SAMSUNG

AI Course

Capstone Project Final Report

For students (instructor review required)

Together for Tomorrow!
Enabling People
Education for Future Generations

Đề tài 15: Xây dựng mô hình gợi ý sách sử dụng phương pháp kết hợp

Nhóm 2

<20/08/2024>

Thành viên nhóm

- Nguyễn Viết Quang
- Trương Thế Việt
- Đồng Anh Quân
- Phạm Tiến Thành
- Lại Nguyên Nam
- Mai Hoàng Tùng

Lời cảm ơn

Chúng em xin gửi lời cảm ơn chân thành đến thầy Tạ Quang Chiểu, người đã không chỉ truyền đạt kiến thức mà còn khơi dậy niềm đam mê học tập trong chúng em suốt khóa học. Từ những nền tảng toán học chuyên ngành cho học máy, đến phân tích dữ liệu thống kê, rồi nâng cao hơn với machine learning, NLP, neural network và deep learning, thầy đã dẫn dắt chúng em từng bước trên con đường khám phá trí tuệ nhân tạo, cho đến dự án capstone này.

Chúng em cũng xin bày tỏ lòng biết ơn sâu sắc đến thầy Vũ Thành Vinh, người đã luôn tận tâm hướng dẫn và hỗ trợ chúng em trong quá trình hoàn thiện dự án này.

Xin gửi lời cảm ơn đặc biệt đến toàn thể đội ngũ của Samsung Innovation Campus (SIC). Cảm ơn SIC đã mang đến một chương trình học tập AI toàn diện và chất lượng, không chỉ cung cấp nền tảng kiến thức hệ thống trên multi campus mà còn tạo điều kiện cho chúng em tiếp cận các bài giảng và tài liệu bổ ích trên LinkedIn. Chúng em cũng trân trọng những buổi kỹ năng mềm trực tiếp và các khóa học trực tuyến trên LinkedIn được SIC gợi ý, giúp chúng em phát triển toàn diện hơn.

Cuối cùng, xin cảm ơn SIC vì 4 tháng vừa qua đã đồng hành và giúp chúng em tiến bộ vượt bậc trong lĩnh vực AI. Chúng em hy vọng sẽ có cơ hội áp dụng những kiến thức và kỹ năng đã học vào thực tế, đóng góp cho sự phát triển của cộng đồng AI.

Mục lục

Bảng chú giải

Thuật ngữ	Chú giải
ISBN (*International Standard BookNumber)*	Mã số tiêu chuẩn quốc tế cho sách
Book-Title	Tiêu đề của sách
Year-Of-Publication	Năm phát hành sách
Image-URL-S	Đường dẫn hình ảnh kích thước S
Image-URL-M	Đường dẫn hình ảnh kích thước M
Image-URL-L	Đường dẫn hình ảnh kích thước L
User-ID	ID độc giả
Location	Địa chỉ độc giả
Age	Tuổi độc giả
Book-Rating	Số điểm đánh giá sách
num-of-rating	Số lượng đánh giá
PCA (Principal Component Analysis)	Phân tích thành phần chính
Autoencoder	Bộ mã hoá tự động
Mean Squared Error (MSE)	Sai số trung bình căn bậc 2

I. Giới thiệu dự án.

1.1. Thông tin cơ bản

- Tên dự án: Xây dựng mô hình gợi ý sách sử dụng phương pháp KNN.
- Mục tiêu chính: Áp dụng kiến thức đã học trong khóa học, bao gồm tiền xử lý dữ liệu, trực quan hóa, phân tích dữ liệu, và sử dụng mô hình KNN để xây dựng một hệ thống gợi ý sách. Ngoài ra, dự án cũng thử nghiệm và so sánh với mô hình Neural Network (autoencoder) để tham khảo .
- Lĩnh vực: Học máy, Học sâu
- Ý nghĩa thực tiễn: Dự án này có thể được tích hợp vào các hệ thống web thương mại điện tử hoặc các nền tảng đọc sách trực tuyến để gợi ý sách cho người dùng dựa trên sở thích và lịch sử đọc của họ, từ đó nâng cao trải nghiệm người dùng và tăng doanh thu.

1.2. Động lực và Mục tiêu

- Động lực:
- Mong muốn áp dụng kiến thức đã học về học máy và học sâu vào một dự án thực tế.
- Giải quyết bài toán gợi ý sách, một vấn đề phổ biến và có ý nghĩa trong lĩnh vực thương mại điện tử và giải trí.
- Mục tiêu:
- Xây dựng một hệ thống gợi ý sách có khả năng đề xuất sách từ sách người dùng quan tâm.
- So sánh kết quả của mô hình KNN và mô hình Neural Network (autoencoder) để đánh giá ưu nhược điểm của từng phương pháp.
- Tạo một web demo để minh họa và kiểm thử hệ thống gợi ý.

1.3. Thành viên và Phân công công việc

Thành viên	Công việc
Quang (nhóm trưởng)	Trực quan hóa, xây dựng model KNN, xây dựng model Autoencoder và so sánh, xây dựng web demo.
Quân	Tiền xử lý dữ liệu, gộp bảng dữ liệu, trích xuất đặc trưng, mã hóa feature, viết báo cáo, xây dựng web demo .
Tùng	Xây dựng model KNN,trực quan hóa kết quả dự đoán bằng PCA 2D, 3D, viết báo cáo.
Nam	Tiền xử lý dữ liệu, trực quan hóa, làm slide,xây dựng web demo.
Thành	Tiền xử lý dữ liệu, trực quan hóa theo nhiều bảng, phân tích dữ liệu, viết báo cáo, xây dựng web demo.
Việt	Xây dựng model KNN, model selection, tối ưu mô hình, viết báo cáo.

1.4. Lịch trình và Mốc quan trọng

Lịch trình:

- Tuần 1: Thu thập dữ liệu
- Tuần 2-3: Xử lý dữ liệu
- Tuần 4-5: Xây dựng mô hình
- Tuần 6: Tạo web demo
- Tuần 7: Viết báo cáo và slide

Mốc quan trọng:

- Hoàn thành thu thập dữ liệu
- Hoàn thành xử lý dữ liệu
- Hoàn thành xây dựng mô hình KNN và Autoencoder
- Hoàn thành web demo
- Hoàn thành báo cáo và slide

II. Tiền xử lý và trực quan hoá dữ liệu

1. Xử lý từng bảng dữ liệu

1.1 Đối với bảng dữ liệu “BX-Books.csv”

- Hình ảnh sau khi đọc file

```
books = pd.read_csv(
    'data/BX-Books.csv',
    sep=";", on_bad_lines='skip',
    low_memory=False ,
    encoding='latin-1')
```

```
books.head()
```

	ISBN	Book-Title	Book-Author	Year-Of-Publication	Publisher	Image-URL-S	Image-URL-M
0	0195153448	Classical Mythology	Mark P. O. Morford	2002	Oxford University Press	http://images.amazon.com/images/P/0195153448.0...	http://images.amazon.com/images/P/0195153448.0...
1	0002005018	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada	http://images.amazon.com/images/P/0002005018.0...	http://images.amazon.com/images/P/0002005018.0...
2	0060973129	Decision in Normandy	Carlo D'Este	1991	HarperPerennial	http://images.amazon.com/images/P/0060973129.0...	http://images.amazon.com/images/P/0060973129.0...
3	0374157065	Flu: The Story of the Great Influenza Pandemic...	Gina Bari Kolata	1999	Farrar Straus Giroux	http://images.amazon.com/images/P/0374157065.0...	http://images.amazon.com/images/P/0374157065.0...
4	0393045218	The Mummies of Urumchi	E. J. W. Barber	1999	W. W. Norton & Company	http://images.amazon.com/images/P/0393045218.0...	http://images.amazon.com/images/P/0393045218.0...

- Dữ liệu cột “Image-URL-S” và “Image-URL-M” là những cột không cần thiết và cần bỏ vì các cột này là URL của những bức ảnh có kích thước S và M khi triển khai các bức ảnh này bé và khó nhìn

- Hình ảnh sau khi thực hiện bỏ các cột: Dữ liệu chỉ còn 6 cột (như hình dưới): Các cột “Image-URL-S” và “Image-URL-M” đã được loại bỏ ra khỏi bảng dữ liệu

```
# after remove
books = books[['ISBN','Book-Title', 'Book-Author', 'Year-Of-Publication', 'Publisher','Image-URL-L']]
```

```
books.shape
```

```
(271360, 6)
```

```
books.head()
```

	ISBN	Book-Title	Book-Author	Year-Of-Publication	Publisher	Image-URL-L
0	0195153448	Classical Mythology	Mark P. O. Morford	2002	Oxford University Press	http://images.amazon.com/images/P/0195153448.0...
1	0002005018	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada	http://images.amazon.com/images/P/0002005018.0...
2	0060973129	Decision in Normandy	Carlo D'Este	1991	HarperPerennial	http://images.amazon.com/images/P/0060973129.0...
3	0374157065	Flu: The Story of the Great Influenza Pandemic...	Gina Bari Kolata	1999	Farrar Straus Giroux	http://images.amazon.com/images/P/0374157065.0...
4	0393045218	The Mummies of Urumchi	E. J. W. Barber	1999	W. W. Norton & Company	http://images.amazon.com/images/P/0393045218.0...

- Ta nhận thấy sau khi loại bỏ các cột không cần thiết bảng dữ liệu của chúng ta đã gọn hơn và giảm được 1 phần dữ liệu trong bảng, điều này giúp cho ta có thể thuận lợi cho các bước xử lý tiếp theo

- Với tên của những cột trong bảng dữ liệu “BX-Books.csv” ta sẽ bằng những tên mới gồm:

- “Book-Title” : “title”
- “Book-Author” : ”author”
- “Year-Of-Publication” : “year”
- “Publisher” : ”publisher”
- “Image-URL-L” : “image_url”

- Hình ảnh trước khi chưa thay đổi tên cột

```
books.head()
```

	ISBN	Book-Title	Book-Author	Year-Of-Publication	Publisher	Image-URL-S	Image-URL-M
0	0195153448	Classical Mythology	Mark P. O. Morford	2002	Oxford University Press	http://images.amazon.com/images/P/0195153448.0...	http://images.amazon.com/images/P/0195153448.0...
1	0002005018	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada	http://images.amazon.com/images/P/0002005018.0...	http://images.amazon.com/images/P/0002005018.0...
2	0060973129	Decision in Normandy	Carlo D'Este	1991	HarperPerennial	http://images.amazon.com/images/P/0060973129.0...	http://images.amazon.com/images/P/0060973129.0...
3	0374157065	Flu: The Story of the Great Influenza Pandemic...	Gina Bari Kolata	1999	Farrar Straus Giroux	http://images.amazon.com/images/P/0374157065.0...	http://images.amazon.com/images/P/0374157065.0...
4	0393045218	The Mummies of Urumchi	E. J. W. Barber	1999	W. W. Norton & Company	http://images.amazon.com/images/P/0393045218.0...	http://images.amazon.com/images/P/0393045218.0...

```
books.shape
```

```
(271360, 8)
```

- Hình ảnh sau khi thay đổi tên cột

	ISBN	title	author	year	publisher	image_url
0	0195153448	Classical Mythology	Mark P. O. Morford	2002	Oxford University Press	http://images.amazon.com/images/P/0195153448.0...
1	0002005018	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada	http://images.amazon.com/images/P/0002005018.0...
2	0060973129	Decision in Normandy	Carlo D'Este	1991	HarperPerennial	http://images.amazon.com/images/P/0060973129.0...
3	0374157065	Flu: The Story of the Great Influenza Pandemic...	Gina Bari Kolata	1999	Farrar Straus Giroux	http://images.amazon.com/images/P/0374157065.0...
4	0393045218	The Mummies of Urumchi	E. J. W. Barber	1999	W. W. Norton & Company	http://images.amazon.com/images/P/0393045218.0...

=> Thay đổi tên cột giúp ta sẽ dễ dàng ghi nhớ và tên sẽ ngắn gọn hơn thuận tiện cho việc thao tác tiếp theo

1.2. Với bảng dữ liệu “BX-Users.csv”

- Hình ảnh sau khi đọc file

```
users = pd.read_csv('data/BX-Users.csv', sep=";", on_bad_lines='skip',low_memory=False , encoding='latin-1')
```

```
users.shape
```

```
(278858, 3)
```

```
users.head()
```

	User-ID	Location	Age
0	1	nyc, new york, usa	NaN
1	2	stockton, california, usa	18.0
2	3	moscow, yukon territory, russia	NaN
3	4	porto, v.n.gaia, portugal	17.0
4	5	farnborough, hants, united kingdom	NaN

- Ta thấy dữ liệu của bảng gồm 3 cột thông tin “User-ID”, “Location”, “Age” là 3 cột thông tin cần thiết nên ta không loại bỏ các cột thông tin nào

- Ta thực hiện đổi tên các cột bằng tên mới gồm:

- “User-ID” : ”user_id”
- “Location” : “location”
- “Age” : “age”

- Hình ảnh thực hiện

```
# Lets remane some wierd columns name
users.rename(columns={"User-ID":'user_id',
                      'Location':'location',
                     "Age":'age'},inplace=True)
```

```
users.head(2)
```

	user_id	location	age
0	1	nyc, new york, usa	NaN
1	2	stockton, california, usa	18.0

1.3. Đối với bảng dữ liệu “BX-Book-Ratings.csv”

Rating dataframe

```
# Now Load the third dataframe
ratings = pd.read_csv('data/BX-Book-Ratings.csv', sep=";", on_bad_lines='skip',low_memory=False , encoding='latin-1')
```

```
ratings.shape
```

```
(1149780, 3)
```

```
ratings.columns
```

```
Index(['User-ID', 'ISBN', 'Book-Rating'], dtype='object')
```

```
ratings.head()
```

	user_id	ISBN	rating
1456	277427	002542730X	10
1457	277427	0026217457	0
1458	277427	003008685X	8
1459	277427	0030615321	0
1460	277427	0060002050	0

- Ta thấy dữ liệu của bảng gồm 3 cột thông tin “User-ID”, “ISBN”, “rating” là 3 cột thông tin cần thiết nên ta không loại bỏ các cột thông tin nào.

=> Sau khi thao tác xong ta đã có 3 bảng dữ liệu với các kích thước dữ liệu như hình sau:

```
print(f'book dataframe {books.shape}\nusers dataframe {users.shape}\nratings dataframe {ratings.shape}')
```

```
book dataframe (271360, 6)
users dataframe (278858, 3)
ratings dataframe (1149780, 3)
```

2. Nhóm các bảng dữ liệu có liên quan

- Sau khi xử lý dữ liệu của 3 bảng ta thấy dữ liệu của 3 bảng vẫn rất lớn, điều này sẽ gây ra việc training model sẽ tốn thời gian và tài nguyên của máy, khó tìm kiếm được các dữ liệu đặc trưng cần thiết. Vậy nên chúng ta cần phải nhóm dữ liệu vào thành 1 bảng bao gồm đầy đủ các dữ liệu cần thiết từ đó tìm ra tổng số đánh giá sách của các user để gợi ý những cuốn sách dựa trên dữ liệu đầu vào

- Ta thấy do số lượng dữ liệu trong bảng ratings khá lớn gồm (114978 dòng, 3 cột) vậy chúng ta cần lọc ra những “user_id” mà có số lượng đánh giá trên 200 cuốn sách

- Hình ảnh sau khi thực hiện lọc:

```
x = ratings['user_id'].value_counts() > 200
x = x[x]
```

```
x
```

```
user_id
11676     True
198711    True
153662    True
98391     True
35859     True
          ...
59727     True
188951    True
268622    True
9856      True
155916    True
Name: count, Length: 899, dtype: bool
```

- Sau khi lọc xong “user_id” có đánh giá trên 200 ta sẽ lưu những “user_id” vừa lọc vào biến y và kiểm tra xem những “user_id” của bảng ratings đó có trong y hay không và gán lại vào bảng dữ liệu ban đầu

- Hình ảnh sau khi thực hiện:

```
y= x.index
y
```

```
Index([ 11676, 198711, 153662,  98391,  35859, 212898, 278418,  76352, 110973,
       235105,
       ...
        88793, 274808,  44296,  28634,  73681,  59727, 188951, 268622,   9856,
       155916],
      dtype='int64', name='user_id', length=899)
```

```
ratings = ratings[ratings['user_id'].isin(y)]
```

```
ratings
```

	user_id	ISBN	rating
1456	277427	002542730X	10
1457	277427	0026217457	0
1458	277427	003008685X	8
1459	277427	0030615321	0
1460	277427	0060002050	0
...	...	...	...
1147612	275970	3829021860	0
1147613	275970	4770019572	0
1147614	275970	896086097	0
1147615	275970	9626340762	8
1147616	275970	9626344990	0

```
ratings.shape
```

```
(526356, 3)
```

```
ratings['user_id'].unique().shape
```

```
(899,)
```

- Dữ liệu này sau khi xử lý chỉ còn lại (526356 dòng, 3 cột) và có 899 giá trị duy nhất

- Ta sẽ thực hiện nối bảng dữ liệu “ratings” và “books” lại dựa vào dữ liệu từ cột “ISBN”

```
ratings_with_books = ratings.merge(books, on='ISBN')
```

```
ratings_with_books.head()
```

	user_id	ISBN	rating	title	author	year	publisher	image_url
0	277427	002542730X	10	Politically Correct Bedtime Stories: Modern Ta...	James Finn Garner	1994	John Wiley & Sons Inc	http://images.amazon.com/images/P/002542730X.0...
1	277427	0026217457	0	Vegetarian Times Complete Cookbook	Lucy Moll	1995	John Wiley & Sons	http://images.amazon.com/images/P/0026217457.0...
2	277427	003008685X	8	Pioneers	James Fenimore Cooper	1974	Thomson Learning	http://images.amazon.com/images/P/003008685X.0...
3	277427	0030615321	0	Ask for May, Settle for June (A Doonesbury book)	G. B. Trudeau	1982	Henry Holt & Co	http://images.amazon.com/images/P/0030615321.0...
4	277427	0060002050	0	On a Wicked Dawn (Cynster Novels)	Stephanie Laurens	2002	Avon Books	http://images.amazon.com/images/P/0060002050.0...

```
ratings_with_books.shape
```

```
(487671, 8)
```

- Tiếp theo ta sẽ nhóm các cột “rating” và “title” lại thành 1 nhóm để đếm số lượng đánh giá của mỗi cuốn sách là bao nhiêu và tạo thành 1 bảng dữ liệu mới

```
number_rating = ratings_with_books.groupby('title')['rating'].count().reset_index()
```

```
number_rating.head()
```

	title	rating
0	A Light in the Storm: The Civil War Diary of ...	2
1	Always Have Popsicles	1
2	Apple Magic (The Collector's series)	1
3	Beyond IBM: Leadership Marketing and Finance ...	1
4	Clifford Visita El Hospital (Clifford El Gran...	1

```
number_rating.shape
```

```
(160269, 2)
```

```
number_rating.rename(columns={'rating':'num_of_rating'},inplace=True)
```

```
number_rating.head()
```

	title	num_of_rating
0	A Light in the Storm: The Civil War Diary of ...	2
1	Always Have Popsicles	1
2	Apple Magic (The Collector's series)	1
3	Beyond IBM: Leadership Marketing and Finance ...	1
4	Clifford Visita El Hospital (Clifford El Gran...	1

- Sau đó ta sẽ nối 2 bảng “num_of_rating” và “ratings_with_books” lại với nhau dựa trên dữ liệu từ cột “title”

- Hình ảnh sau khi nối 2 bảng lại

```
final_rating = ratings_with_books.merge(number_rating, on='title')
```

```
final_rating.head()
```

	user_id	ISBN	rating	title	author	year	publisher	image_url	num_of_rating
0	277427	002542730X	10	Politically Correct Bedtime Stories: Modern Ta...	James Finn Garner	1994	John Wiley & Sons Inc	http://images.amazon.com/images/P/002542730X.0...	82
1	277427	0026217457	0	Vegetarian Times Complete Cookbook	Lucy Moll	1995	John Wiley & Sons	http://images.amazon.com/images/P/0026217457.0...	7
2	277427	003008685X	8	Pioneers	James Fenimore Cooper	1974	Thomson Learning	http://images.amazon.com/images/P/003008685X.0...	1
3	277427	0030615321	0	Ask for May, Settle for June (A Doonesbury book)	G. B. Trudeau	1982	Henry Holt & Co	http://images.amazon.com/images/P/0030615321.0...	1
4	277427	0060002050	0	On a Wicked Dawn (Cynster Novels)	Stephanie Laurens	2002	Avon Books	http://images.amazon.com/images/P/0060002050.0...	13

```
final_rating.shape
```

```
(487671, 9)
```

- Ta sẽ lọc ra những số lượng đánh giá >= 50 để làm cho dữ liệu cân bằng tránh việc dữ liệu bị quá nghiêng về 1 bên

```
# Lets take those books which got at least 50 rating of user
final_rating = final_rating[final_rating['num_of_rating'] >= 50]
```

```
final_rating.head()
```

	user_id	ISBN	rating	title	author	year	publisher	image_url	num_of_rating
0	277427	002542730X	10	Politically Correct Bedtime Stories: Modern Ta...	James Finn Garner	1994	John Wiley & Sons Inc	http://images.amazon.com/images/P/002542730X.0...	82
13	277427	0060930535	0	The Poisonwood Bible: A Novel	Barbara Kingsolver	1999	Perennial	http://images.amazon.com/images/P/0060930535.0...	133
15	277427	0060934417	0	Bel Canto: A Novel	Ann Patchett	2002	Perennial	http://images.amazon.com/images/P/0060934417.0...	108
18	277427	0061009059	9	One for the Money (Stephanie Plum Novels (Pape...	Janet Evanovich	1995	HarperTorch	http://images.amazon.com/images/P/0061009059.0...	108
24	277427	006440188X	0	The Secret Garden	Frances Hodgson Burnett	1998	HarperTrophy	http://images.amazon.com/images/P/006440188X.0...	79

```
final_rating.shape
```

```
(61853, 9)
```

- Sau đó ta sẽ xoá đi những giá trị trùng lặp giữa 2 cột “user_id” và “title”

```
# lets drop the duplicates
final_rating.drop_duplicates(['user_id','title'],inplace=True)
```

```
final_rating.shape
```

```
(59850, 9)
```

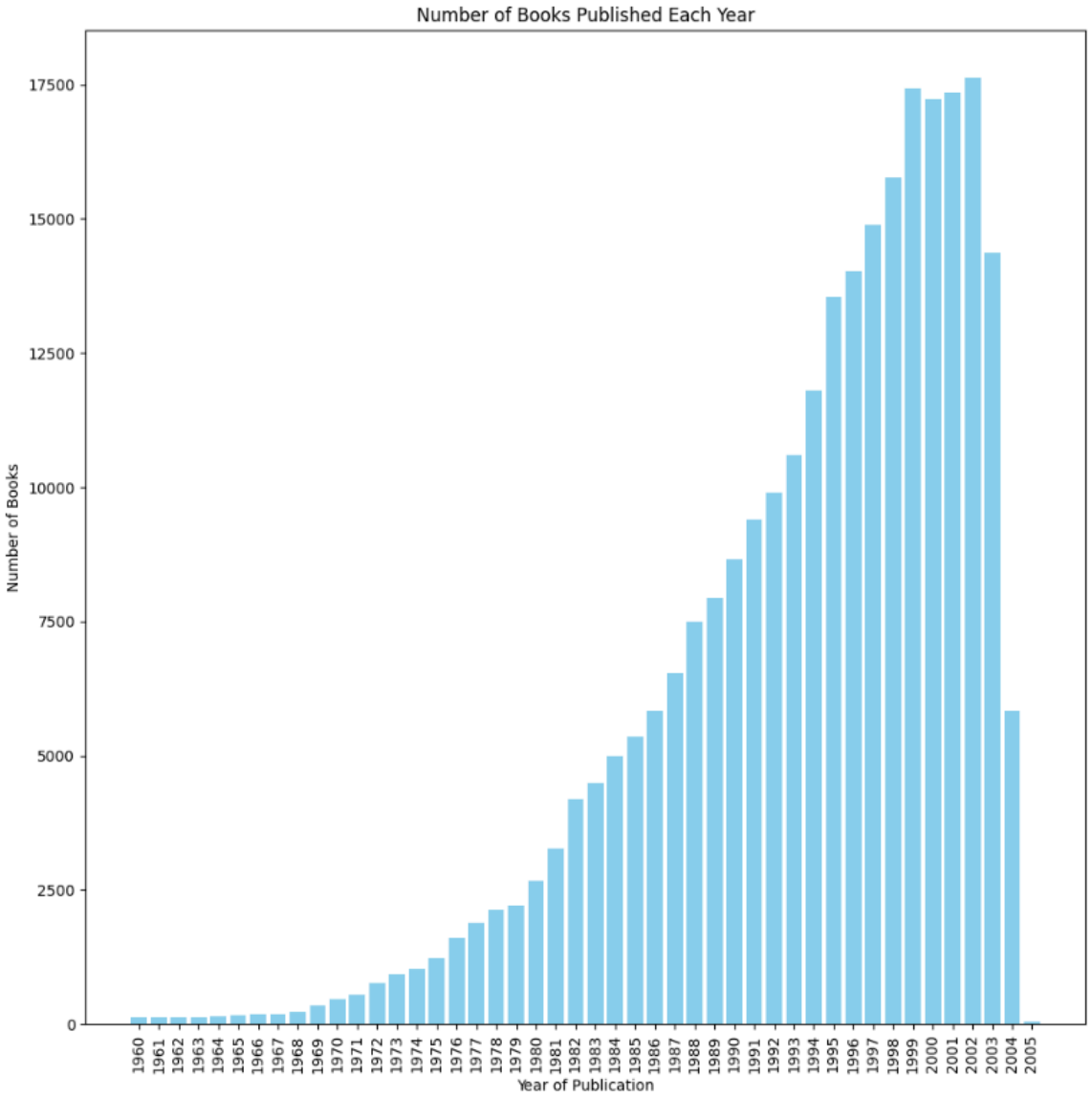
```
final_rating.head()
```

	user_id	ISBN	rating	title	author	year	publisher	image_url	num_of_rating
0	277427	002542730X	10	Politically Correct Bedtime Stories: Modern Ta...	James Finn Garner	1994	John Wiley & Sons Inc	http://images.amazon.com/images/P/002542730X.0...	82
13	277427	0060930535	0	The Poisonwood Bible: A Novel	Barbara Kingsolver	1999	Perennial	http://images.amazon.com/images/P/0060930535.0...	133
15	277427	0060934417	0	Bel Canto: A Novel	Ann Patchett	2002	Perennial	http://images.amazon.com/images/P/0060934417.0...	108
18	277427	0061009059	9	One for the Money (Stephanie Plum Novels (Pape...	Janet Evanovich	1995	HarperTorch	http://images.amazon.com/images/P/0061009059.0...	108
24	277427	006440188X	0	The Secret Garden	Frances Hodgson Burnett	1998	HarperTrophy	http://images.amazon.com/images/P/006440188X.0...	79

=> Sau khi xử lý ta được 1 bảng dữ liệu tên là “final_rating”chỉ còn (59850 dòng, 9 cột), ta đã giảm được 1 số lượng lớn dữ liệu từ các bảng ban đầu, điều này giúp cho bước traning model sẽ nhanh và chính xác hơn

3. Khai phá và trực quan hóa dữ liệu

3.1 Số lượng sách xuất bản mỗi năm

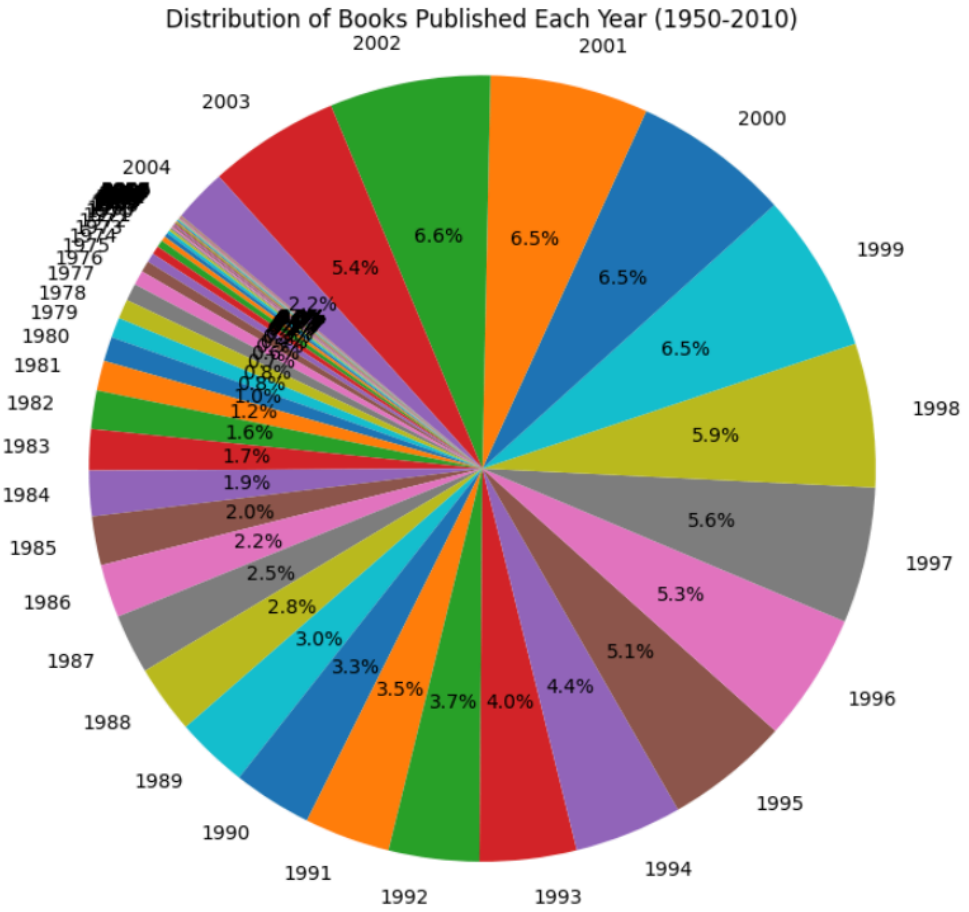


Dựa vào bảng dữ liệu và hình ảnh trực quan ta thấy rằng:

- Số lượng xuất bản sách mỗi năm là một con số quan trọng phản ánh sự phát triển và biến động của ngành xuất bản. Chúng được biểu thị qua các biểu đồ hoặc số thống kê, số lượng trợ giúp xuất bản mà chúng tôi hiểu rõ hơn về xu hướng đọc sách của chúng và hoạt động của các nhà xuất bản.
- Những năm có sự tăng đột biến về số lượng sách xuất bản có thể chỉ ra sự phát triển mạnh mẽ trong lĩnh vực sáng tạo.

=> Thông qua việc phân tích số lượng bản xuất bản theo từng năm, chúng tôi có thể nhận dạng các xu hướng dài hạn, đánh giá mức độ ảnh hưởng của yếu tố xã hội và công nghệ

3.2 Tỉ lệ phân phối sách xuất bản hàng năm(1950-2010)



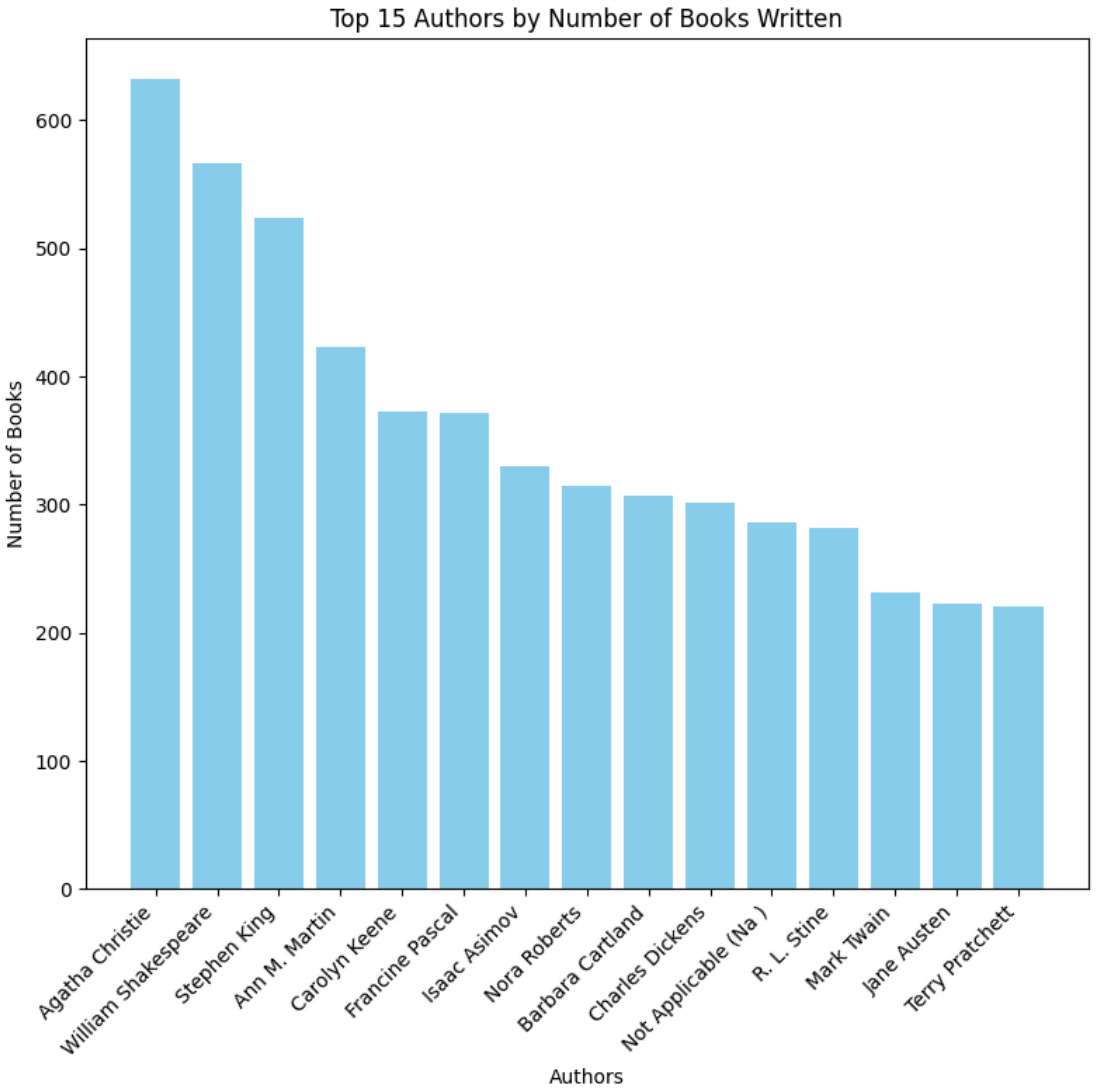
Dựa vào bảng dữ liệu và hình ảnh trực quan ta thấy rằng:

- Phân phối hợp lệ các bản xuất bản hàng năm từ năm 1950 đến năm 2010 đã tìm thấy những thay đổi rõ ràng trong các bản xuất bản lớn qua các thập kỷ.
- Trong giai đoạn đầu, số lượng xuất bản bản có xu hướng tăng đều đặn, Phản ánh sự phát triển mạnh mẽ của văn hóa đọc và sự tăng trưởng nhu cầu thông tin trong xã hội.
- Những năm 1990 và đầu thế kỷ 21 đã chứng minh kiến trúc tăng cường biến đổi về số lượng sách, hỗ trợ cho sự phát triển của các công ty xuất bản lớn và mở rộng thị trường toàn cầu.
- Trong khi đó, những năm gần cuối thập niên 2000 đã ghi nhận sự điều chỉnh lại khi ngành xuất bản phải phù hợp với sự cạnh tranh từ sách điện tử và các phương tiện tiện ích truyền thông số.

=> Tỷ lệ phân phối sách trong suốt hơn 60 năm qua phản ánh sự thay đổi trong xu hướng đọc sách, phát triển công nghệ và những ảnh

hưởng kinh tế và xã hội lên bản xuất bản chuyên ngành, cung cấp cái nhìn sâu sắc về tiến trình hóa nền văn hóa

3.3 Top 15 tác giả viết nhiều sách nhất

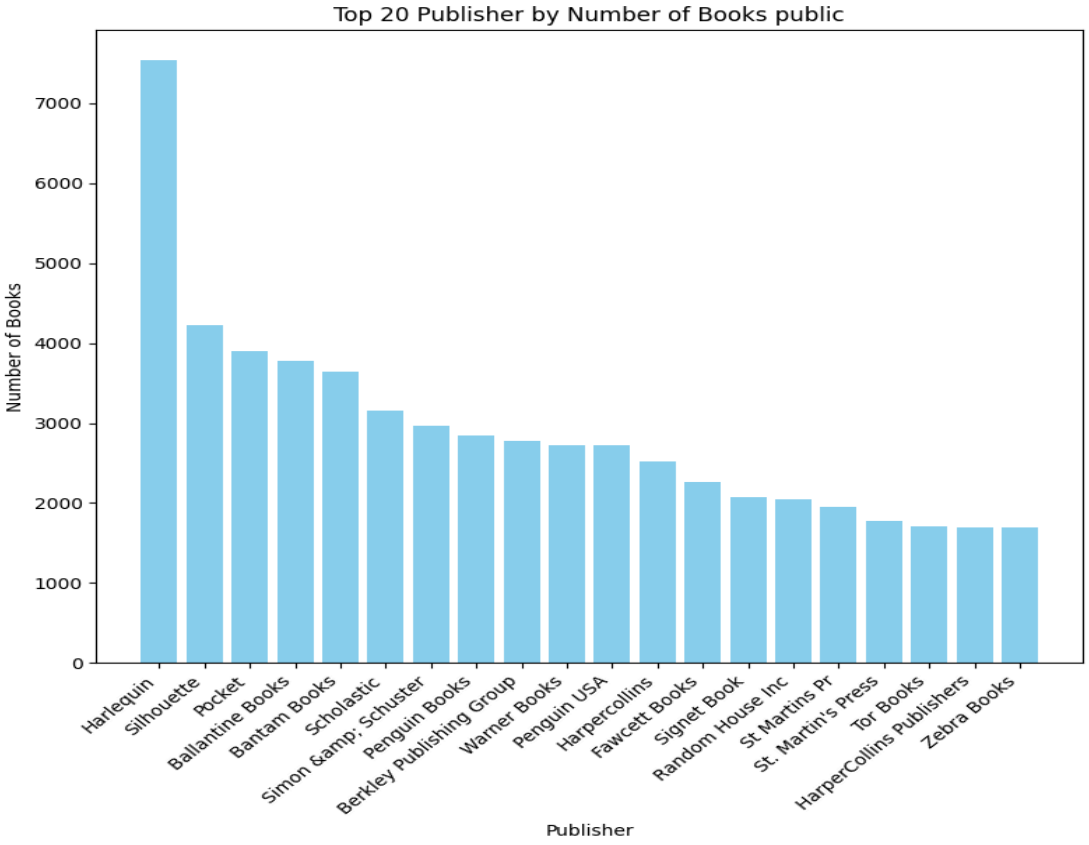


Dựa vào bảng dữ liệu và hình ảnh trực quan ta thấy rằng:

- Những tác giả này thường sở hữu một sự nghiệp dài và đầy sáng tạo, với hàng trăm tác phẩm trải nghiệm dài qua nhiều thể loại khác nhau, từ tiểu thuyết, sách khoa học, đến sách thiếu nhi.
- Sự liên tục hiện diện của họ trong danh sách này chứng tỏ khả năng duy trì chất lượng và sự thay đổi mới trong lối viết, đồng thời tạo ra một lượng lớn tài liệu phong phú cho độc giả.

=> Họ không chỉ đóng góp cho kho tàng văn học mà còn hình thành và ảnh hưởng sâu rộng đến các xu hướng đọc sách và nhu cầu văn học của công chúng.

3.4 Top 20 nhà xuất bản có lượng sách được xuất hiện nhiều nhất

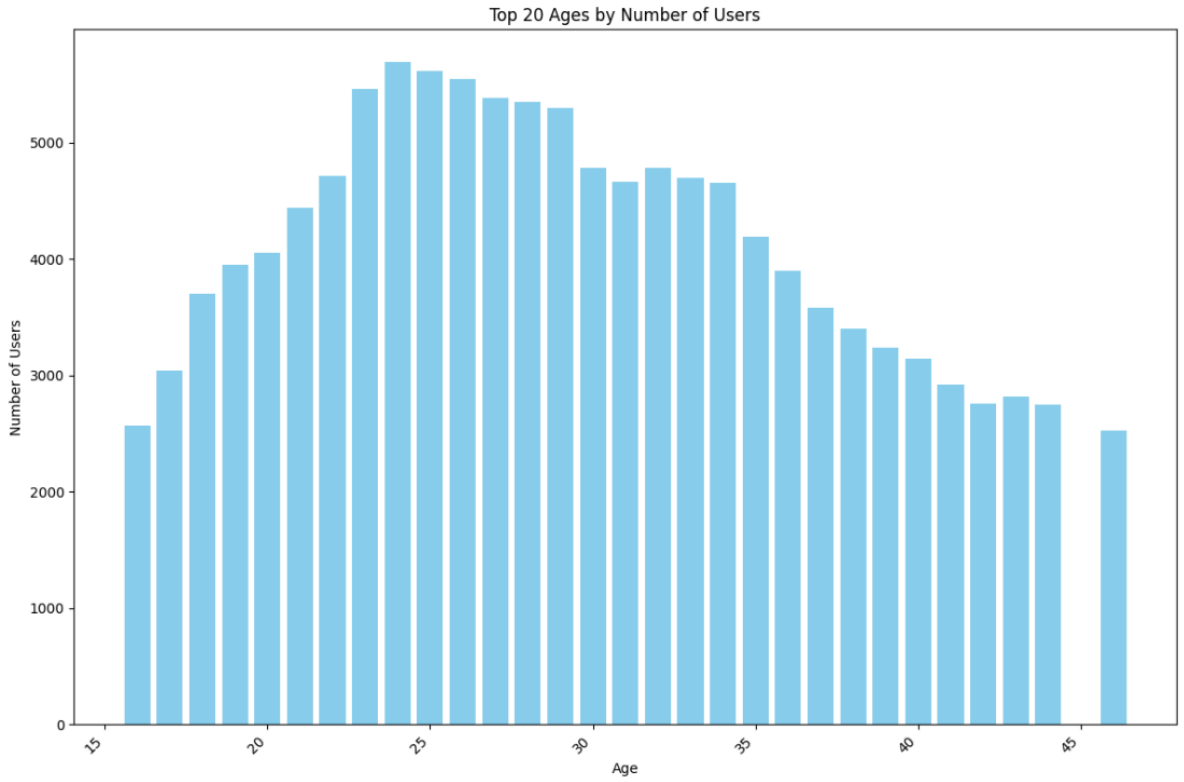


Dựa vào bảng dữ liệu và hình ảnh trực quan ta thấy rằng:

- Danh sách 20 nhà xuất bản hàng đầu có số lượng sách được xuất bản nhiều nhất cung cấp cái nhìn toàn diện về tác động của ảnh và mức độ rộng rãi của các tổ chức xuất bản hàng đầu trên thị trường.
- Các nhà xuất bản này không chứng minh rõ ràng khả năng sản xuất số lượng lớn sách nhưng vẫn cho thấy sự đa dạng và phong phú trong danh mục sách của họ. Việc xuất hiện nhiều sách trên thị trường phản ánh sự thành công trong công việc duy trì chất lượng và phù hợp với nhu cầu đa dạng của độc giả.
- Các tổ chức này thường có khả năng phát hiện và phát triển các hoạt động mới, đồng thời giữ vững hiệu quả và uy tín của mình trong ngành. Họ đóng vai trò quan trọng trong công việc định hình xu hướng học văn bản và đóng góp vào sự phát triển của nền văn hóa đọc.

=> Phân tích danh sách 20 nhà xuất bản hàng đầu giúp chúng tôi hiểu rõ hơn về cấu trúc và sự phát triển của các sản phẩm chuyên ngành, đồng thời cung cấp thông tin về các chủ sở hữu tổ chức đang dẫn đầu.

3.5 Top 20 độ tuổi có số lượng người dùng lớn nhất



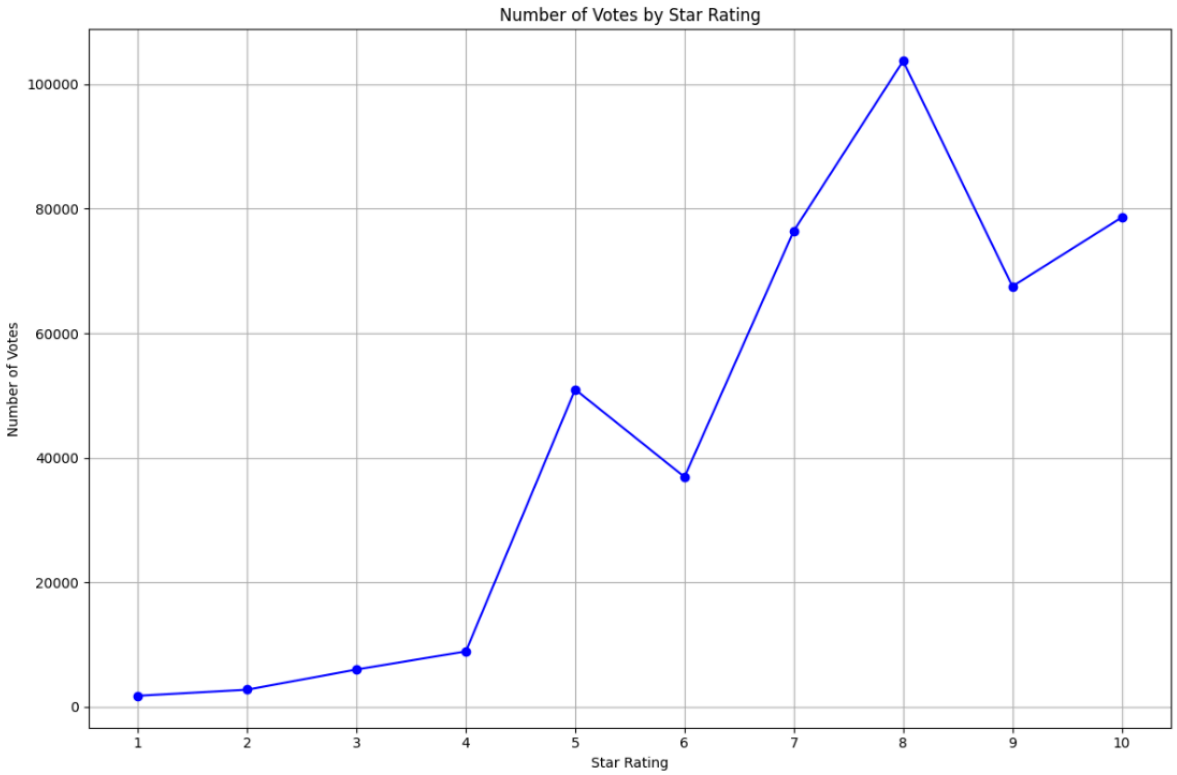
Dựa vào bảng dữ liệu và hình ảnh trực quan ta thấy rằng:

- Danh sách top 20 độ tuổi có số lượng người dùng lớn nhất trong việc tìm kiếm sách mang lại cái nhìn rõ nét về các nhóm tuổi có sự quan tâm và nhu cầu đọc sách cao nhất.
- Những độ tuổi này thường bao gồm các nhóm người từ thanh thiếu niên, sinh viên đại học đến người trưởng thành và trung niên, là những đối tượng có nhu cầu tìm kiếm tri thức, giải trí hoặc phát triển cá nhân thông qua việc đọc.
- Các nhóm tuổi này có thể ưu tiên các thể loại sách khác nhau, từ sách giáo dục, sách phát triển bản thân, tiểu thuyết, đến sách chuyên ngành, phản ánh các giai đoạn khác nhau của cuộc sống và mối quan tâm cá nhân.
- Việc nhận diện các độ tuổi có số lượng người dùng lớn nhất trong việc tìm kiếm sách không chỉ giúp các nhà xuất bản và

nhà sách điều chỉnh chiến lược tiếp cận mà còn cung cấp thông tin quý giá cho các dịch vụ thư viện và nền tảng trực tuyến, nhằm đáp ứng tốt hơn nhu cầu của độc giả.

=> Hiểu rõ sở thích và hành vi tìm kiếm sách của từng nhóm tuổi là chìa khóa để phát triển thị trường, tối ưu hóa danh mục sản phẩm, và mang đến những trải nghiệm đọc phù hợp, phong phú cho từng đối tượng.

3.6 Tỉ lệ phiếu bầu trên thang 10 sao



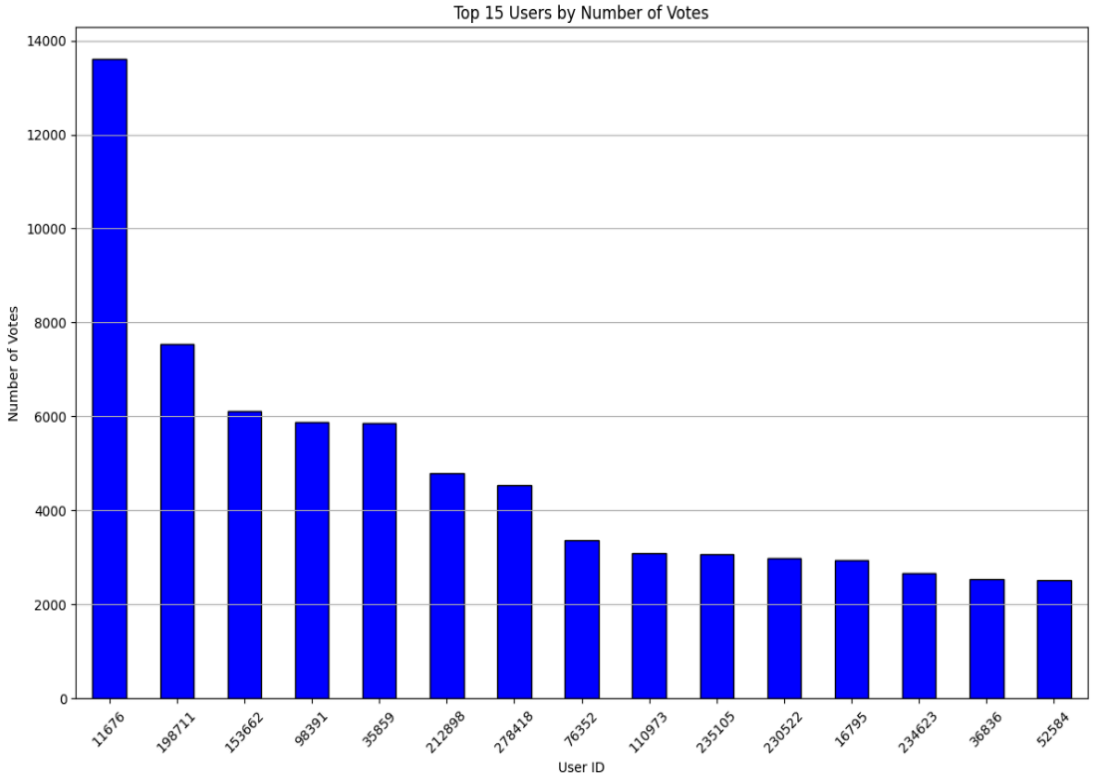
Dựa vào bảng dữ liệu và hình ảnh trực quan ta thấy rằng:

- Tỉ lệ phiếu bầu trên thang 10 sao trong việc đánh giá sách là một công cụ quan trọng để đo lường mức độ hài lòng của độc giả và chất lượng của cuốn sách.
- Thang điểm này cho phép độc giả dễ dàng biểu đạt cảm nhận của mình, từ rất không hài lòng (1 sao) đến cực kỳ hài lòng (10 sao). Khi một cuốn sách nhận được nhiều phiếu bầu cao, chẳng hạn như 8, 9, hoặc 10 sao, điều này cho thấy nội dung, cốt truyện, và phong cách viết đã đáp ứng tốt kỳ vọng của người đọc.

- Ngược lại, nếu cuốn sách nhận nhiều phiếu bầu thấp, đó có thể là dấu hiệu cho thấy cần phải cải thiện về cốt truyện, nhân vật, hoặc cách diễn đạt.

=> Tỉ lệ phiếu bầu trên thang 10 sao không chỉ giúp độc giả khác tham khảo để lựa chọn sách phù hợp với sở thích của mình, mà còn cung cấp thông tin quý giá cho các tác giả và nhà xuất bản trong việc điều chỉnh và nâng cao chất lượng tác phẩm, đáp ứng tốt hơn nhu cầu của độc giả.

3.7 Top 15 người dùng hàng đầu theo số phiếu bầu



Dựa vào bảng dữ liệu và hình ảnh trực quan ta thấy rằng:

- Danh sách top 15 người dùng hàng đầu theo số phiếu bầu trong việc đọc sách là một thước đo quan trọng phản ánh sự tham gia và ảnh hưởng của các độc giả tích cực trong cộng đồng yêu sách.
- Những người dùng này không chỉ đọc nhiều sách mà còn đóng góp đáng kể bằng cách chia sẻ đánh giá và ý kiến của mình qua các phiếu bầu. Với số lượng phiếu bầu lớn, họ có vai trò quan trọng trong việc định hình nhận thức về chất

lượng sách, giúp những độc giả khác dễ dàng lựa chọn những cuốn sách phù hợp với sở thích cá nhân.

- Các đánh giá từ nhóm người dùng hàng đầu này thường được cộng đồng tin tưởng và coi trọng, bởi họ có kinh nghiệm đọc sâu rộng và khả năng nhận xét khách quan, chi tiết.

=> Sự hiện diện của họ trong top 15 không chỉ cho thấy đam mê và cam kết với việc đọc, mà còn nhấn mạnh vai trò của họ trong việc tạo nên một môi trường văn học phong phú, nơi mà ý kiến và trải nghiệm đọc được chia sẻ và lan tỏa rộng rãi.

III. Training model KNN và đánh giá.

1. Các thư viện sử dụng

1.1. Pandas

Pandas là một thư viện dữ liệu mã nguồn mở mạnh mẽ trong Python, được thiết kế để dễ dàng thao tác, phân tích và quản lý dữ liệu có cấu trúc. Nó cung cấp hai cấu trúc dữ liệu chính:

- **Series**: Một mảng một chiều có nhãn, tương tự như một cột trong bảng, giúp quản lý các dữ liệu đơn lẻ một cách hiệu quả.
- **DataFrame**: Một cấu trúc hai chiều giống như bảng tính, có khả năng lưu trữ và thao tác với các tập dữ liệu lớn, hỗ trợ nhiều kiểu dữ liệu khác nhau.

Pandas tích hợp chặt chẽ với các thư viện Python khác như NumPy để cung cấp các thao tác dữ liệu hiệu quả. Nó cung cấp các công cụ linh hoạt để lọc, nhóm, và tổng hợp dữ liệu, giúp người dùng dễ dàng chuyển đổi và làm sạch dữ liệu trước khi phân tích hoặc mô hình hóa.

1.2. Matplotlib

Matplotlib là một thư viện Python chuyên dụng cho việc tạo ra các biểu đồ và hình ảnh trực quan từ dữ liệu. Nó hỗ trợ một loạt các loại biểu đồ từ cơ bản đến phức tạp, bao gồm:

- **Biểu đồ đường**: Thể hiện sự thay đổi của một hoặc nhiều biến số theo thời gian.
- **Biểu đồ phân tán (scatter plot)**: Dùng để thể hiện mối quan hệ giữa hai biến số.
- **Biểu đồ cột và biểu đồ thanh (bar chart)**: Thể hiện so sánh giữa các nhóm dữ liệu.

Matplotlib có khả năng tùy biến cao, cho phép người dùng điều chỉnh từng chi tiết của biểu đồ, từ kích thước, màu sắc, đến kiểu đường nét. Khả năng xuất ra nhiều định dạng khác nhau như PNG, PDF, SVG giúp dễ dàng tích hợp vào các báo cáo và ấn phẩm khoa học.

1.3. Scikit-learn

Scikit-learn là một thư viện Python mạnh mẽ và toàn diện, cung cấp các công cụ cho học máy và khai thác dữ liệu. Nó bao gồm một loạt các thuật toán phổ biến, chẳng hạn như:

- **Phân loại (Classification)**: Thuật toán như SVM, Random Forest, và k-Nearest Neighbors giúp phân loại dữ liệu vào các nhóm khác nhau.
- **Hồi quy (Regression)**: Các mô hình như Linear Regression, Ridge, và Lasso giúp dự đoán các giá trị liên tục.
- **Phân cụm (Clustering)**: Thuật toán như K-means và DBSCAN được sử dụng để nhóm dữ liệu thành các cụm có ý nghĩa.

Thư viện này cũng hỗ trợ các công cụ để đánh giá và lựa chọn mô hình, bao gồm cross-validation và các thước đo đánh giá hiệu suất, giúp tối ưu hóa quá trình xây dựng mô hình.

1.4. SciPy

SciPy là một thư viện Python được xây dựng trên NumPy, cung cấp các chức năng cấp cao cho khoa học tính toán. Nó bao gồm nhiều mô-đun để giải quyết các bài toán phức tạp:

- **Tối ưu hóa (Optimization)**: Bao gồm các thuật toán tối ưu hóa hàm số như minimize, fmin, hỗ trợ tìm điểm cực tiểu và cực đại của các hàm số trong không gian nhiều chiều.
- **Tích phân (Integration)**: Các công cụ như quad, dblquad giúp tính toán tích phân xác định với độ chính xác cao, hữu ích trong các bài toán vật lý và kỹ thuật.
- **Đại số tuyến tính (Linear Algebra)**: Hỗ trợ giải các hệ phương trình tuyến tính, tìm giá trị riêng, và nhiều phép biến đổi ma trận khác.

Ngoài ra, SciPy còn cung cấp các công cụ xử lý tín hiệu và hình ảnh, phân tích thống kê, và hỗ trợ làm việc với ma trận thưa (sparse matrices), cực kỳ hữu ích cho các bài toán yêu cầu lưu trữ và xử lý dữ liệu kích thước lớn.

1.5. sklearn.neighbors.NearestNeighbors

Model NearestNeighbors trong Scikit-learn là một công cụ hiệu quả để thực hiện các thuật toán tìm kiếm hàng xóm gần nhất, được sử dụng phổ biến trong cả phân loại và hồi quy:

- **K-Nearest Neighbors (KNN)**: Đây là một thuật toán phi tham số dùng để phân loại hoặc hồi quy dữ liệu bằng cách xem xét các điểm gần nhất trong không gian đặc trưng. Nó hoạt động dựa trên khoảng cách giữa các điểm dữ liệu và có thể dễ dàng tùy chỉnh bằng các hàm đo khoảng cách khác nhau, như Euclidean hoặc Manhattan.
- **Ưu điểm**: Thuật toán KNN đơn giản, dễ triển khai và không yêu cầu giả định mạnh mẽ về phân phối dữ liệu. Nó thường được sử dụng như một phương pháp cơ sở trong các bài toán học máy.

Mô-đun này cũng có thể xử lý dữ liệu kích thước lớn một cách hiệu quả thông qua việc sử dụng cấu trúc dữ liệu cây (k-d tree, Ball Tree) để tăng tốc quá trình tìm kiếm hàng xóm gần nhất.

2. Principal Component Analysis (PCA)

2.1. Khái niệm

Principal Component Analysis (PCA) là một kỹ thuật phân tích dữ liệu dùng để giảm số lượng biến (đặc trưng) trong tập dữ liệu bằng cách chuyển đổi các biến gốc thành các thành phần chính không tương quan, trong đó các thành phần này giải thích nhiều biến thiên nhất trong dữ liệu.

2.2. Chức Năng

- **Giảm chiều dữ liệu**: Giảm số lượng đặc trưng (biến) trong dữ liệu mà vẫn giữ lại phần lớn thông tin quan trọng.
- **Tìm thành phần chính**: Xác định các hướng (thành phần chính) trong không gian dữ liệu mà dữ liệu phân tán nhiều nhất.
- **Tăng cường khả năng trực quan**: Giúp giảm số chiều của dữ liệu để dễ dàng trực quan hóa.

2.3. Cơ Chế Hoạt Động

1. **Chuẩn hóa dữ liệu**: Loại bỏ trung bình của từng đặc trưng.
2. **Tính ma trận hiệp phương sai**: Đo lường mối quan hệ giữa các đặc trưng.
3. **Tính toán các giá trị eigen và vector eigen**: Xác định các thành phần chính.
4. **Chọn các thành phần chính**: Chọn các thành phần có giá trị eigen lớn nhất để giữ lại.
5. **Chiếu dữ liệu**: Biến đổi dữ liệu vào không gian của các thành phần chính đã chọn.

2.4. Ưu, nhược điểm

Ưu Điểm

- **Giảm chiều dữ liệu**: Giúp giảm số lượng biến mà không làm mất nhiều thông tin.
- **Giảm thiểu đa cộng tuyến**: Loại bỏ các đặc trưng tương quan mạnh.
- **Cải thiện hiệu suất học máy**: Giảm tải tính toán và giúp các thuật toán học máy hoạt động hiệu quả hơn.

Nhược Điểm

- **Mất thông tin**: Một số thông tin có thể bị mất khi giảm số chiều.
- **Khó giải thích**: Các thành phần chính không luôn dễ hiểu hoặc có thể không có ý nghĩa rõ ràng.
- **Yêu cầu chuẩn hóa dữ liệu**: Kết quả phụ thuộc vào việc dữ liệu có được chuẩn hóa hay không.

2.5. Các Phương Thức PCA

1. **PCA Cơ Bản**: Phương pháp truyền thống dựa trên tính toán ma trận hiệp phương sai và vector eigen.
2. **SVD (Phân Tích Giá trị đơn) PCA**: Sử dụng phân tích giá trị đơn (Singular Value Decomposition) thay cho ma trận hiệp phương sai, có thể hiệu quả hơn trong một số trường hợp.
3. **Kernel PCA**: Mở rộng PCA bằng cách sử dụng hàm hạt nhân để xử lý dữ liệu phi tuyến, giúp phát hiện cấu trúc phi tuyến trong dữ liệu.
4. **Sparse PCA**: Thêm điều kiện phân tách để giữ cho các thành phần chính trở nên thưa thớt hơn, giúp dễ dàng giải thích hơn.

2.6. Thực Thi và Công Cụ

- **Thư viện phần mềm**: PCA được hỗ trợ bởi nhiều thư viện trong các ngôn ngữ lập trình phổ biến như:
  - **Python**: scikit-learn, numpy, pandas
  - **R**: prcomp, PCA
  - **MATLAB**: pca function
- **Khả năng tương thích**: PCA có thể hoạt động với dữ liệu lớn và phức tạp nhưng hiệu quả tính toán có thể bị ảnh hưởng nếu số lượng biến rất lớn so với số lượng mẫu.

2.7. Ứng Dụng Thực Tế

- **Nhận diện mẫu**: Trong phân tích hình ảnh và nhận diện đối tượng.
- **Kết hợp dữ liệu**: Trong việc kết hợp và phân tích dữ liệu từ các nguồn khác nhau.
- **Giảm số chiều dữ liệu**: Trong các bài toán học máy như phân loại và hồi quy.

2.8. Áp dụng trong Traning Model:

Out[63]:

user_id	254	2276	2766	2977	3363	3757	4017	4385	6242	6251	...	274004	274061	274301	274308	274808	27597
title																	
1984	9.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	0
1st to Die: A Novel	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	Na
2nd Chance	NaN	10.0	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	0.0	NaN	Na
4 Blondes	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	0.0	...	NaN	NaN	NaN	NaN	NaN	Na
84 Charing Cross Road	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	10
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
Year of Wonders	NaN	NaN	NaN	7.0	NaN	NaN	NaN	NaN	7.0	NaN	...	NaN	NaN	NaN	NaN	NaN	0
You Belong To Me	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	Na
Zen and the Art of Motorcycle Maintenance: An Inquiry into Values	NaN	NaN	NaN	NaN	0.0	NaN	NaN	NaN	NaN	0.0	...	NaN	NaN	NaN	NaN	NaN	0
Zoya	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	Na
\O\" Is for Outlaw"	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	8.0	NaN	NaN	Na

742 rows × 888 columns

Như hình trên đó là bảng dữ liệu chứa thông tin về đánh giá của người dùng đối với các cuốn sách. Cụ thể:

- **Các hàng (title):** Đại diện cho các tiêu đề sách.
- **Các cột (user_id):** Đại diện cho các ID người dùng.
- **Các ô trong bảng:** Chứa các điểm đánh giá mà người dùng dành cho cuốn sách. Nếu ô chứa "NaN" (Not a Number), điều đó có nghĩa là người dùng chưa đánh giá cuốn sách đó.

Có 742 đầu sách và 888 user tương ứng với 742 hàng x 888 cột. Mỗi đầu sách tương ứng với 1 vector có 888 chiều.

In [65]:
```
book_pivot.fillna(0, inplace=True)
```

In [69]:
```
book_pivot.head()
```

Out[69]:

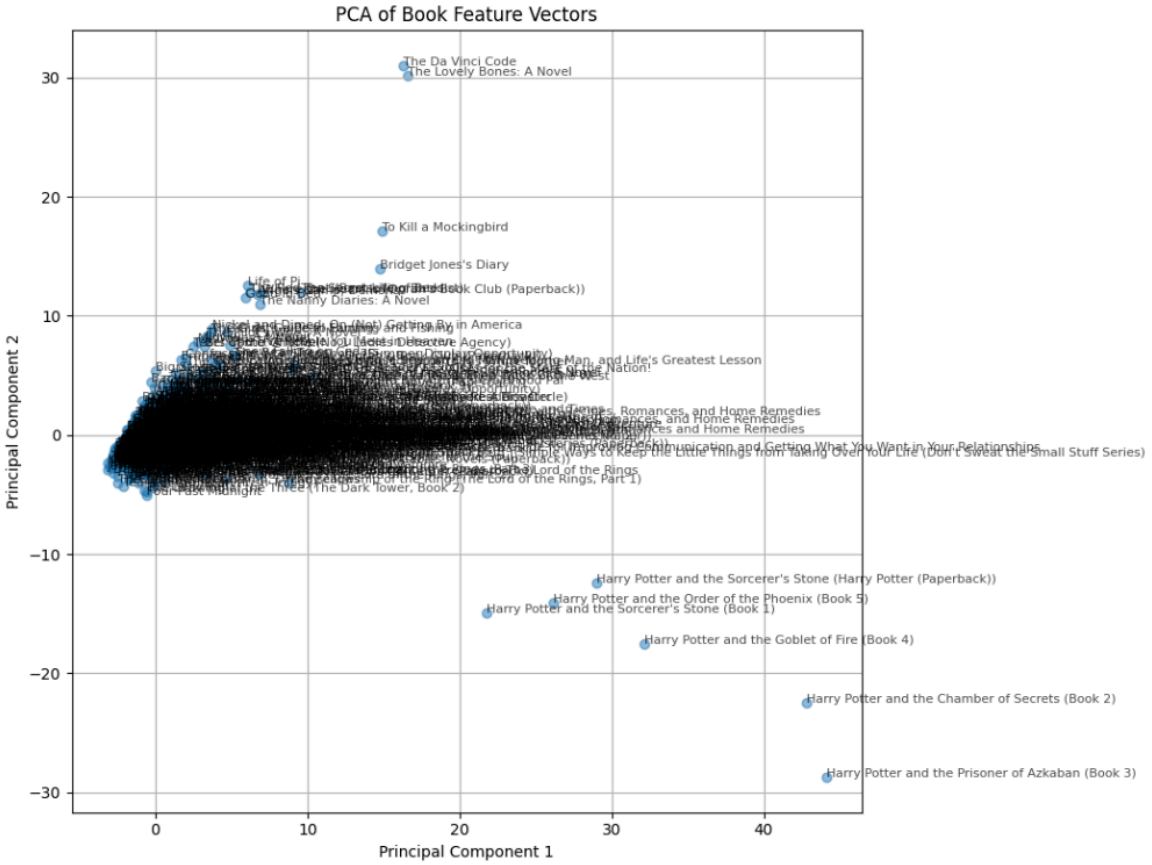
user_id	254	2276	2766	2977	3363	3757	4017	4385	6242	6251	...	274004	274061	274301	274308	274808	275970	27
title																		
1984	9.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	
1st to Die: A Novel	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	
2nd Chance	0.0	10.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	
4 Blondes	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	0.0	
84 Charing Cross Road	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	0.0	0.0	0.0	0.0	10.0	

- book_pivot.fillna(0, inplace=True): Phương thức này được sử dụng để thay thế tất cả các giá trị bị thiếu (NaN) trong DataFrame bằng một giá trị cụ thể, ở đây là 0.
- book_pivot.head(): Chỉ định 5 hàng đầu tiên trong bảng dữ liệu.

```python
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler

# Assuming book_pivot is already created
# Fill NaN values with 0 (or you can use another strategy)
book_pivot_filled = book_pivot.fillna(0)

# Standardize the data
scaler = StandardScaler()
book_pivot_scaled = scaler.fit_transform(book_pivot_filled)

# Perform PCA to reduce to 2 dimensions
pca = PCA(n_components=2)
book_pivot_pca = pca.fit_transform(book_pivot_scaled)

# Create a DataFrame for the PCA results
pca_df = pd.DataFrame(book_pivot_pca, index=book_pivot_filled.index, columns=['PC1', 'PC2'])

# Plotting the PCA results
plt.figure(figsize=(8, 8))
plt.scatter(pca_df['PC1'], pca_df['PC2'], alpha=0.5)
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.title('PCA of Book Feature Vectors')
plt.grid(True)
plt.tight_layout()

# Annotate points with movie titles
for i, title in enumerate(pca_df.index):
    plt.annotate(title, (pca_df['PC1'][i], pca_df['PC2'][i]), fontsize=8, alpha=0.7)

plt.show()
```

- scaler = StandardScaler()
  book_pivot_scaled = scaler.fit_transform(book_pivot_filled): Chuẩn hóa dữ liệu

- pca = PCA(n_components=2): Tạo một đối tượng PCA với số lượng thành phần chính là 2. Điều này có nghĩa là chúng ta sẽ giảm dữ liệu xuống còn 2 chiều.
- book_pivot_pca = pca.fit_transform(book_pivot_scaled): thực hiện PCA và giảm số chiều của dữ liệu từ book_pivot_scaled xuống còn 2 thành phần chính (book_pivot_pca).
- pca_df = pd.DataFrame(book_pivot_pca, index=book_pivot_filled.index, columns=['PC1', 'PC2']): Tạo một DataFrame mới ‘pca_df’ chứa hai thành phần chính (PC1 và PC2) để lưu trữ kết quả của PCA.
- Phần còn lại sẽ phụ trách vẽ biểu đồ kết quả PCA, ghi chú các điểm dữ liệu và hiển thị đồ thị.

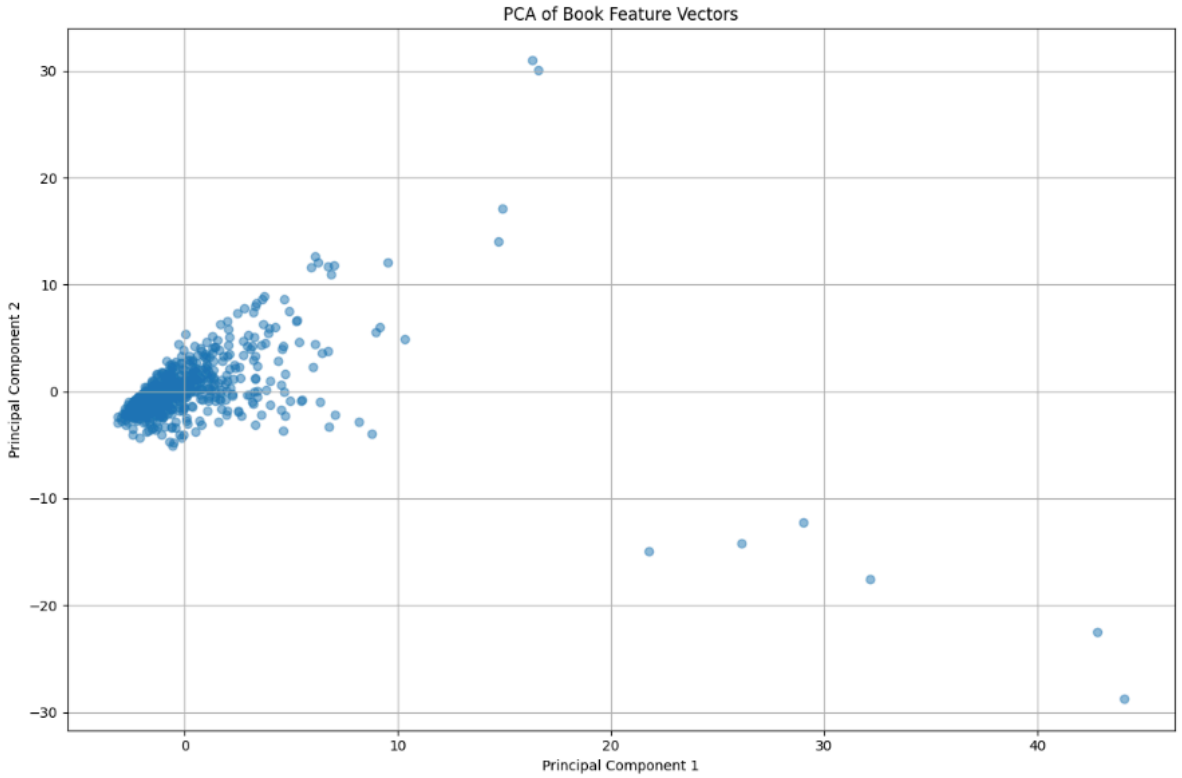


Ở trên là biểu đồ kết quả PCA trên vector đặc trưng của các đầu sách. Biểu đồ thể hiện sự phân bố của các đầu sách dựa theo 2 thành phần.

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler

# Assuming book_pivot is already created
# Fill NaN values with 0 (or you can use another strategy)
book_pivot_filled = book_pivot.fillna(0)

# Add an index column starting from 0
book_pivot_filled['index'] = range(len(book_pivot_filled))

# Standardize the data
scaler = StandardScaler()
book_pivot_scaled = scaler.fit_transform(book_pivot_filled.drop('index', axis=1))

# Perform PCA to reduce to 2 dimensions
pca = PCA(n_components=2)
book_pivot_pca = pca.fit_transform(book_pivot_scaled)

# Create a DataFrame for the PCA results
pca_df = pd.DataFrame(book_pivot_pca, index=book_pivot_filled['index'], columns=['PC1', 'PC2'])

# Plotting the PCA results
plt.figure(figsize=(12, 8))
plt.scatter(pca_df['PC1'], pca_df['PC2'], alpha=0.5)
plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.title('PCA of Book Feature Vectors')
plt.grid(True)
plt.tight_layout()
plt.show()
```

- book_pivot_filled['index'] = range(len(book_pivot_filled)): Thêm một cột index vào DataFrame book_pivot_filled, với giá trị là số nguyên liên tiếp bắt đầu từ 0.
- pca_df = pd.DataFrame(book_pivot_pca, index=book_pivot_filled['index'], columns=['PC1', 'PC2']):
    + Tạo một DataFrame mới **“pca_df”** để lưu trữ kết quả PCA.
    + **“index=book_pivot_filled['index']”** sử dụng cột index từ DataFrame gốc làm chỉ mục cho DataFrame PCA mới.
    + **“columns=['PC1', 'PC2']”** đặt tên cho các cột trong DataFrame mới tương ứng với hai thành phần chính.

Phần còn lại sẽ phụ trách vẽ biểu đồ PCA với các vector đã loại bỏ tên để giúp cho biểu đồ trở nên trực quan hóa hơn.



3. Thuật toán K-Nearest Neighbors (KNN)

A. Giới Thiệu Thuật Toán K-Nearest Neighbors (KNN)

3.1. Giới thiệu về Thuật toán K-Nearest Neighbors (KNN)

K-Nearest Neighbors (KNN) là một thuật toán học máy đơn giản nhưng mạnh mẽ, được sử dụng rộng rãi trong các bài toán phân loại và hồi quy. Thuật toán này dựa trên nguyên lý khoảng cách, hoạt động theo phương pháp tìm kiếm các điểm dữ liệu gần nhất trong không gian đặc trưng.

3.2. Cơ chế hoạt động

3.2.1. Tìm K Hàng Xóm Gần Nhất

- Khoảng cách: Khi cần phân loại hoặc dự đoán cho một điểm dữ liệu mới, KNN sẽ tính toán khoảng cách giữa điểm đó và tất cả các

điểm dữ liệu trong tập huấn luyện. Khoảng cách thường được đo bằng các hàm như Euclidean, Manhattan, hoặc Minkowski.
- Lựa chọn K điểm gần nhất: KNN sau đó sẽ lựa chọn K điểm dữ liệu gần nhất với điểm mới dựa trên khoảng cách đã tính.

3.2.2. Bỏ Phiếu hoặc Tính Trung Bình

- Phân loại: KNN xác định nhãn của điểm dữ liệu mới dựa trên đa số phiếu từ K điểm dữ liệu gần nhất. Điểm dữ liệu mới sẽ nhận nhãn mà xuất hiện nhiều nhất trong K điểm gần nhất.
- Hồi quy: KNN tính giá trị dự đoán cho điểm dữ liệu mới bằng cách trung bình giá trị của K điểm gần nhất.

3.3. Ưu điểm và Nhược điểm

3.3.1. Ưu điểm

- Đơn giản và Dễ Hiểu: KNN là một thuật toán đơn giản, dễ triển khai và trực quan, không yêu cầu nhiều giả định về phân phối dữ liệu.
- Không Cần Huấn Luyện: KNN không có giai đoạn huấn luyện cụ thể, giúp giảm thiểu rủi ro overfitting.
- Linh Hoạt: KNN có thể sử dụng cho cả bài toán phân loại và hồi quy, và dễ dàng mở rộng để làm việc với nhiều loại dữ liệu khác nhau.

3.3.2. Nhược điểm

- Tốc Độ Chậm: KNN có thể trở nên rất chậm khi xử lý các tập dữ liệu lớn vì phải tính toán khoảng cách từ điểm mới đến tất cả các điểm dữ liệu trong tập huấn luyện.
- Nhạy Cảm Với Nhiễu: KNN có thể bị ảnh hưởng mạnh bởi nhiễu trong dữ liệu, đặc biệt khi K nhỏ.
- Yêu Cầu Bộ Nhớ Cao: KNN yêu cầu lưu trữ toàn bộ dữ liệu huấn luyện, do đó có thể tiêu tốn nhiều bộ nhớ khi dữ liệu lớn.

3.4. Các Biến Thể và Cải Tiến

3.4.1. K-D Tree và Ball Tree

- K-D Tree: Đây là một cấu trúc dữ liệu cây phân tách không gian theo từng chiều để tăng tốc độ tìm kiếm các điểm gần nhất. K-D Tree hoạt động hiệu quả với dữ liệu có số chiều thấp (dưới 20 chiều). Tuy nhiên, khi số chiều tăng cao, hiệu suất của K-D Tree giảm đi đáng kể.
- Ball Tree: Ball Tree chia không gian thành các "quả cầu" để tối ưu hóa việc tìm kiếm hàng xóm. Cấu trúc này hiệu quả với dữ liệu có chiều thấp đến trung bình. Tuy nhiên, nó cũng gặp khó khăn khi dữ liệu có số chiều rất cao và không phù hợp khi sử dụng các khoảng cách không phải Euclidean.

3.4.2. Weighting (Sử dụng trọng số)

Thay vì áp dụng nguyên tắc đa số phiếu đơn giản, KNN có thể cải thiện độ chính xác bằng cách áp dụng trọng số dựa trên khoảng cách. Các điểm gần hơn sẽ có trọng số cao hơn, nghĩa là ảnh hưởng lớn hơn đến kết quả phân loại hoặc hồi quy.

3.4.3. Approximate Nearest Neighbors (ANN)

ANN là một phương pháp tìm kiếm gần đúng các hàng xóm với tốc độ cao hơn, đặc biệt hữu ích trong các ứng dụng thời gian thực. Dù tốc độ nhanh hơn, độ chính xác của ANN có thể thấp hơn so với các phương pháp tìm kiếm chính xác như brute-force hoặc K-D Tree.

3.5. Ứng Dụng Thực Tế

3.5.1. Nhận Dạng Mẫu

- Phân loại văn bản: KNN có thể được sử dụng để phân loại tài liệu hoặc email dựa trên nội dung văn bản.
- Nhận dạng chữ viết tay: KNN là một trong những phương pháp được sử dụng trong nhận dạng chữ viết tay, giúp phân loại các ký tự dựa trên hình dáng của chúng.
- Phân loại hình ảnh: KNN có thể phân loại hình ảnh dựa trên các đặc trưng hình học hoặc pixel.

3.5.2. Phát Hiện Bất Thường

KNN có thể được sử dụng để phát hiện các điểm dữ liệu bất thường (outliers) trong các bài toán như phát hiện gian lận, giám sát điều kiện máy móc, và nhiều ứng dụng khác.

3.6. Các Loại Thuật Toán KNN

KNN có nhiều phương pháp khác nhau để thực hiện tìm kiếm hàng xóm gần nhất, mỗi phương pháp có ưu và nhược điểm riêng:

3.6.1. Brute-force

- Mô tả: Tính toán khoảng cách từ điểm cần dự đoán đến tất cả các điểm dữ liệu trong tập huấn luyện.
- Ưu điểm: Đảm bảo chính xác cao, không bị ảnh hưởng bởi số chiều dữ liệu và tương thích với mọi loại khoảng cách.
- Nhược điểm: Chậm khi xử lý dữ liệu lớn, đòi hỏi nhiều tài nguyên tính toán.

3.6.2. Ball Tree

- Mô tả: Chia không gian thành các "quả cầu" để tối ưu hóa việc tìm kiếm hàng xóm gần nhất.
- Ưu điểm: Hiệu quả với dữ liệu có chiều thấp và trung bình.
- Nhược điểm: Hiệu quả giảm khi dữ liệu có nhiều chiều và khó sử dụng với khoảng cách không phải Euclidean.

3.6.3. K-D Tree

- Mô tả: Phân tách không gian dữ liệu theo từng chiều để tìm kiếm nhanh hơn.
- Ưu điểm: Tốt cho dữ liệu thấp chiều (dưới 20 chiều).
- Nhược điểm: Hiệu quả kém trong không gian cao chiều và không phù hợp với khoảng cách không phải Euclidean.

3.6.4. Approximate Nearest Neighbors (ANN)

- Mô tả: Tìm kiếm gần đúng các hàng xóm với tốc độ cao hơn.
- Ưu điểm: Tốc độ rất nhanh, phù hợp với các ứng dụng thời gian thực.

- Nhược điểm: Độ chính xác có thể thấp hơn phương pháp tìm kiếm chính xác.

B. Áp Dụng Trong Training Model

1. Cài đặt và trực quan hóa.

```
import pandas as pd
import matplotlib.pyplot as plt
from sklearn.decomposition import PCA
from sklearn.preprocessing import StandardScaler
from sklearn.neighbors import NearestNeighbors
from scipy.sparse import csr_matrix
```

khai báo thư viện và các công cụ cần dùng

```
# Assuming book_pivot is already created
# Fill NaN values with 0 (or you can use another strategy)
book_pivot_filled = book_pivot.fillna(0)

# Convert book_pivot to a sparse matrix
book_sparse = csr_matrix(book_pivot_filled)
```

fillna(0): Điền giá trị 0 vào những vị trí có giá trị thiếu (NaN).

csr_matrix(book_pivot_filled): Chuyển đổi ma trận book_pivot_filled thành một ma trận thưa theo định dạng CSR. Điều này giúp tiết kiệm bộ nhớ và tăng tốc độ xử lý, đặc biệt khi ma trận chứa nhiều giá trị 0.

```
# Train KNN
knn = NearestNeighbors(metric='cosine', algorithm='brute')
knn.fit(book_sparse)
```

Trong bài toán đề xuất sách, dữ liệu đầu vào thường là dữ liệu thưa (sparse data) như ma trận người dùng-sản phẩm, với nhiều giá trị bằng 0. Điều này làm cho các thuật toán dựa trên cây như KD-Tree hay

Ball Tree hoạt động kém hiệu quả, đặc biệt trong không gian có nhiều chiều và các điểm dữ liệu phân bố rải rác.

Phương pháp brute-force trở thành lựa chọn hợp lý trong trường hợp này, đặc biệt khi sử dụng khoảng cách Cosine để đo lường sự tương đồng giữa các vector thưa. Brute-force không bị hạn chế bởi loại khoảng cách sử dụng và đảm bảo tìm kiếm hàng xóm gần nhất một cách chính xác, bất kể số chiều của không gian dữ liệu.

Mặc dù phương pháp này có nhược điểm là tốc độ chậm và tiêu tốn nhiều tài nguyên tính toán, nhưng nó lại có ưu điểm vượt trội về độ chính xác, đặc biệt với dữ liệu thưa và khoảng cách không Euclidean. Điều này đảm bảo các khuyến nghị đưa ra là chính xác và nhất quán, điều rất quan trọng trong các ứng dụng như đề xuất sản phẩm, nơi mỗi khuyến nghị sai có thể ảnh hưởng đến trải nghiệm người dùng.

```python
# Sample movie index (replace with the actual index of the sample movie)
sample_movie_index = 241

# Find similar movies
distances, indices = knn.kneighbors(book_sparse[sample_movie_index], n_neighbors=10)
similar_movies_indices = indices.flatten()
```

Python

Giả sử ta chọn bộ sách có chỉ số 241 làm bộ sách mẫu, yêu cầu KNN tìm chỉ số của 10 bộ phim gần nhất với bộ phim mẫu.trong ma trận book_sparse

similar_movies_indices = indices.flatten() :

chuyển mảng similar_movies_indices từ mảng 2D thành mảng 1D

```python
# Standardize the data
scaler = StandardScaler()
book_pivot_scaled = scaler.fit_transform(book_pivot_filled)
```

Python

sử dụng lớp StandardScaler trong sklearn.preprocessing để chuẩn hóa dữ liệu.

Khi chuẩn hóa bằng StandardScaler, dữ liệu sẽ được biến đổi sao cho mỗi đặc trưng có trung bình (mean) bằng 0 và độ lệch chuẩn (standard deviation) bằng 1.

book_pivot_scaled là kết quả chuẩn hóa của ma trận book_pivot_filled. Trong ma trận này,mỗi đặc trưng đã được chuẩn hóa để có trung bình bằng 0 và độ lệch chuẩn bằng 1

```
# Perform PCA to reduce to 2 dimensions
pca = PCA(n_components=2)
book_pivot_pca = pca.fit_transform(book_pivot_scaled)

# Create a DataFrame for the PCA results
pca_df = pd.DataFrame(book_pivot_pca, index=book_pivot_filled.index, columns=['PC1', 'PC2'])
```

Thực hiện PCA để giảm xuống 2 chiều , trả về một ma trận mới book_pivot_pca với chỉ 2 cột tương ứng với 2 thành phần chính.

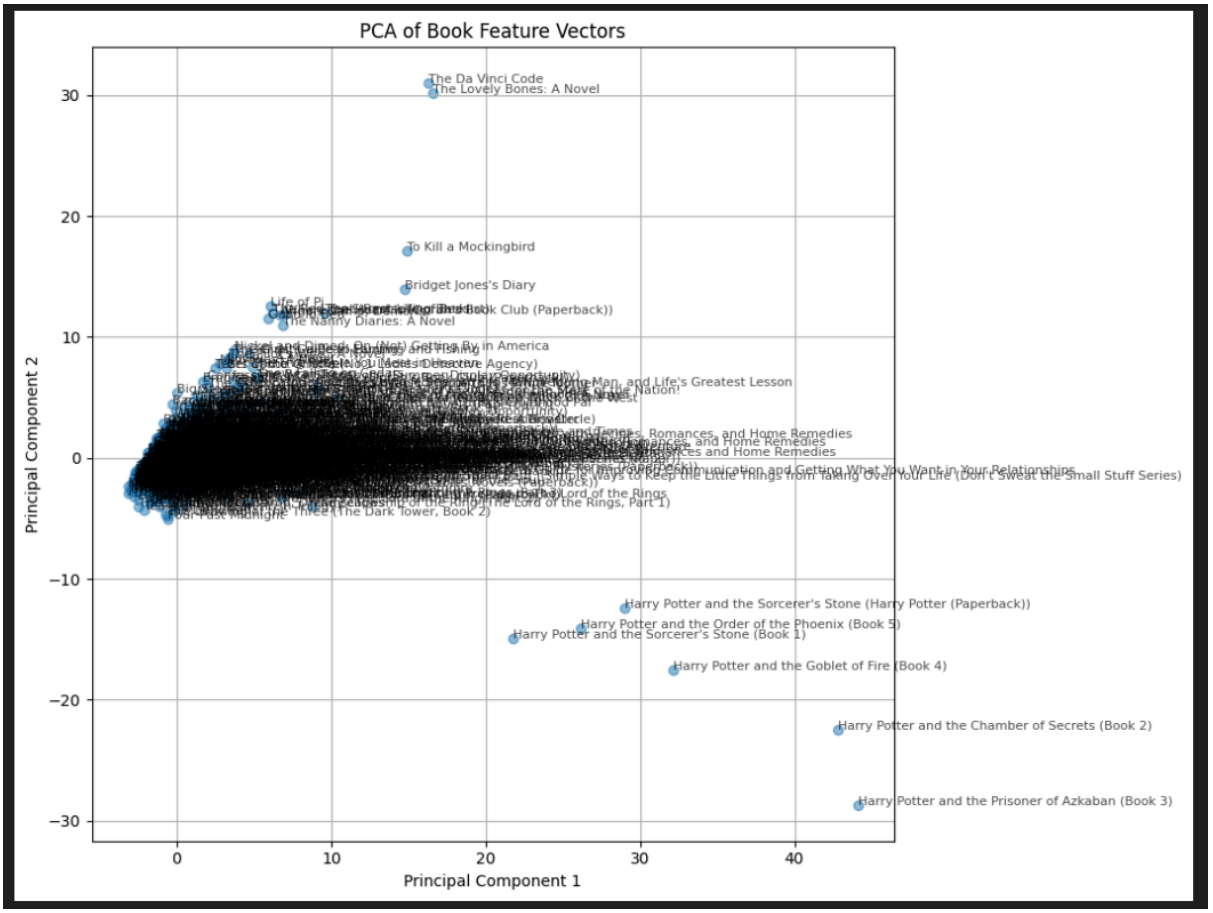
Tạo một DataFrame mới pca_df từ kết quả PCA (book_pivot_pca).

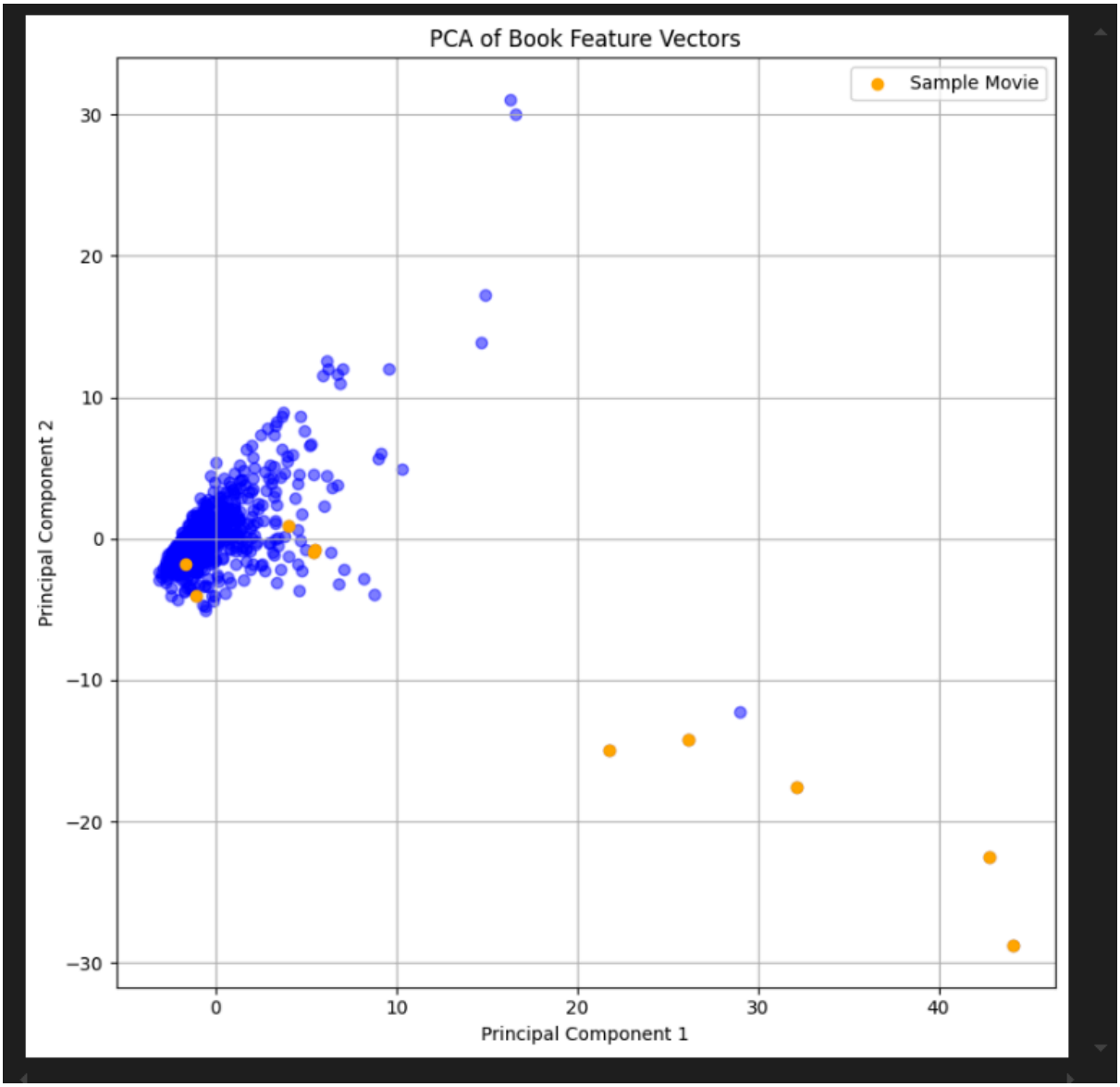
index=book_pivot_filled.index giúp giữ nguyên chỉ số gốc của dữ liệu ban đầu.

columns=['PC1', 'PC2'] đặt tên cho các cột là 'PC1' và 'PC2', đại diện cho hai thành phần chính đầu tiên.

```
# Plotting the PCA results
plt.figure(figsize=(8, 8))

# Plot all points in blue
plt.scatter(pca_df['PC1'], pca_df['PC2'], alpha=0.5, color='blue')

# Highlight the sample movie in orange
plt.scatter(pca_df['PC1'].iloc[sample_movie_index], pca_df['PC2'].iloc[sample_movie_index], color='orange', label='Sample Movie')

# Highlight the similar movies in orange
for idx in similar_movies_indices[1:]:  # Skip the first one as it is the sample movie itself
    plt.scatter(pca_df['PC1'].iloc[idx], pca_df['PC2'].iloc[idx], color='orange')

plt.xlabel('Principal Component 1')
plt.ylabel('Principal Component 2')
plt.title('PCA of Book Feature Vectors')
plt.grid(True)
plt.tight_layout()
plt.legend()
plt.show()
```

Tùy chỉnh để làm nổi bật sách và vẽ biểu đồ các điểm dữ liệu trong không gian 2 chiều .


PCA of Book Feature Vectors
Principal Component 2
Principal Component 1
The Da Vinci Code
The Lovely Bones: A Novel
To Kill a Mockingbird
Bridget Jones's Diary
The Nanny Diaries: A Novel
Harry Potter and the Sorcerer's Stone (Harry Potter (Paperback))
Harry Potter and the Order of the Phoenix (Book 5)
Harry Potter and the Sorcerer's Stone (Book 1)
Harry Potter and the Goblet of Fire (Book 4)
Harry Potter and the Chamber of Secrets (Book 2)
Harry Potter and the Prisoner of Azkaban (Book 3)

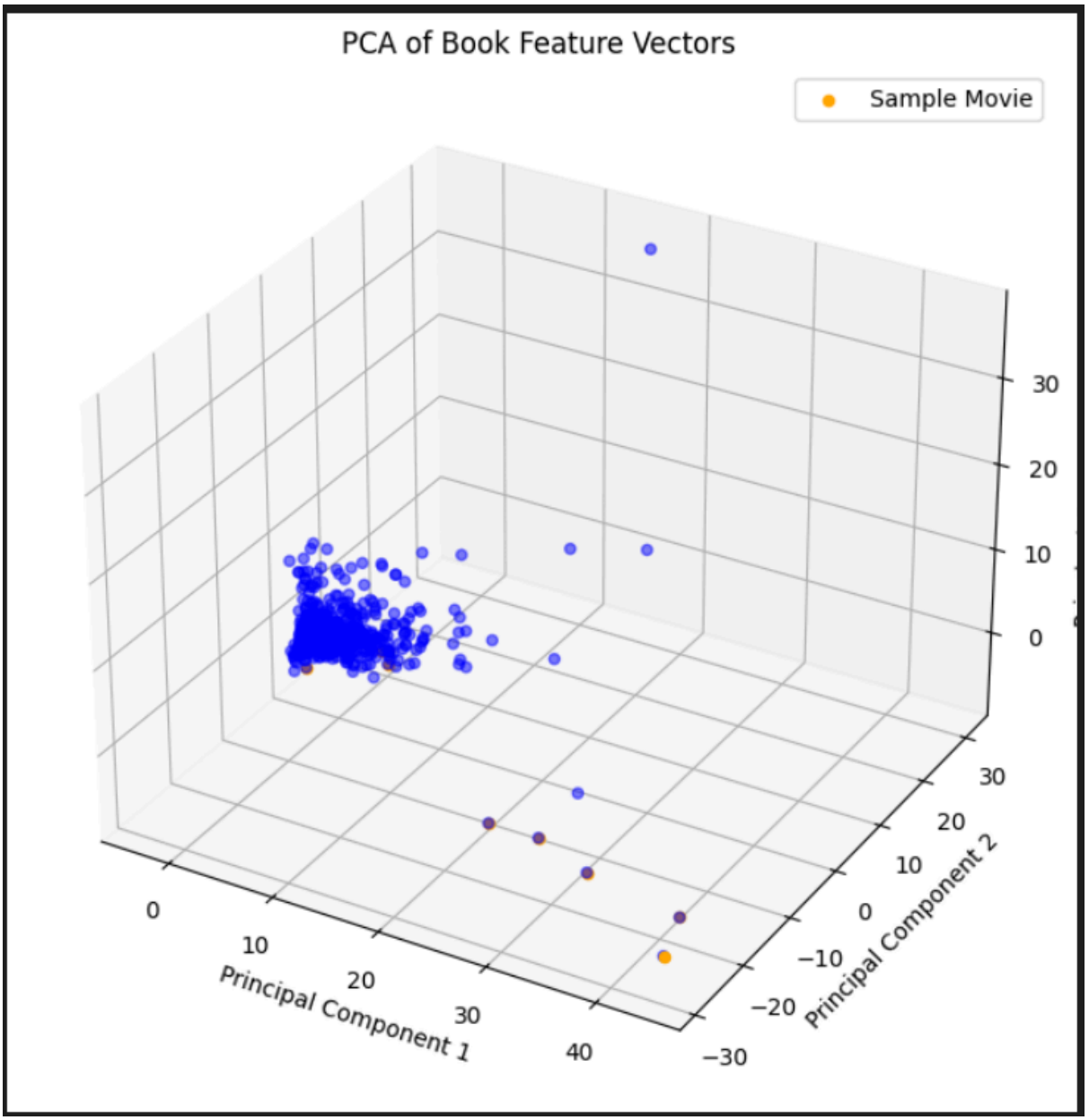


- Biểu đồ được hiển thị , những chấm màu cam là những bộ sách tương tự với Harry Potter “ ,do hiển thị biểu đồ trên không gian 2 chiều nên khoảng cách giữa các điểm màu cam nhìn sẽ xa nhau không được chính xác

```
# Perform PCA to reduce to 3 dimensions
pca = PCA(n_components=3)
book_pivot_pca = pca.fit_transform(book_pivot_scaled)

# Create a DataFrame for the PCA results
pca_df = pd.DataFrame(book_pivot_pca, index=book_pivot_filled.index, columns=['PC1', 'PC2', 'PC3'])
```

```python
# Plotting the PCA results in 3D
fig = plt.figure(figsize=(12, 8))
ax = fig.add_subplot(111, projection='3d')

# Plot all points in blue
ax.scatter(pca_df['PC1'], pca_df['PC2'], pca_df['PC3'], alpha=0.5, color='blue')

# Highlight the sample movie in orange
ax.scatter(pca_df['PC1'].iloc[sample_movie_index], pca_df['PC2'].iloc[sample_movie_index], pca_df['PC3'].iloc[sample_movie_index], color='orange',

# Highlight the similar movies in orange
for idx in similar_movies_indices[1:]:  # Skip the first one as it is the sample movie itself
    ax.scatter(pca_df['PC1'].iloc[idx], pca_df['PC2'].iloc[idx], pca_df['PC3'].iloc[idx], color='orange')

ax.set_xlabel('Principal Component 1')
ax.set_ylabel('Principal Component 2')
ax.set_zlabel('Principal Component 3')
ax.set_title('PCA of Book Feature Vectors')
ax.legend()
plt.show()
```

làm tương tự khi thực hiện PCA để giảm xuống 3 chiều



Dữ liệu trong không gian 3 chiều được hiển thị , những chấm màu cam là những bộ sách tương tự với “harry potter “

2. Kiểm tra gợi ý sách dựa trên mô hình K-Nearest Neighbors (KNN)

```
import numpy as np

def recommend_book_knn(book_id, k=6):
    # Ensure the model is defined and trained
    if 'knn' not in globals():
        raise NameError("The KNN model is not defined. Please define and train the model before calling this function.")

    # Find similar books
    distances, indices = knn.kneighbors(book_sparse[book_id], n_neighbors=k)

    book_name = book_pivot.index[book_id]
    similar_books = []
    for i in range(1, len(indices[0])):  # Start from 1 to exclude the book itself
        similar_book_index = indices[0][i]
        similar_book_name = book_pivot.index[similar_book_index]
        distance = np.linalg.norm(book_pivot.iloc[book_id] - book_pivot.iloc[similar_book_index])
        similar_books.append((similar_book_name, distance))
    total = 0
    print(f"Books similar to '{book_name}':")
    for book, distance in similar_books:
        total += distance
        print(f"{book} - distance {distance:.4f}")
    print(f"Average distance: {total / len(similar_books):.4f}")
```

```
book_id = 0  # Example book index for "Harry Potter and the Sorcerer's Stone (Book 1)"
recommend_book_knn(book_id)
```

1. Đầu tiên kiểm tra xem mô hình KNN có được định nghĩa trong phạm vi toàn cục (global) hay không. Nếu không, nó sẽ ném ra một ngoại lệ (exception) với thông báo yêu cầu định nghĩa và huấn luyện mô hình trước khi gọi hàm này.

```
if 'knn' not in globals():
    raise NameError("The KNN model is not defined. Please define and train the model before calling this function.")
```

2. Tìm kiếm các sách tương tự
   - **“knn.kneighbors”** tìm các sách tương tự cho sách có **“book_id”** dựa trên dữ liệu đã được biến đổi thành một không gian thưa (sparse space) **“book_sparse”**.
   - **“n_neighbors=k”** chỉ định số lượng sách tương tự mà bạn muốn tìm (bao gồm cả chính nó).
   - Kết quả trả về **“distances”** (khoảng cách đến các sách tương tự) và **“indices”** (vị trí của các sách tương tự trong dữ liệu).

```
# Find similar books
distances, indices = knn.kneighbors(book_sparse[book_id], n_neighbors=k)
```

3. Lấy tên sách và tạo danh sách các sách tương tự
   - **“book_name”** là tên của cuốn sách bạn muốn tìm các sách tương tự.

- **“similar_books”** là danh sách sẽ chứa tên và khoảng cách của các sách tương tự.
- Vòng lặp **“for”** bắt đầu từ 1 (bỏ qua chính sách hiện tại), lấy tên của các sách tương tự và tính toán khoảng cách giữa sách đó với sách hiện tại bằng cách sử dụng **“np.linalg.norm”**.

```
book_name = book_pivot.index[book_id]
similar_books = []
for i in range(1, len(indices[0])):  # Start from 1 to exclude the book itself
    similar_book_index = indices[0][i]
    similar_book_name = book_pivot.index[similar_book_index]
    distance = np.linalg.norm(book_pivot.iloc[book_id] - book_pivot.iloc[similar_book_index])
    similar_books.append((similar_book_name, distance))
```

4. In ra danh sách tương tự và khoảng cách
   - **“total”** dùng để tính tổng khoảng cách.
   - Hàm **“print”** in ra danh sách các sách tương tự và khoảng cách của chúng với sách hiện tại.
   - Cuối cùng, in ra khoảng cách trung bình giữa các sách tương tự.

```
total = 0
print(f"Books similar to '{book_name}':")
for book, distance in similar_books:
    total += distance
    print(f"{book} - distance {distance:.4f}")
print(f"Average distance: {total / len(similar_books):.4f}")
```

Sau đây là đoạn code thực tế sau khi sử dụng mô hình KNN để đề xuất các cuốn sách dựa trên cuốn sách đầu vào:

```
book_id = 0  # Example book index for "Harry Potter and the Sorcerer's Stone (Book 1)"
recommend_book_knn(book_id)
```

```
Books similar to '1984':
Animal Farm - distance 53.9722
The Catcher in the Rye - distance 60.3407
Lord of the Flies - distance 64.1483
The Handmaid's Tale - distance 63.8279
Slaughterhouse Five or the Children's Crusade: A Duty Dance With Death - distance 54.4243
Average distance: 59.3427
```

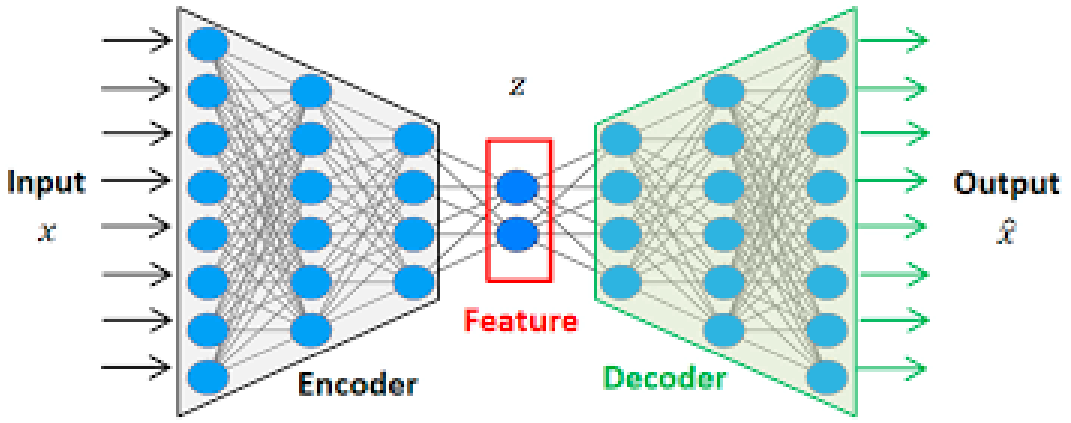
```
recommend_book_knn (3)
```

```
Books similar to '4 Blondes':
The House of the Spirits - distance 30.0666
Pleading Guilty - distance 23.0000
Seabiscuit - distance 35.6090
Bridget Jones: The Edge of Reason - distance 40.1995
Schindler's List - distance 32.7109
Average distance: 32.3172
```

IV. Sử dụng Autoencoder đề xuất sách tương tự.

1. Giới thiệu về Autoencoder

1.1 Khái niệm



Autoencoder là một loại mạng neural không giám sát được thiết kế để học một biểu diễn nén (encoding) của dữ liệu. Cấu trúc cơ bản của autoencoder bao gồm hai phần chính encoder và decoder hoạt động như sau:

- Input và Encoder: Dữ liệu đầu vào (input) được đưa vào mạng encoder. Nhiệm vụ của encoder là chuyển đổi dữ liệu này thành một biểu diễn tiềm ẩn (latent representation), thường có kích thước nhỏ hơn so với dữ liệu gốc. Encoder thực hiện việc này thông qua một loạt các lớp (layers), bao gồm các lớp kết nối đầy đủ (fully connected layers) hoặc các lớp convolutional (trong trường hợp làm việc với dữ liệu hình ảnh).
- Latent Space: Biểu diễn tiềm ẩn (latent space) là một không gian đặc trưng mà ở đó thông tin quan trọng nhất của dữ liệu đầu vào được giữ lại. Latent space có kích thước nhỏ hơn giúp giảm nhiễu và bỏ qua các thông tin không cần thiết trong dữ liệu gốc.
- Decoder: Sau khi dữ liệu được mã hóa vào latent space, nó sẽ được chuyển đến mạng decoder. Decoder sẽ giải mã (decode) biểu diễn tiềm ẩn này để tái tạo lại dữ liệu gốc. Mục tiêu của quá trình này là tái tạo lại dữ liệu sao cho giống với dữ liệu đầu vào ban đầu nhất có thể.

- Loss và cập nhật tham số: Để đánh giá mức độ tái tạo chính xác của autoencoder, một hàm loss (thường là Mean Squared Error - MSE) được sử dụng để so sánh dữ liệu đầu ra từ decoder với dữ liệu đầu vào ban đầu. Hàm loss này cho biết mức độ sai lệch giữa dữ liệu gốc và dữ liệu tái tạo. Các tham số của encoder và decoder sau đó sẽ được cập nhật thông qua quá trình backpropagation, nhằm giảm giá trị của hàm loss trong quá trình huấn luyện.
- Mục tiêu chính của autoencoder: Học cách nắm bắt các đặc trưng quan trọng nhất của dữ liệu đầu vào và bỏ qua các thông tin không cần thiết, từ đó có thể tái tạo lại dữ liệu với độ chính xác cao.

1.2 Ứng dụng

Autoencoder có nhiều ứng dụng đa dạng, bao gồm:

- Nén dữ liệu: Giảm kích thước dữ liệu để lưu trữ và truyền tải hiệu quả hơn
- Loại bỏ nhiễu: Tái tạo dữ liệu sạch từ dữ liệu nhiễu
- Tạo dữ liệu mới: Sinh ra các mẫu dữ liệu mới tương tự dữ liệu huấn luyện
- Học đặc trưng: Trích xuất các đặc trưng hữu ích cho các tác vụ học máy khác

2. Áp dụng Autoencoder vào bài toán đề xuất sách

- **Biểu diễn sách bằng Autoencoder**: Mỗi cuốn sách được biểu diễn bằng một vectơ đặc trưng (feature vector) dựa trên các đánh giá của người dùng. Autoencoder sẽ học cách nén vectơ đặc trưng này thành một biểu diễn tiềm ẩn (latent representation) có số chiều thấp hơn.
- **Lợi ích của việc sử dụng biểu diễn tiềm ẩn**:
    - **Giảm số chiều dữ liệu**: Biểu diễn tiềm ẩn giúp giảm số chiều của dữ liệu, làm đơn giản hóa quá trình tính toán và giảm thiểu nguy cơ overfitting.
    - **Nắm bắt các đặc trưng quan trọng**: Biểu diễn tiềm ẩn giúp nắm bắt các đặc trưng quan trọng nhất của sách, từ đó cải thiện khả năng đề xuất sách tương tự.

- **Gợi ý sách tương tự**: Sau khi sách được biểu diễn trong không gian tiềm ẩn với số chiều thấp, khoảng cách giữa các biểu diễn tiềm ẩn này được tính toán (thường sử dụng khoảng cách Euclid hoặc cosine). Các cuốn sách có khoảng cách gần nhau trong không gian tiềm ẩn sẽ có nội dung hoặc đặc trưng tương tự, và do đó có thể được đề xuất cho người dùng.

3. Cài đặt và giải thích mã

Cài đặt:

```
# Standardize the data
scaler = StandardScaler()
book_pivot_scaled = scaler.fit_transform(book_pivot)

# Split the data into training and testing sets
X_train, X_test = train_test_split(book_pivot_scaled, test_size=0.2, random_state=42)

# Define the neural network model
model_nn = Sequential([
    Input(shape=(book_pivot_scaled.shape[1],)),
    Dense(128, activation='relu'),
    Dense(64, activation='relu'),
    Dense(32, activation='relu'),
    Dense(64, activation='relu'),
    Dense(128, activation='relu'),
    Dense(book_pivot_scaled.shape[1], activation='linear')
])
```

Giải thích từng dòng code:

- StandardScaler: Chuẩn hóa dữ liệu để tất cả các đặc trưng có cùng một thang đo, giúp quá trình huấn luyện ổn định hơn.
- train_test_split: Chia dữ liệu thành tập huấn luyện và tập kiểm tra để đánh giá hiệu suất của mô hình.
- Sequential: Xây dựng mô hình mạng neural theo kiểu tuần tự.
- Input: Lớp đầu vào xác định hình dạng của dữ liệu đầu vào (số cột trong book_pivot_scaled).
- Dense: Các lớp fully connected với số lượng neuron và hàm kích hoạt tương ứng.
- 32, activation='relu': Lớp nút cổ chai (bottleneck layer) với 32 neuron và hàm kích hoạt ReLU, đây là phần quan trọng nhất của autoencoder, nơi dữ liệu được nén thành một biểu diễn tiềm ẩn.

- book_pivot_scaled.shape[1], activation='linear': Lớp đầu ra có cùng số chiều với dữ liệu đầu vào, sử dụng hàm kích hoạt tuyến tính để tái tạo lại dữ liệu gốc.

4. Huấn luyện và đánh giá mô hình

Cài đặt:

```
# Compile the model
model_nn.compile(optimizer='adam', loss='mse')

# Train the model
model_nn.fit(
    X_train, X_train, epochs=50,
    batch_size=32,
    validation_data=(X_test, X_test)
    )
```

Giải thích

1. model_nn.compile(optimizer='adam', loss='mse'):

- Biên dịch mô hình, chuẩn bị cho quá trình huấn luyện.
- optimizer='adam': Sử dụng thuật toán tối ưu Adam để cập nhật các trọng số của mô hình. Adam là một thuật toán tối ưu phổ biến, hiệu quả trong việc huấn luyện các mạng neural.
- loss='mse': Sử dụng hàm mất mát Mean Squared Error (MSE) để đo lường sự khác biệt giữa dữ liệu gốc (đầu vào) và dữ liệu tái tạo (đầu ra của mô hình). MSE tính toán bình phương trung bình của các sai số giữa các giá trị tương ứng trong hai ma trận.

2. model_nn.fit(X_train, X_train, epochs=50, batch_size=32, validation_data=(X_test, X_test)):

- Huấn luyện mô hình trên tập huấn luyện X_train.
- X_train được sử dụng làm cả đầu vào và đầu ra mong muốn, vì mục tiêu của autoencoder là tái tạo lại dữ liệu gốc.
- epochs=50: Huấn luyện mô hình trong 50 epochs (vòng lặp qua toàn bộ tập huấn luyện).
- batch_size=32: Chia tập huấn luyện thành các batch (nhóm) gồm 32 mẫu để cập nhật trọng số mô hình sau mỗi batch.
- validation_data=(X_test, X_test): Sử dụng tập kiểm tra X_test để đánh giá hiệu suất của mô hình sau mỗi epoch. Điều này giúp theo dõi xem mô hình có bị overfitting (học quá khớp với tập huấn luyện) hay không.

3. Kết luận:

- Quá trình huấn luyện sẽ lặp lại 50 lần (epochs). Trong mỗi epoch, mô hình sẽ xử lý từng batch dữ liệu huấn luyện, tính toán hàm mất mát MSE, và cập nhật các trọng số sử dụng thuật toán Adam để giảm thiểu mất mát.
- Sau mỗi epoch, mô hình sẽ được đánh giá trên tập kiểm tra để tính toán mất mát trên tập kiểm tra. Điều này giúp bạn theo dõi sự tiến triển của mô hình và phát hiện sớm overfitting.
- Bạn có thể sử dụng các kỹ thuật khác để đánh giá mô hình, chẳng hạn như trực quan hóa dữ liệu tái tạo và so sánh với dữ liệu gốc, hoặc sử dụng các số liệu đánh giá khác như độ chính xác tái tạo.
- Nếu mô hình bị overfitting, bạn có thể thử các kỹ thuật regularization (chẳng hạn như dropout) hoặc giảm số lượng epochs.

5. Đề xuất sách sử dụng biểu diễn tiềm ẩn từ Autoencoder

5.1. Lưu model

Sau khi huấn luyện mô hình autoencoder, chúng ta đã có thể trích xuất các biểu diễn tiềm ẩn (embeddings) của từng cuốn sách. Các biểu diễn tiềm ẩn này chứa đựng thông tin quan trọng về các đặc trưng của sách, được học từ dữ liệu đánh giá của người dùng. Chúng ta sẽ sử dụng các biểu diễn tiềm ẩn này để tìm kiếm các sách tương tự và đưa ra đề xuất. Việc lưu model và sử dụng thư viện có sẵn load_model giúp chúng ta sử dụng model và không cần training lại.

Cài đặt:

```python
model_nn.save('trained_model.keras')
```

Python

```python
from tensorflow.keras.models import load_model

# Load the model
loaded_model = load_model('trained_model.keras')
# Extract embeddings
# Use the layers up to the embedding layer
embedding_model = Sequential(loaded_model.layers[:-3])
book_embeddings = loaded_model.predict(book_pivot_scaled)
```

Python

```
24/24 [==============================] - 0s 3ms/step
```

Giải thích:

- load_model('trained_model.keras'): Tải mô hình đã được huấn luyện trước đó từ tệp trained_model.keras. Điều này giúp tiết kiệm thời gian vì chúng ta không cần phải huấn luyện lại mô hình.
- embedding_model = Sequential(loaded_model.layers[:-3]): Tạo một mô hình mới chỉ bao gồm các lớp từ đầu đến lớp nút cổ chai (bottleneck layer). Mô hình này sẽ được sử dụng để trích xuất các biểu diễn tiềm ẩn.
- book_embeddings = loaded_model.predict(book_pivot_scaled): Sử dụng mô hình embedding_model để dự đoán các biểu diễn tiềm ẩn cho tất cả các cuốn sách trong book_pivot_scaled.

5.2. Hàm find_similar_movies_nn

Hàm find_similar_movies_nn được sử dụng để tìm kiếm và đề xuất các cuốn sách tương tự dựa trên biểu diễn tiềm ẩn đã được trích xuất từ autoencoder. Chúng ta sẽ điểm qua các thành phần chính của hàm này để hiểu rõ hơn cách hoạt động của nó.

Cài đặt:

```
def find_similar_movies_nn(book_index, top_n=5):
    # Ensure the book index is valid
    if book_index < 0 or book_index >= len(book_pivot.index):
        raise ValueError(f"Book index {book_index} is out of range.")

    # Get the book name
    book_name = book_pivot.index[book_index]

    # Find similar books
    book_embedding = book_embeddings[book_index].reshape(1, -1)
    similarities = cosine_similarity(book_embedding, book_embeddings).flatten()
    similar_indices = similarities.argsort()[-top_n-1:-1][::-1]  # Exclude the book itself
    similar_books = []
    for similar_index in similar_indices:
        similar_book_name = book_pivot.index[similar_index]
        distance = np.linalg.norm(book_pivot.iloc[book_index] - book_pivot.iloc[similar_index])
        similar_books.append((similar_book_name, distance))
    total = 0
    # Print the similar books in the specified format
    print(f"Books similar to '{book_name}':")
    for book, distance in similar_books:
        total += distance
        print(f"{book} - distance {distance:.4f}")
    print(f"Average distance: {total/top_n:.4f}")
```

Giải thích:

Hàm find_similar_movies_nn sử dụng biểu diễn tiềm ẩn từ autoencoder và độ tương tự cosine để tìm kiếm và đề xuất top_n cuốn sách tương tự nhất cho một cuốn sách mục tiêu. Hàm này cũng tính toán và in ra khoảng cách trung bình giữa cuốn sách mục tiêu và các cuốn sách tương tự, cung cấp thêm thông tin về mức độ tương đồng giữa chúng.

Lưu ý: Mã nguồn đầy đủ của hàm này và các hàm khác sẽ được đính kèm ở cuối báo cáo để người đọc có thể tham khảo chi tiết.

6. So sánh kết quả và đánh giá mô hình

6.1 So sánh tổng quan về KNN và Autoencoder

a) Tổng quan

Trong dự án này, chúng ta đã khám phá hai phương pháp khác nhau để đề xuất sách: KNN (K-Nearest Neighbors) và Autoencoder. Mỗi phương pháp có những ưu điểm và nhược điểm riêng, ảnh hưởng đến hiệu suất và khả năng ứng dụng của chúng trong các tình huống khác nhau.

b) KNN

Ưu điểm:

- Đơn giản và dễ hiểu, dễ dàng triển khai và giải thích.
- Không cần huấn luyện mô hình phức tạp, chỉ cần tính toán khoảng cách giữa các điểm dữ liệu.
- Có thể hoạt động tốt khi dữ liệu có cấu trúc rõ ràng và các đặc trưng có ý nghĩa trực tiếp.

Nhược điểm:

- Hiệu suất có thể bị ảnh hưởng bởi sự lựa chọn số lượng láng giềng (K) và độ đo khoảng cách.
- Không hiệu quả khi số chiều dữ liệu lớn (vấn đề "lời nguyền số chiều").
- Không thể nắm bắt các mối quan hệ phức tạp và phi tuyến tính trong dữ liệu.

c) Autoencoder

Ưu điểm:

- Có khả năng học các biểu diễn tiềm ẩn nén, nắm bắt các đặc trưng trừu tượng và phức tạp của dữ liệu.
- Giảm số chiều dữ liệu, giúp cải thiện hiệu suất tính toán và giảm thiểu nguy cơ overfitting.
- Có thể hoạt động tốt với dữ liệu có cấu trúc phức tạp và các mối quan hệ phi tuyến tính.

Nhược điểm:

- Cần huấn luyện mô hình phức tạp, có thể tốn thời gian và tài nguyên tính toán.
- Việc giải thích biểu diễn tiềm ẩn có thể khó khăn.

- Hiệu suất có thể bị ảnh hưởng bởi sự lựa chọn kiến trúc mạng và các siêu tham số.

d) Phân tích so sánh:

KNN phù hợp hơn khi dữ liệu có cấu trúc đơn giản và các đặc trưng có ý nghĩa rõ ràng. Tuy nhiên, nó có thể gặp khó khăn khi số chiều dữ liệu lớn hoặc khi dữ liệu có các mối quan hệ phức tạp.

Autoencoder có khả năng học các biểu diễn phức tạp hơn của dữ liệu, nhưng đòi hỏi quá trình huấn luyện phức tạp hơn và có thể khó giải thích. Nó phù hợp hơn khi dữ liệu có cấu trúc phức tạp hoặc khi cần giảm số chiều dữ liệu.

Tiếp theo, chúng ta sẽ phân tích cụ thể hơn về hiệu suất của hai mô hình này dựa trên kết quả thử nghiệm trên tập dữ liệu ở mục 6.2.

6.2. Ví dụ

Đây là kết quả của ví dụ gợi ý quyển sách tương với quyển ‘1984’ có index = 0.

```
[144]
#recommend_book
recommend_book_knn(0)
```

```
Books similar to '1984':
Animal Farm - distance 53.9722
The Catcher in the Rye - distance 60.3407
Lord of the Flies - distance 64.1483
The Handmaid's Tale - distance 63.8279
Slaughterhouse Five or the Children's Crusade: A Duty Dance With Death - distance 54.4243
Average distance: 59.3427
```

```
[145]
find_similar_movies_nn(0)
```

```
Books similar to '1984':
Foucault's Pendulum - distance 49.1019
Lord of the Flies - distance 64.1483
Shopaholic Takes Manhattan (Summer Display Opportunity) - distance 60.8687
The Great Gatsby - distance 55.9911
Waiting to Exhale - distance 49.1019
Average distance: 55.8424
```

a) Phân tích ví dụ

Từ kết quả thử nghiệm trên cuốn sách "1984" với index 0, chúng ta có thể thấy rõ sự khác biệt trong danh sách sách đề xuất giữa hai mô hình KNN và Autoencoder:

- **KNN:** Đề xuất các cuốn sách như "Animal Farm", "The Catcher in the Rye", "The Handmaid's Tale", và "Slaughterhouse-Five". Average distance là 59.3427.
- **Autoencoder:** Đề xuất các cuốn sách như "Foucault's Pendulum", "Shopaholic Takes Manhattan", "The Great Gatsby", và "Waiting to Exhale". Average distance là 55.8424.

b) Đánh giá dựa trên Average distance:

- Average distance thể hiện khoảng cách trung bình giữa cuốn sách mục tiêu và các cuốn sách được đề xuất. Khoảng cách càng nhỏ, các đề xuất càng được coi là "gần" hoặc "tương tự" hơn với cuốn sách mục tiêu.
- Trong trường hợp này, Autoencoder có Average distance thấp hơn KNN (55.8424 so với 59.3427). Điều này cho thấy, ít nhất là trong ví dụ này, Autoencoder có thể đang tạo ra các đề xuất có liên quan hơn.

c) Giải thích sự khác biệt:

- KNN dựa trên sự tương đồng trực tiếp giữa các vectơ đặc trưng của sách.
- Autoencoder học một biểu diễn tiềm ẩn nén, có thể nắm bắt các đặc trưng phức tạp và trừu tượng hơn của sách mà KNN có thể bỏ lỡ. Điều này có thể dẫn đến các đề xuất khác biệt và tiềm năng là tốt hơn.

Tuy nhiên, cần lưu ý:

- Đây chỉ là một ví dụ duy nhất. Để đưa ra kết luận chắc chắn về mô hình nào tốt hơn, cần đánh giá trên nhiều cuốn sách khác nhau và sử dụng các số liệu đánh giá khác nhau (chẳng hạn như precision, recall, NDCG, v.v.).
- Việc một mô hình có Average distance thấp hơn không đảm bảo rằng nó luôn tạo ra đề xuất tốt hơn trong mọi trường hợp. Chất lượng đề xuất còn phụ thuộc vào nhiều yếu tố khác, bao gồm cả dữ liệu đầu vào, cách người dùng đánh giá sự tương đồng giữa các sách, và mục tiêu cụ thể của hệ thống đề xuất.

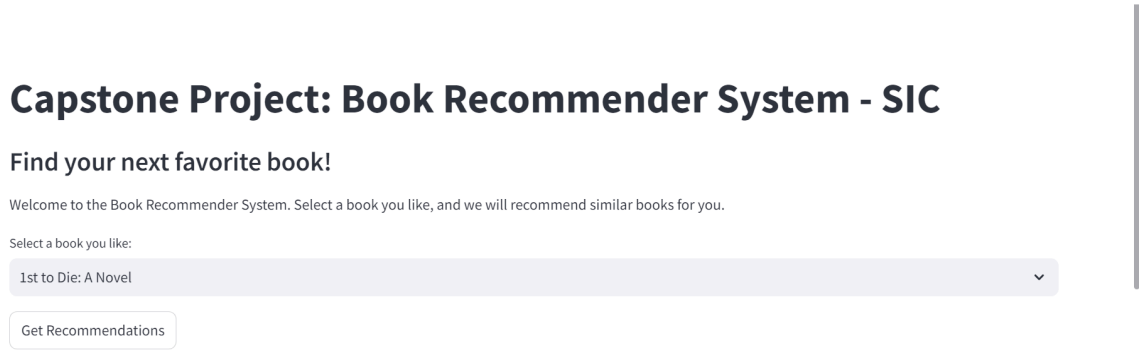
d) Tóm lại:

- Trong ví dụ này, Autoencoder cho thấy tiềm năng tạo ra các đề xuất có liên quan hơn so với KNN, dựa trên Average distance thấp hơn.
- Tuy nhiên, cần đánh giá thêm để xác định mô hình nào thực sự hiệu quả hơn trong việc đáp ứng nhu cầu và sở thích của người dùng.

V. Trang web demo.

Link dự án và video demo: Tại đây.

1. Trang chủ tìm kiếm



2. Đề xuất tìm kiếm

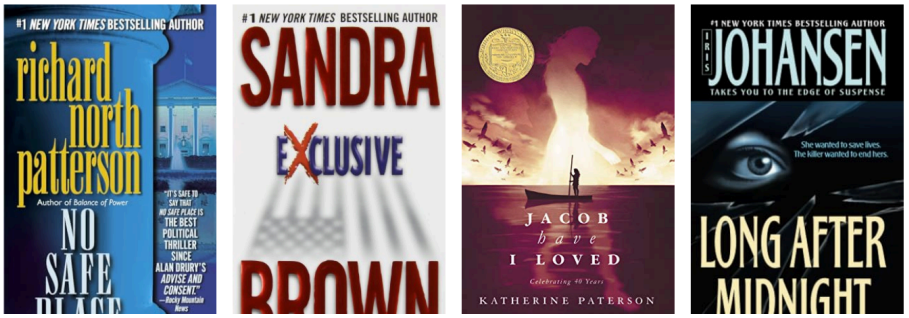
3. Hiện các đầu sách tương tự.

Select a book you like:

A Civil Action

Get Recommendations

No Safe Place | Exclusive | Jacob Have I Loved | Long After Midnight | The Cradle Will Fall



4. Lịch sử tìm kiếm.

Previously Recommended Books

4 Blondes: 4 Blondes, No Safe Place, Pleading Guilty, Long After Midnight, Exclusive, Lake Wobegon days
A Case of Need: A Case of Need, Exclusive, Jacob Have I Loved, Pleading Guilty, No Safe Place, The Cradle Will Fall
4 Blondes: 4 Blondes, No Safe Place, Pleading Guilty, Long After Midnight, Exclusive, Lake Wobegon days
A Bend in the Road: A Bend in the Road, Exclusive, The Cradle Will Fall, No Safe Place, Family Album, Lake Wobegon days
A Heartbreaking Work of Staggering Genius: A Heartbreaking Work of Staggering Genius, No Safe Place, A Civil Action, Long After Midni
A Fine Balance: A Fine Balance, Exclusive, No Safe Place, Long After Midnight, Deck the Halls (Holiday Classics), The Cradle Will Fal
A Case of Need: A Case of Need, Exclusive, Jacob Have I Loved, Pleading Guilty, No Safe Place, The Cradle Will Fall
A Walk to Remember: A Walk to Remember, The Rescue, The Killing Game: Only One Can Win...and the Loser Dies, The Cradle Will Fall, Gr
A Civil Action: A Civil Action, No Safe Place, Exclusive, Jacob Have I Loved, Long After Midnight, The Cradle Will Fall