hoạt hình sản xuất tại Nhật Bản***

Phim hoạt hình Nhật Bản (anime) nổi tiếng trên toàn thế giới và có một cộng đồng người hâm mộ đông đảo. Nhiều anime đã trở thành hiện tượng văn hóa và có sức ảnh hưởng lớn, không chỉ trong lĩnh vực giải trí mà còn trong âm nhạc, thời trang, game và các lĩnh vực khác.

Anime được tạo ra bằng cách sử dụng các kỹ thuật hoạt hình truyền thống hoặc kỹ thuật hoạt hình số, và thường có đặc điểm riêng về nét vẽ, phong cách và biểu đạt.Hiện nay, hãng sản xuất anime nổi tiếng nhất tại Nhật là Ghibli với nhiều bộ phim khá nổi tiếng như: “Spirited Away”, “Grave of the Fireflies”, “My Neighbor Totoro”,… được cả thế giới công nhận.

### **Một số trang web xem Anime trực tuyến**

Đây là một số trang web phổ biến để xem anime trực tuyến. Những trang web này được người dùng tin dùng vì độ uy tín và những bộ phim chất lượng cao:

* **Crunchyroll:** Crunchyroll là một trong những trang web xem anime hàng đầu, cung cấp nhiều bộ anime phổ biến và mới nhất. Trang web này cung cấp cả phiên bản miễn phí và phiên bản trả phí với nội dung chất lượng cao và các tính năng bổ sung.
* **Funimation:** Funimation là một trang web chuyên về anime, cung cấp nhiều bộ anime đa dạng và phụ đề trong nhiều ngôn ngữ khác nhau. Trang web này cũng có phiên bản miễn phí và phiên bản trả phí với nội dung chất lượng cao.
* **Netflix:** Netflix cũng là một nền tảng phổ biến để xem anime. Họ cung cấp một loạt các bộ anime phổ biến và độc quyền, cũng như các bộ phim và chương trình truyền hình khác. Netflix yêu cầu một khoản phí hàng tháng để truy cập vào nội dung của họ.
* **AnimeLab:** AnimeLab là một trang web xem anime phổ biến ở Úc và New Zealand. Họ cung cấp nhiều bộ anime từ các nhà sản xuất hàng đầu và có cả phiên bản miễn phí và phiên bản trả phí.

# **CHƯƠNG II. PHÂN TÍCH DỮ LIỆU**

## **Bộ dữ liệu sử dụng trong bài**

### **Giới thiệu về bộ dữ liệu**

Bộ dữ liệu sử dụng trong bài tập là bộ dữ liệu **“Anime Recommendation Database 2020”**. Đây là bộ dữ liệu được lấy từ trang web **Kaggle**, chứa thông tin về 17.562 anime và sở thích từ 325.772 người dùng khác nhau tại trang web đánh giá phim anime myanimelist.net

A screenshot of a computer

Description automatically generated with medium confidence

***Hình 4. Bộ dữ liệu Anime Recommendation Database 2020***

### **Các trường trong bộ dữ liệu**

A screen shot of a computer

Description automatically generated with low confidence

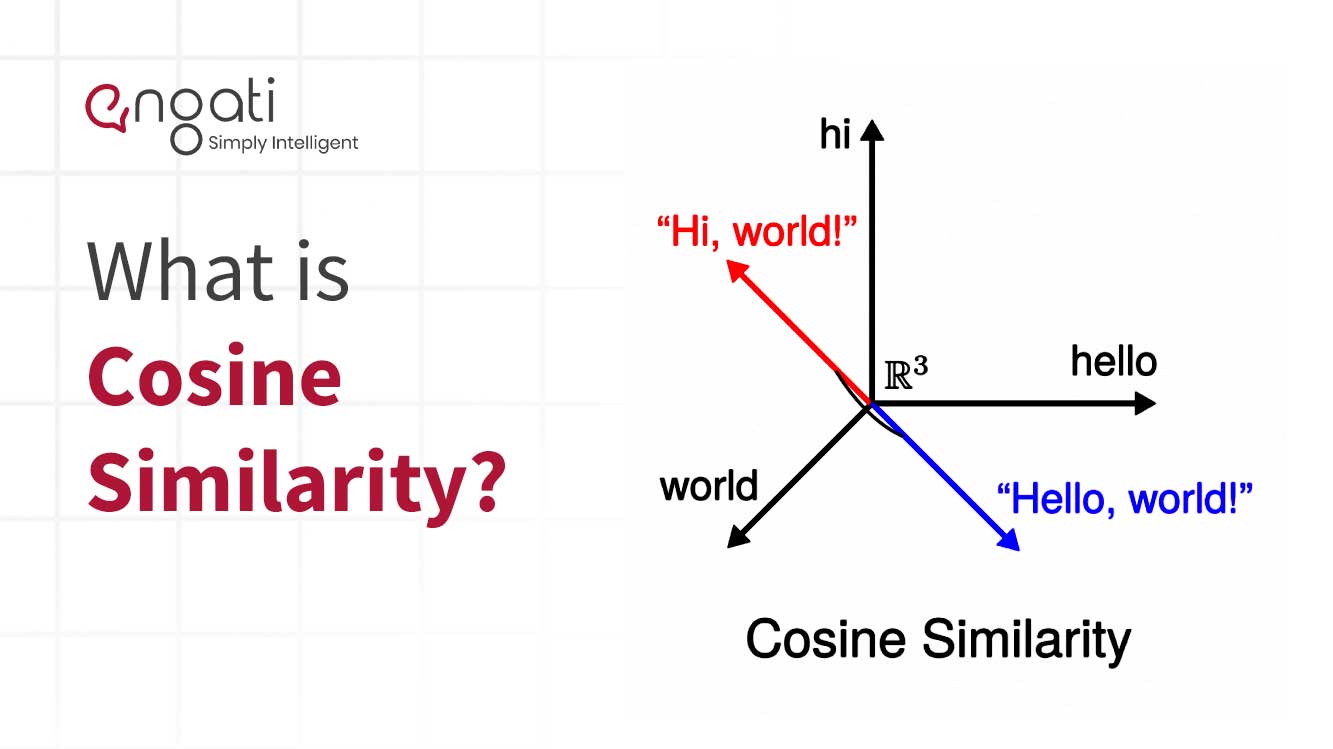
***Hình 5. Bộ dữ liệu anime.csv sau khi tải về***

* **MAL\_ID:** Mã số của bộ phim trên trang web “My Anime List”
* **Name:** Tên của phim
* **Score:** Số điểm trung bình của phim trên thang điểm 10
* **Genres:** Thể loại của bộ phim (Hài, Hành động, Lãng mạn, Kinh dị, …)
* **English name:** Tên bộ phim theo tiếng Anh
* **Japanese name:** Tên bộ phim theo tiếng Nhật
* **Type:** Hình thức công chiếu của phim (Chiếu TV, chiếu rạp, phần phim mở rộng,…)
* **Episodes:** Số tập của phim
* **Aired:** Thời gian công chiếu phim
* **Premiered:** Mùa công chiếu phim (Mùa hè năm 1990, Mùa đông năm 2006,…)
* **Producers:** Nhà sản xuất
* **Licensors:** Đơn vị cấp phép công chiếu
* **Studios:** Hãng phim
* **Source:** Nguồn gốc của phim (Truyện tranh, tiểu thuyết, game,…)
* **Duration:** Độ dài mỗi tập
* **Rating:** Xếp hạng độ tuổi
* **Ranked:** Xếp hạng của phim
* **Popularity:** Độ phổ biến của phim, xếp hạng dựa vào số người dùng thêm vào danh sách xem của họ
* **Members:** Số lượng thành viên cộng đồng nằm trong “nhóm” của bộ phim này
* **Favorites:** Số người thêm bộ phim vào danh sách yêu thích
* **Watching:** Số người đang xem phim
* **Completed:** Só người đã xem xong bộ phim
* **On-Hold:** Số người đang tạm dừng xem
* **Dropped:** Số người đã bỏ xem phim
* **Plan to Watch:** Số người lên kế hoạch xem phim
* **Score-10:** Số người đánh giá 10 cho phim
* **Score-9:** Số người đánh giá 9 cho phim
* **Score-8:** Số người đánh giá 8 cho phim
* **Score-7:** Số người đánh giá 7 cho phim
* **Score-6:** Số người đánh giá 6 cho phim
* **Score-5:** Số người đánh giá 5 cho phim
* **Score-4:** Số người đánh giá 4 cho phim
* **Score-3:** Số người đánh giá 3 cho phim
* **Score-2:** Số người đánh giá 2 cho phim
* **Score-1:** Số người đánh giá 1 cho phim

## **Thuật toán tính toán khoảng cách cosine**

### **2.2.1. Khái niệm**

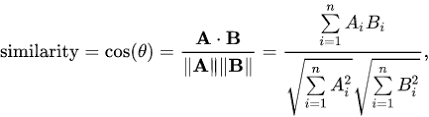
Thuật toán tính toán khoảng cách cosine (Cosine Similarity) là một phương pháp đo đạc mức độ tương đồng giữa hai vectơ trong không gian đa chiều. Nó được sử dụng rộng rãi trong lĩnh vực xử lý ngôn ngữ tự nhiên và khai phá dữ liệu, thường được sử dụng để so sánh sự tương đồng giữa các văn bản, từ ngữ hoặc vectơ biểu diễn của đối tượng trong không gian đa chiều.



***Hình 6. Thuật toán Cosine Similarity***

Để tính toán khoảng cách cosine giữa hai vectơ, ta thực hiện các bước sau:

1. **Chuẩn hóa các vectơ:** Đầu tiên, chúng ta chuẩn hóa các vectơ đầu vào để chúng có cùng độ dài hoặc độ dài tương đương. Điều này đảm bảo rằng chỉ số cosine similarity sẽ nằm trong khoảng [-1, 1]. Để chuẩn hóa một vectơ, ta chia nó cho độ dài của vectơ đó. Độ dài của một vectơ có thể được tính bằng cách lấy căn bậc hai của tổng bình phương các thành phần của vectơ.
2. **Tính tích vô hướng:** Tiếp theo, ta tính tích vô hướng (dot product) của hai vectơ đã được chuẩn hóa. Tích vô hướng của hai vectơ a và b được tính bằng cách lấy tổng của tích các thành phần tương ứng của hai vectơ đó.
3. **Tính khoảng cách cosine:** Cuối cùng, ta tính khoảng cách cosine bằng cách chia tích vô hướng cho tích của độ dài hai vectơ. Khoảng cách cosine giữa hai vectơ a và b được tính bằng công thức sau:



Kết quả là một số trong khoảng [-1, 1], với 1 đại diện cho hai vectơ hoàn toàn tương đồng, 0 đại diện cho hai vectơ không tương đồng và -1 đại diện cho hai vectơ hoàn toàn đối ngược nhau.

### **2.2.2. Ý nghĩa của thuật toán Cosine Similarity**

Thuật toán Cosine Similarity có nhiều ứng dụng trong các lĩnh vực khác nhau. Dưới đây là một số ví dụ phổ biến về ứng dụng của thuật toán cosine similarity:

* **Xử lý ngôn ngữ tự nhiên:** Trong xử lý ngôn ngữ tự nhiên, cosine similarity được sử dụng để đo đạc mức độ tương đồng giữa các văn bản. Nó có thể được áp dụng để tìm kiếm văn bản tương tự, phân loại văn bản, gom cụm văn bản, hoặc xây dựng các hệ thống gợi ý dựa trên nội dung.
* **Hệ thống gợi ý:** Thuật toán cosine similarity được sử dụng trong hệ thống gợi ý để tìm các mục tương tự hoặc người dùng tương đồng. Ví dụ, trên các trang thương mại điện tử, nó có thể được sử dụng để gợi ý sản phẩm tương tự cho người dùng dựa trên lịch sử mua hàng của họ hoặc dựa trên sự tương đồng về mặt nội dung.
* **Phân cụm dữ liệu:** Cosine similarity cũng được sử dụng trong phân cụm dữ liệu để nhóm các điểm dữ liệu có đặc trưng tương tự. Thuật toán có thể giúp xác định sự tương đồng giữa các đối tượng và gom nhóm chúng lại với nhau.
* **Hệ thống lọc cộng tác:** Trong hệ thống lọc cộng tác, cosine similarity có thể được sử dụng để tính toán sự tương đồng giữa các người dùng hoặc các mục tiêu. Kết quả này sau đó có thể được sử dụng để tạo ra các gợi ý cá nhân hóa cho người dùng hoặc tìm ra các người dùng tương tự.
* **Trích xuất thông tin:** Cosine similarity cũng có thể được sử dụng để trích xuất thông tin từ các văn bản. Nó có thể đo đạc mức độ tương tự giữa các văn bản và từ đó xác định được sự liên quan và tính chất của các đối tượng trong văn bản.

## **Thuật toán CountVectorizer**

### **2.3.1. Khái niệm**

Thuật toán CountVectorizer là một phương pháp trong xử lý ngôn ngữ tự nhiên để biểu diễn văn bản thành vectơ đặc trưng dựa trên tần số xuất hiện của các từ trong văn bản. Nó là một phần của thư viện scikit-learn (sklearn) của Python và được sử dụng rộng rãi trong các tác vụ như phân loại văn bản, gom cụm và trích xuất thông tin.

Cách hoạt động của CountVectorizer:

1. **Chia tách văn bản:** Đầu tiên, CountVectorizer sẽ chia tách các văn bản thành các "token" (có thể là từ, ký tự, hoặc n-gram, phụ thuộc vào cấu hình).
2. **Xây dựng từ điển:** CountVectorizer xây dựng một từ điển của các từ duy nhất trong tập dữ liệu. Mỗi từ sẽ được gán một chỉ mục dcuy nhất.
3. **Tính toán tần số xuất hiện:** CountVectorizer sử dụng từ điển để tính toán tần số xuất hiện (số lần xuất hiện) của mỗi từ trong mỗi văn bản.
4. **Biểu diễn thành vector:** Kết quả là một ma trận mà mỗi hàng đại diện cho một văn bản và mỗi cột đại diện cho tần số xuất hiện của một từ trong văn bản tương ứng.

CountVectorizer là một công cụ mạnh mẽ để biểu diễn văn bản thành dạng số học, tuy nhiên, nó không lưu trữ thông tin về tần số xuất hiện tương đối giữa các văn bản và không xem xét các khía cạnh ngữ nghĩa của từ.

### **2.3.2. Ví dụ về CountVectorizer**

Xem xét một vài văn bản mẫu từ một tài liệu (mỗi văn bản là một phần tử danh sách):

**document = [ “One Geek helps Two Geeks”, “Two Geeks help Four Geeks”, “Each Geek helps many other Geeks at GeeksforGeeks.”]**

CountVectorizer tạo một ma trận (matrix) trong đó mỗi từ duy nhất được biểu thị bằng một cột (column) của ma trận và mỗi mẫu văn bản từ tài liệu là một hàng (row) trong ma trận. Giá trị của mỗi ô (cell) là số lượng từ trong mẫu văn bản cụ thể đó:

A screenshot of a computer

Description automatically generated with low confidence

* Có 12 từ duy nhất trong tài liệu, được biểu diễn dưới dạng các cột của bảng.
* Có 3 mẫu văn bản trong tài liệu, mỗi mẫu được biểu thị dưới dạng các hàng của bảng.
* Mỗi ô chứa một số, đại diện cho số lượng từ trong văn bản cụ thể đó.
* Tất cả các từ đã được chuyển đổi thành chữ thường.
* Các từ trong các cột đã được sắp xếp theo thứ tự bảng chữ cái.

Bên trong CountVectorizer, những từ này không được lưu trữ dưới dạng chuỗi. Thay vào đó, chúng được cung cấp một giá trị chỉ số (index) cụ thể. Trong trường hợp này, "at" sẽ có chỉ số 0, "each" sẽ có chỉ số 1, "four" sẽ có chỉ số 2, v.v. Biểu diễn thực tế đã được hiển thị trong bảng dưới đây.

A grid of numbers on a white background

Description automatically generated with medium confidence

# **CHƯƠNG III. THIẾT KẾ CHƯƠNG TRÌNH**

## **3.1. Ngôn ngữ lập trình và môi trường làm việc**

### **3.1.1. Visual Studio Code**

**Visual Studio Code** là một [trình soạn thảo mã nguồn](https://vi.wikipedia.org/wiki/Tr%C3%ACnh_so%E1%BA%A1n_th%E1%BA%A3o_m%C3%A3_ngu%E1%BB%93n" \o "Trình soạn thảo mã nguồn) được phát triển bởi [Microsoft](https://vi.wikipedia.org/wiki/Microsoft) dành cho [Windows](https://vi.wikipedia.org/wiki/Microsoft_Windows), [Linux](https://vi.wikipedia.org/wiki/Linux) và [macOS](https://vi.wikipedia.org/wiki/MacOS). Nó hỗ trợ chức năng debug, đi kèm với [Git](https://vi.wikipedia.org/wiki/Git_(ph%E1%BA%A7n_m%E1%BB%81m)), có chức năng nổi bật cú pháp ([syntax highlighting](https://vi.wikipedia.org/w/index.php?title=Syntax_highlighting&action=edit&redlink=1)), tự hoàn thành mã thông minh, [snippets](https://vi.wikipedia.org/w/index.php?title=Snippets&action=edit&redlink=1), và [cải tiến mã nguồn](https://vi.wikipedia.org/wiki/C%E1%BA%A3i_ti%E1%BA%BFn_m%C3%A3_ngu%E1%BB%93n" \o "Cải tiến mã nguồn). Nó cũng cho phép tùy chỉnh, do đó, người dùng có thể thay đổi theme, phím tắt, và các tùy chọn khác. Nó miễn phí và là [phần mềm mã nguồn mở](https://vi.wikipedia.org/wiki/Ph%E1%BA%A7n_m%E1%BB%81m_t%E1%BB%B1_do_ngu%E1%BB%93n_m%E1%BB%9F" \o "Phần mềm tự do nguồn mở) theo [giấy phép MIT](https://vi.wikipedia.org/wiki/Gi%E1%BA%A5y_ph%C3%A9p_MIT" \o "Giấy phép MIT), mặc dù bản phát hành của [Microsoft](https://vi.wikipedia.org/wiki/Microsoft) là theo [giấy phép phần mềm](https://vi.wikipedia.org/wiki/Gi%E1%BA%A5y_ph%C3%A9p_ph%E1%BA%A7n_m%E1%BB%81m" \o "Giấy phép phần mềm) miễn phí.

**Visual Studio Code** được dựa trên [Electron](https://en.wikipedia.org/wiki/Electron_(software_framework)), một nền tảng được sử dụng để triển khai các ứng dụng [Node.js](https://vi.wikipedia.org/wiki/Node.js) [máy tính cá nhân](https://vi.wikipedia.org/wiki/M%C3%A1y_t%C3%ADnh_c%C3%A1_nh%C3%A2n" \o "Máy tính cá nhân) chạy trên động cơ bố trí Blink. Mặc dù nó sử dụng nền tảng Electronnhưng phần mềm này không phải là một bản khác của [Atom](https://vi.wikipedia.org/wiki/Atom), nó thực ra được dựa trên trình biên tập của [Visual Studio Online](https://vi.wikipedia.org/wiki/Microsoft_Visual_Studio) (tên mã là "Monaco").

Graphical user interface, text, application

Description automatically generated

***Hình 7. Logo Visual Studio Code và giao diện làm việc***

**Visual Studio Code** có thể được mở rộng qua [plugin](https://vi.wikipedia.org/wiki/Plugin). Điều này giúp bổ sung thêm chức năng cho trình biên tậpvà hỗ trợ thêm ngôn ngữ. Một tính năng đáng chú ý là khả năng tạo phần mở rộng để phân tích mã, như là các linter và công cụ phân tích, sử dụng [Language Server Protocol](https://vi.wikipedia.org/w/index.php?title=Language_Server_Protocol&action=edit&redlink=1).

**Visual Studio Code** là một trình biên tập mã. Nó hỗ trợ nhiều ngôn ngữ và chức năng tùy vào ngôn ngữ sử dụng theo như trong bảng sau. Nhiều chức năng của Visual Studio Code không hiển thị ra trong các menu tùy chọn hay [giao diện người dùng](https://vi.wikipedia.org/wiki/Giao_di%E1%BB%87n_ng%C6%B0%E1%BB%9Di_d%C3%B9ng" \o "Giao diện người dùng). Thay vào đó, chúng được gọi thông qua khung nhập lệnh hoặc qua một tập tin [.json](https://vi.wikipedia.org/wiki/JSON) (ví dụ như tập tin tùy chỉnh của người dùng). Khung nhập lệnh là một giao diện theo dòng lệnh. Tuy nhiên, nó biến mất khi người dùng nhấp bất cứ nơi nào khác, hoặc nhấn tổ hợp phím để tương tác với một cái gì đó ở bên ngoài đó. Tương tự như vậy với những dòng lệnh tốn nhiều thời gian để xử lý. Khi thực hiện những điều trên thì quá trình xử lý dòng lệnh đó sẽ bị hủy.

### **3.1.2. Python**

**Python** là một [ngôn ngữ lập trình](https://vi.wikipedia.org/wiki/Ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh" \o "Ngôn ngữ lập trình) bậc cao cho các mục đích lập trình đa năng, do [Guido van Rossum](https://vi.wikipedia.org/w/index.php?title=Guido_van_Rossum&action=edit&redlink=1) tạo ra và lần đầu ra mắt vào năm [1991](https://vi.wikipedia.org/wiki/1990). Python được thiết kế với ưu điểm mạnh là dễ đọc, dễ học và dễ nhớ. Python là ngôn ngữ có hình thức rất sáng sủa, cấu trúc rõ ràng, thuận tiện cho người mới học lập trình và là ngôn ngữ lập trình dễ học; được dùng rộng rãi trong phát triển [trí tuệ nhân tạo](https://vi.wikipedia.org/wiki/Tr%C3%AD_tu%E1%BB%87_nh%C3%A2n_t%E1%BA%A1o" \o "Trí tuệ nhân tạo). Cấu trúc của Python còn cho phép người sử dụng viết mã lệnh với số lần gõ phím tối thiểu.

**Python** luôn được xếp hạng vào những ngôn ngữ lập trình phổ biến nhất.

**Python** hoàn toàn [tạo kiểu động](https://vi.wikipedia.org/w/index.php?title=T%E1%BA%A1o_ki%E1%BB%83u_%C4%91%E1%BB%99ng&action=edit&redlink=1" \o "Tạo kiểu động (trang không tồn tại)) và dùng cơ chế [cấp phát bộ nhớ tự động](https://vi.wikipedia.org/wiki/Qu%E1%BA%A3n_l%C3%BD_b%E1%BB%99_nh%E1%BB%9B" \o "Quản lý bộ nhớ); do vậy nó tương tự như [Perl](https://vi.wikipedia.org/wiki/Perl), [Ruby](https://vi.wikipedia.org/wiki/Ruby_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)), [Scheme](https://vi.wikipedia.org/wiki/Scheme), [Smalltalk](https://vi.wikipedia.org/wiki/Smalltalk), và [Tcl](https://vi.wikipedia.org/wiki/Tcl" \o "Tcl). Python được phát triển trong một dự án mã mở, do [tổ chức phi lợi nhuận](https://vi.wikipedia.org/wiki/T%E1%BB%95_ch%E1%BB%A9c_phi_l%E1%BB%A3i_nhu%E1%BA%ADn" \o "Tổ chức phi lợi nhuận) Python Software Foundation quản lý.

Logo, company name

Description automatically generated

***Hình 8. Ngôn ngữ lập trình Python***

**Python** là một [ngôn ngữ lập trình đa mẫu hình](https://vi.wikipedia.org/wiki/M%E1%BA%ABu_h%C3%ACnh_l%E1%BA%ADp_tr%C3%ACnh" \o "Mẫu hình lập trình). [Lập trình hướng đối tượng](https://vi.wikipedia.org/wiki/L%E1%BA%ADp_tr%C3%ACnh_h%C6%B0%E1%BB%9Bng_%C4%91%E1%BB%91i_t%C6%B0%E1%BB%A3ng" \o "Lập trình hướng đối tượng) và [lập trình cấu trúc](https://vi.wikipedia.org/wiki/L%E1%BA%ADp_tr%C3%ACnh_c%E1%BA%A5u_tr%C3%BAc" \o "Lập trình cấu trúc) được hỗ trợ hoàn toàn, và nhiều tính năng của nó cũng hỗ trợ lập trình hàm và [lập trình hướng khía cạnh](https://vi.wikipedia.org/wiki/L%E1%BA%ADp_tr%C3%ACnh_h%C6%B0%E1%BB%9Bng_kh%C3%ADa_c%E1%BA%A1nh" \o "Lập trình hướng khía cạnh) (bao gồm [siêu lập trình](https://vi.wikipedia.org/wiki/L%E1%BA%ADp_tr%C3%ACnh_meta" \o "Lập trình meta) và [siêu đối tượng](https://vi.wikipedia.org/w/index.php?title=Si%C3%AAu_%C4%91%E1%BB%91i_t%C6%B0%E1%BB%A3ng&action=edit&redlink=1" \o "Siêu đối tượng (trang không tồn tại)). Các mẫu hình khác cũng được hỗ trợ thông qua các phần mở rộng, bao gồm [thiết kế theo hợp đồng](https://vi.wikipedia.org/w/index.php?title=Thi%E1%BA%BFt_k%E1%BA%BF_theo_h%E1%BB%A3p_%C4%91%E1%BB%93ng&action=edit&redlink=1" \o "Thiết kế theo hợp đồng (trang không tồn tại)) và [lập trình logic](https://vi.wikipedia.org/wiki/L%E1%BA%ADp_tr%C3%ACnh_logic" \o "Lập trình logic).

**Python** sử dụng [kiểu động](https://vi.wikipedia.org/wiki/H%E1%BB%87_th%E1%BB%91ng_ki%E1%BB%83u" \o "Hệ thống kiểu) và một dạng kết hợp giữa [đếm tham chiếu](https://vi.wikipedia.org/w/index.php?title=%C4%90%E1%BA%BFm_tham_chi%E1%BA%BFu&action=edit&redlink=1" \o "Đếm tham chiếu (trang không tồn tại)) và bộ dọn rác kiểm tra theo chu kì để [quản lí bộ nhớ](https://vi.wikipedia.org/wiki/Qu%E1%BA%A3n_l%C3%AD_b%E1%BB%99_nh%E1%BB%9B" \o "Quản lí bộ nhớ). Nó cũng có tính năng [phân giải tên](https://vi.wikipedia.org/w/index.php?title=Ph%C3%A2n_gi%E1%BA%A3i_t%C3%AAn&action=edit&redlink=1" \o "Phân giải tên (trang không tồn tại)) động ([liên kết muộn](https://vi.wikipedia.org/w/index.php?title=Li%C3%AAn_k%E1%BA%BFt_mu%E1%BB%99n&action=edit&redlink=1" \o "Liên kết muộn (trang không tồn tại))), cho phép liên kết các tên biến và phương thức trong quá trình thực thi chương trình.

Triết lý căn bản của ngôn ngữ Python được trình bày trong tài liệu *The*[*Zen of Python*](https://vi.wikipedia.org/wiki/Zen_of_Python) (*PEP 20*), có dạng [thơ Haiku](https://vi.wikipedia.org/wiki/Haiku" \o "Haiku), tóm gọn như sau:

* Đẹp đẽ tốt hơn xấu xí
* Minh bạch tốt hơn ngầm định
* Đơn giản tốt hơn phức tạp
* Phức tạp tốt hơn rắc rối
* Tính dễ đọc rất quan trọng.

Thay vì tích hợp hết tất cả các tính năng vào phần cốt lõi, Python được thiết kế để [dễ dàng mở rộng](https://vi.wikipedia.org/w/index.php?title=T%C3%ADnh_m%E1%BB%9F_r%E1%BB%99ng&action=edit&redlink=1" \o "Tính mở rộng (trang không tồn tại)) (bằng các mô đun). Tính mô đun nhỏ gọn này đã làm cho Python trở nên phổ biến như là một cách thêm các giao diện lập trình được vào các ứng dụng hiện có. Tầm nhìn của Van Rossum về một ngôn ngữ có phần lõi nhỏ với một thứ viện chuẩn rộng lớn và một trình thông dịch dễ dàng mở rộng bắt nguồn từ việc ông nản lòng trước [ABC](https://vi.wikipedia.org/w/index.php?title=ABC_(ng%C3%B4n_ng%E1%BB%AF_l%E1%BA%ADp_tr%C3%ACnh)&action=edit&redlink=1), một ngôn ngữ lập trình tán thành hướng tiếp cận ngược lại. Python thường được mô tả là một ngôn ngữ "tặng kèm pin" nhờ vào [thư viện chuẩn](https://vi.wikipedia.org/wiki/Th%C6%B0_vi%E1%BB%87n_chu%E1%BA%A9n" \o "Thư viện chuẩn) bao quát của nó.

**Python** nỗ lực hướng đến một cú pháp đơn giản hơn, gọn gàng hơn trong khi vẫn cho các nhà phát triển lựa chọn phương pháp viết mã của họ. Đối lập với khẩu hiệu "[có nhiều hơn một cách để làm việc này](https://vi.wikipedia.org/w/index.php?title=C%C3%B3_nhi%E1%BB%81u_h%C6%A1n_m%E1%BB%99t_c%C3%A1ch_%C4%91%E1%BB%83_l%C3%A0m_vi%E1%BB%87c_n%C3%A0y&action=edit&redlink=1" \o "Có nhiều hơn một cách để làm việc này (trang không tồn tại))", triết lý thiết kể của Python lại nằm trong châm ngôn "chỉ nên có một— và tốt nhất là chỉ một—cách rõ ràng để làm việc này". [Alex Martelli](https://vi.wikipedia.org/w/index.php?title=Alex_Martelli&action=edit&redlink=1), một Viện sĩ (Fellow) tại Tổ chức Phần mềm Python (Python Software Foundation) và là một tác giả viết sách Python, viết rằng "Mô tả một thứ gì đó là "tài tình" *không* được coi là một lời khen ngợi trong văn hoá Python."

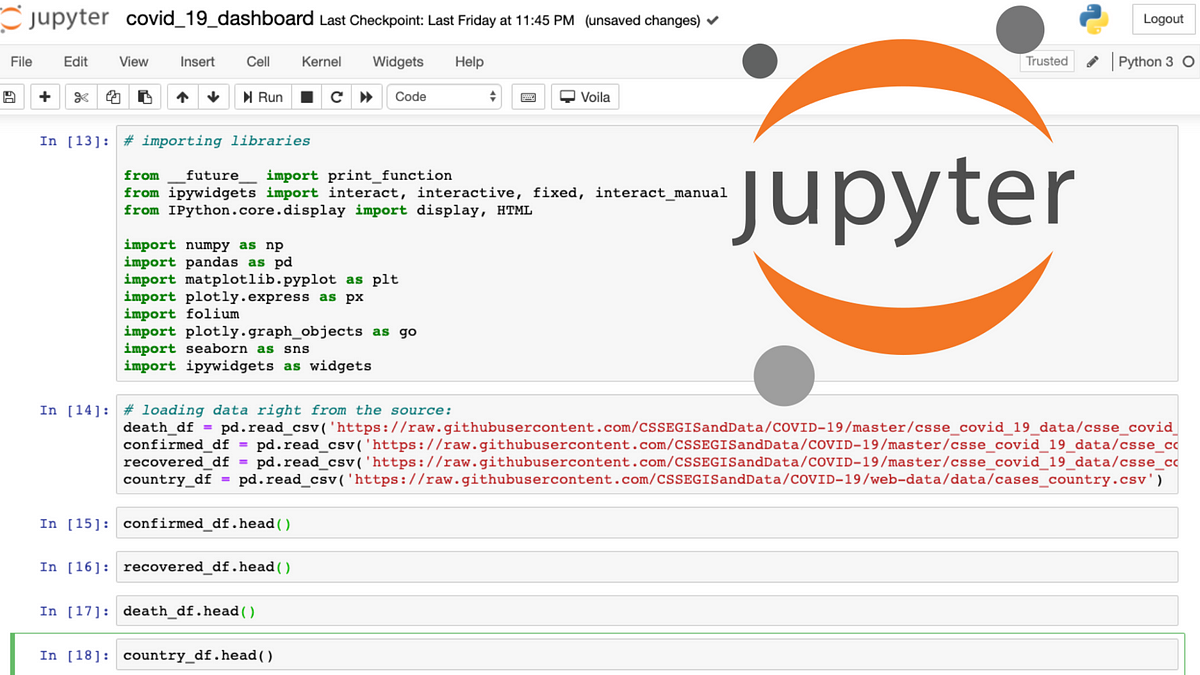
### **3.1.3. Jupyter Notebook**

Jupyter là một thuật ngữ được ghép từ ba ngôn ngữ lập trình Julia, Python và R. Trước đây Jupyter Notebook có tên là IPython Notebook, đến năm 2014 tách ra khỏi IPython và đổi tên thành Jupyter Notebook.

Jupyter Notebook là một nền tảng tính toán khoa học mã nguồn mở, bạn có thể sử dụng để tạo và chia sẻ các tài liệu có chứa code trực tiếp, phương trình, trực quan hóa dữ liệu và văn bản tường thuật.

Jupyter Notebook được coi là môi trường điện toán tương tác đa ngôn ngữ, hỗ trợ hơn 40 ngôn ngữ lập trình cho người dùng.

Jupyter Notebook cho phép bạn viết và thực thi mã trong các ô (cells) riêng lẻ và xem kết quả ngay lập tức. Bạn có thể sử dụng nhiều ngôn ngữ lập trình khác nhau như Python, R, Julia và nhiều ngôn ngữ khác. Điều này rất hữu ích khi bạn muốn thực hiện các tính toán, trực quan hóa dữ liệu, viết báo cáo hoặc chia sẻ ý tưởng của mình.



***Hình 9. Cấu trúc chia các ô riêng lẻ của Jupyter Notebook***

Jupyter Notebook được viết bằng các ngôn ngữ như Python, R và Julia, nền tảng này hiện đang được sử dụng rộng rãi. Bên cạnh đó, Jupyter còn tạo ra tài liệu, trực quan hóa dữ liệu và lưu trữ chúng một cách dễ dàng hơn rất nhiều. Dưới đây là một số lợi ích mà Jupyter Notebook mang lại:

* **Phân tích khám phá dữ liệu (Exploratory Data Analysis):** Jupyter cho phép người dùng xem kết quả của code in-line (mã inline) mà không cần phụ thuộc vào các phần khác của code. Trong Notebook mọi ô của code có thể được kiểm tra bất cứ lúc nào, điều này đã giúp Jupyter trở nên khác biệt so với các ID như Pycharm, VSCode. Việc Jupyter có thể xuất code in-line đã giúp ích rất nhiều trong quá trình phân tích khám phá dữ liệu (EDA).
* **Bộ đệm dễ dàng trong ô tích hợp:** Từng ô tự duy trì trạng thái hoạt động sẽ hơi khó, nhưng với Jupyter, công việc này sẽ được thực hiện tự động. Vì Jupyter lưu trữ kết quả hoạt động của mọi ô đang chạy, cho dù là code đang đào tạo mô hình machine learning hay code đang tải xuống gigabyte dữ liệu từ một máy chủ từ xa.
* **Độc lập ngôn ngữ:** Jupyter Notebook ở định dạng JSON, vì thế nó được biết đến là một nền tảng độc lập cũng như độc lập về ngôn ngữ.
* **Trực quan hóa dữ liệu (Data Visualisation):** Jupyter Notebook hỗ trợ trực quan hóa dữ liệu và hiển thị thêm một số đồ họa và biểu đồ. Những điều này được tạo ra từ code với sự trợ giúp của các mô -đun như Matplotlib, Plotly hoặc Bokeh. Ngoài ra, Jupyter còn cho phép người dùng cùng chia sẻ code và bộ dữ liệu hoặc thay đổi tương tác với nhau.
* **Tương tác trực tiếp với code:** Jupyter Notebook sử dụng "ipywidgets" packages, cung cấp cho người dùng giao diện chuẩn nhằm khám phá sự tương tác trực tiếp với code và với dữ liệu. Người dùng có thể chỉnh sửa và chạy code, làm cho code của Jupyter non-static. Ngoài ra, nó còn cho phép người dùng kiểm soát nguồn đầu vào của code và phản hồi lại trực tiếp trên trình duyệt.
* **Các mẫu code tài liệu:** Jupyter giúp người dùng dễ dàng giải thích từng dòng code của họ với các phản hồi được đính kèm. Dù trong code đã có đầy đủ các chức năng nhưng người dùng vẫn có thể tăng thêm sự tương tác bằng các lời giải thích.

## **Code chương trình**

import numpy as np

import pandas as pd

import re

import string

from sklearn.feature\_extraction import text

from sklearn.feature\_extraction.text import CountVectorizer

from sklearn.metrics.pairwise import cosine\_similarity

Đầu tiên ta thêm các thư viện cần thiết cho bài toán như Numpy và Pandas để làm việc với dữ liệu. Thư viện String để xử lý các chuỗi. Trong thư viện Sklearn, ta sử dụng các hàm như text để làm việc với văn bản, hàm CountVectorizer là hàm đếm tần suất xuất hiện của từ trong văn bản. Cuối cùng là hàm cosine\_similarity là hàm so sánh khoảng cách cosine.

anime = pd.read\_csv("D:\Visual studio code\CheckertheAnime\henime2024.csv")

anime.head()

Ta bắt đầu đọc file henime2024.csv đã tải, gán nó với DataFrame có tên anime, sau đó xuất ra 5 hàng đầu tiên của DataFrame.

A screenshot of a computer program

Description automatically generated with medium confidence

anime = anime[['Name', 'Genres', 'Type', 'Episodes']]

anime.head()

Trong DataFrame này, ta sẽ làm việc với các cột Tên phim, Thể loại, Hình thức công chiếu và Số tập phim. Ta chỉ giữ lại 4 cột này và xuất ra màn hình.

A screenshot of a computer

Description automatically generated

def clean(text):

    text = str(text).lower()

    text = re.sub('\[.\*?\]', '', text)

    text = re.sub('https?://\S+|www\.\S+', '', text)

    text = re.sub('<.\*?>+', '', text)

    text = re.sub('[%s]' % re.escape(string.punctuation), '', text)

    text = re.sub('\n', '', text)

    text = re.sub('\w\*\d\w\*', '', text)

    return text

anime["cleaned\_Name"] = anime["Name"].apply(clean)

anime["cleaned\_Genres"] = anime["Genres"].apply(clean)

anime["cleaned\_NameGenres"] = anime["cleaned\_Name"] + " " + anime["cleaned\_Genres"]

anime.head()

Tiếp theo ta xây dựng một hàm có tên clean(). Hàm này sẽ thực hiện nhiệm vụ lọc các văn bản, loại bỏ các dấu câu, kí tự đặc biệt, biến chữ hoa thành chữ thường.

Ta áp dụng hàm clean() cho hai cột “Name” và “Genres”, tạo hai cột mới có tên “cleaned\_Name” và “cleaned\_Genres” chứa các chuỗi đã được lọc. Cuối cùng tạo một cột mới có tên “cleaned\_NameGenres” là sự kết hợp của hai chuỗi “cleaned\_Name” và “cleaned\_Genres” và được cách nhau bởi dấu “ ”.

**A screenshot of a computer

Description automatically generated with medium confidence**

indices = pd.Series(anime.index,index=anime[‘cleaned\_NameGenres’]).drop\_duplicates()

Ta tạo một Serries có tên “indices” có công dụng để tìm kiếm nhanh dựa trên cột “cleaned\_NameGenres” vừa tạo. drop\_duplicates() để loại bỏ các thành phần trùng lặp. Có thể hiểu đơn giản là dòng code này sẽ gán một mã số riêng cho mỗi chuỗi của cột “cleaned\_NameGenres”. Mỗi mã số này sẽ là duy nhất, do vậy ta có thể trỏ đến mã số để lấy thông tin của chuỗi.

def get\_similarity(title):

    new\_title = False

    feature = anime["cleaned\_NameGenres"].tolist()

if not(title in feature):

        new\_title = True

        feature.append(title)

    tfidf = text.CountVectorizer()

    tfidf\_matrix = tfidf.fit\_transform(feature)

    similarity = cosine\_similarity(tfidf\_matrix)

    del tfidf

    del tfidf\_matrix

    if (new\_title):

        del feature

        return similarity[len(similarity) – 1]

    else:

        del feature

        index = pd.Series(indices[title])

        return similarity[index[0]]

Ta xây dựng hàm có tên get\_similarity() dùng để tính độ tương đồng giữa anime nhập vào từ bàn phím và danh sách phim có trong dữ liệu. Đầu tiên ta tạo một biến có tên “new\_title” có giá trị mặc định là False. Biến này sẽ được dùng để kiểm tra xem anime nhập vào có tồn tại trong dữ liệu hay không.

Tạo một list danh sách có tên “feature” bằng cách lấy dữ liệu của cột “cleaned\_NameGenres” và chuyển thành danh sách list.

Một hàm if để kiểm tra xem anime nhập vào có tồn tại trong dữ liệu hay không. Nếu không tồn tại, biến “new\_title” sẽ chuyển thành giá trị True, sau đó thêm anime vừa nhập vào danh sách “feature” sử dụng “append”.

Tạo một đối tượng CountVectorizer có tên “tfidf” để biểu diễn các chuỗi thành ma trận đếm. “tfidf” hay còn gọi là TF-IDF, là một phương pháp phổ biến trong xử lý ngôn ngữ tự nhiên (NLP) được sử dụng để đánh giá độ quan trọng của một từ trong văn bản. Trong phương pháp này, mỗi từ sẽ được đánh giá điểm theo hai yếu tố chính: tần số của từ đó trong văn bản (TF) và tầm quan trong của từ đó trong văn bản (IDF). Tiếp đến ta biến đổi danh sách “feature” thành ma trận đếm sử dụng “fit\_transform” của đối tượng CountVectorizer. Đặt tên ma trận vừa tạo là “tfidf\_matrix”. Cuối cùng tính tính ma trận độ tương đồng Cosine từ ma trận “tfidf\_matrix” và đặt tên là “similarity”.

Ta sử dụng if-else để lấy độ tương đồng “similarity” vừa tính. Nếu anime vừa nhập không tồn tại trong dữ liệu (new\_title = True), anime này sẽ nằm cuối cùng trong ma trận tương đồng “similarity” bằng cách lấy độ dài của “similarity” trừ đi 1.

Nếu anime có tồn tại trong dữ liệu, ta sẽ tìm mã số của anime đó, sử dụng chỉ số “indices”. Cuối cùng trả về hàng tương ứng với chỉ số “index[0]” trong “similarity”. Đây là chỉ số tương đồng giữa anime vừa nhập và các anime khác trong dữ liệu.

def Anime\_Recommendation():

    name = input("Nhập tên Anime (có thể trống): ")

    genres = input("Nhập thể loại phim (có thể trống): ")

    cleaned\_name = clean(name)

    cleaned\_genres = clean(genres)

    cleaned\_NameGenres = cleaned\_name + " " + cleaned\_genres

    similarity = get\_similarity(cleaned\_NameGenres)

    similarity\_scores = list(enumerate(similarity))

    similarity\_scores = sorted(similarity\_scores, key=lambda x: x[1], reverse=True)

    similarity\_scores = filter(lambda x: x[1] > 0, similarity\_scores)

    similarity\_scores = list(similarity\_scores)

movieindices = [i[0] for i in similarity\_scores if i[0] < len(anime)]

    scores = [i[1] for i in similarity\_scores if i[0] < len(anime)]

    result = pd.DataFrame([anime.iloc[i] for i in movieindices])

    result['score'] = scores

    result = result['Name'].values.tolist()

print("Anime Recommendation: ", name)

    print("Tên phim:", name)

    print("Thể loại phim:", genres)

    print("Các Anime gợi ý:")

    print(" ")

    if (len(result) > 10):

        for i in range(0,10):

            print(result[i])

    else:

        for i in range(0, len(result)):

            print(result[i])

    del similarity

    del similarity\_scores

    del movieindices

    del scores

    del result

Anime\_Recommendation()

Cuối cùng là xây dựng hàm Anime\_Recommendation() dùng để gợi ý Anime cho người dùng. Ta tạo hai biến “name” và “genres” có thể nhập từ bàn phím để nhập vào tên anime và thể loại muốn gợi ý. Ta áp dụng hàm clean() cho hai chuỗi vừa nhập để lọc văn bản, sau đó gán hai chuỗi đã sửa vào hai biến “cleaned\_name” và “clean\_genres”. Tạo chuỗi kết hợp giữa hai chuỗi trên, có tên là “cleaned\_NameGenres”, được cách nhau bởi dấu “ ”.

Ta gọi hàm “get\_similarity” đã tạo bên trên với tên “similarity” để tính độ tương đồng giữa chuỗi “cleaned\_NameGenres” vừa nhập và các chuỗi khác trong dữ liệu. Tạo một danh sách có tên “similarity\_scores” chứa 2 thành phần là chỉ số “indices” và điểm số tương đồng từ “enumerate(similarity)”. Ta chọn chỉ số điểm số tương đồng sử dụng hàm “lambda”, sử dụng “sorted” để sắp xếp theo giá trị giảm dần. Sử dụng hàm “filter” để chỉ giữ lại các điểm số tương đồng lớn hơn 0

Tạo một biến “movieindices” là một danh sách chứa chỉ số “indices” của anime trong “similarity\_score” mà có chỉ số nhỏ hơn độ dài của anime.csv. Tạo biến “scores” là danh sách chưa điểm số tương đồng của anime trong “similarity\_score” mà có chỉ số nhỏ hơn độ dài của anime.csv.

Ta tạo một DataFrame mới có tên “result” bằng cách sử dụng vòng lặp for để lấy chỉ số trong “movieindices” và trích xuất hàng tương ứng trong anime.csv. Gán cột “score” cho DataFrame “result” với các giá trị từ danh sách “scores”. Chọn cột “Name” và chuyển đổi thành danh sách list.

Ta in ra màn hình tên anime và thể loại vừa nhập. Sử dụng if-else để in ra danh sách tên anime gợi ý. Hàm if-else dùng để xuất ra 10 phim hoặc nhỏ hơn 10 phim được gợi ý.

Cuối cùng chạy hàm Anime\_Recommendation() để hệ thống gợi ý những bộ Anime có cùng thể loại hoặc giống với tên của phim vừa nhập vào.

# **CHƯƠNG IV. THỰC NGHIỆM VÀ KẾT LUẬN**

## **4.1. Chạy chương trình**

A screenshot of a phone

Description automatically generated with low confidence

Sau khi chạy chương trình, màn hình sẽ hiện lên khung text để người dùng có thể nhập dữ liệu từ bàn phím. Đầu tiên là nhập tên Anime, ta sẽ thử mới cái tên “one piece”.

A screenshot of a computer

Description automatically generated with medium confidence

Nhấn Enter, ta sẽ đến với khung text kế tiếp. Tại đây ta nhập vào thể loại ta muốn tìm và gợi ý. Nếu nhập nhiều thể loại, các thể loại sẽ được cách nhau bởi dấu “,”. Tại đây ta sẽ nhập vào thể loại Hành động, Hài hước và Phiêu lưu.s

A screenshot of a computer

Description automatically generated

Hệ thống sẽ chạy và đưa ra kết quả gợi ý bao gồm 10 phim. Theo kết quả, các phim gợi ý đều là các phim One Piece từ chiếu TV, chiếu rạp đến các phần phim mở rộng. Theo cá nhân mình, các phim gợi ý đều có liên quan đến phim mình vừa nhập, mình sẽ tìm và xem những phim này.

## **4.2. Kết quả đã đạt được và hạn chế**

Bài tập **“Gợi ý Anime theo sở thích người xem”** là một bài tập hay và có tính ứng dụng cao trong đời sống. Qua môn học Khoa học dữ liệu và bài tập lớn này, em đã có thêm kiến thức về xử lý dữ liệu thô, cách làm việc với dữ liệu một cách khoa học và tạo ra một chương trình gợi ý (Recommendation System) đơn giản. Chương trình đã lấy dữ liệu từ trang web Kaggle và gợi ý ra được những phim có cùng tên và thể loại.

Bên cạnh những kết quả đã đạt được, chương trình vẫn còn những điểm hạn chế. Chương trình lấy bộ dữ liệu đã cũ từ năm 2020, vì thế những phim ra mắt sau này sẽ không có trong dữ liệu. Chương trình chưa có tính ứng dụng cao vì cần phải nhập tên và thể loại thủ công từ bàn phím, còn lạc hậu đối với những hệ thống đang được các trang web xem phim lớn sử dụng.

## **4.3. Hướng phát triển của bài tập**

Trong tương lai em sẽ phát triển thêm giao diện người dùng cho chương trình. Ngoài ra, dữ liệu sẽ được nâng cấp, sửa các lỗi phát sinh, kết hợp thêm nhiều hàm và câu lệnh để phù hợp với xu hướng ngày nay, cố gắng để chương trình ngày càng được hoàn thiện và có thể ứng dụng vào thực tiễn.

Trong quá trình làm bài tập tiểu luận, rất khó để tránh khỏi các sai sót không đáng có. Em xin cảm ơn thầy **Nguyễn Tuấn Linh** và các bạn đã giúp đỡ em trong quá trình nghiên cứu và hoàn thành bài tập.

# **TÀI LIỆU THAM KHẢO**

<https://www.kaggle.com/datasets/hernan4444/anime-recommendation-database-2020>

<https://www.kaggle.com/code/christopherreswara/aol-ml-tf-idf>

<https://scikit-learn.org/stable/>

<https://www.dictionary4it.com/term/CountVectorizer-7850/>

<https://www.geeksforgeeks.org/cosine-similarity/>

<https://www.youtube.com/watch?v=hM8UdUt1PqU>

<https://myanimelist.net/>