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: Xây dựng hệ thống đè xuất sơn*

Sinh viên: Dương Quang Sáng

Lớp: K55KMT

Giáo viên hướng dẫn: Nguyễn Văn Huy

**Thái Nguyên - 2023**

**\_– 2023**

**TRƯỜNG ĐHKTCN CỘNG HOÀ XÃ HỘI CHỦ NGHĨA VIỆT NAM**

**KHOA ĐIỆN TỬ *Độc lập – Tự do – Hạnh phúc***

**BÀI TẬP LỚN**

MÔN HỌC: KHOA HỌC DỮ LIỆU

**BỘ MÔN: TIN HỌC CÔNG NGHIỆP**

*Sinh viên: Dương Quang Sáng Lớp: K55KMT*

*MSSV: K1954 8010 6019 Ngành: Kỹ thuật máy tính*

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

1.Tên đề tài : Xây dựng hệ thống đề xuất sơn

.

2. Yêu cầu: Đọc được dữ liệu tải về. Xây dựng được chương trình demo, nhập mã sản phẩm học từ khóa sản phẩm từ bàn phím. Chương trình sẽ xuất ra các sản phẩm tương đồng cho khách hàng.

|  |  |
| --- | --- |
| 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)*

MỤC LỤC

[LỜI NÓI ĐẦU 6](#_Toc136406629)

[CHƯƠNG I: GIỚI THIỆU CHUNG 6](#_Toc136406630)

[1.1.Hệ thống đề xuất sơn. 6](#_Toc136406631)

[1.1.1.Hệ thống đề xuất là gì ? 6](#_Toc136406632)

[1.1.2.Hệ thống đề xuất sơn là gì ? 7](#_Toc136406633)

[1.1.3.Ý nghĩ của hệ thống đề xuất sơn 9](#_Toc136406634)

[1.1.4.Thuật toán được sử dụng trong bài. 10](#_Toc136406635)

[CHƯƠNG II. PHÂN TÍCH DỮ LIỆU 11](#_Toc136406636)

[2.1.Bộ dữ liệu được sử dụng trong bài. 11](#_Toc136406637)

[2.1.1.Giới thiệu về bộ dữ liệu. 11](#_Toc136406638)

[2.1.2. Các trường trong bộ dữ liệu 11](#_Toc136406639)

[2.2. Thuật toán tính toán khoảng cách cosine 12](#_Toc136406640)

[2.2.1.Khái niệm 12](#_Toc136406641)

[2.2.2. Ý nghĩa của thuật toán Cosine Similarity 13](#_Toc136406642)

[2.3.Thuật toán CountVectorizer 14](#_Toc136406643)

[2.3.1.Khái niệm 14](#_Toc136406644)

[2.3.2.Ví dụ về CountVectorizer 14](#_Toc136406645)

[CHƯƠNG III. THIẾT KẾ CHƯƠNG TRÌNH 16](#_Toc136406646)

[3.1.Ngôn ngữ lập trình và môi trường làm việc 16](#_Toc136406647)

[3.1.1. Visual Studio Code 16](#_Toc136406648)

[3.1.2. Python 17](#_Toc136406649)

[3.1.3. Jupyter Notebook 19](#_Toc136406650)

[3.2.Code chương trình 20](#_Toc136406651)

[CHƯƠNG IV. THỰC NGHIỆM VÀ KẾT LUẬN 23](#_Toc136406652)

[4.1. Chạy chương trình 23](#_Toc136406653)

[4.2.Kết quả đã đạt được và hạn chế 24](#_Toc136406654)

[4.3.Hướng phát triển của bài tập 25](#_Toc136406655)

[TÀI LIỆU THAM KHẢO 26](#_Toc136406656)

# LỜI MỞ ĐẦU

Trong xã hội ngày nay, với sự phát triển không ngừng của công nghệ và sự bùng nổ của thông tin, việc tìm kiếm và khám phá các sản phẩm đã trở thành một phần quan trọng trong cuộc sống hàng ngày của chúng ta. Tuy nhiên, trong một thị trường đa dạng và phong phú như hiện nay, việc đối mặt với sự lựa chọn đồ sộ có thể là một nhiệm vụ khó khăn đối với người tiêu dùng.

Đó là lý do tại sao hệ thống đề xuất sản phẩm đã nổi lên như một công nghệ quan trọng để giúp chúng ta đối mặt với vấn đề này. Cụ thể hệ thống đề xuất ở đây là đề xuất sơn.

Với sự kết hợp của các phương pháp máy học, khai phá dữ liệu và lọc cộng tác, hệ thống đề xuất sơn ngày nay có khả năng cung cấp những gợi ý cá nhân hóa, phù hợp với nhu cầu và sở thích của từng người dùng. Nó giúp chúng ta tiết kiệm thời gian và nỗ lực trong việc tìm kiếm và khám phá những sản phẩm sơn mới, đồng thời mang đến trải nghiệm mua sắm trực tuyến tiện lợi và tối ưu.

Trong phạm vi bài báo cáo này em xin được trình bày quá trình thực hiện bài tập **“Đề xuất sơn”** 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

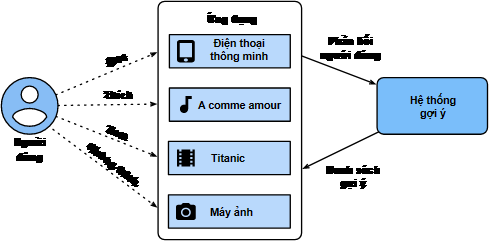
# 1.1.Hệ thống đề xuất sơn

## Hệ thống đề xuất là gì ?

Trong thập kỷ vừa qua, mạng Internet đã phát triển thành một nền tảng cho các dịch vụ trực tuyến quy mô lớn, đồng thời thay đổi sâu sắc cách ta giao tiếp, đọc tin tức, mua sắm và xem phim. Trong khi đó, một lượng lớn chưa từng có các sản phẩm (chúng tôi sử dụng từ sản phẩm (item) cho phim ảnh, tin tức, sách và hàng hóa) được bày bán trực tuyến yêu cầu một hệ thống có thể giúp ta tìm những sản phẩm ưa thích hơn.

Do đó, hệ thống đề xuất là công cụ lọc thông tin mạnh mẽ có thể thúc đẩy các dịch vụ cá nhân hóa và cung cấp trải nghiệm riêng biệt cho từng người dùng. Nói ngắn gọn, hệ thống đề xuất đóng vai trò nòng cốt trong việc tận dụng nguồn dữ liệu dồi dào hiện có để giúp việc đưa ra lựa chọn dễ dàng hơn.

Ngày nay, hệ thống đề xuất là thành phần trung tâm của nhiều nhà cung cấp dịch vụ trực tuyến như Amazon, Netflix, và YouTube. Nhớ lại ví dụ Amazon đưa ra đề xuất các sách Học sâu trong [Fig. 1.3.3](https://d2l.aivivn.com/chapter_introduction/index_vn.html#subsec-recommender-systems). Có hai lợi ích của việc sử dụng hệ thống đề xuất: Một mặt, nó có thể giảm lượng lớn công sức tìm kiếm sản phẩm của người dùng và giảm thiểu vấn đề quá tải thông tin. Mặt khác, nó có thể tăng giá trị kinh doanh cho các nhà cung cấp dịch vụ trực tuyến và trở thành nguồn doanh thu quan trọng. Chương này sẽ giới thiệu những khái niệm cơ bản, các mô hình cổ điển và những bước tiến gần đây của học sâu trong lĩnh vực hệ thống đề xuất.



***Hình 1. Ví dụ về hệ thống đề xuất***

### **Hệ thống đề xuất sơn là gì ?**

Hệ thống đề xuất sơn là một loại hệ thống thông tin có khả năng dự đoán, gợi ý hoặc đề xuất các sản phẩm sơn cho người dùng dựa trên sở thích, hành vi mua hàng trước đó hoặc các thông tin ngữ cảnh khác. Mục tiêu của hệ thống đề xuất sản phẩm là cung cấp cho người dùng những gợi ý hữu ích và cá nhân hóa để giúp họ tìm kiếm, khám phá và mua các sản phẩm một cách hiệu quả.

Hệ thống đề xuất sơn có thể áp dụng trong lĩnh vực đồ gỗ mỹ nghệ và ngành công nghiệp khác như sơn nhà sơn xe.. . Các phương pháp và thuật toán trong hệ thống đề xuất sản phẩm có thể sử dụng các kỹ thuật máy học, khai phá dữ liệu, lọc cộng tác, lọc dựa trên nội dung và nhiều phương pháp khác để tạo ra các đề xuất phù hợp với từng người dùng.

Tùy thuộc vào mục đích và ngữ cảnh sử dụng, hệ thống đề xuất sản phẩm sơn có thể áp dụng các phương pháp như hệ thống đề xuất dựa trên nội dung, hệ thống đề xuất dựa trên người dùng, hệ thống đề xuất kết hợp, hệ thống đề xuất theo lượt xem và nhiều hơn nữa. Mỗi phương pháp có ưu điểm và hạn chế riêng, và phụ thuộc vào đặc điểm của dữ liệu và yêu cầu của hệ thống mà chúng ta có thể lựa chọn phương pháp phù hợp.

Hệ thống đề xuất sơn có thể sử dụng nhiều phương pháp khác nhau để đưa ra các đề xuất sản phẩm . Dưới đây là một số phương pháp phổ biến được sử dụng:

* **Lịch sử mua hàng:** Hệ thống gợi ý có thể dựa trên lịch sử mua hàng của người dùng để đưa ra các sản phẩm tương tự hoặc sản phẩm mà người dùng có thể quan tâm.
* **Đánh giá và phản hồi:** Hệ thống có thể sử dụng thông tin về đánh giá và phản hồi của người dùng về các sản phẩm để gợi ý các sản phẩm tương tự hoặc sản phẩm được đánh giá cao.
* **Sở thích cá nhân:** Hệ thống có thể thu thập thông tin về sở thích cá nhân của người dùng, ví dụ như các danh mục sản phẩm mà người dùng thường xem, mua hàng hoặc đánh giá.
* **Thuộc tính sản phẩm:** Hệ thống có thể xem xét các thuộc tính và đặc trưng của sản phẩm để tìm ra các sản phẩm tương tự. Điều này có thể bao gồm mô tả sản phẩm, từ khóa, danh mục sản phẩm, giá cả và các thuộc tính khác.
* **Kết nối xã hội:** Hệ thống có thể sử dụng thông tin về mạng xã hội của người dùng, ví dụ như kết nối bạn bè, quan tâm chung, hoặc hoạt động chia sẻ, để gợi ý các sản phẩm được ưa thích bởi cộng đồng xung quanh người dùng.

### **Ý nghĩa của hệ thống đề xuất sơn**

Hệ thống đề xuất sơn có ý nghĩa quan trọng đối với những người làm sơn và đối tượng sử dụng khác nhau. Dưới đây là một số ý nghĩa của hệ thống đề xuất sơn:

* + Cải thiện trải nghiệm người dùng: Hệ thống gợi ý sản phẩm giúp cung cấp cho người dùng những sản phẩm phù hợp với sở thích và nhu cầu của họ. Điều này giúp tăng trải nghiệm cá nhân, tiết kiệm thời gian tìm kiếm và giúp người dùng khám phá những sản phẩm mới và thú vị.
  + Tăng doanh số bán hàng: Bằng cách đề xuất những sản phẩm liên quan hoặc tương tự, hệ thống đề xuất sơn có thể tăng khả năng chuyển đổi người dùng thành khách hàng và tăng doanh số bán hàng. Việc gợi ý những sản phẩm phù hợp giúp tăng khả năng mua hàng và tạo sự hài lòng cho khách hàng.
  + Tối ưu hóa tỷ lệ chuyển đổi: Hệ thống đề xuất có thể tối ưu hóa tỷ lệ chuyển đổi bằng cách đưa ra những gợi ý phù hợp và hấp dẫn đối với từng khách hàng. Việc gợi ý sản phẩm phù hợp giúp khách hàng cảm thấy đáng quan tâm và gia tăng khả năng mua hàng.
  + Tăng tính cạnh tranh và trung thành khách hàng: Hệ thống đề xuất sơn có thể giúp tăng tính cạnh tranh của doanh nghiệp bằng cách cung cấp những đè xuất sản phẩm sơn tốt nhất. Điều này có thể tạo sự ưu ái và trung thành từ phía khách hàng, giúp tạo lợi thế cạnh tranh trong thị trường.
  + Phân tích dữ liệu và hiểu khách hàng: Hệ thống đề xuất sơn thu thập thông tin từ hành vi và sở thích của khách hàng để phân tích và hiểu rõ hơn về họ. Điều này giúp doanh nghiệp có cái nhìn sâu sắc về khách hàng và tạo ra các chiến lược kinh doanh phù hợp.

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

thuật toán cosine similarity (tương đồng cosine) được sử dụng để tính độ tương đồng giữa người dùng và giữa các sản phẩm.

Thuật toán cosine similarity được sử dụng để đo độ tương đồng dựa trên góc giữa hai vector. Nó được áp dụng trong bài toán gợi ý sản phẩm để tính toán sự tương đồng giữa các người dùng hoặc các sản phẩm dựa trên hành vi mua hàng. Kết quả của thuật toán là một ma trận tương đồng mà mỗi phần tử đại diện cho độ tương đồng giữa hai người dùng hoặc hai sản phẩm tương ứng.

Sau đó, trong đoạn code, các sản phẩm tương tự nhất được tìm kiếm bằng cách sắp xếp dựa trên độ tương đồng và lấy ra top 10 sản phẩm có độ tương đồng cao nhất.

# CHƯƠNG II. PHÂN TÍCH DỮ LIỆU

## Bộ dữ liệu sử dụng trong bài

### **Giới thiệu về bộ dữ liệu**

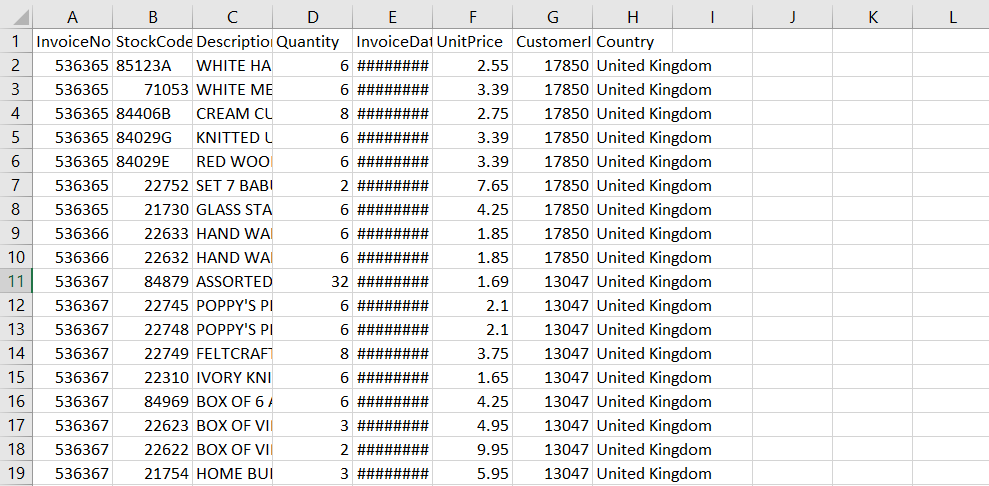
Bộ dữ liệu sử dụng trong bài tập là bộ dữ liệu **“Online Retail Dataset”**. Đây là bộ dữ liệu được lấy từ trang web **Kaggle**, chứa thông tin về các thông tin về sản phẩm mà khách hàng .

Ảnh có chứa văn bản, phần mềm, Biểu tượng máy tính, Phần mềm đa phương tiện

Mô tả được tạo tự động

***Hình 2. Bộ dữ liệu Anime Recommendation Database 2020***

### **Các trường trong bộ dữ liệu**



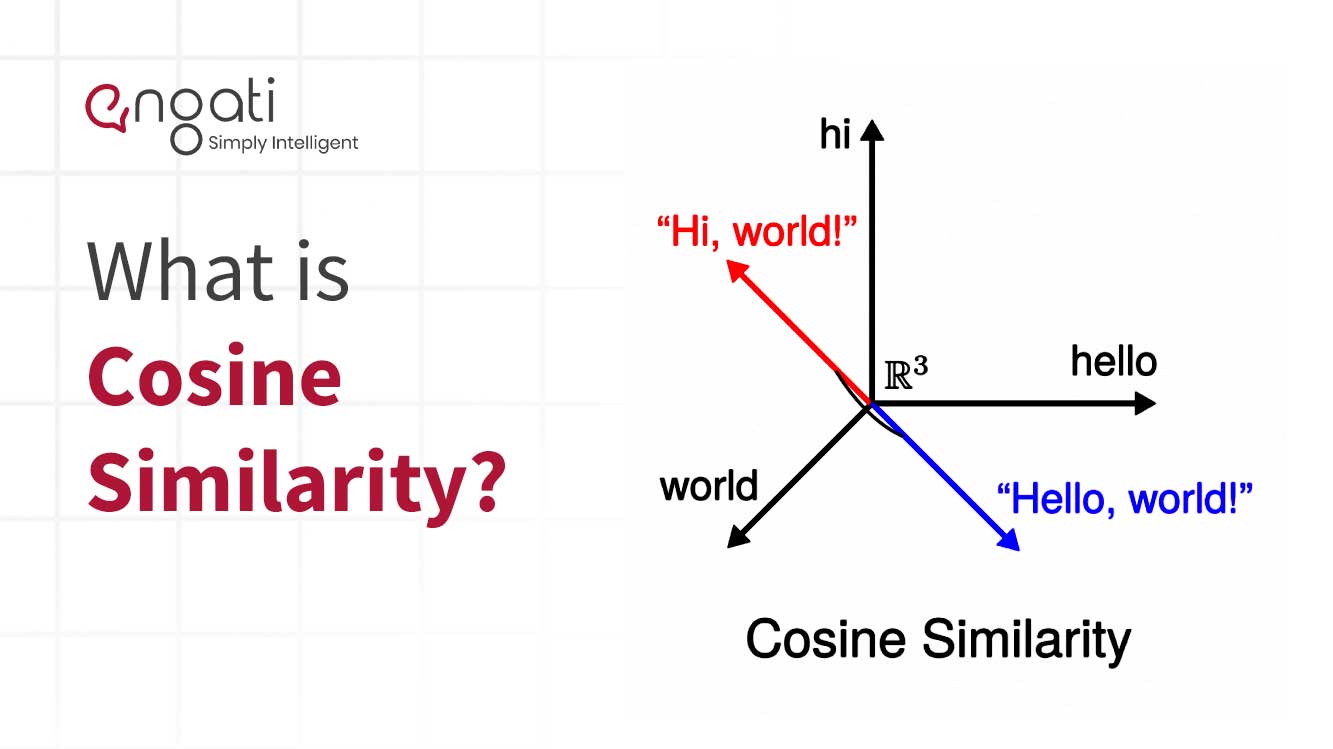
***Hình 3. Bộ dữ liệu anime.csv sau khi tải về***

* **InvoiceNo :** Mã hóa đơn sản phẩm của khách hàng
* **StockCode :** Mã sản phẩm
* **Description :** Miêu tả sản phẩm
* **Quantity :** Số lượng sản phẩm
* **InvoiceDate :** ngày lập hóa đơn
* **UnitPrice :** giá hóa đơn
* **CustomerID :** Mã khách hàng
* **Country :** Quốc gia

## Thuật toán tính toán khoảng cách cosine

### **2.2.1. Khái niệm**

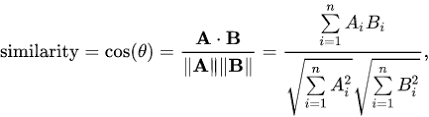
Thuật toán tính toán khoảng cách cosine (Cosine Similarity) là một phương pháp đo đạc mức độ tương đồng giữa hai vectơ trong không gian đa chiều. Nó được sử dụng rộng rãi trong lĩnh vực xử lý ngôn ngữ tự nhiên và khai phá dữ liệu, thường được sử dụng để so sánh sự tương đồng giữa các văn bản, từ ngữ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



***Hình 4. 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 5. 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 6. 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."

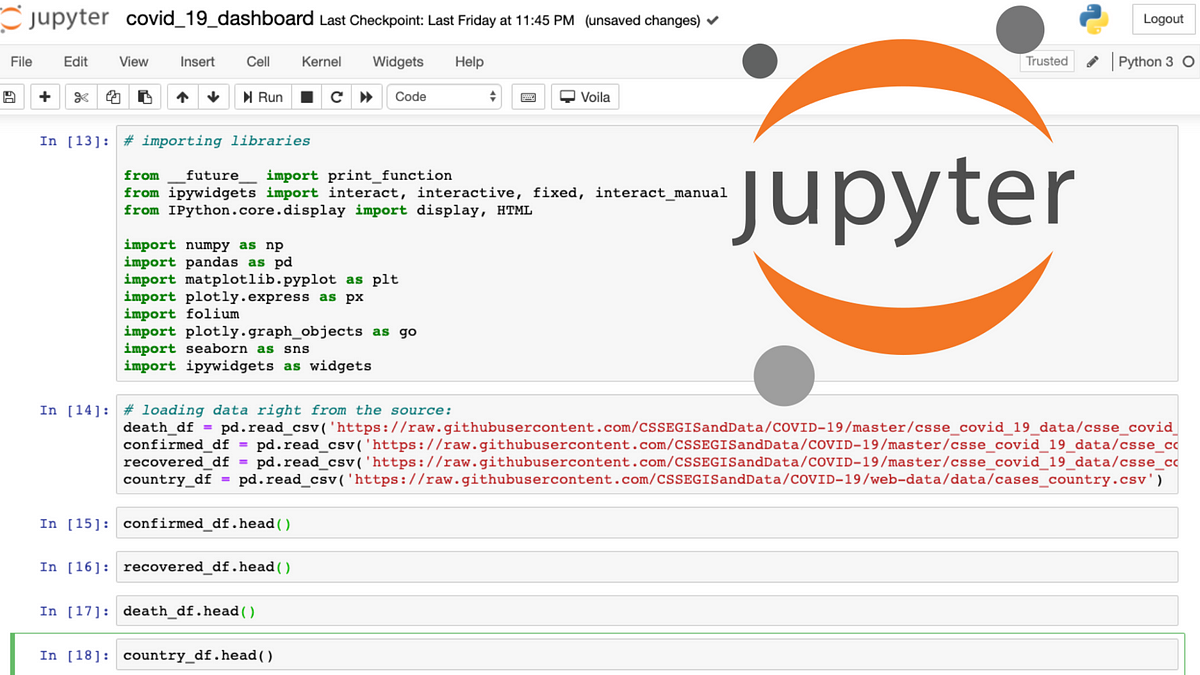
### **3.1.3. Jupyter Notebook**

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 7. 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

from datetime import datetime  
import pandas as pd  
from sklearn.metrics.pairwise import cosine\_similarity  
  
dateparse = lambda x: datetime.strptime(x, '%m/%d/%Y %H:%M')  
df = pd.read\_csv("C:\\Users\\hoabu\\baitaplon\\OnlineRetail.csv", parse\_dates=['InvoiceDate'], date\_parser=dateparse, encoding='unicode\_escape')

* Thư viện **datetime** được import để làm việc với thời gian.
* Thư viện **pandas** được import để làm việc với dữ liệu dạng bảng.
* **cosine\_similarity** từ **sklearn.metrics.pairwise** được import để tính độ tương đồng cosine giữa các vector.
* Hàm **dateparse** được định nghĩa để chuyển đổi định dạng ngày tháng trong dữ liệu.
* Dòng cuối cùng đọc dữ liệu từ tệp CSV và lưu vào DataFrame **df** với việc áp dụng hàm **dateparse** cho cột 'InvoiceDate'.

df = df.loc[df['Quantity'] > 0]  
df1 = df.dropna(subset=['CustomerID'])  
customer\_item\_matrix = df1.pivot\_table(index='CustomerID', columns='StockCode', values='Quantity', aggfunc='sum')  
customer\_item\_matrix = customer\_item\_matrix.applymap(lambda x: 1 if x > 0 else 0)

* Dòng đầu tiên lọc các hàng trong DataFrame **df** sao cho giá trị cột 'Quantity' lớn hơn 0, loại bỏ các hàng có số lượng không hợp lệ.
* Dòng thứ hai loại bỏ các hàng có giá trị bị thiếu (NaN) trong cột 'CustomerID'.
* Dòng thứ ba tạo ma trận pivot từ DataFrame **df1** với chỉ số (index) là 'CustomerID', cột là 'StockCode', và giá trị là 'Quantity', sử dụng hàm tổng (**aggfunc='sum'**) để tính tổng số lượng sản phẩm.
* Dòng cuối cùng áp dụng hàm lambda để biến các giá trị trong ma trận thành 1 nếu giá trị ban đầu lớn hơn 0, ngược lại thành 0.

user\_to\_user\_sim\_matrix = pd.DataFrame(cosine\_similarity(customer\_item\_matrix))  
user\_to\_user\_sim\_matrix.columns = customer\_item\_matrix.index  
user\_to\_user\_sim\_matrix['CustomerID'] = customer\_item\_matrix.index  
user\_to\_user\_sim\_matrix = user\_to\_user\_sim\_matrix.set\_index('CustomerID')

* Dòng thứ hai đặt tên cột của **user\_to\_user\_sim\_matrix** bằng các giá trị của chỉ số (index) của **customer\_item\_matrix**.
* Dòng thứ ba thêm cột 'CustomerID' vào **user\_to\_user\_sim\_matrix** và gán giá trị của chỉ số (index) của **customer\_item\_matrix** vào cột này.
* Dòng cuối cùng đặt cột 'CustomerID' làm chỉ số (index) của **user\_to\_user\_sim\_matrix**.

item\_item\_sim\_matrix = pd.DataFrame(cosine\_similarity(customer\_item\_matrix.T))  
item\_item\_sim\_matrix.columns = customer\_item\_matrix.T.index  
item\_item\_sim\_matrix['StockCode'] = customer\_item\_matrix.T.index  
item\_item\_sim\_matrix = item\_item\_sim\_matrix.set\_index('StockCode')

* Dòng thứ hai tính độ tương đồng cosine giữa các sản phẩm bằng cách áp dụng hàm **cosine\_similarity** lên ma trận chuyển vị của **customer\_item\_matrix**.
* Dòng thứ ba đặt tên cột của **item\_item\_sim\_matrix** bằng các giá trị của chỉ số (index) chuyển vị của **customer\_item\_matrix**.
* Dòng thứ tư thêm cột 'StockCode' vào **item\_item\_sim\_matrix** và gán giá trị của chỉ số (index) chuyển vị của **customer\_item\_matrix** vào cột này.
* Dòng cuối cùng đặt cột 'StockCode' làm chỉ số (index) của **item\_item\_sim\_matrix**.

while True:  
 product\_code = input("Nhập mã sản phẩm (hoặc 'q' để thoát): ")  
 if product\_code == 'q':  
 break  
  
 if product\_code not in item\_item\_sim\_matrix.index:  
 print("Không tìm thấy mã sản phẩm.")  
 continue  
  
 top\_10\_similar\_items = list(  
 item\_item\_sim\_matrix  
 .loc[product\_code]  
 .sort\_values(ascending=False)  
 .iloc[:10]  
 .index  
 )  
  
 print("Các sản phẩm tương tự:")  
 print(df.loc[df['StockCode'].isin(top\_10\_similar\_items), ['StockCode', 'Description']].drop\_duplicates().set\_index('StockCode'))

* Dòng 3: Kiểm tra xem mã sản phẩm nhập vào (**product\_code**) có tồn tại trong chỉ số (index) của DataFrame **item\_item\_sim\_matrix** hay không. Nếu không tồn tại, nghĩa là không có thông tin về mã sản phẩm này trong ma trận độ tương đồng giữa các sản phẩm. Trong trường hợp này, in ra thông báo "Không tìm thấy mã sản phẩm" và sử dụng lệnh **continue** để bỏ qua phần còn lại của vòng lặp và chờ người dùng nhập mã sản phẩm khác.
* Dòng 5-9: Nếu mã sản phẩm tồn tại trong **item\_item\_sim\_matrix**, chúng ta tiến hành tìm kiếm các sản phẩm tương tự. Đầu tiên, chúng ta truy cập hàng tương ứng với mã sản phẩm trong **item\_item\_sim\_matrix** bằng cách sử dụng phương thức **.loc[product\_code]**. Sau đó, chúng ta sắp xếp giá trị độ tương đồng của các sản phẩm theo thứ tự giảm dần bằng phương thức **.sort\_values(ascending=False)**. Tiếp theo, chúng ta chỉ lấy 10 sản phẩm đầu tiên trong danh sách này bằng phương thức **.iloc[:10]**. Cuối cùng, chúng ta trích xuất chỉ số (mã sản phẩm) của các sản phẩm này bằng phương thức **.index** và lưu trữ chúng trong danh sách **top\_10\_similar\_items**.
* Dòng 11-14: In ra thông báo "Các sản phẩm tương tự" và hiển thị thông tin của các sản phẩm tương tự. Đầu tiên, chúng ta lọc các hàng trong DataFrame **df** mà có mã sản phẩm thuộc danh sách **top\_10\_similar\_items**, sử dụng điều kiện **df['StockCode'].isin(top\_10\_similar\_items)**. Sau đó, chúng ta chỉ lấy cột 'StockCode' (mã sản phẩm) và 'Description' (mô tả sản phẩm) bằng cách sử dụng **['StockCode', 'Description']**. Tiếp theo, chúng ta loại bỏ các hàng trùng lặp bằng phương thức **.drop\_duplicates()**. Cuối cùng, chúng ta đặt chỉ số của DataFrame là 'StockCode' bằng cách sử dụng **.set\_index('StockCode')** và in ra kết quả

# 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.

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 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/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/>