**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 phim netflix cho người xem*

Sinh viên: Phạm Như Thanh

Lớp: K55KMT

Giáo viên hướng dẫn: Nguyễn Văn Huy

**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: Phạm Như Thanh Lớp: K55KMT*

*MSSV: K1954 8010 6021 Ngành: Kỹ thuật máy tính*

*Ngày giao đề: 24/04/2023 Ngày hoàn thành: 30/05/2023*

1.Tên đề tài : Gợi ý Phim netflix 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. Netflix***

***Hình 4. . Bộ dữ liệu Dataset: Netflix***

***Hình 5. Bộ dữ liệu Netflix.csv***

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

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

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

**MỤC LỤC**

[1](#_Toc136421938)

[**LỜI MỞ ĐẦU** 6](#_Toc136421939)

[**CHƯƠNG I. GIỚI THIỆU CHUNG** 7](#_Toc136421940)

[**1.1.** **Hệ thống gợi ý phim** 7](#_Toc136421941)

[**1.1.1.** **Hệ thống gợi ý là gì ?** 7](#_Toc136421942)

[**1.1.2.** **Hệ thống gợi ý phim là gì ?** 8](#_Toc136421943)

[**1.1.3.** **Ý nghĩa của hệ thống gợi ý phim** 9](#_Toc136421944)

[**1.1.4.** **Thuật toán được sử dụng trong bài tập** 10](#_Toc136421945)

[**1.2.** **Netflix** 10](#_Toc136421946)

[**1.2.1.** **Netflix là gì ?** 10](#_Toc136421947)

[**1.2.2.** **Một số trang web xem Netflix trực tuyến** 11](#_Toc136421948)

[**CHƯƠNG II. PHÂN TÍCH DỮ LIỆU** 12](#_Toc136421949)

[**2.1.** **Bộ dữ liệu sử dụng trong bài** 12](#_Toc136421950)

[**2.1.1.** **Giới thiệu về bộ dữ liệu** 12](#_Toc136421951)

[**2.1.2.** **Các trường trong bộ dữ liệu** 13](#_Toc136421952)

[**2.2.** **Thuật toán tính toán khoảng cách cosine** 14](#_Toc136421953)

[**2.2.1.**  **Khái niệm** 14](#_Toc136421954)

[**2.2.2. Ý nghĩa của thuật toán Cosine Similarity** 15](#_Toc136421955)

[**2.3.** **Thuật toán TF-IDF** 16](#_Toc136421956)

[**2.3.1.**  **Khái niệm** 16](#_Toc136421957)

[**2.3.2.**  **ứng dụng** 17](#_Toc136421958)

[**CHƯƠNG III. THIẾT KẾ CHƯƠNG TRÌNH** 18](#_Toc136421959)

[**3.1. Ngôn ngữ lập trình và môi trường làm việc** 18](#_Toc136421960)

[**3.1.1. Visual Studio Code** 18](#_Toc136421961)

[**3.1.2. Python** 19](#_Toc136421962)

[**3.2.** **Code chương trình** 21](#_Toc136421963)

[**CHƯƠNG IV. THỰC NGHIỆM VÀ KẾT LUẬN** 35](#_Toc136421964)

[**4.1. Chạy chương trình** 35](#_Toc136421965)

[**4.2. Kết quả đã đạt được và hạn chế** 36](#_Toc136421966)

[**4.3. Hướng phát triển của bài tập** 37](#_Toc136421967)

[**TÀI LIỆU THAM KHẢO** 37](#_Toc136421968)

# **LỜI MỞ ĐẦU**

Ngày nay, với sự phát triển vượt bậc của công nghệ và Internet, dịch vụ xem phim trực tuyến đã trở thành một phần không thể thiếu trong cuộc sống hàng ngày của chúng ta. Netflix, một trong những nền tảng phát sóng phim trực tuyến hàng đầu thế giới, đã thu hút hàng triệu người dùng trên toàn thế giới bằng thư viện phim và chương trình đa dạng, từ phim điện ảnh đến chương trình truyền hình.

Tuy nhiên, đôi khi việc tìm kiếm một bộ phim phù hợp để xem có thể trở nên khó khăn với lượng thông tin lớn và sự đa dạng của thư viện phim Netflix. Điều này đã khơi nguồn cho sự phát triển của các hệ thống gợi ý phim, dựa trên các thuật toán và kỹ thuật xử lý dữ liệu trong lĩnh vực Data Science.

Trong tiểu luận này, chúng ta sẽ tìm hiểu về cách sử dụng Data Science để xây dựng một hệ thống gợi ý phim Netflix. Chúng ta sẽ khám phá các kỹ thuật tiền xử lý dữ liệu, tính toán độ tương đồng giữa các phim, và xây dựng mô hình dựa trên ma trận TF-IDF và cosine similarity. Qua đó, chúng ta sẽ có khả năng gợi ý các phim tương tự cho người dùng dựa trên phim mà họ đã xem hoặc yêu thích.

Trong phạm vi bài báo cáo này em xin được trình bày quá trình thực hiện bài tập **“Gợi ý phim Netflix cho 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**

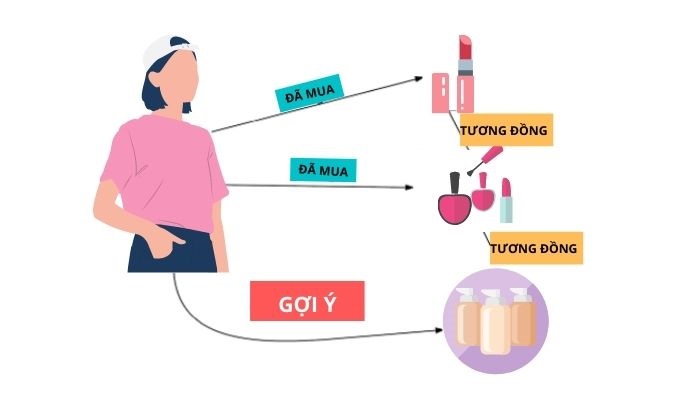
## **Hệ thống gợi ý phim**

### **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 là **Content-based Filtering**. Cụ thể, các bộ phim trong tập dữ liệu được phân tích và trích xuất các đặc trưng như thể loại, đạo diễn, diễn viên, đánh giá, mô tả, và được kết hợp thành một "bag of words" (túi từ). Tiếp theo, sử dụng TF-IDF (Term Frequency-Inverse Document Frequency) để chuyển đổi bag of words thành ma trận TF-IDF. Ma trận này sẽ được sử dụng để tính toán độ tương đồng cosine giữa các bộ phim. Cuối cùng, khi người dùng nhập tên một bộ phim, thuật toán sẽ tìm kiếm bộ phim tương tự dựa trên độ tương đồng cosine và đưa ra các đề xuất phim tương tự.

## **Netflix**

### **Netflix là gì ?**

Netflix là một dịch vụ phát trực tuyến (streaming) cho phép người dùng xem phim, chương trình truyền hình, và nội dung giải trí trên Internet. Netflix cung cấp một thư viện rộng lớn các bộ phim, series truyền hình, show thực tế và nhiều loại nội dung khác từ khắp nơi trên thế giới. Người dùng có thể truy cập vào nền tảng của Netflix thông qua các thiết bị như máy tính, điện thoại thông minh, máy tính bảng, smart TV và các thiết bị phát đa phương tiện khác.



***Hình 3. Netflix***

Netflix không chỉ cung cấp các nội dung phổ biến từ các nhà sản xuất truyền hình và điện ảnh hàng đầu, mà còn sản xuất và phân phối các nội dung gốc của riêng mình, được gọi là "Netflix Originals". Các bộ phim và series truyền hình này thường được sản xuất bởi Netflix hoặc hợp tác với các đội ngũ sản xuất và diễn viên tài năng trên toàn cầu.

Netflix đã trở thành một trong những dịch vụ phát trực tuyến phổ biến nhất trên thế giới, với hàng triệu người dùng trên hầu hết các quốc gia. Mô hình kinh doanh của Netflix cho phép người dùng đăng ký một gói thành viên và truy cập vào toàn bộ thư viện nội dung của họ với một khoản phí hàng tháng. Người dùng có thể xem các phim và chương trình truyền hình mà họ muốn, vào bất kỳ thời điểm nào và từ bất kỳ địa điểm nào có kết nối Internet.

### **Một số trang web xem phim**

Một số trang web tương tự Netflix trong việc cung cấp dịch vụ phát trực tuyến nội dung giải trí. Dưới đây là một số trang web phổ biến như Netflix:

**1. Amazon Prime Video:** Dịch vụ phát trực tuyến của Amazon, cung cấp nhiều bộ phim, series truyền hình và chương trình giải trí. Nó cũng có các bộ phim và series gốc do Amazon sản xuất.

**2. Hulu:** Một dịch vụ phát trực tuyến tại Hoa Kỳ, cung cấp các bộ phim, series truyền hình và nội dung truyền hình từ các kênh truyền hình cáp và đài phát thanh.

**3. Disney+:** Dịch vụ phát trực tuyến của Disney, tập trung vào nội dung của Walt Disney Studios, Marvel, Star Wars và National Geographic.

**4. HBO Max:** Dịch vụ phát trực tuyến của HBO, cung cấp các bộ phim, series truyền hình, nội dung gốc và nội dung từ các kênh truyền hình HBO.

**5. Apple TV+:** Dịch vụ phát trực tuyến của Apple, cung cấp nội dung gốc từ Apple và hợp tác với các nhà sản xuất và diễn viên nổi tiếng.

**6. YouTube Premium:** Dịch vụ trả phí của YouTube, cung cấp nội dung không quảng cáo, phim, series truyền hình và nội dung gốc từ người sáng tạo YouTube.

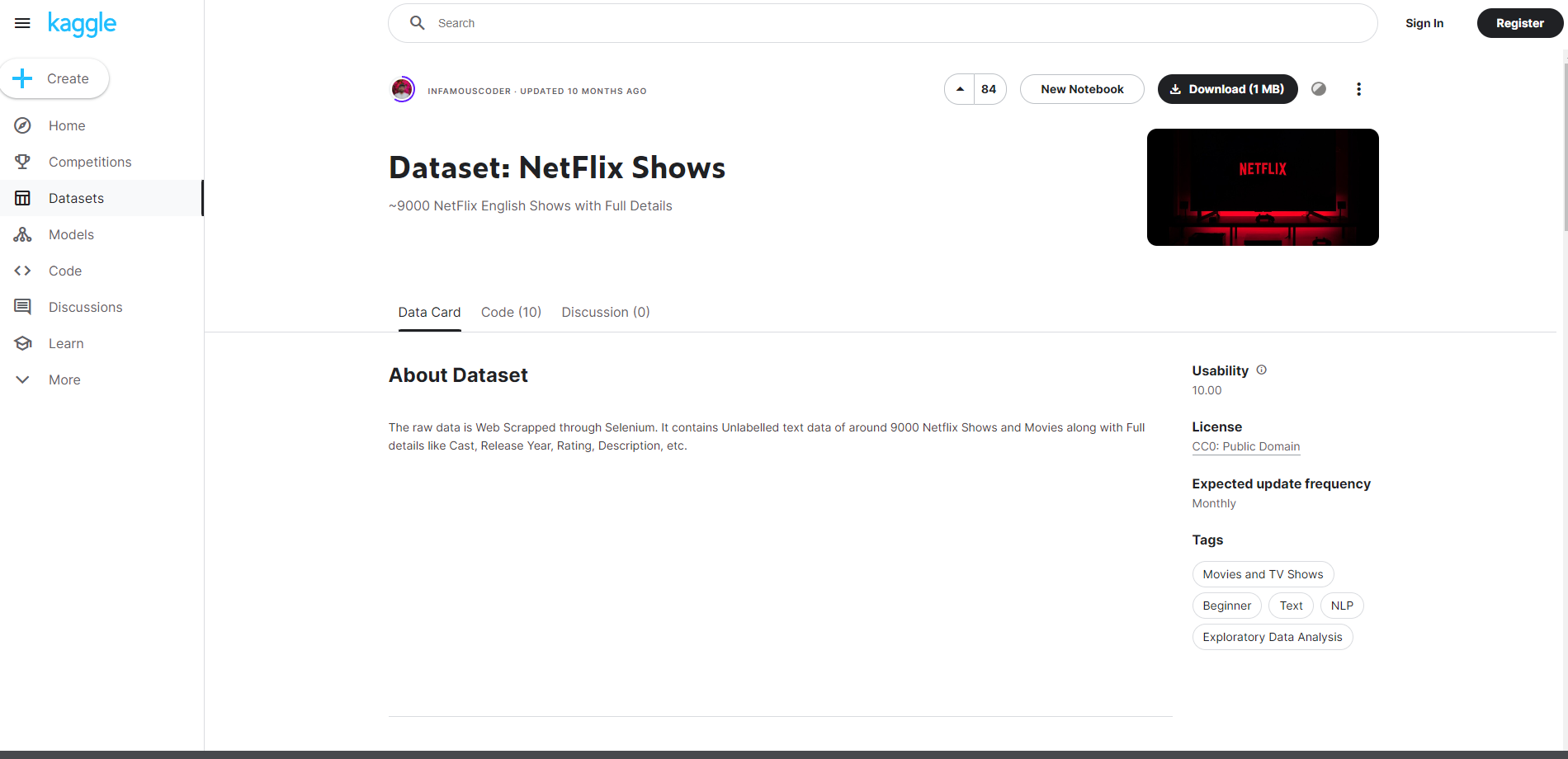
**7. Vudu:** Dịch vụ mua và thuê phim trực tuyến, cung cấp một thư viện rộng lớn các bộ phim và chương trình truyền hình.

# **CHƯƠNG II. PHÂN TÍCH DỮ LIỆU**

## **Bộ dữ liệu sử dụng trong bài**

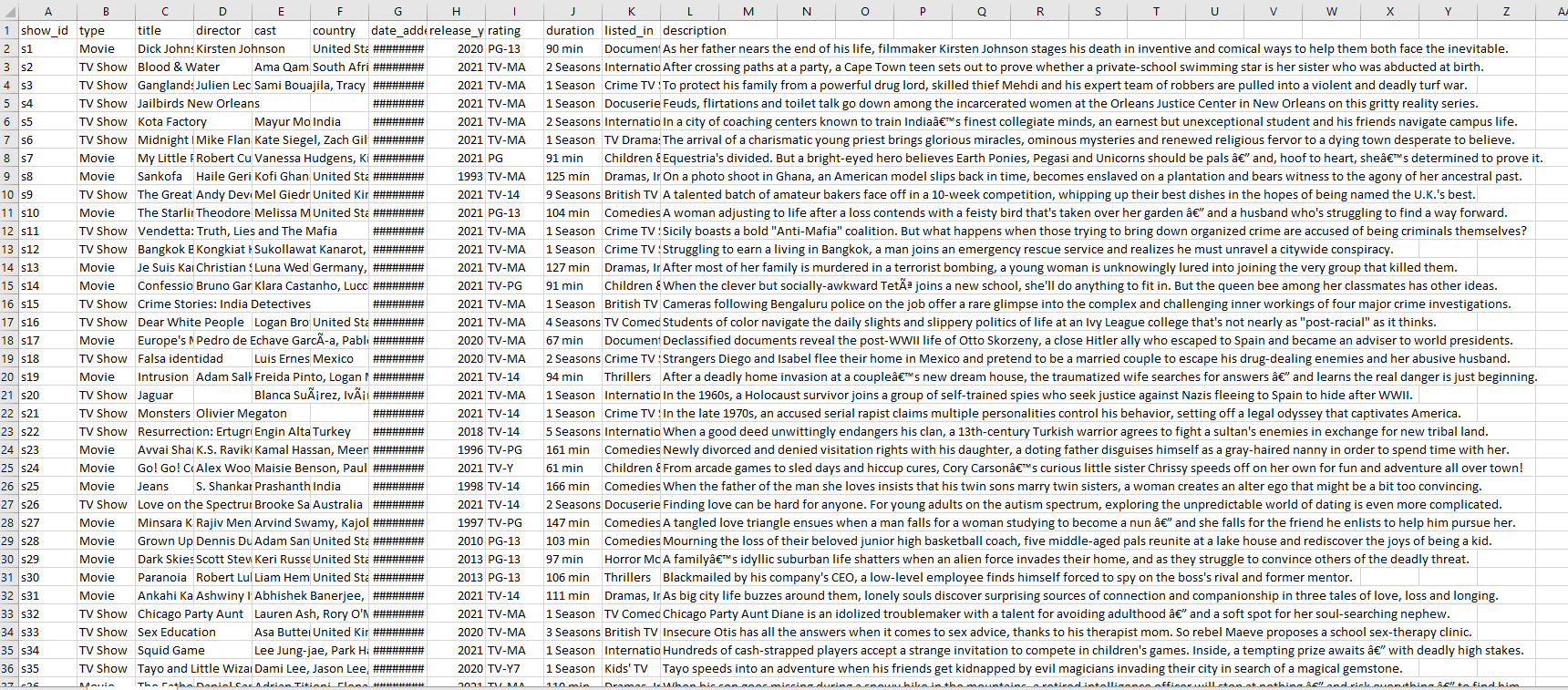
### **Giới thiệu về bộ dữ liệu**

Bộ dữ liệu sử dụng trong bài tập là được lấy từ trang web **Kaggle**, chứa thông tin về 8.807 bộ phim và show truyền hình trên Netflix.



***Hình 4. Bộ dữ liệu Dataset: Netflix***

### **Các trường trong bộ dữ liệu**

A screen shot of a computer

Description automatically generated with low confidence

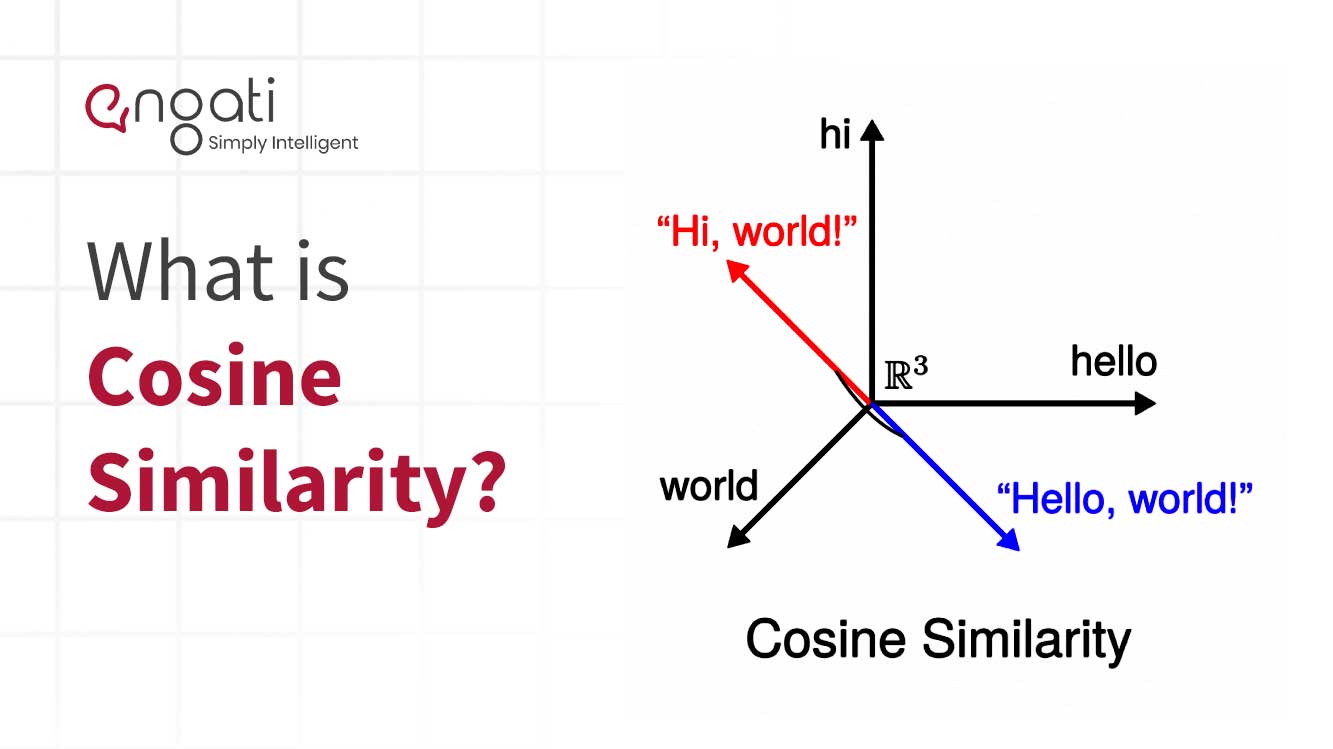
***Hình 5. Bộ dữ liệu netflix.csv sau khi tải về***

* **Show\_id:** Mã số của bộ phim trên netflix
* **Type:** Hình thức Movie hoặc TV show
* **Title:** Tiêu đề bộ phim
* **Director:** Đạo diễn
* **Cast:** Dàn diễn viên
* **Country:** Đất nước
* **Date\_added:** Ngày cập nhật
* **Release\_year:** Năm phát hành
* **Rating:** Đánh giá
* **Duration:** Thời lượng
* **Listed\_in:** Thể loại phim
* **Description:** Mô tả

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

### **2.2.1. Khái niệm**

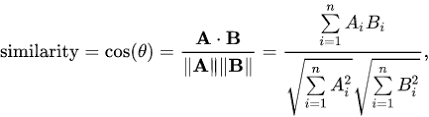
Thuật toán tính toán khoảng cách cosine (Cosine Similarity) là một phương pháp đo đạc mức độ tương đồng giữa hai vectơ trong không gian đa chiều. Nó được sử dụng rộng rãi trong lĩnh vực xử lý ngôn ngữ tự nhiên và khai phá dữ liệu, thường được sử dụng để so sánh sự tương đồng giữa các văn bản, từ ngữ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

### **2.3.1. Khái niệm**

TF-IDF (viết tắt của term frequency – inverse document frequency) là một phương thức thống kê thường được sử dụng trong mảng truy xuất thông tin (information retrieval) và khai phá dữ liệu văn bản (text mining) để đánh giá mức độ quan trọng của một cụm từ đối với một tài liệu cụ thể trong một tập hợp bao gồm nhiều tài liệu. Khái niệm này đã xuất hiện từ rất sớm trong các lĩnh vực nghiên cứu khác nhau, chẳng hạn như ngôn ngữ học (linguistics) và cấu trúc thông tin (information architecture), nhờ vào khả năng hỗ trợ xử lý nhiều tập tài liệu với số lượng lớn trong một khoảng thời gian ngắn.

Các máy tìm kiếm thường sử dụng các biến số khác nhau của thuật toán TF-IDF như là một phần trong cơ chế xếp hạng. Bằng cách gán cho các tài liệu một mức điểm số về độ liên quan (relevance score), chúng có thể đưa ra các kết quả tìm kiếm thích hợp chỉ trong phần triệu giây.

**TF- term frequency** – tần số xuất hiện của 1 từ trong 1 văn bản. Cách tính:



* Thương của số lần xuất hiện 1 từ trong văn bản và số lần xuất hiện nhiều nhất của một từ bất kỳ trong văn bản đó. (giá trị sẽ thuộc khoảng [0, 1])
* **f(t,d)** - số lần xuất hiện từ t trong văn bản **d**.
* **max{f(w,d):w∈d}** - số lần xuất hiện nhiều nhất của một từ bất kỳ trong văn bản.

**IDF – inverse document frequency.** Tần số nghịch của 1 từ trong tập văn bản (corpus).

Tính IDF để giảm giá trị của những từ phổ biến. Mỗi từ chỉ có 1 giá trị IDF duy nhất trong tập văn bản.



* : - tổng số văn bản trong tập **D**
* |{�∈�:�∈�}|: - số văn bản chứa từ nhất định, với điều kiện � xuất hiện trong văn bản d (i.e., tf(�,�)≠0). Nếu từ đó không xuất hiện ở bất cứ 1 văn bản nào trong tập thì mẫu số sẽ bằng 0 => phép chia cho không không hợp lệ, vì thế người ta thường thay bằng mẫu thức 1+|{�∈�:�∈�}|.

Cơ số logarit trong công thức này không thay đổi giá trị của 1 từ mà chỉ thu hẹp khoảng giá trị của từ đó. Vì thay đổi cơ số sẽ dẫn đến việc giá trị của các từ thay đổi bởi một số nhất định và tỷ lệ giữa các trọng lượng với nhau sẽ không thay đổi. (nói cách khác, thay đổi cơ số sẽ không ảnh hưởng đến tỷ lệ giữa các giá trị IDF). Tuy nhiên việc thay đổi khoảng giá trị sẽ giúp tỷ lệ giữa IDF và TF tương đồng để dùng cho công thức TF-IDF như bên dưới.

Giá trị **TF-IDF**:



Những từ có giá trị TF-IDF cao là những từ xuất hiện nhiều trong văn bản này, và xuất hiện ít trong các văn bản khác. Việc này giúp lọc ra những từ phổ biến và giữ lại những từ có giá trị cao (từ khoá của văn bản đó)

Ví dụ, TF-IDF từ lâu đã là một phần trong cơ chế xếp hạng của Google. Google sử dụng TF-IDF để xác định xem những cụm từ nào có liên quan (hoặc không liên quan) về mặt chủ đề bằng cách phân tích tần suất một cụm từ xuất hiện trên một trang (term frequency – TF) và tần suất ước tính xuất hiện trên một trang trung bình, trong một tập hợp lớn hơn bao gồm nhiều tài liệu (inverse document frequency – IDF).

### **2.3.2. ứng dụng**

IDF có ứng dụng trong máy tìm kiếm. Ví dụ, khi người dùng gửi một truy vấn đến máy tìm kiếm, hệ thống cần biết từ nào là từ người dùng quan tâm nhất. Chẳng hạn: truy vấn của người dùng là "làm thế nào để sửa máy ủi". Sau khi tách từ, chúng ta sẽ có tập các từ: làm, thế nào, để, sửa, máy ủi. Trong các từ này, "máy ủi" sẽ có IDF cao nhất. Hệ thống sẽ lấy ra tất cả các văn bản có chứa từ máy ủi và sau đó mới thực hiện việc đánh giá và so sánh dựa trên toàn bộ câu truy vấn.

# **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) 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). 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) theo [giấy phép MIT](https://vi.wikipedia.org/wiki/Gi%E1%BA%A5y_ph%C3%A9p_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) 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) 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").



***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). 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) 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). 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) 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); 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). 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) Python Software Foundation quản lý.



***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). [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) và [lập trình cấu trúc](https://vi.wikipedia.org/wiki/L%E1%BA%ADp_tr%C3%ACnh_c%E1%BA%A5u_tr%C3%BAc) 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) (bao gồm [siêu lập trình](https://vi.wikipedia.org/wiki/L%E1%BA%ADp_tr%C3%ACnh_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). 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) và [lập trình logic](https://vi.wikipedia.org/wiki/L%E1%BA%ADp_tr%C3%ACnh_logic).

**Python** sử dụng [kiểu động](https://vi.wikipedia.org/wiki/H%E1%BB%87_th%E1%BB%91ng_ki%E1%BB%83u) 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) 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). 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) 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)), 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), 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) (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) 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)", 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."

## **Code chương trình**

* **“import string**

**import numpy as np**

**import pandas as pd**

**import sklearn**

**import warnings**

**warnings.filterwarnings('ignore')”**

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

import warnings: Import module warnings để điều chỉnh cách xử lý các cảnh báo trong Python.

warnings.filterwarnings('ignore'): Đặt cấu hình để tắt cảnh báo trong quá trình chạy code.

* **“df=pd.read\_csv("C:\\Users\\phamt\\Downloads\\netflixData.csv\\netflix\_titles.csv")”**

df.head() Ta bắt đầu đọc file netflix.csv đã tải, gán nó với DataFrame có tên df, sau đó xuất ra 5 hàng đầu tiên của DataFrame.



* **“df.describe(include='all')”**

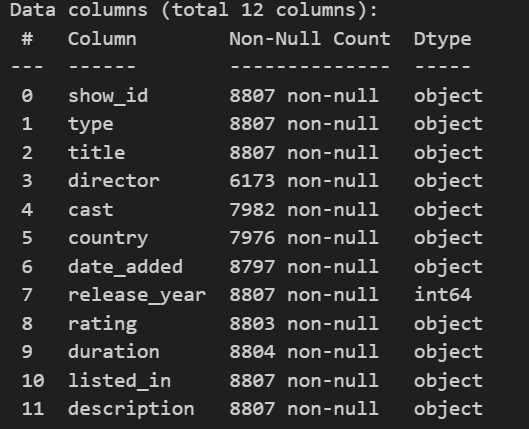
được sử dụng để tạo báo cáo thống kê mô tả về dữ liệu trong DataFrame df.

Phương thức describe() của pandas được gọi trên DataFrame để tính toán các thống kê mô tả như số lượng, giá trị trung bình, độ lệch chuẩn, giá trị tối thiểu, giá trị tối đa và các phần vị trí (25%, 50%, 75%) cho các cột trong DataFrame.

Đối số include='all' được sử dụng để bao gồm cả các cột không phải là số trong báo cáo thống kê mô tả. Mặc định, chỉ các cột số liệu được bao gồm trong báo cáo. Khi sử dụng include='all', báo cáo sẽ cung cấp thông tin về cả các cột không phải là số liệu, bao gồm số lượng giá trị duy nhất, giá trị phổ biến nhất và tần suất của chúng.

Kết quả trả về sẽ là một DataFrame chứa báo cáo thống kê mô tả về dữ liệu trong df, bao gồm các cột và các thống kê tương ứng.

* **“df.info()”**



Khi gọi df.info(), nó sẽ hiển thị các thông tin sau:

Số lượng dòng trong DataFrame.

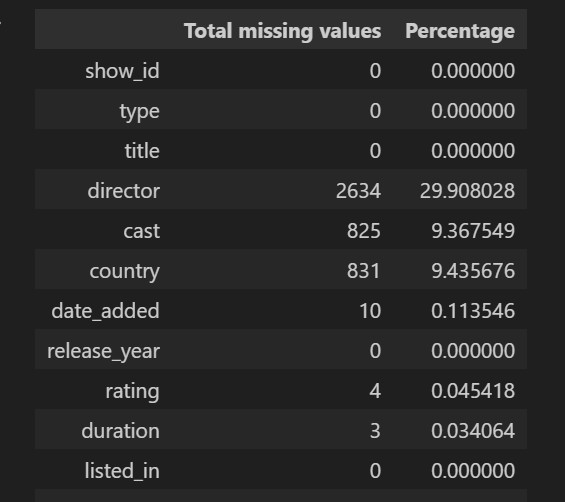
Số lượng cột trong DataFrame.

Tên cột và các thông tin liên quan như số lượng giá trị không null (không thiếu), kiểu dữ liệu và bộ nhớ sử dụng.

Tổng số lượng bộ nhớ sử dụng bởi DataFrame.

* **“pd.DataFrame({'Total missing values':df.isna().sum(),**

**'Percentage':(df.isna().sum()/len(df))\*100})**”



Đoạn mã “pd.DataFrame({'Total missing values': df.isna().sum(), 'Percentage': (df.isna().sum()/len(df))\*100})” tạo ra một DataFrame mới để hiển thị thông tin về số lượng giá trị thiếu (null) trong từng cột của DataFrame df và tỷ lệ phần trăm giá trị thiếu so với tổng số lượng dòng trong DataFrame.

“df.isna().sum()” là một phương thức của DataFrame df kết hợp với phương thức sum(). Nó được sử dụng để đếm số lượng giá trị thiếu (null) trong từng cột của DataFrame. Kết quả là một Series, trong đó chỉ số của Series là tên các cột và giá trị là số lượng giá trị thiếu trong từng cột.

“(df.isna().sum()/len(df))\*100” tính tỷ lệ phần trăm giá trị thiếu trong từng cột. Đầu tiên, df.isna().sum() đếm số lượng giá trị thiếu trong từng cột. Sau đó, chia cho len(df) (tổng số lượng dòng trong DataFrame) để tính tỷ lệ phần trăm, và cuối cùng nhân với 100 để đưa ra kết quả dưới dạng phần trăm. Kết quả cũng là một Series với chỉ số là tên các cột và giá trị là tỷ lệ phần trăm giá trị thiếu trong từng cột.

“pd.DataFrame(...)” tạo ra một DataFrame mới từ các Series trên. Chỉ số của DataFrame là tên các cột, và các cột trong DataFrame bao gồm 'Total missing values' (số lượng giá trị thiếu) và 'Percentage' (tỷ lệ phần trăm).

Đoạn mã trên cung cấp thông tin về số lượng giá trị thiếu và tỷ lệ phần trăm giá trị thiếu trong từng cột của DataFrame, giúp phân tích dữ liệu và xử lý giá trị thiếu khi cần thiết

* **“new\_df = df[['title', 'type', 'director', 'cast', 'rating', 'listed\_in', 'description']]**

**new\_df.set\_index('title', inplace=True)**

**new\_df.head()”**



Đoạn mã “new\_df = df[['title', 'type', 'director', 'cast', 'rating', 'listed\_in', 'description']] tạo ra một DataFrame” mới có tên là new\_df từ DataFrame ban đầu df. DataFrame new\_df chỉ bao gồm các cột 'title', 'type', 'director', 'cast', 'rating', 'listed\_in' và 'description' từ DataFrame df.

Sau đó, new\_df.set\_index('title', inplace=True) sử dụng cột 'title' làm chỉ số (index) của DataFrame new\_df. Tham số inplace=True được truyền vào để thay đổi DataFrame new\_df gốc mà không cần tạo một DataFrame mới. Chức năng này cho phép truy xuất các dòng trong DataFrame new\_df bằng giá trị của cột 'title' thay vì chỉ số dòng mặc định.

Cuối cùng, new\_df.head() được sử dụng để hiển thị năm dòng đầu tiên của DataFrame new\_df.

* **“new\_df.fillna('',inplace=True)”**

Đoạn mã “new\_df.fillna('', inplace=True)” được sử dụng để thay thế các giá trị thiếu (NaN) trong DataFrame new\_df bằng giá trị rỗng (''). Tham số inplace=True được truyền vào để thay đổi DataFrame new\_df gốc mà không cần tạo một DataFrame mới. Chức năng này giúp chuẩn bị dữ liệu cho các bước xử lý tiếp theo bằng cách xử lý các giá trị thiếu trong DataFrame. Trong trường hợp này, giá trị rỗng ('') được sử dụng để đại diện cho các giá trị bị thiếu trong DataFrame.

* **“def separate(texts):**

**t = []**

**for text in texts.split(','):**

**t.append(text.replace(' ', '').lower())**

**return ' '.join(t)**

**def remove\_space(texts):**

**return texts.replace(' ', '').lower()**

**def remove\_punc(texts):**

**return texts.translate(str.maketrans('','',string.punctuation)).lower()”**

Các hàm separate, remove\_space và remove\_punc được định nghĩa trong đoạn mã trên để thực hiện các xử lý văn bản trên các chuỗi dữ liệu.

Hàm separate(texts): Nhận đầu vào là một chuỗi texts và thực hiện các bước sau:

Tách chuỗi thành các thành phần con bằng cách sử dụng dấu phẩy (,) làm dấu phân cách. Điều này giả định rằng chuỗi texts chứa các thành phần được phân tách bằng dấu phẩy.

Loại bỏ khoảng trắng trong mỗi thành phần bằng cách sử dụng phương thức replace(' ', '').

Chuyển đổi thành phần thành chữ thường bằng phương thức lower().

Trả về chuỗi đã được xử lý bằng cách sử dụng phương thức join(t) để kết hợp các thành phần lại thành một chuỗi mới, với các thành phần cách nhau bởi dấu cách.

Hàm remove\_space(texts): Nhận đầu vào là một chuỗi texts và thực hiện các bước sau:

Loại bỏ khoảng trắng trong chuỗi bằng cách sử dụng phương thức replace(' ', '').

Chuyển đổi chuỗi thành chữ thường bằng phương thức lower().

Trả về chuỗi đã được xử lý.

Hàm remove\_punc(texts): Nhận đầu vào là một chuỗi texts và thực hiện các bước sau:

Loại bỏ các dấu câu trong chuỗi bằng cách sử dụng phương thức translate(str.maketrans('', '', string.punctuation)). Điều này sử dụng phương thức str.maketrans() để tạo bảng dịch với các dấu câu được xóa.

Chuyển đổi chuỗi thành chữ thường bằng phương thức lower().

Trả về chuỗi đã được xử lý.

* **“new\_df['type'] = new\_df['type'].apply(remove\_space)**

**new\_df['director'] = new\_df['director'].apply(separate)**

**new\_df['cast'] = new\_df['cast'].apply(separate)**

**new\_df['rating'] = new\_df['rating'].apply(remove\_space)**

**new\_df['listed\_in'] = new\_df['listed\_in'].apply(separate)**

**new\_df['description'] = new\_df['description'].apply(remove\_punc)**

* **“new\_df.head()”**



Đoạn mã trên thực hiện các xử lý văn bản trên các cột trong new\_df để chuẩn bị dữ liệu cho việc tạo bộ từ vựng và tính độ tương đồng cosine.

new\_df['type'] = new\_df['type'].apply(remove\_space): Áp dụng hàm remove\_space lên cột 'type' trong new\_df. Hàm này sẽ loại bỏ khoảng trắng trong chuỗi và chuyển đổi chuỗi thành chữ thường. Kết quả được gán lại vào cột 'type'.

new\_df['director'] = new\_df['director'].apply(separate): Áp dụng hàm separate lên cột 'director' trong new\_df. Hàm này sẽ tách các thành phần trong chuỗi bằng dấu phẩy, loại bỏ khoảng trắng và chuyển đổi thành chữ thường cho mỗi thành phần. Kết quả được gán lại vào cột 'director'.

new\_df['cast'] = new\_df['cast'].apply(separate): Áp dụng hàm separate lên cột 'cast' trong new\_df. Hàm này sẽ tách các thành phần trong chuỗi bằng dấu phẩy, loại bỏ khoảng trắng và chuyển đổi thành chữ thường cho mỗi thành phần. Kết quả được gán lại vào cột 'cast'.

new\_df['rating'] = new\_df['rating'].apply(remove\_space): Áp dụng hàm remove\_space lên cột 'rating' trong new\_df. Hàm này sẽ loại bỏ khoảng trắng trong chuỗi và chuyển đổi chuỗi thành chữ thường. Kết quả được gán lại vào cột 'rating'.

new\_df['listed\_in'] = new\_df['listed\_in'].apply(separate): Áp dụng hàm separate lên cột 'listed\_in' trong new\_df. Hàm này sẽ tách các thành phần trong chuỗi bằng dấu phẩy, loại bỏ khoảng trắng và chuyển đổi thành chữ thường cho mỗi thành phần. Kết quả được gán lại vào cột 'listed\_in'.

new\_df['description'] = new\_df['description'].apply(remove\_punc): Áp dụng hàm remove\_punc lên cột 'description' trong new\_df. Hàm này sẽ loại bỏ các dấu câu trong chuỗi và chuyển đổi chuỗi thành chữ thường. Kết quả được gán lại vào cột 'description'.

Cuối cùng, new\_df.head() được sử dụng để hiển thị 5 dòng đầu tiên của new\_df sau khi đã xử lý.

* **“new\_df['bag\_of\_words'] = ''**

**for i, row in enumerate(new\_df.iterrows()):**

**string = ''**

**for col in new\_df.columns:**

**if row[1][col] == '':**

**continue**

**else:**

**string += row[1][col] + ' '**

**new\_df['bag\_of\_words'][i] = string.strip()**

**new\_df.drop(new\_df.columns[:-1], axis=1, inplace=True)”**

Đoạn mã trên thực hiện các bước sau:

new\_df['bag\_of\_words'] = '': Tạo một cột mới trong new\_df có tên là 'bag\_of\_words' và khởi tạo giá trị mặc định là chuỗi rỗng.

Sử dụng vòng lặp enumerate(new\_df.iterrows()) để duyệt qua từng dòng trong new\_df và lấy chỉ mục và nội dung của từng dòng.

Khởi tạo một chuỗi rỗng string.

Traversing qua các cột trong new\_df.columns (title, type, director, cast, rating, listed\_in, description), kiểm tra giá trị của từng cột trong dòng hiện tại (row[1][col]).

Nếu giá trị của cột là rỗng (''), bỏ qua và tiếp tục với cột tiếp theo.

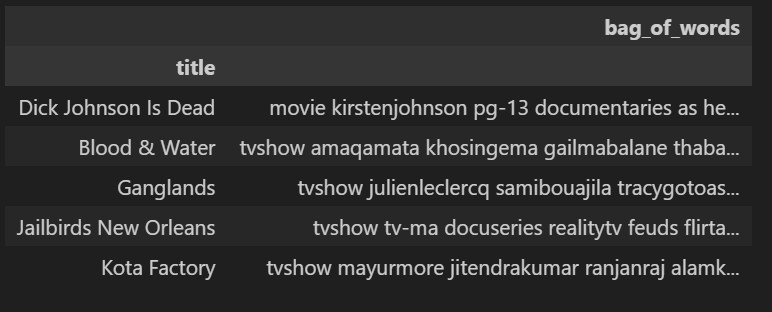
Nếu giá trị của cột không rỗng, thêm giá trị đó vào chuỗi string, cách nhau bởi một khoảng trắng.

Gán giá trị của chuỗi string đã được xây dựng vào cột 'bag\_of\_words' tại chỉ mục tương ứng (new\_df['bag\_of\_words'][i] = string.strip()).

Cuối cùng, sử dụng new\_df.drop(new\_df.columns[:-1], axis=1, inplace=True) để loại bỏ các cột không cần thiết (title, type, director, cast, rating, listed\_in, description) và giữ lại chỉ cột 'bag\_of\_words'.

Cuối cùng, new\_df.head() được sử dụng để hiển thị 5 dòng đầu tiên của new\_df sau khi đã thêm cột 'bag\_of\_words' và loại bỏ các cột không cần thiết.

* **“new\_df.head()”**



Hiển thị 5 dòng đầu tiên của DataFrame "new\_df" sau khi thực hiện các thay đổi.

* **“from sklearn.feature\_extraction.text import TfidfVectorizer**

**from sklearn.metrics.pairwise import cosine\_similarity”**

from sklearn.feature\_extraction.text import TfidfVectorizer: Đây là một class trong sklearn được sử dụng để chuyển đổi văn bản thành vectơ đặc trưng TF-IDF (Term Frequency-Inverse Document Frequency). TF-IDF là một phương pháp đánh giá sự quan trọng của một từ trong một văn bản dựa trên tần suất xuất hiện của từ đó trong văn bản và tần suất xuất hiện của từ đó trong toàn bộ tập văn bản. TfidfVectorizer thực hiện việc tính toán và trích xuất các đặc trưng TF-IDF từ văn bản.

from sklearn.metrics.pairwise import cosine\_similarity: Đây là một hàm trong sklearn được sử dụng để tính độ tương đồng cosine giữa các vectơ đặc trưng. Độ tương đồng cosine là một phương pháp đo lường sự tương đồng giữa hai vectơ dựa trên góc giữa chúng trong không gian đa chiều. Hàm cosine\_similarity trong sklearn tính toán ma trận độ tương đồng cosine giữa các vectơ đặc trưng để đo lường sự tương đồng giữa các mẫu trong không gian đặc trưng.

* **“tfid = TfidfVectorizer()**

**tfid\_matrix = tfid.fit\_transform(new\_df['bag\_of\_words'])”**

Chúng ta tạo một đối tượng TfidfVectorizer có tên là tfid bằng cách khởi tạo một phiên bản của lớp TfidfVectorizer từ sklearn. Sau đó, chúng ta sử dụng phương thức fit\_transform của đối tượng tfid để biến đổi dữ liệu văn bản trong cột 'bag\_of\_words' của DataFrame new\_df thành ma trận đặc trưng TF-IDF.

Cụ thể, fit\_transform sẽ thực hiện hai công việc:

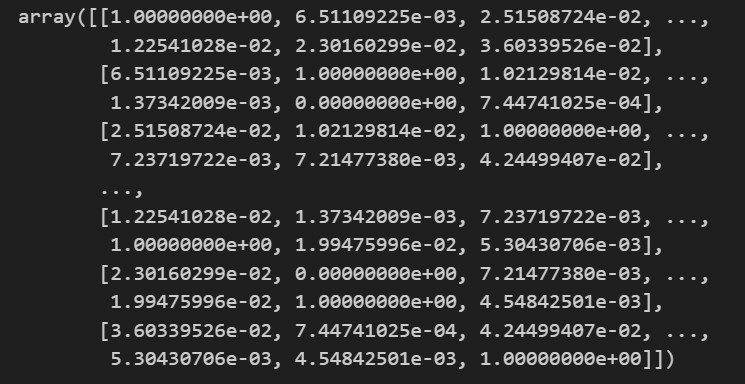
Phương thức fit sẽ học từ điển từ vựng từ dữ liệu văn bản đầu vào. Nó xác định các từ duy nhất trong tất cả các văn bản và xây dựng một từ điển từ vựng.

Phương thức transform sẽ biến đổi dữ liệu văn bản thành ma trận đặc trưng TF-IDF, sử dụng từ điển từ vựng đã học từ bước trước đó. Kết quả là một ma trận thưa (sparse matrix) có kích thước là số lượng văn bản x số lượng từ trong từ điển.

Cuối cùng, ma trận đặc trưng TF-IDF được lưu trong biến tfid\_matrix để sử dụng trong các phân tích và tính toán tương đồng sau này.

* **“cosine\_sim = cosine\_similarity(tfid\_matrix, tfid\_matrix)**

**cosine\_sim”**



Sử dụng hàm cosine\_similarity từ sklearn để tính toán ma trận tương đồng cosine giữa các vector đặc trưng trong tfid\_matrix. cosine\_similarity nhận đầu vào là hai ma trận hoặc mảng có cùng số chiều, và trả về một ma trận tương đồng cosine có kích thước là số lượng vector trong đầu vào.

Ở đây, chúng ta đưa tfid\_matrix vào hai lần để tính toán tương đồng cosine giữa các vector trong tfid\_matrix với chính nó. Kết quả là ma trận cosine\_sim, trong đó mỗi phần tử (i, j) thể hiện mức độ tương đồng cosine giữa vector thứ i và vector thứ j trong tfid\_matrix.

Ma trận cosine\_sim được sử dụng sau này để xác định độ tương đồng giữa các phim dựa trên vector đặc trưng TF-IDF của chúng.

* **“final\_df = df[['title', 'type']]”**

tạo một DataFrame mới có tên là final\_df từ DataFrame gốc df. DataFrame mới này chỉ chứa hai cột là 'title' và 'type', được lựa chọn từ df.

Các cột được chọn là 'title' và 'type' để sử dụng trong việc hiển thị kết quả gợi ý sau này.

* **“def recommendation(title, total\_result=5, threshold=0.5):**

**idx = final\_df[final\_df['title'] == title].index[0]**

**final\_df['similarity'] = cosine\_sim[idx]**

**sort\_final\_df = final\_df.sort\_values(by='similarity', ascending=False)[1:total\_result+1]**

**result\_list = []**

**movies = sort\_final\_df['title'][sort\_final\_df['type'] == 'Movie']**

**tv\_shows = sort\_final\_df['title'][sort\_final\_df['type'] == 'TV Show']**

**if len(movies) != 0:**

**result\_list.append('Similar Movie(s) list:')**

**for i, movie in enumerate(movies):**

**result\_list.append('{}. {}'.format(i+1, movie))**

**result\_list.append('')**

**else:**

**result\_list.append('Similar Movie(s) list:')**

**result\_list.append('-\n')**

**if len(tv\_shows) != 0:**

**result\_list.append('Similar TV\_show(s) list:')**

**for i, tv\_show in enumerate(tv\_shows):**

**result\_list.append('{}. {}'.format(i+1, tv\_show))**

**else:**

**result\_list.append('Similar TV\_show(s) list:')**

**result\_list.append('-')**

**return result\_list”**

Hàm có tên là recommendation để thực hiện quá trình gợi ý phim tương tự. Dưới đây là giải thích từng bước của hàm:

idx = final\_df[final\_df['title'] == title].index[0]: Tìm chỉ mục của phim có tiêu đề trùng khớp với title đầu vào trong DataFrame final\_df. Chúng ta sử dụng chỉ mục này để truy cập vào ma trận cosine\_sim và lấy độ tương tự của phim đó với các phim khác.

final\_df['similarity'] = cosine\_sim[idx]: Gán các giá trị tương tự từ ma trận cosine\_sim vào cột 'similarity' trong DataFrame final\_df.

sort\_final\_df=final\_df.sort\_values(by='similarity', ascending=False)[1:total\_result+1]: Sắp xếp DataFrame final\_df theo cột 'similarity' giảm dần và chọn total\_result phim có độ tương tự cao nhất (loại bỏ phim đầu tiên vì đó là chính phim đầu vào).

result\_list = []: Khởi tạo một danh sách rỗng để lưu kết quả gợi ý.

movies = sort\_final\_df['title'][sort\_final\_df['type'] == 'Movie']: Lấy danh sách các phim trong sort\_final\_df có loại là 'Movie'.

tv\_shows = sort\_final\_df['title'][sort\_final\_df['type'] == 'TV Show']: Lấy danh sách các phim trong sort\_final\_df có loại là 'TV Show'.

Kiểm tra và thêm các phim tương tự vào danh sách kết quả result\_list. Nếu danh sách phim tương tự không rỗng, chúng ta thêm tiêu đề "Similar Movie(s) list:" vào result\_list và liệt kê các phim trong danh sách movies. Tương tự, chúng ta kiểm tra danh sách phim truyền hình tương tự và thêm tiêu đề "Similar TV\_show(s) list:" vào result\_list và liệt kê các phim trong danh sách tv\_shows.

Trả về danh sách kết quả result\_list chứa các phim tương tự được gợi ý.

* **“import tkinter as tk**

**import pandas as pd**

**from fuzzywuzzy import fuzz**

**from fuzzywuzzy import process**

**def recommend\_movie():**

**title = entry.get()**

**result\_text.delete('1.0', tk.END) # Xóa nội dung hiển thị trước đó**

**# Tìm kiếm các phim gần đúng với tên nhập vào**

**fuzzy\_results = process.extract(title, df['title'], limit=10, scorer=fuzz.ratio)**

**similar\_movies = [result[0] for result in fuzzy\_results]**

**# Hiển thị thông tin về các phim tương tự**

**for movie\_title in similar\_movies:**

**movie\_info = df[df['title'] == movie\_title]**

**if not movie\_info.empty:**

**result\_text.insert(tk.END, "Tên phim: " + movie\_info['title'].iloc[0] + "\n")**

**result\_text.insert(tk.END, "Loại: " + movie\_info['type'].iloc[0] + "\n")**

**result\_text.insert(tk.END, "Đạo diễn: " + str(movie\_info['director'].iloc[0]) + "\n")**

**result\_text.insert(tk.END, "Diễn viên: " + str(movie\_info['cast'].iloc[0]) + "\n")**

**result\_text.insert(tk.END, "Đánh giá: " + movie\_info['rating'].iloc[0] + "\n")**

**result\_text.insert(tk.END, "Thể loại: " + str(movie\_info['listed\_in'].iloc[0]) + "\n")**

**result\_text.insert(tk.END, "Mô tả: " + movie\_info['description'].iloc[0] + "\n")**

**result\_text.insert(tk.END, "-----------------------------------\n")**

**else:**

**result\_text.insert(tk.END, "Không tìm thấy thông tin về phim: " + movie\_title + "\n")**

**result\_text.insert(tk.END, "\n")**

**# Tạo giao diện**

**window = tk.Tk()**

**window.title("Netflix Movie Recommendation")**

**window.geometry("400x300")**

**label = tk.Label(window, text="Nhập tên phim:")**

**label.pack()**

**entry = tk.Entry(window)**

**entry.pack()**

**button = tk.Button(window, text="Đề xuất", command=recommend\_movie)**

**button.pack()**

**result\_text = tk.Text(window)**

**result\_text.pack()**

**# Đọc dữ liệu từ file CSV**

**df = pd.read\_csv("netflix\_titles.csv")**

**window.mainloop()”**

Sử dụng thư viện tkinter để tạo một giao diện đơn giản cho việc gợi ý phim từ dữ liệu trong file CSV. Dưới đây là giải thích từng phần của mã:

Import các thư viện cần thiết:

tkinter để tạo giao diện người dùng.

pandas để làm việc với dữ liệu dạng DataFrame.

fuzzywuzzy để thực hiện tìm kiếm các phim gần đúng dựa trên sự tương đồng chuỗi.

Định nghĩa hàm recommend\_movie(): Đây là hàm được gọi khi người dùng nhấn vào nút "Đề xuất". Hàm này thực hiện các bước sau:

Lấy tên phim từ trường nhập liệu (entry).

Xóa nội dung hiển thị trước đó trong result\_text.

Sử dụng fuzzywuzzy để tìm kiếm các phim gần đúng với tên nhập vào. Giới hạn kết quả tìm kiếm là 10 phim, và sử dụng hàm tính điểm tương đồng fuzz.ratio.

Hiển thị thông tin về các phim tương tự trong result\_text. Đầu tiên, lặp qua danh sách các phim tương tự đã tìm được. Tiếp theo, tìm thông tin về phim trong DataFrame df dựa trên tên phim. Nếu tìm thấy thông tin, hiển thị thông tin về phim bao gồm tiêu đề, loại, đạo diễn, diễn viên, đánh giá, thể loại và mô tả. Nếu không tìm thấy thông tin, hiển thị thông báo không tìm thấy.

Cuối cùng, thêm dòng trống vào result\_text.

Tạo giao diện sử dụng tkinter:

Tạo một cửa sổ (window) với tiêu đề "Netflix Movie Recommendation" và kích thước 400x300.

Tạo một nhãn (label) để nhắc người dùng nhập tên phim.

Tạo một trường nhập liệu (entry) để người dùng nhập tên phim.

Tạo một nút (button) với văn bản "Đề xuất" và liên kết nó với hàm recommend\_movie.

Tạo một vùng văn bản (result\_text) để hiển thị thông tin về các phim tương tự.

Đọc dữ liệu từ file CSV:

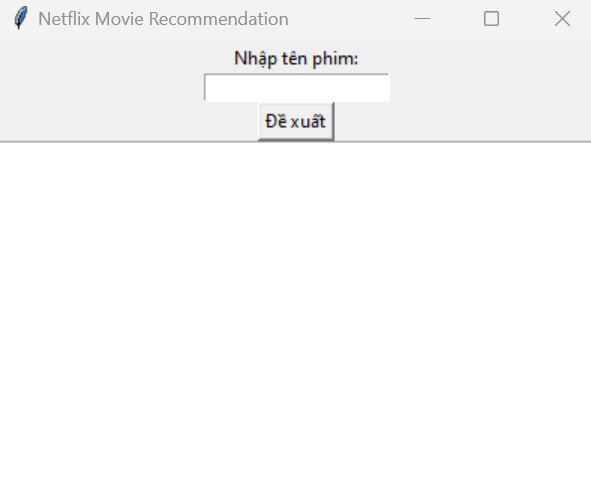
Sử dụng pd.read\_csv để đọc dữ liệu từ file CSV và lưu vào DataFrame df.

Kích hoạt vòng lặp chính của giao diện (window.mainloop()).

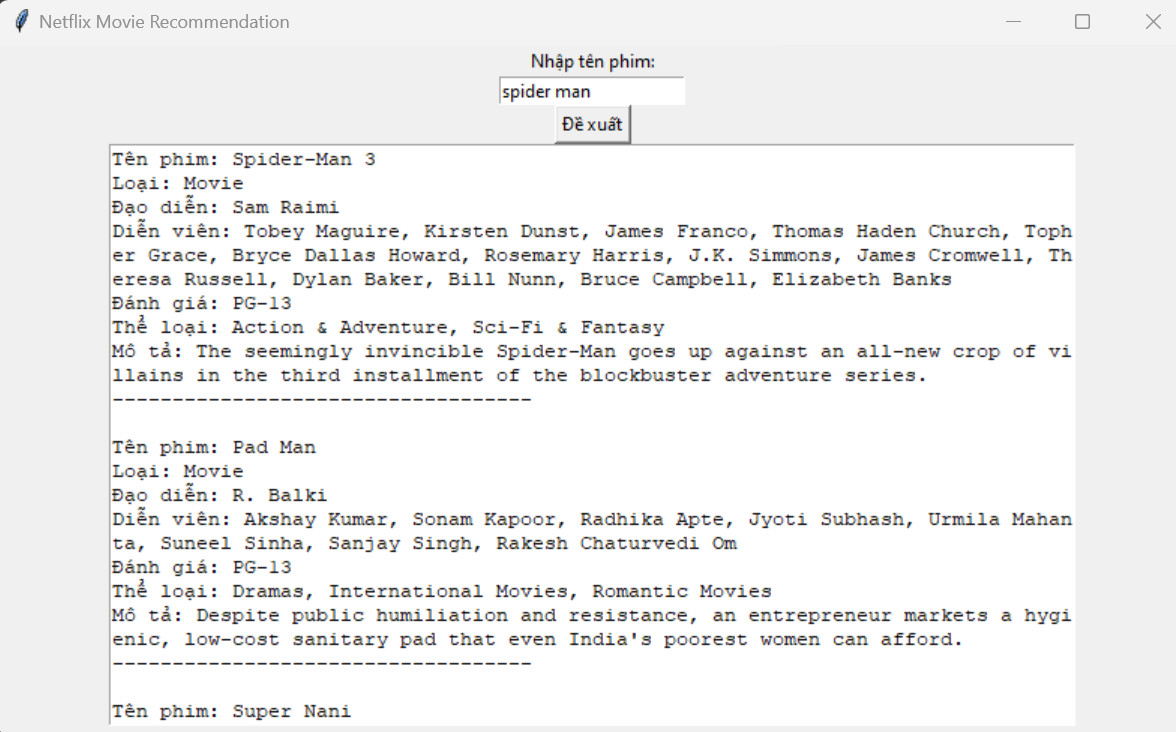
# **CHƯƠNG IV. THỰC NGHIỆM VÀ KẾT LUẬN**

## **4.1. Chạy chương trình**

Chạy chương trình



Sau khi chạy chương trình, màn hình sẽ hiện lên box tìm kiếm để người dùng có thể nhập dữ liệu từ bàn phím. Ta nhập thử phim “ Spider man”.



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

Bài tập **“Gợi ý phim netflix cho 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 Văn Huy** 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/infamouscoder/dataset-netflix-shows?resource=download>

https://www.kaggle.com/code/matteofasuloo/in-depth-eda-of-netflix-movies-tv-shows

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

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