



T. C.
ERCIYES ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
BİLGİSAYAR MÜHENDİSLİĞİ BÖLÜMÜ

DESIGN PROJECT DERSİ
FİNAL RAPORU

“Book
Recommendation
System

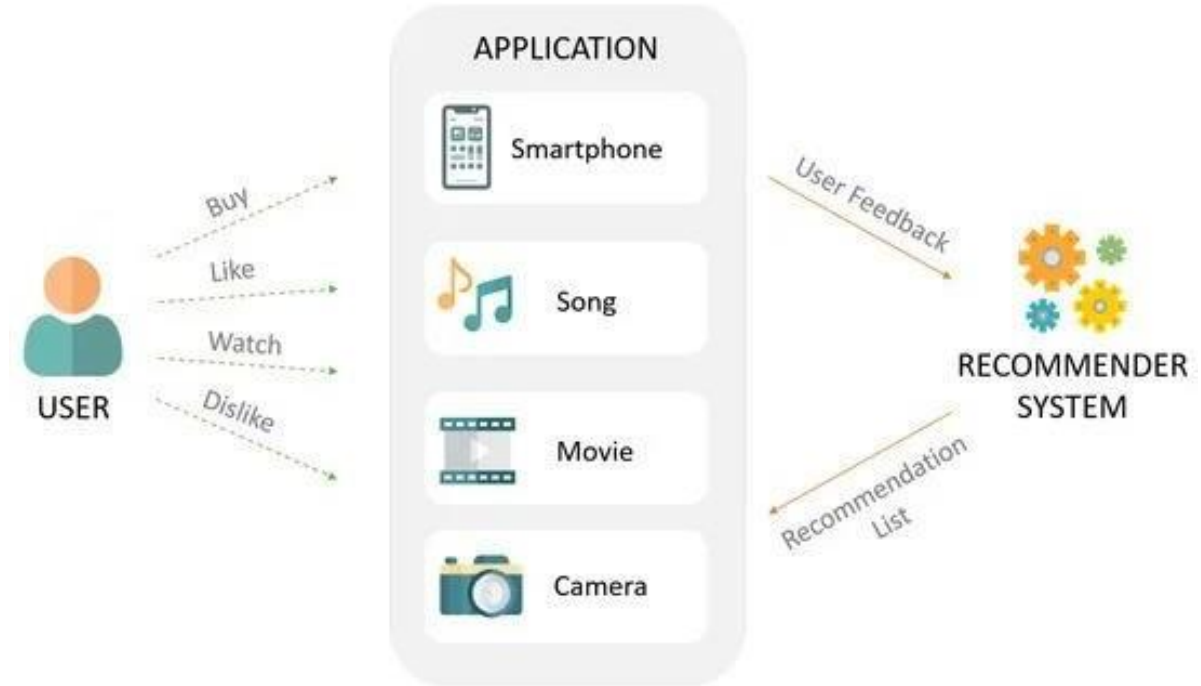
”PROJE ID:
24221702

HAZIRLAYAN:
1030520856 KADİR AYDIN
1030520480 HAKKI CAN KAFALI

HAZİRAN
2024

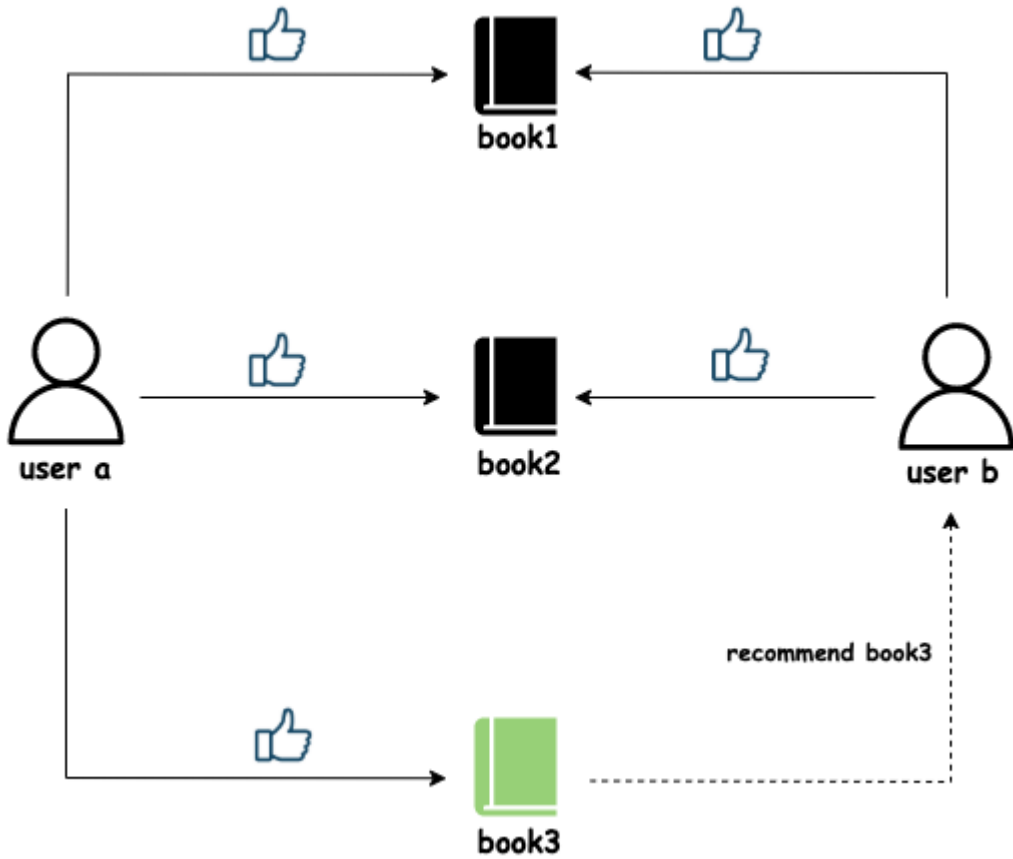
Recommendation System (Öneri Sistemi) Nedir?

Bir öneri sistemi, kullanıcıların beğendiği öğeleri ve diğer öğeleri inceleyerek kullanıcılara benzer öğeleri önerebilen bir yazılımdır. Bu sistemler genellikle bir veri tabanına dayalıdır ve kullanıcıların beğendiği öğeleri ve diğer öğeleri bir vektör olarak gösterir. Daha sonra cosine similarity gibi bir yöntemle bu vektörler arasındaki benzerliği ölçer ve kullanıcılara benzer öğeleri önerebilir. Örneğin, bir kitap öneri sistemi kullanıcıların beğendiği kitapları ve diğer kitapları inceleyerek benzer kitapları önerebilir. Benzer şekilde, bir film öneri sistemi de kullanıcıların beğendiği filmleri ve diğer filmleri inceleyerek benzer filmleri önerebilir. Bu sistemler, kullanıcıların beğendiği öğeleri ve diğer öğeleri inceleyerek kullanıcıların zevklerini anlayarak daha doğru önerilerde bulunabilir.



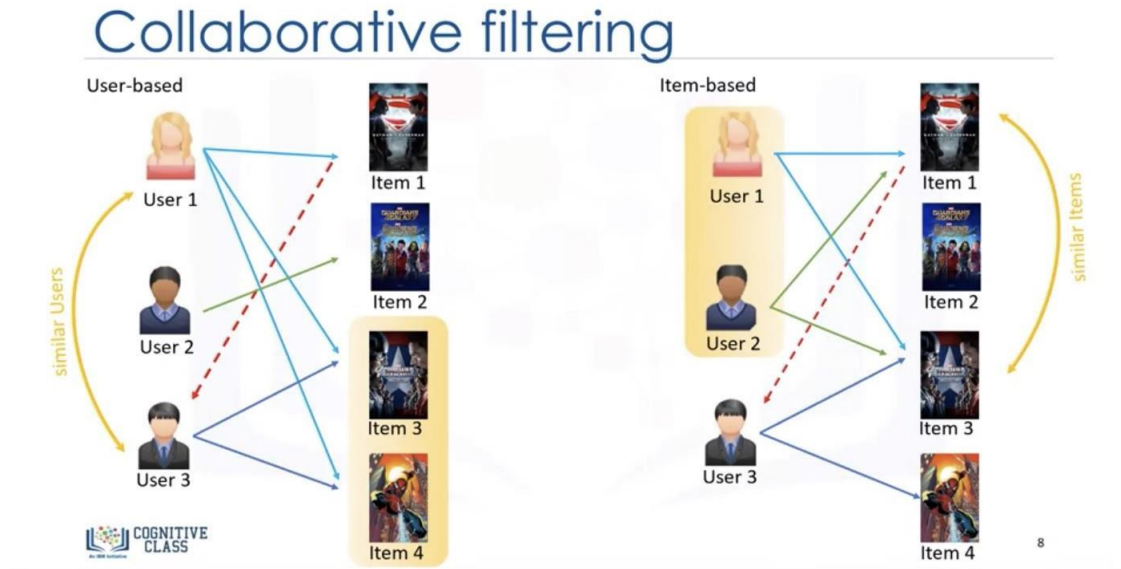
Kitap Öneri Sistemi Nedir?

Bir kitap öneri sistemi, kullanıcıların beğendiği kitaplar ve diğer kitaplar arasındaki benzerliği ölçerek, kullanıcılara benzer filmleri önerebilen bir yazılımdır. Bu sistemler genellikle bir veri tabanına dayalıdır ve kullanıcıların beğendiği kitapları ve diğer kitapları bir vektör olarak gösterir. Daha sonra cosine similarity gibi bir yöntemle bu vektörler arasındaki benzerliği ölçer ve kullanıcılara benzer kitaplar önerebilir. Bu sistemler, kullanıcıların beğendiği kitapları ve diğer kitapları inceleyerek kullanıcıların zevklerini anlayarak daha doğru önerilerde bulunabilir.



Collaborative Filtering Nedir?

Collaborative filtering, veri madenciliği ve makine öğrenimi alanlarında kullanılan bir yöntemdir. Bu yöntem, bir kullanıcının bir ürünü seçmesi veya beğenmesi gibi verilerden yola çıkarak benzer zevklere sahip diğer kullanıcıların seçim ve beğenilerini tahmin etmeye çalışır. Bu sayede, bir kullanıcının diğer kullanıcıların seçim ve beğenilerini dikkate alarak kendi seçimlerini yapmasına yardımcı olunur.



Collaborative filtering yöntemi, genellikle öneri sistemleri konusunda kullanılır. Örneğin, bir online alışveriş sitelerinde bir kullanıcının alışveriş geçmişi ve beğenileri dikkate alınarak kendisine özel öneriler sunulmasında bu yöntem kullanılır. Böylece, kullanıcıların daha kolay ve doğru ürünleri bulmaları ve satın almaları sağlanır.

Collaborative filtering yöntemi, genellikle iki farklı yöntemle uygulanır. Bunlar, "user-based collaborative filtering" ve "item-based collaborative filtering" olarak adlandırılır. User-based collaborative filtering yönteminde, bir kullanıcının benzer zevklere sahip diğer kullanıcıları bulunur ve bu kullanıcıların seçim ve beğenileri dikkate alınarak öneriler sunulur. Item-based collaborative filtering yönteminde ise, bir ürünü beğenen kullanıcıların benzer ürünleri beğendiği düşünülerek öneriler sunulur.

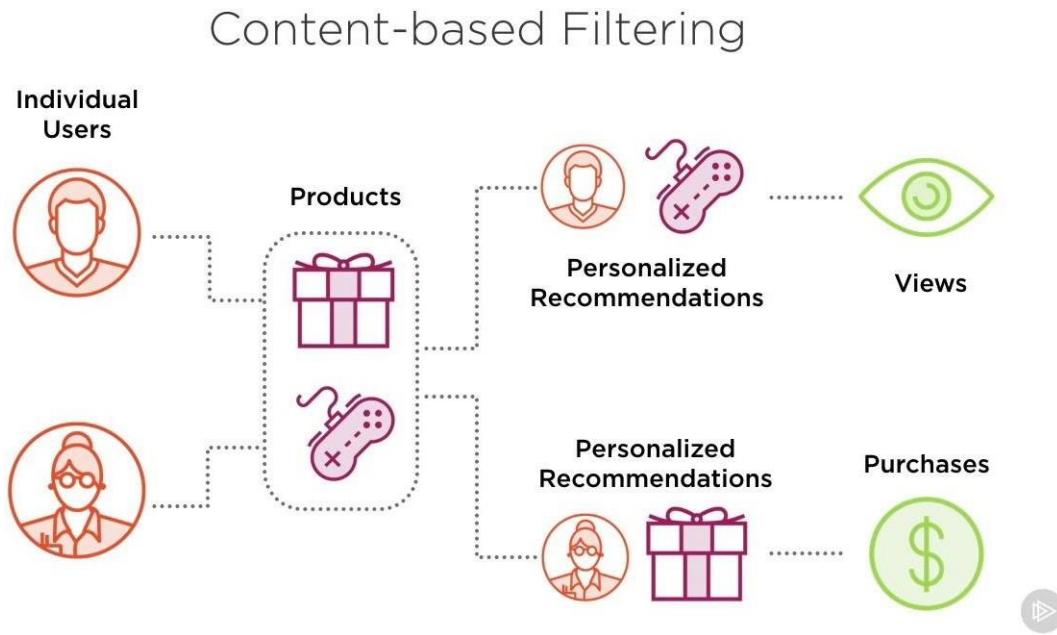
Collaborative filtering yöntemini uygularken, öncelikle veri kümesinde bulunan kullanıcıların seçim ve beğenilerini incelenir ve bu veriler matris halinde temsil edilir. Daha sonra, bu matriste bulunan elemanların boş olmayan kısımları

dikkate alınarak kullanıcılar ve ürünler arasındaki benzerlikler hesaplanır. Bu benzerlikler, genellikle Pearson korelasyon katsayısı veya cosine benzerliği gibi metriklerle ölçülür. Son olarak, bu benzerlikler dikkate alınarak öneriler sunulur.

Content Based Filtering Nedir?

Content based filtering, veri madenciliği ve makine öğrenimi alanlarında kullanılan bir yöntemdir. Bu yöntem, bir ürünün kendine özgü özelliklerini dikkate alarak benzer ürünleri tahmin etmeye çalışır. Örneğin, bir kitabı okuyan bir kullanıcının beğendiği bir başka kitabı tahmin etmek için, okuduğu kitabın yazarı, türü veya konusu gibi özellikleri dikkate alınabilir.

Content based filtering yöntemi, genellikle öneri sistemleri konusunda kullanılır. Örneğin, bir online alışveriş sitelerinde bir kullanıcının alışveriş geçmişi ve beğenileri dikkate alınarak benzer ürünleri önerebilir. Böylece, kullanıcıların daha kolay ve doğru ürünleri bulmaları ve satın almaları sağlanır.

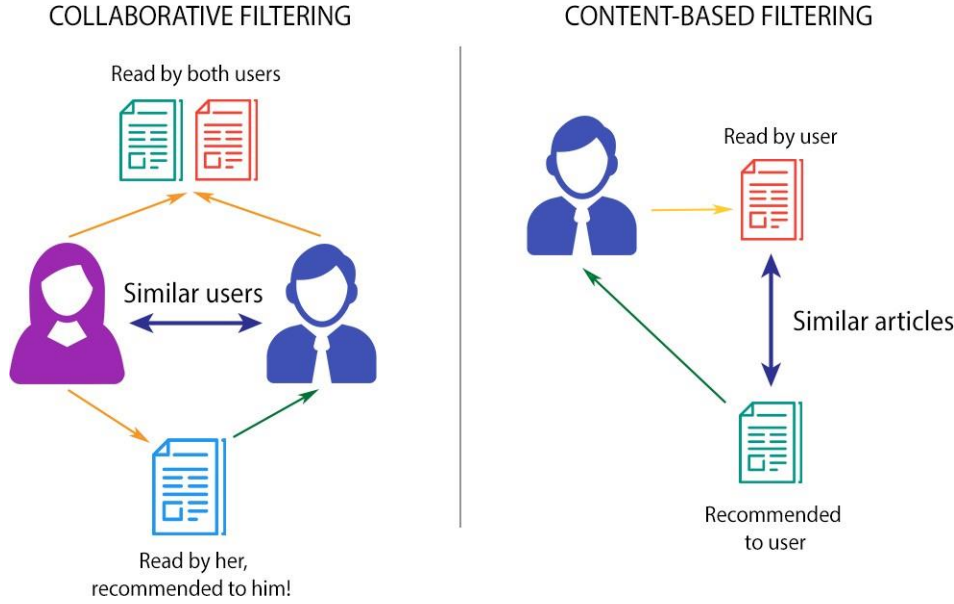


Content based filtering yöntemini uygularken, öncelikle ürünlerin özellikleri belirlenir ve bu özellikler matris halinde temsil edilir. Bu matriste, her satır bir ürünü, her sütun ise bir özelliği temsil eder. Daha sonra, bu matriste bulunan elemanların boş olmayan kısımları dikkate alınarak ürünler arasındaki benzerlikler hesaplanır. Bu benzerlikler, genellikle cosine benzerliği veya Jaccard benzerliği gibi metriklerle ölçülür. Son olarak, bu benzerlikler dikkate alınarak öneriler sunulur.

Content based filtering yönteminde, ürünlerin özellikleri belirlenirken oldukça önemlidir. Bu özellikler, ürünlerin niteliklerini veya özelliklerini ifade eden kelimeler veya etiketler olarak belirlenir. Örneğin, bir kitap için "macera",

"romantik", "bilim kurgu" gibi kelimeler ürünün özellikleri olarak belirlenebilir. Bu özellikler, genellikle ürünlerin metin içeriği ya da kullanıcıların verdiği etiketler gibi verilerden çıkarılır.

KISACA FİLTRELEME YÖNTEMLERİNİN FARKLARI



PROJE KONUMUZ

Kaggle'dan alınan veri setindeki verileri kullanarak, kullanıcıya kitap ismi soran ve verdiği kitap ismine göre Collaborative filtering yöntemi kullanarak kitap ismi öneren bir sistemdir. Ayrıca veri setine göre en popüler 50 kitabı listeleyen bir sistemdir.

VERİLER

Kaggle verimizin linki : <https://www.kaggle.com/datasets/arashnic/book-recommendation-dataset>

Kullandığımız verilerden biri Books.csv adlı veri setinden sağlanmıştır. Veri setinde 271360 farklı satır içerir. Tablo sütunları ISBN, Book-Title, Book-Author, Year-Of-Publication, Publisher, Image-URL-S, Image-URL-M, Image-URL-L. Toplamda 8 sütun bulunmaktadır.

Kullandığımız verilerden biri Users.csv adlı veri setinden sağlanmıştır. Veri setinde 278858 farklı satır içerir. Tablo sütunları User-ID, Location, Age Toplamda 3 sütun bulunmaktadır.

Kullandığımız verilerden biri Ratings.csv adlı veri setinden sağlanmıştır. Veri setinde 1149780 farklı satır içerir. Tablo sütunları User-ID, ISBN, Book-Rating Toplamda 3 sütün bulunmaktadır.

PROJENİN AMACI

Bu projenin amacı, bir kullanıcın kitap okumak istediğinde Collaborative filtering yöntemiyle filtreleme yapan, ilgi alanına göre mümkün olan en yüksek doğru sınıflandırma yüzdesine sahip kitaplar önermektir.

PROJE KODLAR VE AÇIKLAMALARI

Projemizde Python programlama dilini kullandık.

```
import numpy as np
import pandas as pd
```

Bu kodlar, numpy ve pandas kütüphanelerini projemize dahil ediyor.

numpy: Sayısal hesaplamalar için kullanılan bir kütüphanedir. Genellikle çok boyutlu diziler ve matrisler üzerinde işlemler yapmak için kullanılır.

pandas: Veri manipülasyonu ve veri analizi için kullanılan bir kütüphanedir. Verileri okumak, yazmak, filtrelemek, gruplamak ve daha birçok işlemi kolayca yapmamızı sağlar.

```
books = pd.read_csv('Books.csv')
users = pd.read_csv('Users.csv')
ratings = pd.read_csv('Ratings.csv')
```

Bu kod parçacığı, CSV dosyalarını okuyarak veri setlerini yüklüyor.

pd.read_csv('Books.csv'): 'Books.csv' dosyasını okuyarak, kitap verilerini içeren bir DataFrame'e (books) dönüştürüyor.

pd.read_csv('Users.csv'): 'Users.csv' dosyasını okuyarak, kullanıcı verilerini içeren bir DataFrame'e (users) dönüştürüyor.

pd.read_csv('Ratings.csv'): 'Ratings.csv' dosyasını okuyarak, kullanıcıların kitaplara verdiği puanları içeren bir DataFrame'e (ratings) dönüştürüyor.

```
books.head()
```

Bu kod, books veri çerçevesinin ilk birkaç satırını görüntüler. Bu, veri setinin nasıl yapılandırıldığını ve hangi tür bilgilerin bulunduğunu anlamamıza yardımcı olur.

	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...

```
users.head()
```

Bu kod, users veri çerçevesinin ilk birkaç satırını görüntüler. Bu, veri setinin nasıl yapılandırıldığını ve hangi tür bilgilerin bulunduğunu anlamamıza yardımcı olur.

	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

```
ratings.head()
```

Bu kod, ratings veri çerçevesinin ilk birkaç satırını görüntüler. Bu, veri setinin nasıl yapılandırıldığını ve hangi tür bilgilerin bulunduğunu anlamamıza yardımcı olur.

	User-ID	ISBN	Book-Rating
0	276725	034545104X	0
1	276726	0155061224	5
2	276727	0446520802	0
3	276729	052165615X	3
4	276729	0521795028	6

```
print(books.shape)
print(ratings.shape)
print(users.shape)
```

Bu kod parçasığı, her bir veri çerçevesinin boyutunu (satır ve sütun sayısını) görüntüler.

Bu bilgiler, veri setlerinin ne kadar büyük olduğunu ve hangi tür işlemlerin yapılacağını anlamamıza yardımcı olur.

```
(271360, 8)
(1149780, 3)
(278858, 3)
```

```
books.isnull().sum()
```

Bu kod, books veri çerçevesindeki her sütunda eksik değerlerin sayısını hesaplar.

isnull() yöntemi, her hücrenin boş olup olmadığını kontrol eder ve sonuç olarak bir boolean DataFrame döndürür. Ardından, sum() yöntemi bu boolean değerleri toplar, True değerleri 1 olarak ve False değerleri 0 olarak kabul ederek eksik değerlerin toplam sayısını verir.

Bu, veri setindeki her sütunda kaç tane eksik değer olduğunu görmemizi sağlar. Eksik değerler, veri temizliği ve analizinde dikkate alınması gereken önemli bir konudur. Eğer eksik değerler varsa, bunlarla nasıl başa çıkılacağını belirlemek önemlidir.

```
ISBN                0
Book-Title          0
Book-Author        1
Year-Of-Publication 0
Publisher           2
Image-URL-S         0
Image-URL-M         0
Image-URL-L         3
dtype: int64
```

```
users.isnull().sum()
```

```
User-ID            0
Location           0
Age               110762
dtype: int64
```

```
ratings.isnull().sum()
```

```
User-ID      0
ISBN         0
Book-Rating  0
dtype: int64
```

```
books.duplicated().sum()
```

Bu kod, books veri çerçevesindeki yinelenen satırların sayısını hesaplar.

duplicated() yöntemi, her satırın bir önceki satırla aynı olup olmadığını kontrol eder ve sonuç olarak bir boolean Serisi döndürür. sum() yöntemi bu boolean değerleri toplar, yinelenen satırların toplam sayısını verir.

Bu, veri setinde kaç tane yinelenen satır olduğunu görmemizi sağlar.

```
0
```

```
ratings.duplicated().sum()
```

```
0
```

```
users.duplicated().sum()
```

```
0
```

Popularity Based System (Popüler 50 kitabı listeleyen kodlar)

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

Bu kod, ratings ve books veri çerçevelerini 'ISBN' sütunu üzerinde birleştirerek yeni bir veri çerçevesi oluşturuyor. Bu yeni veri çerçevesi, kitap derecelendirmeleriyle ilgili her bir girişi, ilgili kitabın bilgileriyle birleştirir.

merge() fonksiyonu, belirli bir sütuna göre iki veri çerçevesini birleştirir. Burada, 'ISBN' sütununa göre birleştirme yapıyoruz. Bu, her bir derecelendirme girişinin, ilgili kitabın bilgileriyle birleştirileceği anlamına gelir.

Sonuç olarak, ratings_with_name adında bir yeni veri çerçevesi elde ederiz, bu çerçeve her bir derecelendirme girişini, ilgili kitabın bilgileriyle birleştirir. Bu, kitap tavsiye sistemi oluştururken kitapların özelliklerini kullanmamıza olanak sağlar.

```
num_rating_df = ratings_with_name.groupby('Book-Title').count()['Book-Rating'].reset_index()
num_rating_df.rename(columns= {'Book-Rating': 'num_ratings'}, inplace=True)
num_rating_df
```

Bu kod, her bir kitaba yapılan derecelendirmelerin sayısını içeren bir DataFrame oluşturuyor.

İlk olarak, ratings_with_name veri çerçevesi üzerinde 'Book-Title' sütunuyla grupta yapılıyor ve her bir grup için 'Book-Rating' sütunundaki değerlerin sayısı hesaplanır. Böylece, her bir kitaba yapılan toplam derecelendirme sayısını içeren bir Seri elde edilir.

Daha sonra, bu Seri, 'num_ratings' olarak adlandırılan yeni bir sütunla birlikte bir DataFrame'e dönüştürülür. Bu işlem reset_index() fonksiyonu ile yapılır.

Son olarak, 'Book-Rating' sütununun adı 'num_ratings' olarak değiştirilir.

Bu şekilde, her bir kitaba yapılan toplam derecelendirme sayısını içeren bir DataFrame elde edilir.

	Book-Title	num_ratings
0	A Light in the Storm: The Civil War Diary of ...	4
1	Always Have Popsicles	1
2	Apple Magic (The Collector's series)	1
3	Ask Lily (Young Women of Faith: Lily Series, ...	1
4	Beyond IBM: Leadership Marketing and Finance ...	1
...
241066	Ä?Ä?Ipiraten.	2
241067	Ä?Ä?rger mit Produkt X. Roman.	4
241068	Ä?Ä?sterlich leben.	1
241069	Ä?Ä?stlich der Berge.	3
241070	Ä?Ä?thique en toc	2

241071 rows × 2 columns

```
avg_rating_df = ratings_with_name.groupby('Book-Title').mean()['Book-Rating'].reset_index()
avg_rating_df.rename(columns= {'Book-Rating': 'avg_rating'}, inplace=True)
avg_rating_df
```

Bu kod, her bir kitabın ortalama derecelendirmesini içeren bir DataFrame oluşturuyor.

İlk olarak, ratings_with_name veri çerçevesi üzerinde 'Book-Title' sütunuyla grupta yapılar ve her bir grup için 'Book-Rating' sütunundaki değerlerin ortalaması hesaplanır. Böylece, her bir kitaba ait ortalama derecelendirme elde edilir.

Daha sonra, bu ortalama derecelendirme değerleri, 'avg_rating' olarak adlandırılan yeni bir sütunla birlikte bir DataFrame'e dönüştürülür. Bu işlem reset_index() fonksiyonu ile yapılır.

Son olarak, 'Book-Rating' sütununun adı 'avg_rating' olarak değiştirilir.

Bu şekilde, her bir kitabın ortalama derecelendirmesini içeren bir DataFrame elde edilir.

	Book-Title	avg_rating
0	A Light in the Storm: The Civil War Diary of ...	2.250000
1	Always Have Popsicles	0.000000
2	Apple Magic (The Collector's series)	0.000000
3	Ask Lily (Young Women of Faith: Lily Series, ...	8.000000
4	Beyond IBM: Leadership Marketing and Finance ...	0.000000
...
241066	Ä?Ä?lpiraten.	0.000000
241067	Ä?Ä?rger mit Produkt X. Roman.	5.250000
241068	Ä?Ä?sterlich leben.	7.000000
241069	Ä?Ä?stlich der Berge.	2.666667
241070	Ä?Ä?thique en toc	4.000000

241071 rows × 2 columns

```
popular_df = num_rating_df.merge(avg_rating_df, on = 'Book-Title')
popular_df
```

Bu kod, her bir kitabın toplam derecelendirme sayısını ve ortalama derecelendirmesini içeren iki DataFrame'i birleştirerek bir DataFrame oluşturuyor.

merge() fonksiyonu kullanılarak, num_rating_df ve avg_rating_df veri çerçeveleri 'Book-Title' sütunu üzerinde birleştiriliyor. Bu, her iki veri çerçevesindeki aynı kitaplar üzerinde birleştirme yapılmasını sağlar.

Sonuç olarak, 'popular_df' adında bir DataFrame elde edilir. Bu DataFrame, her bir kitabın toplam derecelendirme sayısını ve ortalama derecelendirmesini içerir.

	Book-Title	num_ratings	avg_rating
0	A Light in the Storm: The Civil War Diary of ...	4	2.250000
1	Always Have Popsicles	1	0.000000
2	Apple Magic (The Collector's series)	1	0.000000
3	Ask Lily (Young Women of Faith: Lily Series, ...	1	8.000000
4	Beyond IBM: Leadership Marketing and Finance ...	1	0.000000
...
241066	Ã?Â?lpiraten.	2	0.000000
241067	Ã?Â?rger mit Produkt X. Roman.	4	5.250000
241068	Ã?Â?sterlich leben.	1	7.000000
241069	Ã?Â?stlich der Berge.	3	2.666667
241070	Ã?Â?thique en toc	2	4.000000

241071 rows × 3 columns


```
popular_df = popular_df[popular_df['num_ratings'] >= 250].sort_values('avg_rating', ascending=False).head(50)
```

Bu kod, popular_df DataFrame'ini filtreleyerek ve sıralayarak en popüler kitapları seçiyor. İşlemler şu şekilde gerçekleştiriliyor:

popular_df DataFrame'indeki num_ratings sütununda 250 veya daha fazla derecelendirme olan kitapları seçiyoruz. Bu, daha geniş bir kullanıcı kitlesi tarafından derecelendirilen kitapları almayı sağlar.

Ardından, bu filtelenmiş verileri ortalama derecelendirme (avg_rating) sütununa göre azalan sırada sıralıyoruz. Bu, en yüksek ortalama derecelendirmeye sahip kitapların en üstte olmasını sağlar.

Son olarak, ilk 50 kitabı (head(50)) alıyoruz.

Bu işlemlerin sonucunda, en popüler 50 kitabı içeren bir DataFrame elde edilir. Bu kitaplar, geniş bir derecelendirme yelpazesine sahip oldukları ve yüksek bir ortalama derecelendirmeye birlikte oldukça popüler oldukları için tavsiye edilebilirler.

```
popular_df = popular_df.merge(books, on = 'Book-Title').drop_duplicates('Book-Title')[['Book-Title', 'Book-Author', 'Image-URL-M', 'num_ratings', 'avg_rating']]
```

Bu kod, popular_df DataFrame'ini 'Book-Title' sütunu üzerinde birleştirerek tekrarlanan girişleri kaldırıyor ve yalnızca belirli sütunları seçiyor. İşlemler şu şekilde gerçekleştiriliyor:

popular_df DataFrame'ini, 'Book-Title' sütunu üzerinde 'books' veri çerçevesiyle birleştiriyoruz. Bu, popüler kitapların detaylarını içeren books veri çerçevesinden ilgili bilgileri almayı sağlar.

drop_duplicates('Book-Title') yöntemini kullanarak tekrarlanan girişleri kaldırıyoruz. Bu, aynı kitaba ait birden fazla girişi olan durumları ele alır ve yalnızca bir girişi tutar.

Son olarak, 'Book-Title', 'Book-Author', 'Image-URL-M', 'num_ratings' ve 'avg_rating' sütunlarını seçiyoruz. Bu, popüler kitapların adını, yazarını, kapağını, toplam derecelendirme sayısını ve ortalama derecelendirmesini içeren bir DataFrame oluşturur.

Bu işlemlerin sonucunda, popüler kitapların belirli bilgilerini içeren bir DataFrame elde edilir. Bu bilgiler, kullanıcılara kitap tavsiye etmek için kullanılabilir.

popular_df

	Book-Title	Book-Author	Image-URL-M	num_ratings	avg_rating
0	Harry Potter and the Prisoner of Azkaban (Book 3)	J. K. Rowling	http://images.amazon.com/images/P/0439136350.0...	428	5.852804
3	Harry Potter and the Goblet of Fire (Book 4)	J. K. Rowling	http://images.amazon.com/images/P/0439139597.0...	387	5.824289
5	Harry Potter and the Sorcerer's Stone (Book 1)	J. K. Rowling	http://images.amazon.com/images/P/0590353403.0...	278	5.737410
9	Harry Potter and the Order of the Phoenix (Boo...	J. K. Rowling	http://images.amazon.com/images/P/043935806X.0...	347	5.501441
13	Harry Potter and the Chamber of Secrets (Book 2)	J. K. Rowling	http://images.amazon.com/images/P/0439064872.0...	556	5.183453
16	The Hobbit : The Enchanting Prelude to The Lor...	J.R.R. TOLKIEN	http://images.amazon.com/images/P/0345339681.0...	281	5.007117
17	The Fellowship of the Ring (The Lord of the Ri...	J.R.R. TOLKIEN	http://images.amazon.com/images/P/0345339703.0...	368	4.948370
26	Harry Potter and the Sorcerer's Stone (Harry P...	J. K. Rowling	http://images.amazon.com/images/P/059035342X.0...	575	4.895652
28	The Two Towers (The Lord of the Rings, Part 2)	J.R.R. TOLKIEN	http://images.amazon.com/images/P/0345339711.0...	260	4.880769
39	To Kill a Mockingbird	Harper Lee	http://images.amazon.com/images/P/0446310786.0...	510	4.700000
47	The Da Vinci Code	Dan Brown	http://images.amazon.com/images/P/0385504209.0...	898	4.642539
53	The Five People You Meet in Heaven	Mitch Albom	http://images.amazon.com/images/P/0786868716.0...	430	4.551163
55	The Catcher in the Rye	J.D. Salinger	http://images.amazon.com/images/P/0316769487.0...	449	4.545657
62	The Lovely Bones: A Novel	Alice Sebold	http://images.amazon.com/images/P/0316666343.0...	1295	4.468726
63	1984	George Orwell	http://images.amazon.com/images/P/0451524934.0...	284	4.454225
72	Prodigal Summer: A Novel	Barbara Kingsolver	http://images.amazon.com/images/P/0060959037.0...	253	4.450593
73	Neverwhere	Neil Gaiman	http://images.amazon.com/images/P/0380789019.0...	265	4.449057
78	The Secret Life of Bees	Sue Monk Kidd	http://images.amazon.com/images/P/0142001740.0...	774	4.447028
84	Stupid White Men ...and Other Sorry Excuses fo...	Michael Moore	http://images.amazon.com/images/P/0060392452.0...	283	4.356890
85	Tuesdays with Morrie: An Old Man, a Young Man,...	MITCH ALBOM	http://images.amazon.com/images/P/0385484518.0...	493	4.354970
88	The Red Tent (Bestselling Backlist)	Anita Diamant	http://images.amazon.com/images/P/0312195516.0...	723	4.334716
89	The Hitchhiker's Guide to the Galaxy	Douglas Adams	http://images.amazon.com/images/P/0671461494.0...	268	4.328358
98	Nickel and Dime: On (Not) Getting By in America	Barbara Ehrenreich	http://images.amazon.com/images/P/0805063897.0...	335	4.289552
100	Into the Wild	Jon Krakauer	http://images.amazon.com/images/P/0385486804.0...	252	4.273810
103	Fahrenheit 451	Ray Bradbury	http://images.amazon.com/images/P/3257208626.0...	409	4.264059
116	Girl with a Pearl Earring	Tracy Chevalier	http://images.amazon.com/images/P/0452282152.0...	526	4.218631
117	Outlander	DIANA GABALDON	http://images.amazon.com/images/P/0440222915.0...	283	4.173145
123	Where the Heart Is (Oprah's Book Club (Paperba...	Billie Letts	http://images.amazon.com/images/P/0446672211.0...	585	4.105983
124	Seabiscuit: An American Legend	LAURA HILLENBRAND	http://images.amazon.com/images/P/0449005615.0...	275	4.098182
126	Life of Pi	Yann Martel	http://images.amazon.com/images/P/0151008116.0...	664	4.088855
130	The Bean Trees	Barbara Kingsolver	http://images.amazon.com/images/P/0060915544.0...	389	4.087404
133	A Child Called \t": One Child's Courage to S...	Dave Pelzer	http://images.amazon.com/images/P/1558743669.0...	265	4.086792
135	ANGELA'S ASHES	Frank McCourt	http://images.amazon.com/images/P/0684874350.0...	279	4.075269
137	Good in Bed	Jennifer Weiner	http://images.amazon.com/images/P/0743418174.0...	490	4.055102
139	Fast Food Nation: The Dark Side of the All-Ame...	Eric Schlosser	http://images.amazon.com/images/P/0060938455.0...	321	4.037383
140	American Gods	Neil Gaiman	http://images.amazon.com/images/P/0380789035.0...	302	4.006623
141	Skipping Christmas	JOHN GRISHAM	http://images.amazon.com/images/P/0385505833.0...	322	4.006211
147	Left Behind: A Novel of the Earth's Last Days ...	Tim Lahaye	http://images.amazon.com/images/P/0842329129.0...	318	4.003145
149	The Golden Compass (His Dark Materials, Book 1)	PHILIP PULLMAN	http://images.amazon.com/images/P/037582345X.0...	336	4.000000
157	The Color Purple	Alice Walker	http://images.amazon.com/images/P/0671617028.0...	314	3.964968
163	Silence of the Lambs	Thomas Harris	http://images.amazon.com/images/P/0312924585.0...	256	3.960938
166	About a Boy	Nick Hornby	http://images.amazon.com/images/P/1573227331.0...	262	3.900763
173	Seven Up (A Stephanie Plum Novel)	Janet Evanovich	http://images.amazon.com/images/P/0312265840.0...	278	3.888489
175	The Alchemist: A Fable About Following Your Dream	Paulo Coelho	http://images.amazon.com/images/P/0062502174.0...	266	3.875940
177	Hard Eight : A Stephanie Plum Novel (A Stephan...	Janet Evanovich	http://images.amazon.com/images/P/0312983867.0...	269	3.825279
182	Suzanne's Diary for Nicholas	James Patterson	http://images.amazon.com/images/P/0316969443.0...	457	3.820569
186	Lord of the Flies	William Gerald Golding	http://images.amazon.com/images/P/0399501487.0...	259	3.818533
192	A Prayer for Owen Meany	John Irving	http://images.amazon.com/images/P/0345361792.0...	413	3.796610
194	The Vampire Lestat (Vampire Chronicles, Book II)	ANNE RICE	http://images.amazon.com/images/P/0345313860.0...	301	3.777409
195	White Oleander : A Novel (Oprah's Book Club)	Janet Fitch	http://images.amazon.com/images/P/0316284955.0...	356	3.772472

Collaborative Filtering Based Recommender (Collabrative fillreleme yaparak girilen kitaba göre önerme)

```
x = ratings_with_name.groupby('User-ID').count()['Book-Rating'] > 200
padhe_likhe_users = x[x].index
```

Bu kod, her kullanıcının kaç kitaba derecelendirme yaptığını sayar, ardından 200'den fazla derecelendirme yapan kullanıcıları belirler. İşlemler şu şekilde gerçekleştirilir:

ratings_with_name DataFrame'i üzerinde 'User-ID' sütunuyla gruptama yapılır. Her bir kullanıcının yaptığı derecelendirme sayısı count() fonksiyonu ile hesaplanır. Ardından, bu derecelendirmelerin 200'den fazla olup olmadığı kontrol edilir (> 200). Sonuçlar, True veya False değerlerini içeren bir Seri olarak alınır. x[x] ifadesi, yalnızca 200'den fazla derecelendirme yapan kullanıcıların indekslerini içeren bir Seri'yi seçer.

Bu indeksler, padhe_likhe_users değişkenine atanır.

Bu işlemler sonucunda, 200'den fazla kitaba derecelendirme yapmış olan kullanıcıların 'User-ID' değerlerini içeren bir dizi elde edilir. Bu kullanıcılar, padhe likhe (eğitimli) kullanıcılar olarak adlandırılabilir ve daha güvenilir bir kitap tavsiye sistemi oluşturmak için kullanılabilirler.

```
filtered_rating = ratings_with_name[ratings_with_name['User-ID'].isin(padhe_likhe_users)]
```

Bu kod, "padhe_likhe_users" olarak adlandırılan eğitimli kullanıcıların yaptığı derecelendirmeleri içeren bir DataFrame oluşturur.

ratings_with_name DataFrame'indeki 'User-ID' sütununu filtrelemek için isin() yöntemi kullanılır. Bu yöntem, belirli bir dizi içindeki değerlere sahip olan satırları seçmemizi sağlar.

Seçilen kullanıcılar, "padhe_likhe_users" değişkenindeki eğitimli kullanıcıların 'User-ID' değerleridir.

Bu filtreleme işlemi sonucunda, yalnızca eğitimli kullanıcıların yaptığı derecelendirmeleri içeren bir DataFrame olan "filtered_rating" oluşturulur.

Bu işlem, daha güvenilir bir kitap tavsiye sistemi oluşturmak için sadece eğitimli kullanıcıların derecelendirmelerini kullanmamızı sağlar. Bu derecelendirmeler, genellikle daha niteliklidir ve daha güvenilir bir kullanıcı kitlesini temsil eder.

```
y = filtered_rating.groupby('Book-Title').count()['Book-Rating'] >= 50
famous_books = y[y].index
```

Bu kod, belirli bir sayıda (50 veya daha fazla) derecelendirme almış olan kitapları seçer.

Öncelikle, "filtered_rating" DataFrame'indeki 'Book-Title' sütunuyla grupta yapılar.

Her bir kitabın aldığı derecelendirme sayısı count() fonksiyonu ile hesaplanır.

Ardından, bu derecelendirmelerin 50 veya daha fazla olup olmadığı kontrol edilir (>= 50).

Sonuçlar, True veya False değerlerini içeren bir Seri olarak alınır.

Bu Seri içindeki True değerlere sahip olan kitapların indeksleri seçilir.

Seçilen indeksler, "famous_books" değişkenine atanır.

Bu işlemler sonucunda, en az 50 derecelendirme almış olan kitapların adlarını içeren bir dizi elde edilir. Bu kitaplar, popülerliklerini kanıtlamış ve kullanıcılar tarafından yaygın olarak derecelendirilmiş olabilirler.

```
final_ratings = filtered_rating[filtered_rating['Book-Title'].isin(famous_books)]
```

Bu kod, en az 50 derecelendirme almış olan popüler kitaplar arasında yer alan derecelendirmeleri içeren bir DataFrame oluşturur.

filtered_rating DataFrame'indeki 'Book-Title' sütunu, "famous_books" değişkeninde belirtilen popüler kitapların adlarıyla eşleştirilir.

Bu eşleşmeyi sağlayan satırlar, isin() yöntemi kullanılarak filtrelenir ve yalnızca popüler kitapların derecelendirmelerini içeren DataFrame'e atanır.

Bu işlem sonucunda, sadece popüler kitaplar arasında yer alan derecelendirmeleri içeren bir DataFrame olan "final_ratings" elde edilir.

```
pt = final_ratings.pivot_table(index='Book-Title', columns='User-ID', values='Book-Rating')
```

Bu kod, "final_ratings" DataFrame'inden bir pivot tablosu oluşturur. Pivot tablosu, kullanıcıların her bir kitaba verdiği derecelendirmeleri içerir.

pivot_table() fonksiyonu, bir DataFrame üzerinde pivot tablosu oluşturmak için kullanılır.

index='Book-Title' parametresi, pivot tablosunun satır indekslerini 'Book-Title' (kitap adı) sütununa ayarlar.

columns='User-ID' parametresi, pivot tablosunun sütun indekslerini 'User-ID' (kullanıcı kimliği) sütununa ayarlar.

values='Book-Rating' parametresi, pivot tablosunun değerlerini 'Book-Rating' (kitap derecelendirmesi) sütununa ayarlar.

Bu işlem sonucunda, her bir kitabın her bir kullanıcı tarafından verilen derecelendirmelerini içeren bir pivot tablosu oluşturulur.

```
pt.fillna(0,inplace=True)
```

Bu kod, pivot tablosundaki eksik değerleri (NaN) sıfır değerleriyle doldurur.

`fillna(0, inplace=True)`: Bu yöntem, pivot tablosundaki tüm eksik değerleri (NaN) sıfır (0) değerleriyle doldurur. `inplace=True` parametresi, değişikliklerin doğrudan orijinal DataFrame'e uygulanmasını sağlar.

Bu işlem sonucunda, pivot tablosundaki tüm eksik değerlerin sıfır değerleriyle doldurulmuş bir versiyon elde edilir. Bu, daha fazla işlem yapmadan önce tablonun tutarlı bir şekilde kullanılmasını sağlar.

pt

User-ID	254	2276	2766	2977	3363	4017	4385	6251	6323	6543	...	271705	273979	274004	274061	274301	274308	275970	277427	277639	27841
Book-Title																					
1984	9.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	10.0	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	9.0	...	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	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	0.0	0.0	0.0	0.0
A Bend in the Road	0.0	0.0	7.0	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	0.0	0.0
...
Year of Wonders	0.0	0.0	0.0	7.0	0.0	0.0	0.0	0.0	0.0	0.0	...	0.0	9.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0
You Belong To Me	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	0.0	0.0	0.0	0.0
Zen and the Art of Motorcycle Maintenance: An Inquiry Into Values	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	0.0	0.0	0.0	0.0
Zoya	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	0.0	0.0	0.0	0.0

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

Bu kod, sklearn kütüphanesinden `cosine_similarity` fonksiyonunu dahil eder. Bu fonksiyon, verilen özellik matrisi üzerinde kosinüs benzerliği hesaplar.

`cosine_similarity`: Bu fonksiyon, verilen iki matris arasındaki kosinüs benzerliğini hesaplar. Kosinüs benzerliği, iki vektör arasındaki açıya dayalı bir benzerlik ölçüsüdür. İki vektörün aynı yöne doğru ne kadar benzediğini ölçer.

Bu fonksiyon, genellikle kullanıcıların veya ürünlerin profillerini temsil eden özellik matrisleri arasındaki benzerlikleri hesaplamak için kullanılır. Özellikle, kitap tavsiye sistemlerinde, kullanıcıların ve kitapların özelliklerini temsil eden matrisler arasındaki benzerlikleri hesaplamak için kullanılabilir.

```
similarity_scores = cosine_similarity(pt)
```

Bu kod, önceden oluşturulmuş olan pivot tablosu üzerinde kosinüs benzerliklerini hesaplar.

cosine_similarity(pt): Bu fonksiyon, pt olarak adlandırılan pivot tablosu üzerinde kosinüs benzerliklerini hesaplar. Bu fonksiyon, pivot tablosundaki her bir satırı bir vektör olarak düşünür ve bu vektörler arasındaki kosinüs benzerliğini hesaplar. Sonuç olarak, her bir satır için diğer satırlarla olan benzerlik ölçüleri içeren bir benzerlik matrisi elde edilir.

Bu işlem sonucunda, pivot tablosundaki kitaplar arasındaki benzerlikleri gösteren bir benzerlik matrisi olan "similarity_scores" oluşturulur.

```
similarity_scores.shape
```

Benzerlik puanlarının şekli, benzerlik matrisinin boyutunu belirtir. Özellikle, her bir satır ve sütun, pivot tablosundaki bir kitabı temsil eder. Dolayısıyla, matrisin şekli, toplam kitap sayısı ile aynı olmalıdır.

"similarity_scores.shape" kodu, benzerlik matrisinin boyutunu döndürür. Bu şekilde, matrisin kaç satır ve sütundan oluştuğunu öğreniriz. Her bir satır ve sütun, pivot tablosundaki bir kitabı temsil eder.

Eğer bu kodu çalıştırsak, dönen değer bir tuple olacaktır. Tuple'ın ilk elemanı satır sayısını (yani toplam kitap sayısını) ve ikinci elemanı sütun sayısını temsil eder. Dolayısıyla, (satır_sayısı, sütun_sayısı) şeklinde bir çıktı elde edilir. Bu çıktı, benzerlik matrisinin boyutunu belirtir.

```
(706, 706)
```

```
def recommend(book_name):
    index = np.where(pt.index == book_name)[0][0]
    similar_items = sorted(list(enumerate(similarity_scores[0])),key = lambda x:x[1], reverse = True)[1:6]

    data = []
    for i in similar_items:
        item = []
        temp_df = books[books['Book-Title'] == pt.index[i][0]]
        item.extend(list(temp_df.drop_duplicates('Book-Title')['Book-Title'].values))
        item.extend(list(temp_df.drop_duplicates('Book-Title')['Book-Author'].values))
        item.extend(list(temp_df.drop_duplicates('Book-Title')['Image-URL-M'].values))

        data.append(item)

    return data
```

Bu fonksiyon, bir kitap adı alır ve bu kitaba benzer olan diğer kitapları önerir.

İlk olarak, pt pivot tablosunda, verilen kitap adının indeksini bulur. Bu, kitabın benzerlik matrisindeki konumunu belirler.

Daha sonra, bu kitaba benzer olan diğer kitapları ve benzerlik puanlarını bulur.

Benzerlik puanlarına göre sıralama yapar ve en benzer olan 5 kitabı seçer.

Her bir benzer kitap için, kitap adı, yazarı ve kapağını içeren bir liste oluşturur.

Bu bilgileri içeren bir liste oluşturur ve döndürür.

Bu fonksiyon, kullanıcının belirli bir kitap için önerilen diğer kitapları almasına yardımcı olur. Bu şekilde, kullanıcıların ilgi alanlarına ve tercihlerine daha uygun kitaplar önerilebilir.

Örnek Kullanım

```
recommend('1984')
```

```
[[ 'Animal Farm',  
  'George Orwell',  
  'http://images.amazon.com/images/P/0451526341.01.MZZZZZZZ.jpg'],  
  ["The Handmaid's Tale",  
  'Margaret Atwood',  
  'http://images.amazon.com/images/P/0449212602.01.MZZZZZZZ.jpg'],  
  ['Brave New World',  
  'Aldous Huxley',  
  'http://images.amazon.com/images/P/0060809833.01.MZZZZZZZ.jpg'],  
  ['The Vampire Lestat (Vampire Chronicles, Book II)',  
  'ANNE RICE',  
  'http://images.amazon.com/images/P/0345313860.01.MZZZZZZZ.jpg'],  
  ['The Hours : A Novel',  
  'Michael Cunningham',  
  'http://images.amazon.com/images/P/0312243022.01.MZZZZZZZ.jpg']]
```

Content-Based Filtreleme Kullanarak Kitap Öneri

```
import pandas as pd  
  
# Veri Setlerini Yükleme  
books = pd.read_csv('Books.csv')  
ratings = pd.read_csv('Ratings.csv')
```

pd.read_csv() fonksiyonu, CSV dosyalarını pandas DataFrame objelerine dönüştürür. İşte bu kodun yaptığı şey:

import pandas as pd: pandas kütüphanesini pd takma adıyla içe aktarıyoruz.

books = pd.read_csv('Books.csv'): 'Books.csv' adlı CSV dosyasını okuyarak veriyi

books adlı bir DataFrame'e yüklüyoruz.

ratings = pd.read_csv('Ratings.csv'): 'Ratings.csv' adlı CSV dosyasını okuyarak veriyi

ratings adlı bir DataFrame'e yüklüyoruz.

```
# Kitap özelliklerini birleştirme
book_features = books[['Book-Title', 'Book-Author', 'Year-Of-Publication', 'Publisher']]
book_features.head()
```

	Book-Title	Book-Author	Year-Of-Publication	Publisher
0	Classical Mythology	Mark P. O. Morford	2002	Oxford University Press
1	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada
2	Decision in Normandy	Carlo D'Este	1991	HarperPerennial
3	Flu: The Story of the Great Influenza Pandemic...	Gina Bari Kolata	1999	Farrar Straus Giroux
4	The Mummies of Urumchi	E. J. W. Barber	1999	W. W. Norton & Company

Bu kod parçası, books DataFrame'indeki belirli sütunları seçerek kitap özelliklerini içeren bir DataFrame oluşturuyor. İşte bu kodun yaptığı işlemler:

books[['Book-Title', 'Book-Author', 'Year-Of-Publication', 'Publisher']]: 'books' DataFrame'inden 'Book-Title' (Kitap Başlığı), 'Book-Author' (Kitap Yazarı), 'Year-Of-Publication' (Yayın Yılı) ve 'Publisher' (Yayınevi) sütunlarını seçiyoruz.
book_features.head(): Oluşturulan DataFrame'in ilk beş satırını görüntülüyoruz.

```
from sklearn.feature_extraction.text import TfidfVectorizer

# Kitap özelliklerini birleştirme
book_features = books[['Book-Title', 'Book-Author', 'Year-Of-Publication', 'Publisher']]
book_features.fillna('', inplace=True) # Eksik değerleri doldur

# Tüm özellikleri birleştirme
book_features['all_features'] = book_features['Book-Title'] + ' ' + book_features['Book-Author'] + ' ' + \
    book_features['Year-Of-Publication'].astype(str) + ' ' + book_features['Publisher']

# TF-IDF vektörleme
tfidf_vectorizer = TfidfVectorizer()
tfidf_matrix = tfidf_vectorizer.fit_transform(book_features['all_features'])

tfidf_matrix.shape
```

Bu kod parçası, TF-IDF (Terim Frekansı-Ters Belge Frekansı) vektörleme yöntemini kullanarak kitap özelliklerini sayısal vektörlere dönüştürüyor. İşte bu kodun yaptığı işlemler:

from sklearn.feature_extraction.text import TfidfVectorizer: Scikit-learn kütüphanesinden TF-IDF vektörleme işlemi için TfidfVectorizer sınıfını içe aktarıyoruz.

book_features.fillna("", inplace=True): book_features DataFrame'indeki eksik değerleri (NaN) boşluklarla dolduruyoruz. Bu, metin verilerini düzgün bir şekilde birleştirirken hata almamızı önler.

book_features['all_features']: 'Book-Title', 'Book-Author', 'Year-Of-Publication' ve 'Publisher' sütunlarını birleştirerek her kitabın tüm özelliklerini içeren bir 'all_features' adında yeni bir sütun oluşturuyoruz.

tfidf_vectorizer = TfidfVectorizer(): TF-IDF vektörleme işlemi için bir TfidfVectorizer nesnesi oluşturuyoruz.

tfidf_matrix = tfidf_vectorizer.fit_transform(book_features['all_features']): TfidfVectorizer'ı kullanarak 'all_features' sütunundaki metin verilerini TF-IDF vektörlerine dönüştürüyoruz. Bu, her kitabı bir TF-IDF vektörüyle temsil eden bir matris oluşturur.

tfidf_matrix.shape: Oluşturulan TF-IDF matrisinin boyutlarını görüntülüyoruz.

Bu kod parçası, kitap özelliklerini TF-IDF vektörlerine dönüştürerek, her bir kitabın içeriğini sayısal bir şekilde temsil etmiş oluruz.

```
from sklearn.neighbors import NearestNeighbors

# NearestNeighbors modeli oluşturma
nn_model = NearestNeighbors(n_neighbors=10, algorithm='auto', metric='cosine')
nn_model.fit(tfidf_matrix)

def get_similar_books(book_title):
    # Kitap adına göre en benzer kitapları bulma
    book_idx = book_features[book_features['Book-Title'] == book_title].index
    if len(book_idx) == 0:
        return []

    book_idx = book_idx[0] # Eşleşen ilk kitabı al
    distances, indices = nn_model.kneighbors(tfidf_matrix[book_idx], n_neighbors=11)

    similar_books = []
    for i in range(1, len(distances.flatten())):
        similar_books.append({
            'title': book_features.iloc[indices.flatten()[i]]['Book-Title'],
            'author': book_features.iloc[indices.flatten()[i]]['Book-Author'],
            'similarity': 1 - distances.flatten()[i] # Benzerlik skoru
        })

    return similar_books

# Örnek olarak bir kitabın en benzer 10 kitabını alalım
similar_books = get_similar_books('Classical Mythology')
similar_books
```

Bu kod parçası, NearestNeighbors algoritmasını kullanarak bir kitaba en yakın benzer kitapları bulmayı sağlar. İşte bu kodun yaptığı işlemler:

from sklearn.neighbors import NearestNeighbors: Scikit-learn kütüphanesinden NearestNeighbors sınıfını içe aktarıyoruz. Bu sınıf, k-NN (en yakın komşu) algoritmasını uygular.

nn_model = NearestNeighbors(n_neighbors=10, algorithm='auto', metric='cosine'): NearestNeighbors modelini oluşturuyoruz.

Bu model, 10 en yakın komşuyu (n_neighbors=10) ve kosinüs benzerliği metriğini (metric='cosine') kullanır.

nn_model.fit(tfidf_matrix): Modeli, TF-IDF matrisi üzerinde eğitiyoruz.

def get_similar_books(book_title):: get_similar_books adında bir fonksiyon

tanımlıyoruz. Bu fonksiyon, belirli bir kitaba en benzer kitapları bulur.

book_idx = book_features[book_features['Book-Title'] == book_title].index: Verilen kitap adına (book_title) göre, veri setindeki ilgili satırın dizinini buluyoruz.

if len(book_idx) == 0:: Eğer belirtilen kitap adı veri setinde yoksa, boş bir liste döndürüyoruz.

distances, indices = nn_model.kneighbors(tfidf_matrix[book_idx], n_neighbors=11): Belirtilen kitabın TF-IDF vektörünü kullanarak, en yakın 11 kitabın indislerini (indices) ve bu kitaplara olan uzaklıkları (distances) buluyoruz.

for i in range(1, len(distances.flatten())):: Döngüyle, en yakın komşuların indislerini ve benzerlik skorlarını bir liste olarak saklıyoruz.

return similar_books: Benzer kitapları içeren bir liste döndürüyoruz.

Bu fonksiyon, verilen bir kitaba en benzer 10 kitabı bulur ve her birini kitap adı, yazarı ve benzerlik skoruyla birlikte bir sözlük içinde döndürür. Son olarak, örneğin 'Classical Mythology' adlı bir kitaba en benzer 10 kitabı bulmak için bu fonksiyonu kullanırız.

KOD ÇIKTISI

```
[{'title': 'Classical mythology',  
  'author': 'Mark P. O Morford',  
  'similarity': 0.7959678751746495},  
{ 'title': 'Classical mythology',  
  'author': 'Mark P.O Morford',  
  'similarity': 0.7910106176477564},  
{ 'title': 'Classical Mythology',  
  'author': 'Mark P. O. Morford',  
  'similarity': 0.7536294519640896},  
{ 'title': 'The Oxford Classical Dictionary',  
  'author': 'Simon Hornblower',  
  'similarity': 0.4218899598093364},  
{ 'title': 'The Oxford Companion to Classical Literature (Oxford Paperbacks)',  
  'author': 'Paul H. Harvey',  
  'similarity': 0.40594388558958916},  
{ 'title': 'The Oxford Companion to Classical Civilization',  
  'author': 'Simon Hornblower',  
  'similarity': 0.3839290774025582},  
{ 'title': 'The Oxford History of the Classical World',  
  'author': 'Oswyn Murray',  
  'similarity': 0.37243474055880377},  
{ 'title': 'Goddesses in World Mythology',  
  'author': 'Martha Ann',  
  'similarity': 0.3723297948193506},  
{ 'title': 'A Dictionary of World Mythology (Oxford Paperback Reference)',  
  'author': 'Arthur Cottrell',  
  'similarity': 0.3682743879229696},  
{ 'title': 'The Dictionary of Classical Mythology',  
  'author': 'JOHN EDWARD ZIMMERMAN',  
  'similarity': 0.3674630549188711}]
```

Arayüz Oluşturma ve Web Sitesi Haline Getirme

Python framework olan flask kullanarak oluşturulmuştur.

Popüler Kitaplar Ekranı

Bu ekran, en popüler 50 kitabı gösterir. Her bir kitabın, kitap adı, yazarı ve kapağı bulunur. Bu şekilde, kullanıcılar popüler kitapları keşfedebilir ve ilgi çekici kitaplara göz atabilirler.

Kitap Öneri Ekranı


Bu ekran, kullanıcının girdiği bir kitaba göre önerilen diğer 5 kitabı gösterir. Kullanıcı, giriş alanına istediği kitabın adını yazarak, bu kitaba benzer olan diğer kitapları keşfedebilir. Önerilen kitaplar, Collaborative Filtering yöntemi kullanılarak belirlenir ve kullanıcının ilgi alanlarına uygun olabilir.

Sitenin Ekran Görüntüleri


My Book Recommender

Home Recommend Contact


TOP 50 BOOKS




Harry Potter and the Prisoner of Azkaban (Book 3)
J. K. Rowling
Votes - 428
Rating - 5.852803738317757




Harry Potter and the Goblet of Fire (Book 4)
J. K. Rowling
Votes - 387
Rating - 5.8242894056847545



Harry Potter and the Sorcerer's Stone (Book 1)
J. K. Rowling
Votes - 278
Rating - 5.737410071942446



Harry Potter and the Order of the Phoenix (Book 5)
J. K. Rowling
Votes - 347
Rating - 5.501440922190202

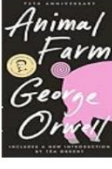


Harry Potter and the Chamber of Secrets (Book 2)
J. K. Rowling
Votes - 556
Rating - 5.183453237410072


My Book Recommender

Home Recommend Contact

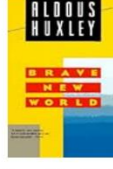
Recommend Books




Animal Farm
George Orwell




The Handmaid's Tale
Margaret Atwood



Brave New World
Aldous Huxley



The Vampire Lestat (Vampire Chronicles, Book II)
ANNE RICE



The Hours : A Novel
Michael Cunningham