Book Recommender System

THAPAR INSTITUTE
OF ENGINEERING & TECHNOLOGY
(Deemed to be University)

Computer Science and Engineering Department

Thapar Institute of Engineering and Technology

(Deemed to be University), Patiala – 147004

Machine Learning Project

Submitted By:

Tanish Modi-102203956

Shivanshu Garg-102253004

Submitted To:

Ms. Kudratdeep Aulakh

Index

1. Introduction

1.1 Book Recommendation Dataset

https://www.kaggle.com/datasets/arashnic/book-recommendation-dataset

1.2 Description

This project is a Book Recommendation System that uses two techniques: Popularity-Based and Collaborative Filtering.

1. Popularity-Based Recommender: Recommends the top books based on the number of ratings and average rating, focusing on books that are most popular among users.
2. Collaborative Filtering: Recommends books by finding similarities between books based on users' ratings using cosine similarity.

The system processes the data, calculates similarity scores, and provides book recommendations. It also saves key data using Pickle for future use.

2. Libraries Used

- Numpy: for performing efficient numerical operations
- Pandas: for loading, cleaning, and manipulating the dataset, making data exploration and transformation
- Sklearn(cosine_similarity): used to measure how similar two non-zero vectors are, based on the cosine of the angle between them
  Cosine Similarity= $(A \cdot B) / (\|A\| \|B\|)$
- Pickle: to save and load pre-processed data

3. Algorithm(s) Used

1. Popularity-Based Recommendation:

This algorithm recommends books based on their overall popularity. Books with more ratings and higher average ratings are considered popular and recommended to all users.

Steps:

1. Merge ratings with book details.
2. Calculate the number of ratings and average rating for each book.
3. Recommend the top books with high ratings and many reviews.

2. Collaborative Filtering (Cosine Similarity):

This method recommends books based on user preferences. It calculates the similarity between books using cosine similarity, which measures the angle between their rating vectors.

Steps:

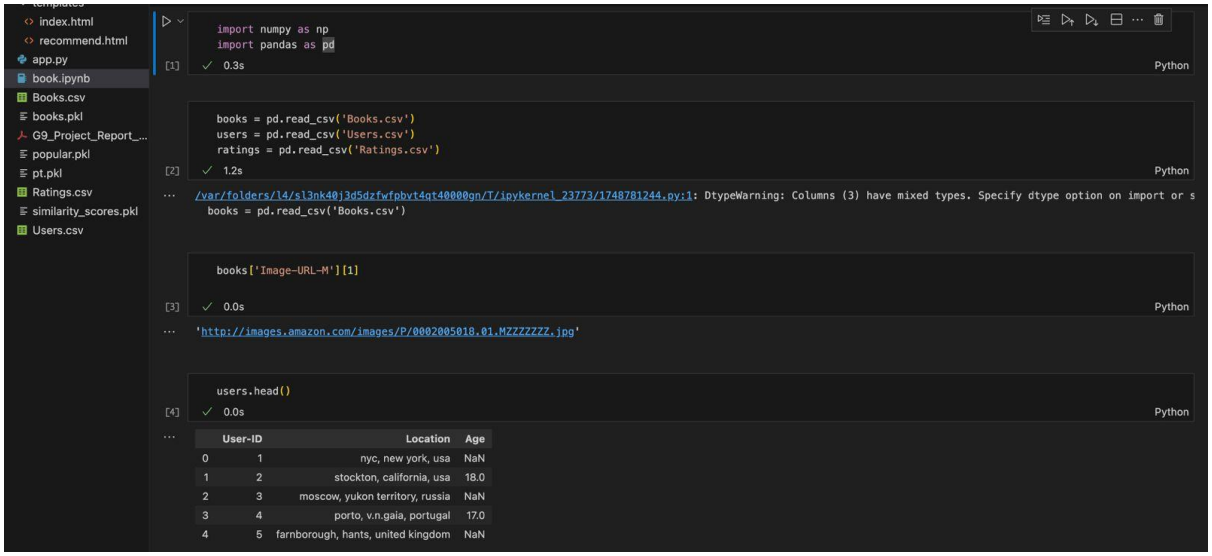
1. Create a user-item matrix with book ratings.
2. Calculate similarity scores between books using cosine similarity.
3. Recommend the most similar books based on a user's previous ratings.

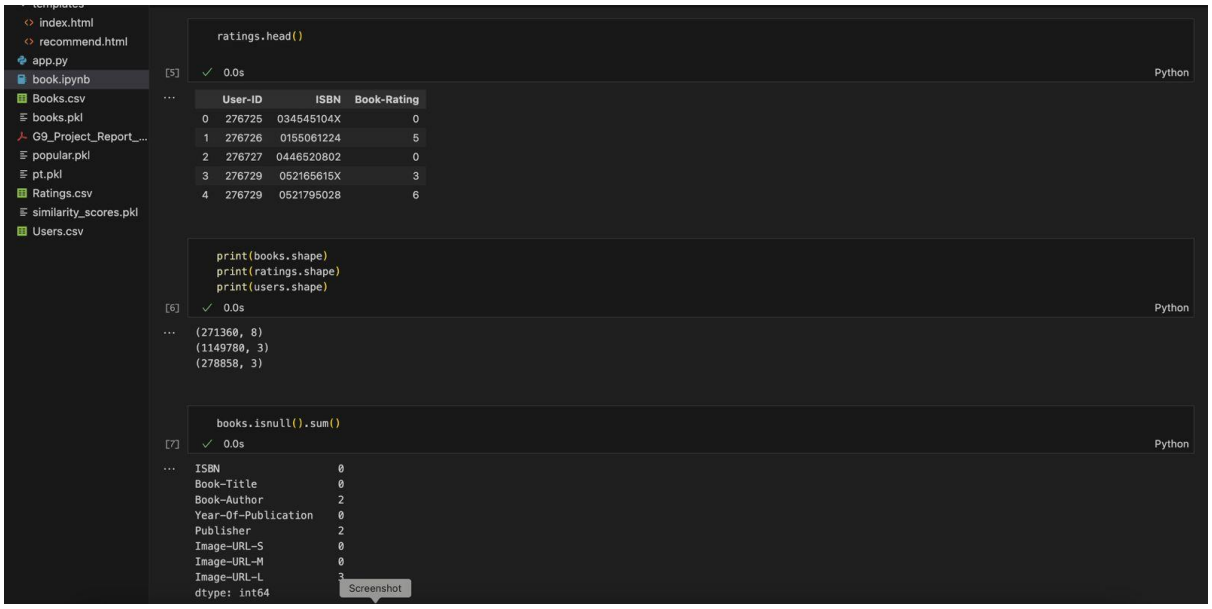
Comparison:

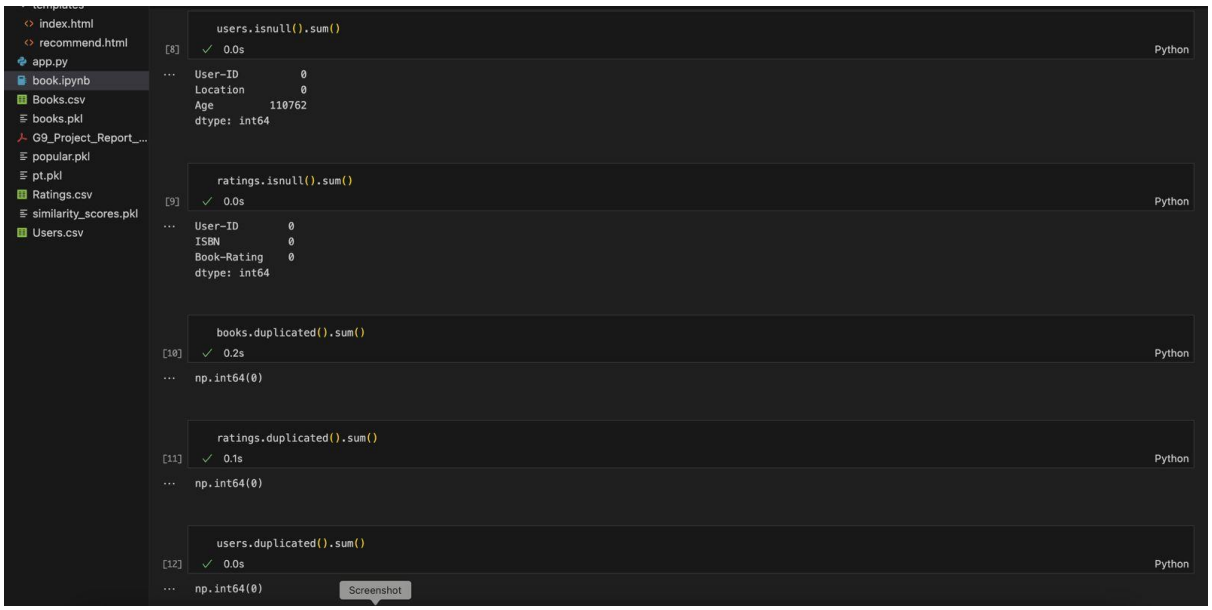
The Book Recommender System uses two algorithms:

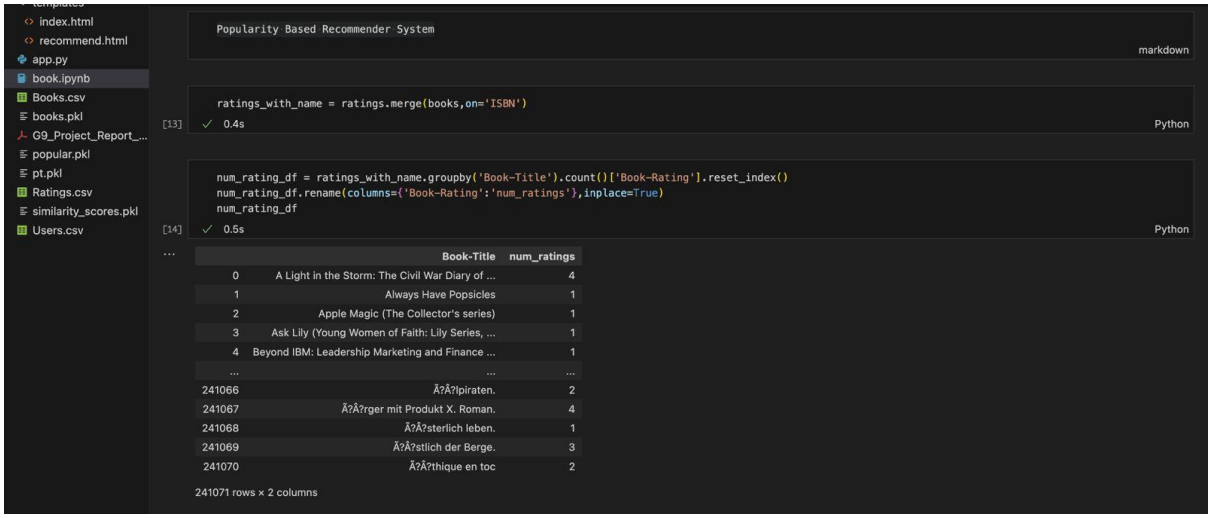
1. Popularity-Based Recommendation: Recommends books based on the number of ratings and average rating, favoring popular books.
2. Collaborative Filtering (Cosine Similarity): Personalizes recommendations by finding similar books based on user ratings using cosine similarity.

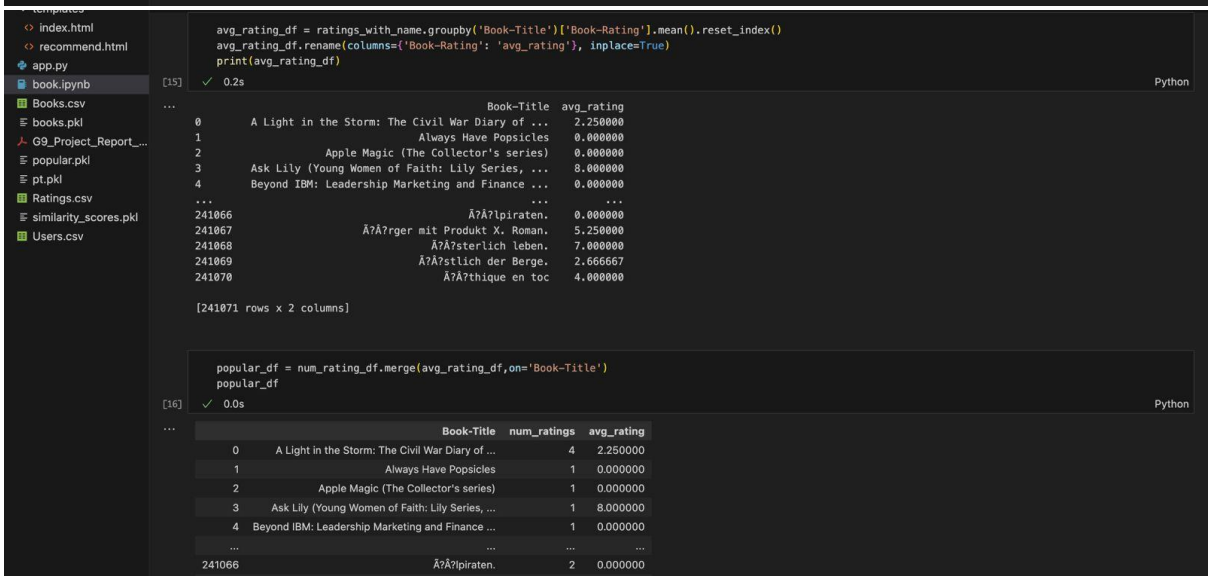
4. Code and Screenshots

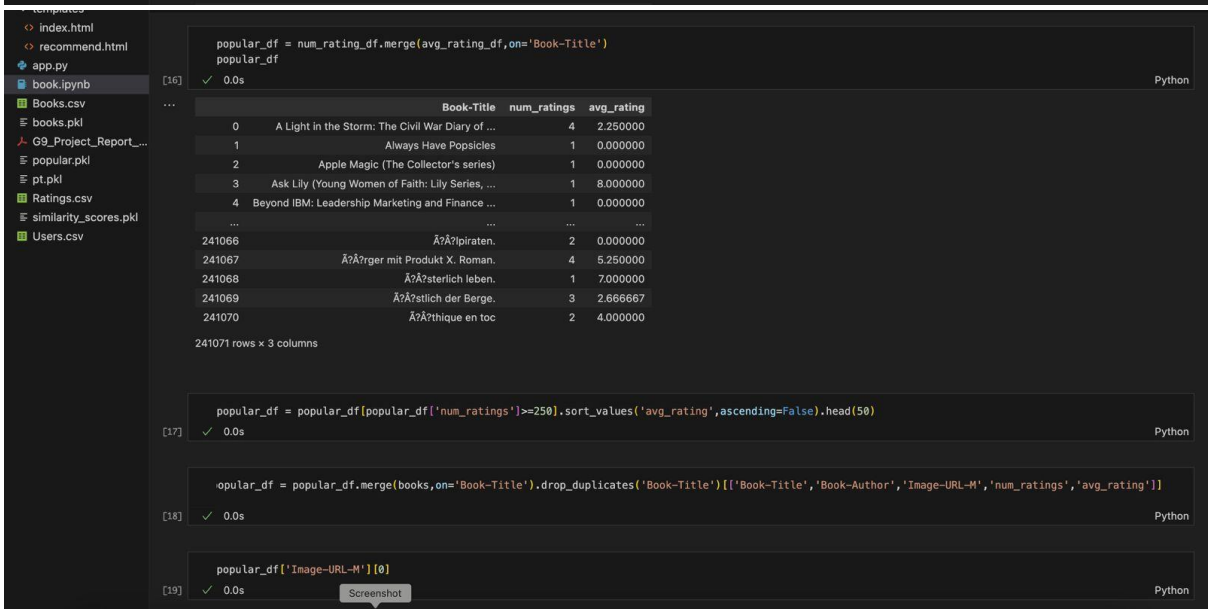


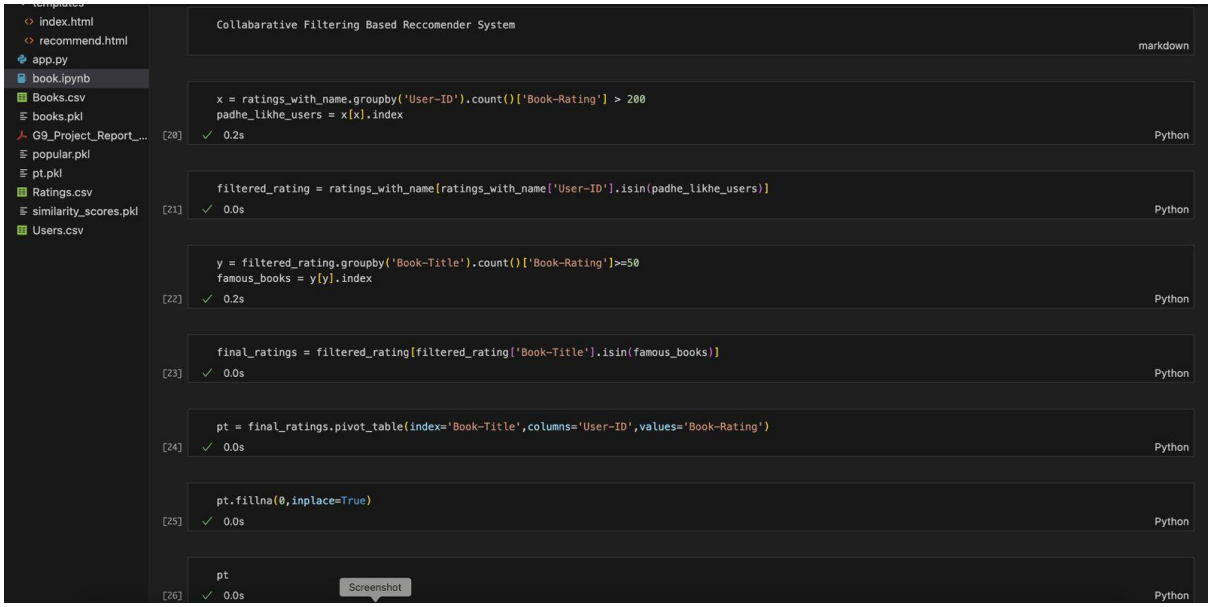


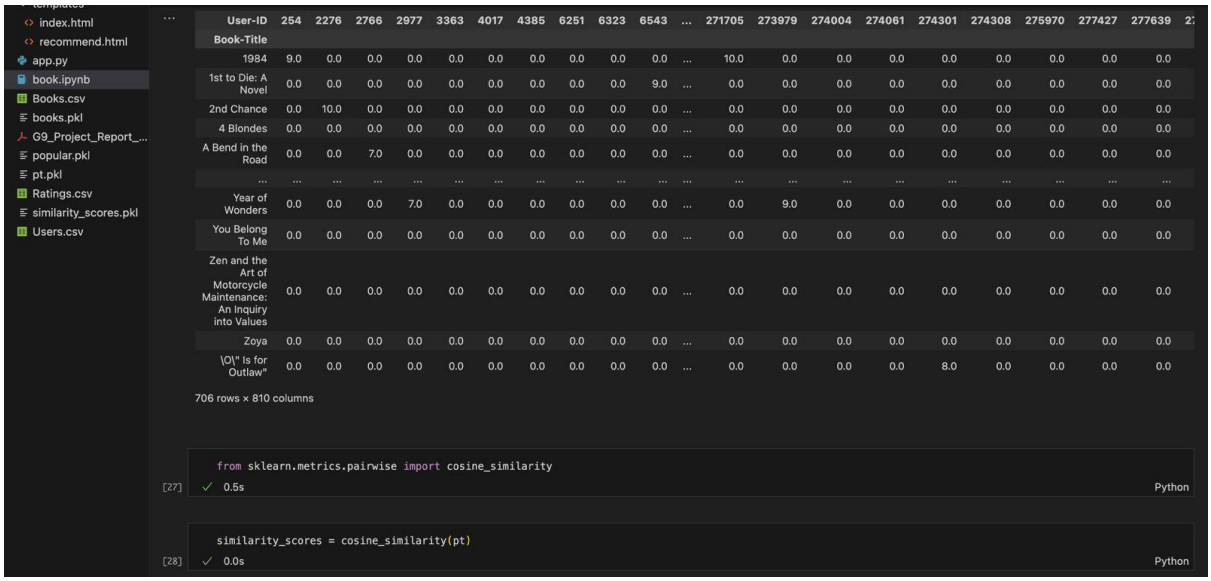



index.html
recommend.html
app.py
book.ipynb
Books.csv
books.pkl
G9_Project_Report_...
popular.pkl
pt.pkl
Ratings.csv
similarity_scores.pkl
Users.csv
Popularity Based Recommender System
markdown
ratings_with_name = ratings.merge(books,on='ISBN')
[13] 0.4s
Python
num_rating_df = ratings_with_name.groupby('Book-Title').count()['Book-Rating'].reset_index()
num_rating_df.rename(columns={'Book-Rating':'num_ratings'},inplace=True)
num_rating_df
[14] 0.5s
Python
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 Ã?Â?lpiraten. 2
241067 Ã?Â?rger mit Produkt X. Roman. 4
241068 Ã?Â?sterlich leben. 1
241069 Ã?Â?stlich der Berge. 3
241070 Ã?Â?thique en toc 2
241071 rows × 2 columns


index.html
recommend.html
app.py
book.ipynb
Books.csv
books.pkl
G9_Project_Report_...
popular.pkl
pt.pkl
Ratings.csv
similarity_scores.pkl
Users.csv
avg_rating_df = ratings_with_name.groupby('Book-Title')['Book-Rating'].mean().reset_index()
avg_rating_df.rename(columns={'Book-Rating': 'avg_rating'}, inplace=True)
print(avg_rating_df)
[15] 0.2s
Python
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 Ã?Â?lpiraten. 0.000000
241067 Ã?Â?rger mit Produkt X. Roman. 5.250000
241068 Ã?Â?sterlich leben. 7.000000
241069 Ã?Â?stlich der Berge. 2.666667
241070 Ã?Â?thique en toc 4.000000
[241071 rows x 2 columns]
popular_df = num_rating_df.merge(avg_rating_df,on='Book-Title')
popular_df
[16] 0.0s
Python
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


index.html
recommend.html
app.py
book.ipynb
Books.csv
books.pkl
G9_Project_Report_...
popular.pkl
pt.pkl
Ratings.csv
similarity_scores.pkl
Users.csv
popular_df = num_rating_df.merge(avg_rating_df,on='Book-Title')
popular_df
[16] 0.0s
Python
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)
[17] 0.0s
Python
popular_df = popular_df.merge(books,on='Book-Title').drop_duplicates('Book-Title')[['Book-Title','Book-Author','Image-URL-M','num_ratings','avg_rating']]
[18] 0.0s
Python
popular_df['Image-URL-M'][0]
[19] 0.0s
Python
Screenshot



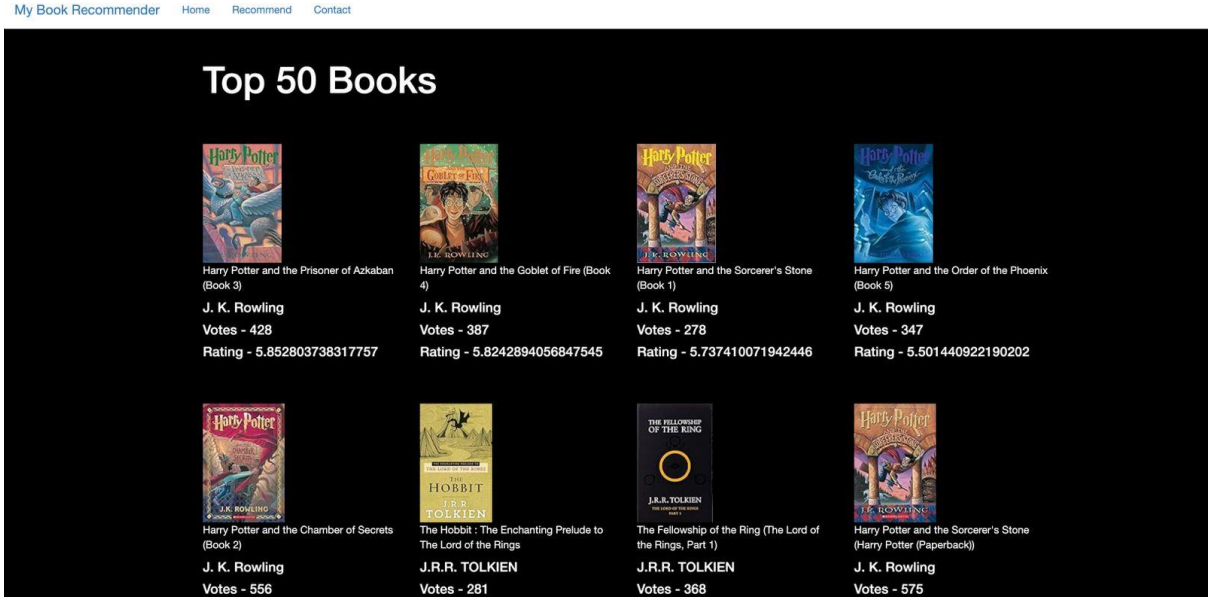


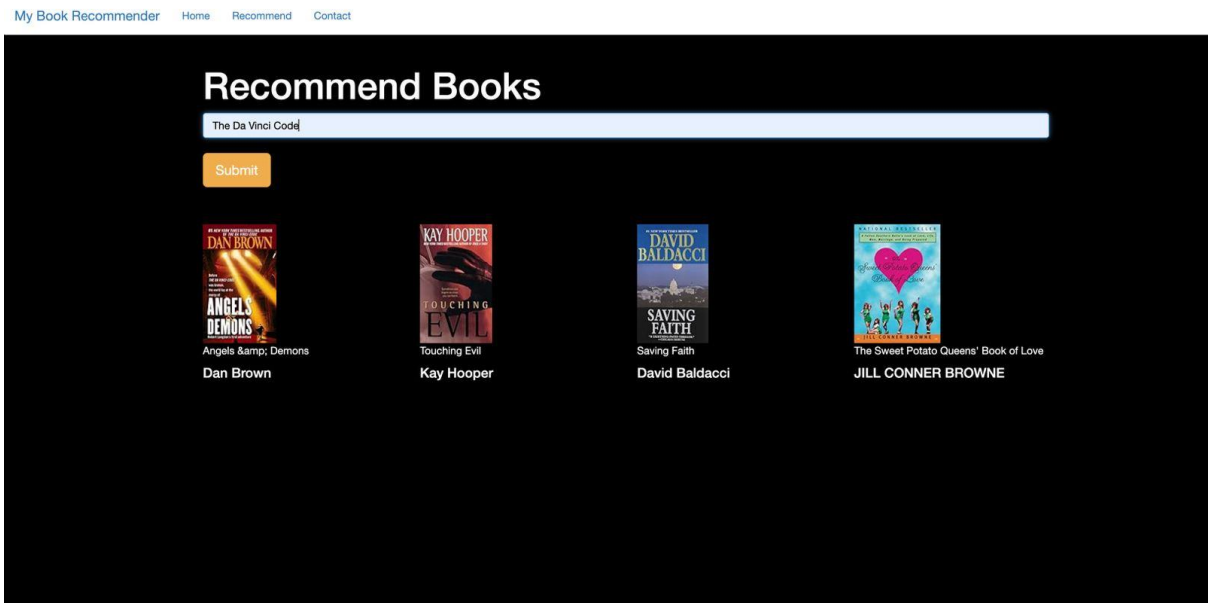
User-ID	254	2276	2766	2977	3363	4017	4385	6251	6323	6543	...	271705	273979	274004	274061	274301	274308	275970	277427	277639	2
Book-Title																					
1984	9.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	...	10.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	
1st to Die: A Novel	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	0.0	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	
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	
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	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...
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	
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	
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	
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\" Is for Outlaw"	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	8.0	0.0	0.0	0.0	0.0	

706 rows × 810 columns


```
def recommend(book_name):
    # index fetch
    index = np.where(pt.index==book_name)[0][0]
    similar_items = sorted(list(enumerate(similarity_scores[index])),key=lambda x:x[1],reverse=True)[1:5]

    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
```

	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.
...	...	...	...	...	...	...	
271354	0449906736	Flashpoints: Promise and Peril in a New World	Robin Wright	1993	Ballantine Books	http://images.amazon.com/images/P/0449906736.0...	http://images.amazon.com/images/P/0449906736.0.
271356	0525447644	From One to One Hundred	Teri Sloat	1991	Dutton Books	http://images.amazon.com/images/P/0525447644.0...	http://images.amazon.com/images/P/0525447644.0.
271357	006008667X	Lily Dale : The True Story of the Town that Ta...	Christine Wicker	2004	HarperSanFrancisco	http://images.amazon.com/images/P/006008667X.0...	http://images.amazon.com/images/P/006008667X.0.
271358	0192126040	Republic (World's Classics)	Plato	1996	Oxford University Press	http://images.amazon.com/images/P/0192126040.0...	http://images.amazon.com/images/P/0192126040.0.
271359	0767409752	A Guided Tour of Rene Descartes' Meditations	Christopher Biffle	2000	McGraw-Hill Humanities/Social Sciences/Languages	http://images.amazon.com/images/P/0767409752.0...	http://images.amazon.com/images/P/0767409752.0.




My Book Recommender
Home
Recommend
Contact
Recommend Books
The Da Vinci Code
Submit
DAN BROWN
ANGELS DEMONS
Angels & Demons
Dan Brown
KAY HOOPER
TOUCHING EVIL
Touching Evil
Kay Hooper
DAVID BALDACCI
SAVING FAITH
Saving Faith
David Baldacci
The Sweet Potato Queens' Book of Love
JILL CONNER BROWNE