Northeastern University, Khoury College of Computer Science

CS 6220 Data Mining - Final Project

Due: April 18, 2024(100 points)

Book Recommendation System

Yu Wang

https://github.com/Titojojo/CS6220-Data-Mining

1 Introduction

The objective of this project is to develop a book recommendation system that can accurately suggest books to users based on user rating histories (collaborative filtering, clustering) and book genre prediction (content-based). The system leverages Book Recommendation Dataset, which contains user demographics, book metadata, and ratings, aiming at recommending the most-likely interested books to the user and providing a personalized reading experience.

With the rise of web services, recommender systems are becoming more and more important in our daily lives. Right now, book recommendations are often made by looking at simple factors like what is popular/trending. This recommendation approach does not dig deep into readers' preferences, often leading to generic suggestions that don't quite hit the mark. This project aims at designing a book recommendation system that digs deeper by using information about users' rating histories and book metadata to make more personalized suggestions. A successful system can help users discover books they really love but might not have found, targeting a better user satisfaction.

2 Background

Recommendation algorithms are widely used in e-commerce websites and e-libraries. The two most popular algorithms are collaborative filtering and cluster models. Other algorithms like the content-recommendation are often used in some cases. This project will explore the following three methods in recommendation systems: collaborative filtering, clustering model, and content-based genre prediction.

Collaborative Filtering:
This method is rooted in the assumption that users who agreed in the past will agree in the future. It aggregates preferences from users with similar rating patterns to make recommendations. Traditional collaborative filtering faces challenges in handling large datasets because it is computationally expensive and typically scales linearly with the number of users and items (Linden, Smith, & York, 2003). Techniques such as dimensionality reduction and clustering can help solve these performance issues by reducing the complexity of user-item matrices.

Clustering Model:
Clustering classifies users into segments based on similarities in their rating behaviors (Xu & Chen, 2005). This approach treats the recommendation as a classification problem. The segments are determined through unsupervised learning algorithms, users in the same cluster are more similar to each other than in other clusters. Without pre-labeled data, we can use clustering algorithms like K-means, hierarchical clustering, and DBSCAN.

Content-Based:
Unlike the other two recommendation methods that rely on user ratings, the content-based method recommends items based on the content of the items. For books, this could mean genres, author styles, or book descriptions (Pazzani & Billsus, 2007). This method is helpful when rating data and user preference data are limited.

3 Approach

3.1 Data and Data Analysis

3.1.1 Understand the Data

The Book Recommendation Dataset provides detailed data in user demographic, book information, and ratings. This provides a solid foundation for developing a personalized book recommendation system. The Book-Crossing dataset comprises 3 files:

- **Books**
  Books dataset contains **271,360** records of data. Books are identified by their respective ISBN. Invalid ISBNs have already been removed from the dataset. Some content-based information is given (Book-Title, Book-Author, Year-Of-Publication, Publisher), obtained from Amazon Web Services. Note that in case of several authors, only the first is provided. URLs linking to cover images are also given, appearing in three different flavors (Image-URL-S, Image-URL-M, Image-URL-L), i.e., small, medium, large. These URLs point to the Amazon web site.

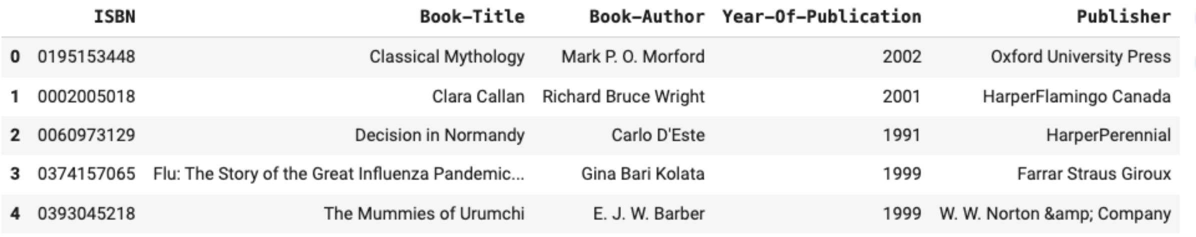
- **Users**
  Users dataset contains **278,858** records of data. User IDs (User-ID) have been anonymized and map to integers. Demographic data is provided (Location, Age) if available. Otherwise, these fields contain NULL-values.

- **Ratings**
  Ratings dataset contains **1,149,780** records of data. The dataset contains the book rating information. Ratings (Book-Rating) are either explicit, expressed on a scale from 1-10 (higher values denoting higher appreciation), or implicit, expressed by 0.

3.1.2 Data Pre-processing

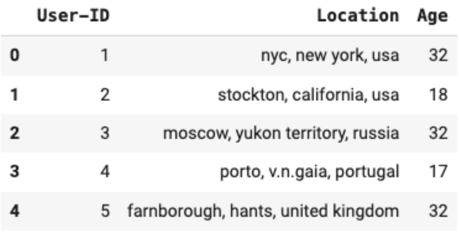
The following operations are performed for data pre-processing to ensure data quality and consistency:

1. **Remove Irrelevant Columns**: Drop columns related to book image URLs, as these are not relevant for our analysis and cannot be processed in the context of our current recommendation algorithms.
2. **Handle Missing Values**: Fill missing values in the 'Author' and 'Publisher' fields using the mode of each respective column.
3. **Adjust Zero-Value Ratings**: Zero ratings indicate implicit ratings rather than explicits rating of zero. Replace zero-ratings with the median of non-zero ratings.
4. **Adjust Missing Ages**: Replace missing values in the 'Age' field with the median age. Filter out entries with non-positive or unrealistically high ages (e.g. people in ages older than 100 and younger than 5 are less likely to make a rating) to ensure the data's realism and relevance.
5. **Handle Invalid Year of Publication**: Remove entries with non-positive years of publication to ensure the data's realism and relevance.
6. **Data Type Conversion**: Convert fields like 'Age', 'Year-Of-Publication', and 'Book-Rating' to integers.
7. **Merge DataFrames**: Combine multiple datasets into a single DataFrame to centralize information during the analysis phase.



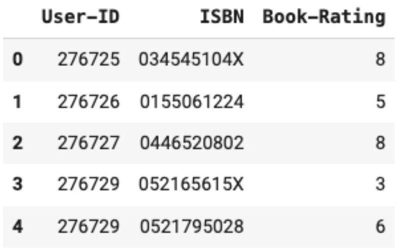
	ISBN	Book-Title	Book-Author	Year-Of-Publication	Publisher
0	0195153448	Classical Mythology	Mark P. O. Morford	2002	Oxford University Press
1	0002005018	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada
2	0060973129	Decision in Normandy	Carlo D'Este	1991	HarperPerennial
3	0374157065	Flu: The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

Figure 3.1: Books



	User-ID	Location	Age
0	1	nyc, new york, usa	32
1	2	stockton, california, usa	18
2	3	moscow, yukon territory, russia	32
3	4	porto, v.n.gaia, portugal	17
4	5	farnborough, hants, united kingdom	32

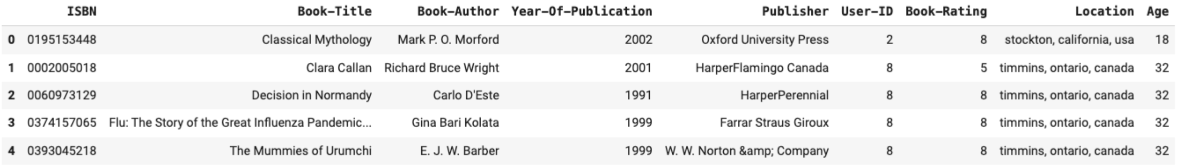
Figure 3.2: Users



	User-ID	ISBN	Book-Rating
0	276725	034545104X	8
1	276726	0155061224	5
2	276727	0446520802	8
3	276729	052165615X	3
4	276729	0521795028	6

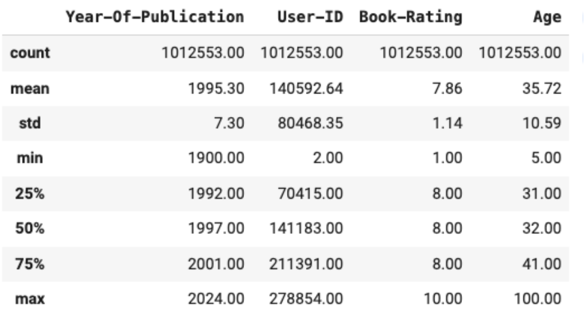
Figure 3.3: Ratings

After merging the datasets, the DataFrame contains the following fields: **ISBN** uniquely identifies each book; **Book-Title** and **Book-Author** provide the title and author's name; **Year-Of-Publication** indicates the year the book was published; **Publisher** provides the publisher name; **User-ID** and the related user demographics represented by **Location** and **Age**; and finally, the **Book-Rating**. We now have a comprehensive dataset that links book information, user demographics, and ratings together.



	ISBN	Book-Title	Book-Author	Year-Of-Publication	Publisher	User-ID	Book-Rating	Location	Age
0	0195153448	Classical Mythology	Mark P. O. Morford	2002	Oxford University Press	2	8	stockton, california, usa	18
1	0002005018	Clara Callan	Richard Bruce Wright	2001	HarperFlamingo Canada	8	5	timmins, ontario, canada	32
2	0060973129	Decision in Normandy	Carlo D'Este	1991	HarperPerennial	8	8	timmins, ontario, canada	32
3	0374157065	Flu: The Story of the Great Influenza Pandemic...	Gina Bari Kolata	1999	Farrar Straus Giroux	8	8	timmins, ontario, canada	32
4	0393045218	The Mummies of Urumchi	E. J. W. Barber	1999	W. W. Norton &amp; Company	8	8	timmins, ontario, canada	32

Figure 3.4: Merged DataFrame



	Year-Of-Publication	User-ID	Book-Rating	Age
count	1012553.00	1012553.00	1012553.00	1012553.00
mean	1995.30	140592.64	7.86	35.72
std	7.30	80468.35	1.14	10.59
min	1900.00	2.00	1.00	5.00
25%	1992.00	70415.00	8.00	31.00
50%	1997.00	141183.00	8.00	32.00
75%	2001.00	211391.00	8.00	41.00
max	2024.00	278854.00	10.00	100.00

Figure 3.5: Overview of the Merged DataFrame

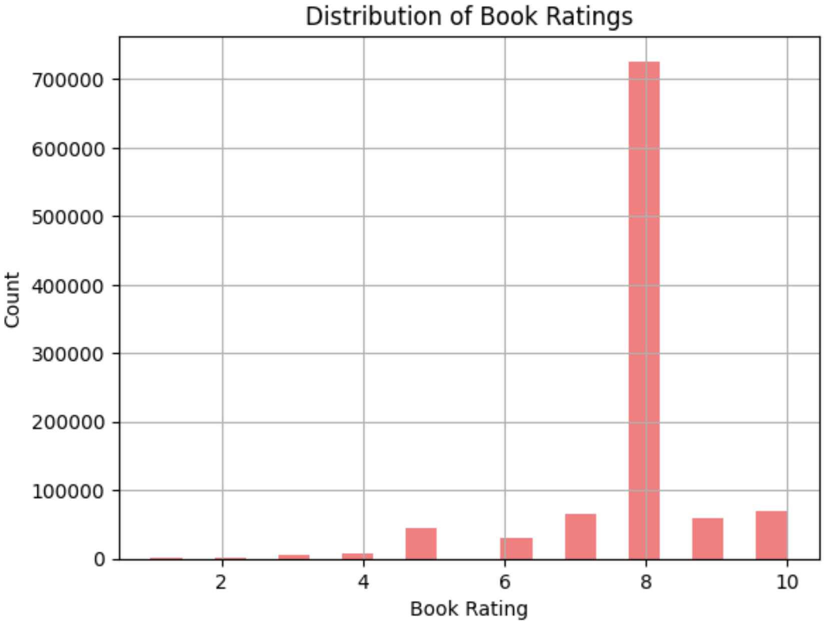
3.1.3 Data Visualization

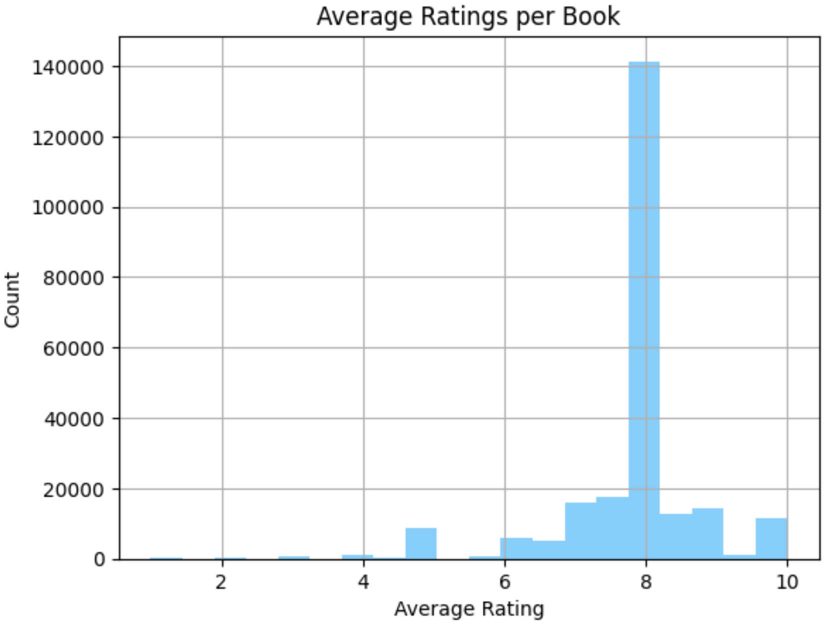
Several insights can been observed through the histogram figures:

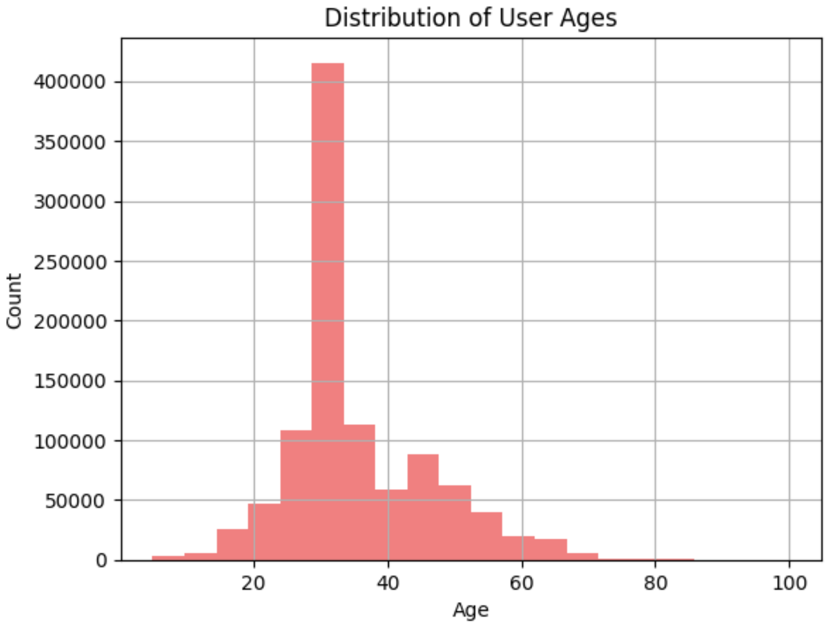
- **Book Ratings**: There is a notably high frequency of the rating 8 that is caused by the pre-processing step where missing ratings were replaced with the median value, which is 8. Excluding the peak at 8, the distribution of book ratings suggests a tendency to rate books on the higher end of the scale.

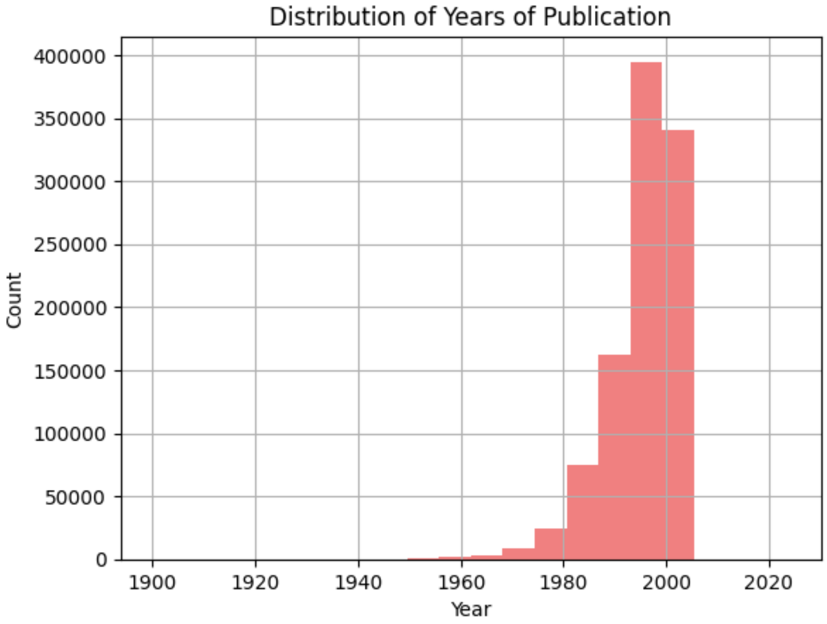
- **Average Ratings per Book**: The distribution of average ratings per book is similar as the distribution of book ratings. Most books have moderate to high average ratings. Books with higher average ratings should be prioritized in recommendation algorithms.

- **User Ages**: There is a notably high frequency of the age 32 that is caused by the pre-processing step where missing ages were replaced with the median value, which is 32. Other than age 32, the remaining age distribution might reflect a diverse user base, mostly between 20 to 55. This age group could potentially be the most active readers or more willing to provide ratings.
- **Years of Publication**: The distribution of the years of publication shows a steady increase in the number of books published over the years, with a sharp rise in 1990s.





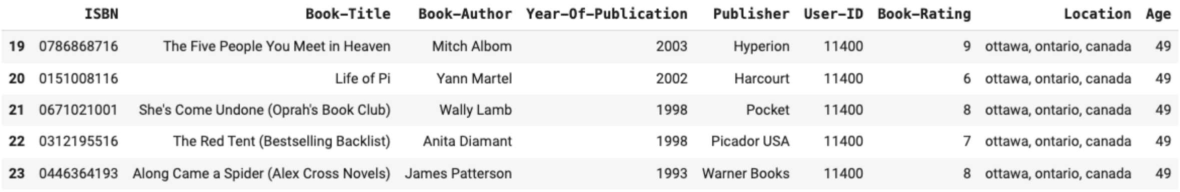




3.1.4 Filter DataFrame for Recommendation

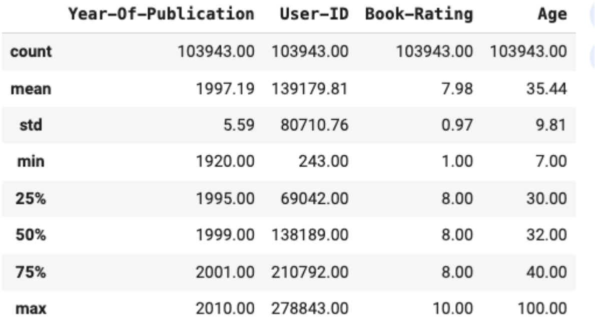
After preprocessing, it can be observed that the DataFrame still remains considerably large, leading to a risk of exhausting RAM resources during analysis. Additionally, books with very few ratings and users who have provided only a limited number of ratings do not offer enough data points to effectively support the recommendation system.

To address these issues, we should further refine our dataset to include only books that have received more than 100 ratings and users who have given more than 50 ratings to ensure we have more reliable data for analysis. The filtered Dataframe has **103,943** pieces of records.



	ISBN	Book-Title	Book-Author	Year-Of-Publication	Publisher	User-ID	Book-Rating	Location	Age
19	0786868716	The Five People You Meet in Heaven	Mitch Albom	2003	Hyperion	11400	9	ottawa, ontario, canada	49
20	0151008116	Life of Pi	Yann Martel	2002	Harcourt	11400	6	ottawa, ontario, canada	49
21	0671021001	She's Come Undone (Oprah's Book Club)	Wally Lamb	1998	Pocket	11400	8	ottawa, ontario, canada	49
22	0312195516	The Red Tent (Bestselling Backlist)	Anita Diamant	1998	Picador USA	11400	7	ottawa, ontario, canada	49
23	0446364193	Along Came a Spider (Alex Cross Novels)	James Patterson	1993	Warner Books	11400	8	ottawa, ontario, canada	49

Figure 3.6: Filtered DataFrame



	Year-Of-Publication	User-ID	Book-Rating	Age
count	103943.00	103943.00	103943.00	103943.00
mean	1997.19	139179.81	7.98	35.44
std	5.59	80710.76	0.97	9.81
min	1920.00	243.00	1.00	7.00
25%	1995.00	69042.00	8.00	30.00
50%	1999.00	138189.00	8.00	32.00
75%	2001.00	210792.00	8.00	40.00
max	2010.00	278843.00	10.00	100.00

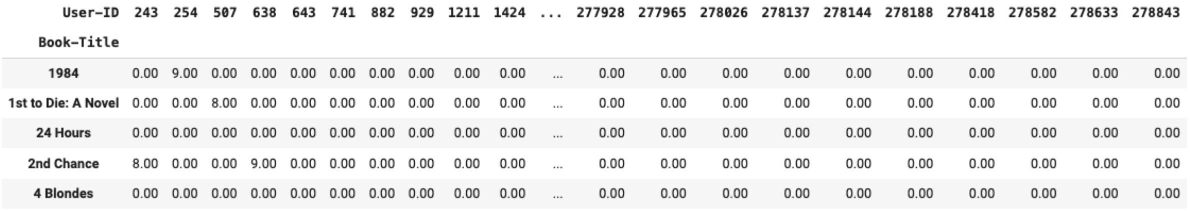
Figure 3.7: Overview of the Filtered DataFrame

3.2 Implementation

3.2.1 Method 1: Collaborative Filtering

Collaborative filtering is a popular technique in recommendation systems. It relies on the assumption that users who have agreed in the past will agree in the future about their preferences.

This method constructs a matrix, with books represented by rows and users by columns, the matrix entries correspond to the ratings that users have given to books. Given the nature of the dataset, most entries in the matrix are missing and we fill the missing values with 0.



User-ID	243	254	507	638	643	741	882	929	1211	1424	...	277928	277965	278026	278137	278144	278188	278418	278582	278633	278843
Book-Title																					
1984	0.00	9.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
1st to Die: A Novel	0.00	0.00	8.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
24 Hours	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
2nd Chance	8.00	0.00	0.00	9.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
4 Blondes	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00

Figure 3.8: Book-User Ratings Matrix

1A: Cosine Similarity Method

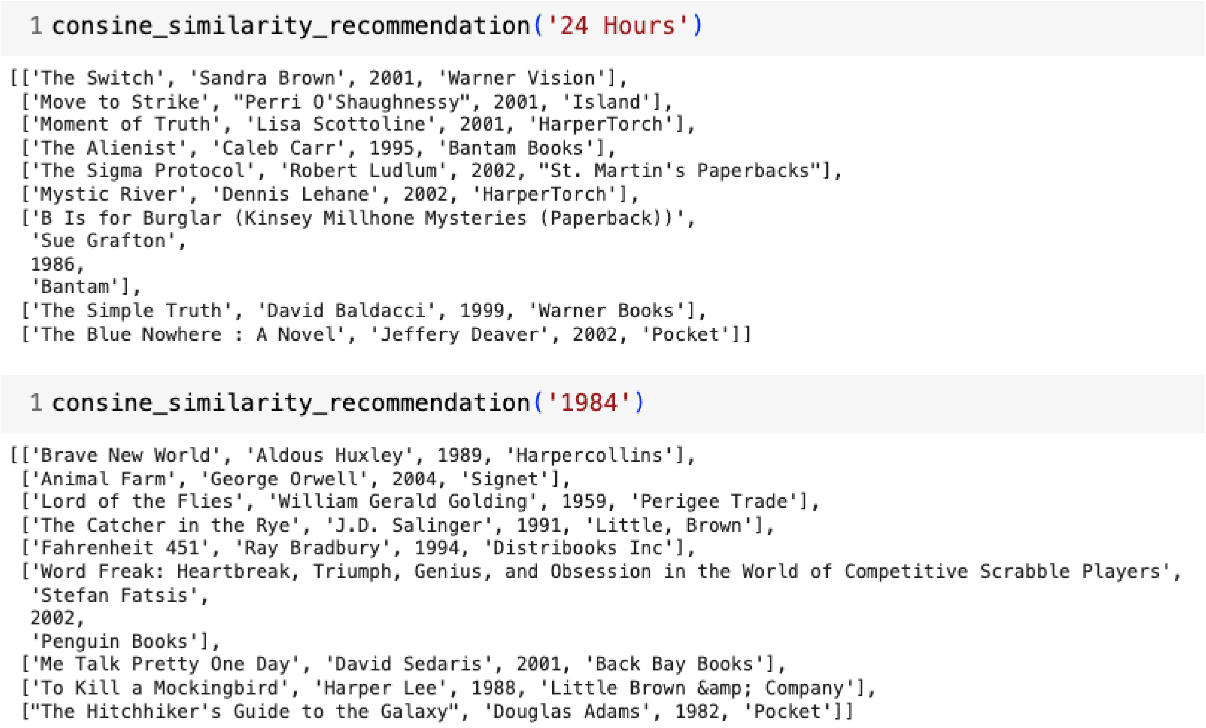
Cosine similarity provides a measure of similarity between two non-zero vectors of an inner product space, it reflects the cosine of the angle between vectors, with a value 1 indicating perfect similarity and a value 0 indicating no similarity. In this method we calculate the cosine similarity between books based on the user ratings, which helps us understand which books are most similar.

The cosine similarity of two vectors A and B can be calculated using the formula:

$$\text{cosine\_similarity}(\mathbf{A}, \mathbf{B}) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\| \|\mathbf{B}\|}$$

Implementation Steps:

1. Construct the matrix: Generate a user-book ratings matrix by pivoting the DataFrame.
2. Compute cosine similarity: Compute cosine similarity between books based on user ratings. Cosine similarity measures how similar the books are in terms of user ratings.
3. Recommendation function: Create the Recommendation function to recommend books based on the cosine similarity scores of the given book title. This function locates a book in the similarity matrix, retrieves the most similar books, and outputs a list of top recommendations to the user.



```
consine_similarity_recommendation('24 Hours')
```

```
[['The Switch', 'Sandra Brown', 2001, 'Warner Vision'],
 ['Move to Strike', "Perri O'Shaughnessy", 2001, 'Island'],
 ['Moment of Truth', 'Lisa Scottoline', 2001, 'HarperTorch'],
 ['The Alienist', 'Caleb Carr', 1995, 'Bantam Books'],
 ['The Sigma Protocol', 'Robert Ludlum', 2002, "St. Martin's Paperbacks"],
 ['Mystic River', 'Dennis Lehane', 2002, 'HarperTorch'],
 ['B Is for Burglar (Kinsey Millhone Mysteries (Paperback))',
  'Sue Grafton',
  1986,
  'Bantam'],
 ['The Simple Truth', 'David Baldacci', 1999, 'Warner Books'],
 ['The Blue Nowhere : A Novel', 'Jeffery Deaver', 2002, 'Pocket']]
```

```
consine_similarity_recommendation('1984')
```

```
[['Brave New World', 'Aldous Huxley', 1989, 'Harpercollins'],
 ['Animal Farm', 'George Orwell', 2004, 'Signet'],
 ['Lord of the Flies', 'William Gerald Golding', 1959, 'Perigee Trade'],
 ['The Catcher in the Rye', 'J.D. Salinger', 1991, 'Little, Brown'],
 ['Fahrenheit 451', 'Ray Bradbury', 1994, 'Distribooks Inc'],
 ['Word Freak: Heartbreak, Triumph, Genius, and Obsession in the World of Competitive Scrabble Players',
  'Stefan Fatsis',
  2002,
  'Penguin Books'],
 ['Me Talk Pretty One Day', 'David Sedaris', 2001, 'Back Bay Books'],
 ['To Kill a Mockingbird', 'Harper Lee', 1988, 'Little Brown &amp; Company'],
 ["The Hitchhiker's Guide to the Galaxy", 'Douglas Adams', 1982, 'Pocket']]
```

Figure 3.9: Example Results of Cosine Similarity Recommendation

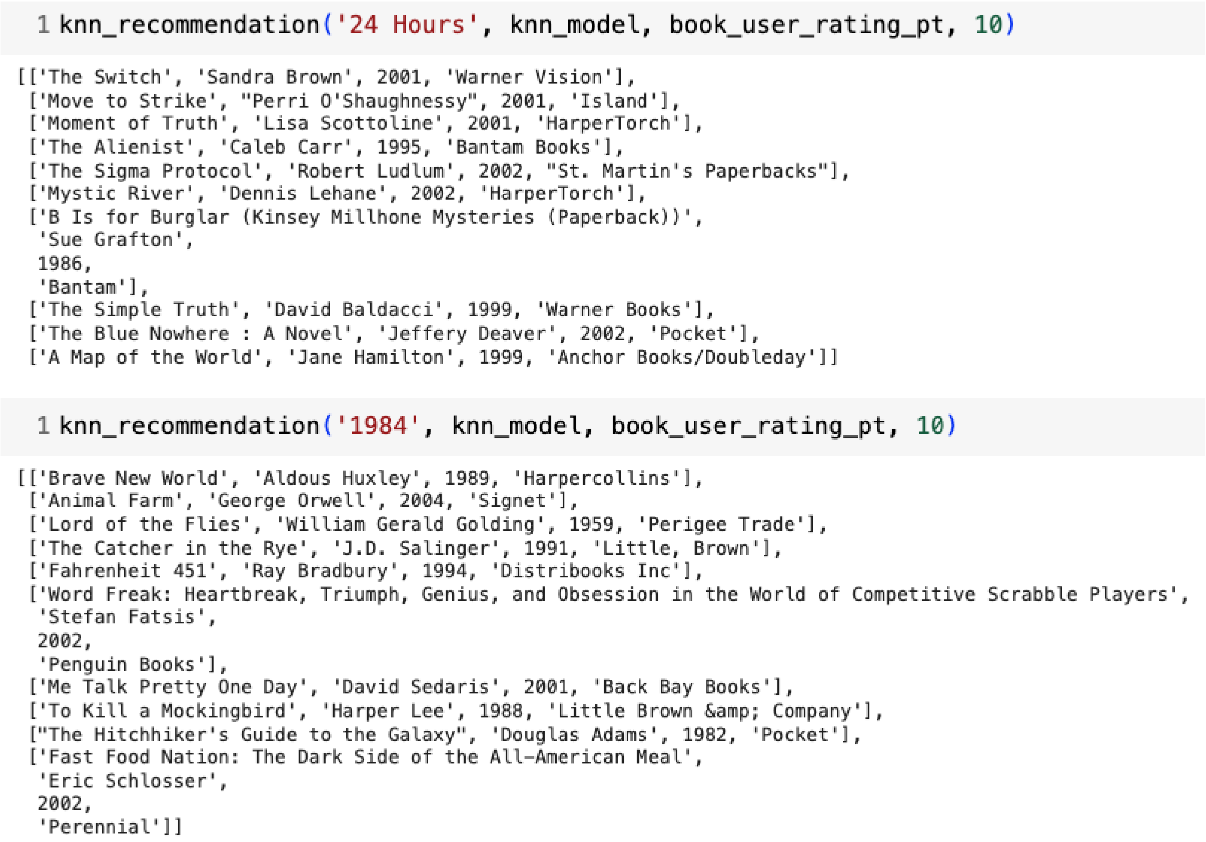
1B: KNN Method

The K-Nearest Neighbors (KNN) is an algorithm used to identify items with the most similarity to a query item. In the context of our Book Recommendation System, it is used to find the k books that have the closest user rating patterns to the given book. The neighbors are selected based on their proximity to the given book using the distance metric defined by the algorithm.

Implementation Steps:

1. Configure the KNN model with the *cosine* metric to calculate the distance to get the nearest neighbors. Set 5 as the number of neighbors to retrieve.

2. Recommendation function: Implement the KNN recommendation function. First, find the index of the given book title, then retrieve the nearest neighbors using the *.kneighbors* method. Outputs a list of top recommendations to the user.



```
knn_recommendation('24 Hours', knn_model, book_user_rating_pt, 10)
```

```
[['The Switch', 'Sandra Brown', 2001, 'Warner Vision'],
 ['Move to Strike', "Perri O'Shaughnessy", 2001, 'Island'],
 ['Moment of Truth', 'Lisa Scottoline', 2001, 'HarperTorch'],
 ['The Alienist', 'Caleb Carr', 1995, 'Bantam Books'],
 ['The Sigma Protocol', 'Robert Ludlum', 2002, "St. Martin's Paperbacks"],
 ['Mystic River', 'Dennis Lehane', 2002, 'HarperTorch'],
 ['B Is for Burglar (Kinsey Millhone Mysteries (Paperback))',
  'Sue Grafton',
  1986,
  'Bantam'],
 ['The Simple Truth', 'David Baldacci', 1999, 'Warner Books'],
 ['The Blue Nowhere : A Novel', 'Jeffery Deaver', 2002, 'Pocket'],
 ['A Map of the World', 'Jane Hamilton', 1999, 'Anchor Books/Doubleday']]
```

```
knn_recommendation('1984', knn_model, book_user_rating_pt, 10)
```

```
[['Brave New World', 'Aldous Huxley', 1989, 'Harpercollins'],
 ['Animal Farm', 'George Orwell', 2004, 'Signet'],
 ['Lord of the Flies', 'William Gerald Golding', 1959, 'Perigee Trade'],
 ['The Catcher in the Rye', 'J.D. Salinger', 1991, 'Little, Brown'],
 ['Fahrenheit 451', 'Ray Bradbury', 1994, 'Distribooks Inc'],
 ['Word Freak: Heartbreak, Triumph, Genius, and Obsession in the World of Competitive Scrabble Players',
  'Stefan Fatsis',
  2002,
  'Penguin Books'],
 ['Me Talk Pretty One Day', 'David Sedaris', 2001, 'Back Bay Books'],
 ['To Kill a Mockingbird', 'Harper Lee', 1988, 'Little Brown &amp; Company'],
 ["The Hitchhiker's Guide to the Galaxy", 'Douglas Adams', 1982, 'Pocket'],
 ['Fast Food Nation: The Dark Side of the All-American Meal',
  'Eric Schlosser',
  2002,
  'Perennial']]
```

Figure 3.10: Example Results of KNN Recommendation

3.2.2 Method 2: Clustering

The clustering algorithm categorizes users into segments based on their book ratings. This approach treats the recommendation as a classification problem. Users in the same cluster share similar preferences and behaviors, and are more alike to each other than in other clusters. By utilizing the clustering algorithm, the system can recommend books that resonate with the specific interests of each user group.

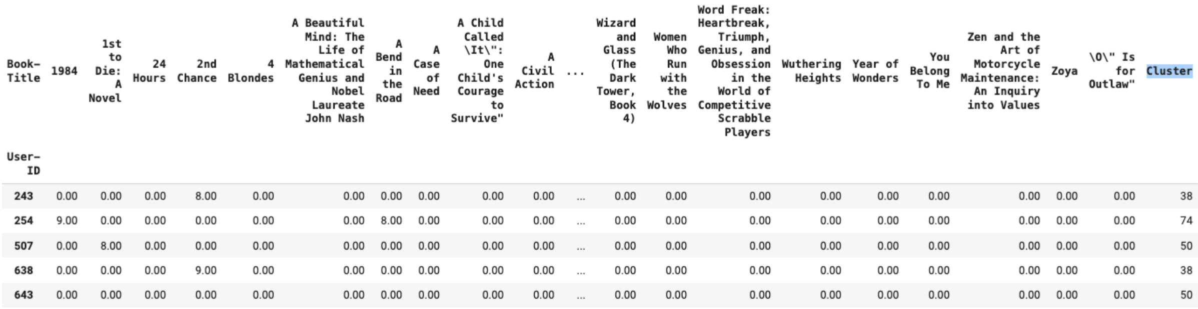
Implementation Steps:

1. User-Book Ratings matrix: Constructs a matrix, with users represented by rows and books by columns, the matrix entries correspond to the ratings that users have given to books. Fill the missing values with 0. Generate a user-book ratings matrix by pivoting the DataFrame.

2. K-Means clustering: Segment the users into 100 clusters using the K-Means algorithm.

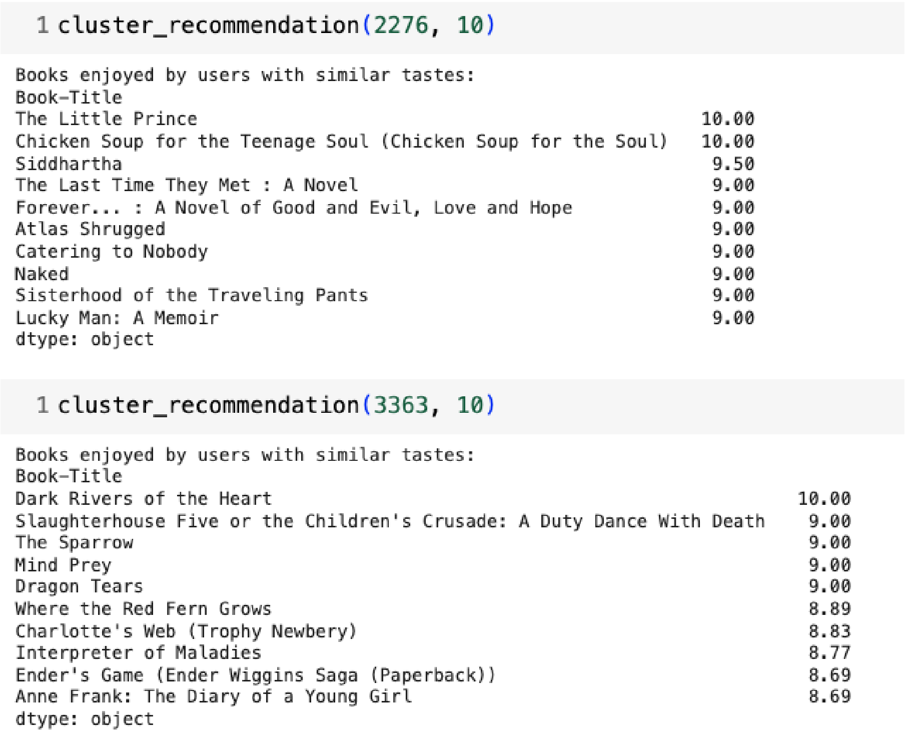
3. Recommendation function: Implement a function to recommend books to a given user based on the preferences within their user cluster. This function locates the cluster of the

given user, computes the average ratings for books within this cluster, and excludes books already rated by the user. And then recommend the top-rated books within the cluster to the user.



Book-Title	1984	1st to Die: A Novel	24 Hours	2nd Chance	4 Blondes	A Beautiful Mind: The Life of Mathematical Genius and Nobel Laureate John Nash	A Bend in the Road	A Case of Need	A Child Called \It\": One Child's Courage to Survive"	A Civil Action	...	Wizard and Glass (The Dark Tower, Book 4)	Women Who Run with the Wolves	Word Freak: Heartbreak, Triumph, Genius, and Obsession in the World of Competitive Scrabble Players	Wuthering Heights	Year of Wonders	You Belong To Me	Zen and the Art of Motorcycle Maintenance: An Inquiry into Values	Zoya	\O\" Is for Outlaw"	Cluster
User-ID																					
243	0.00	0.00	0.00	8.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	38
254	9.00	0.00	0.00	0.00	0.00	0.00	8.00	0.00	0.00	0.00	...	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	74
507	0.00	8.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	50
638	0.00	0.00	0.00	9.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	38
643	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	...	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	50

Figure 3.11: User-Book Ratings Matrix, with Cluster on the rightmost column



```
cluster_recommendation(2276, 10)
```

```
Books enjoyed by users with similar tastes:
Book-Title
The Little Prince                                                  10.00
Chicken Soup for the Teenage Soul (Chicken Soup for the Soul)     10.00
Siddhartha                                                          9.50
The Last Time They Met : A Novel                                    9.00
Forever... : A Novel of Good and Evil, Love and Hope                9.00
Atlas Shrugged                                                      9.00
Catering to Nobody                                                  9.00
Naked                                                               9.00
Sisterhood of the Traveling Pants                                   9.00
Lucky Man: A Memoir                                                 9.00
dtype: object
```

```
cluster_recommendation(3363, 10)
```

```
Books enjoyed by users with similar tastes:
Book-Title
Dark Rivers of the Heart                                                    10.00
Slaughterhouse Five or the Children's Crusade: A Duty Dance With Death      9.00
The Sparrow                                                                  9.00
Mind Prey                                                                    9.00
Dragon Tears                                                                 9.00
Where the Red Fern Grows                                                     8.89
Charlotte's Web (Trophy Newbery)                                             8.83
Interpreter of Maladies                                                      8.77
Ender's Game (Ender Wiggins Saga (Paperback))                                8.69
Anne Frank: The Diary of a Young Girl                                        8.69
dtype: object
```

Figure 3.12: Example Results of Clustering Recommendation

3.2.3 Method 3: Content-Based Using Word2Vec

The content-based approach uses the Word2Vec model to analyze semantic relationships between words in book titles, and map each book to a genre. This approach cleans and tokenizes book titles, then by training the Word2Vec model on the book titles text data, it generates word embeddings that capture the essence of each book's thematic content, and predicts the most similar genre of each book. The model then recommends the top 10 highest-rated books within the genre for a given book.

Implementation Steps:

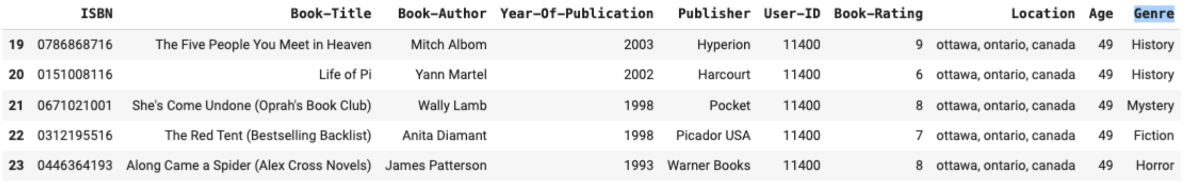
1. Pre-define a list of book genres. This list will be used to train the Word2Vec model and help identify the genre of book titles based on the word content.

2. Preprocess and prepare data: Clean and split the book titles into words, and convert genre labels into lowercase to maintain consistency in the training data. Prepare words for training.

3. Train the Word2Vec model on the aggregated words. The model learns to represent words as vectors and words with similar meanings are positioned closely in the vector space.

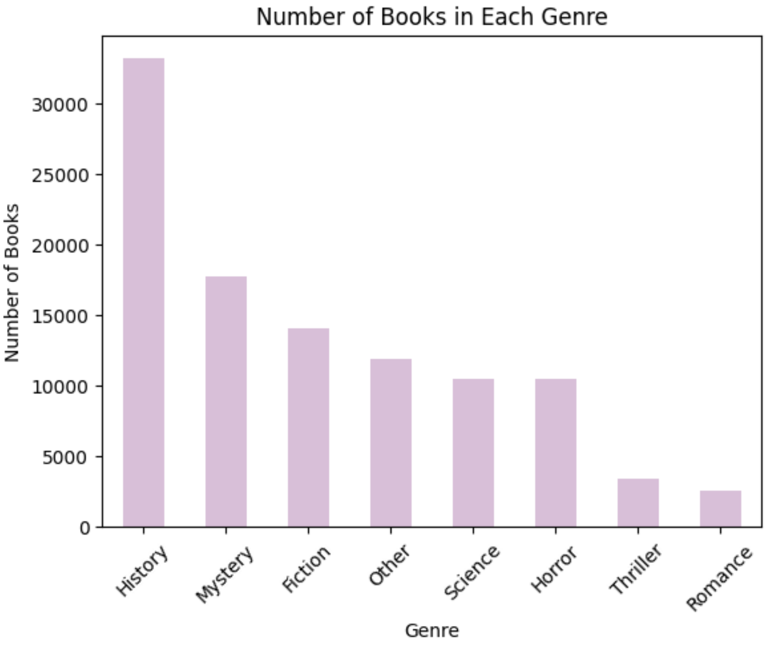
4. Genre prediction: Implement a function to predict the genre of a book based on its title string. The function cleans the title, converts it into vectors using the Word2Vec model, and compares the average vector with the genre vectors to find the predicted genre. Apply the function and get all books predicted.

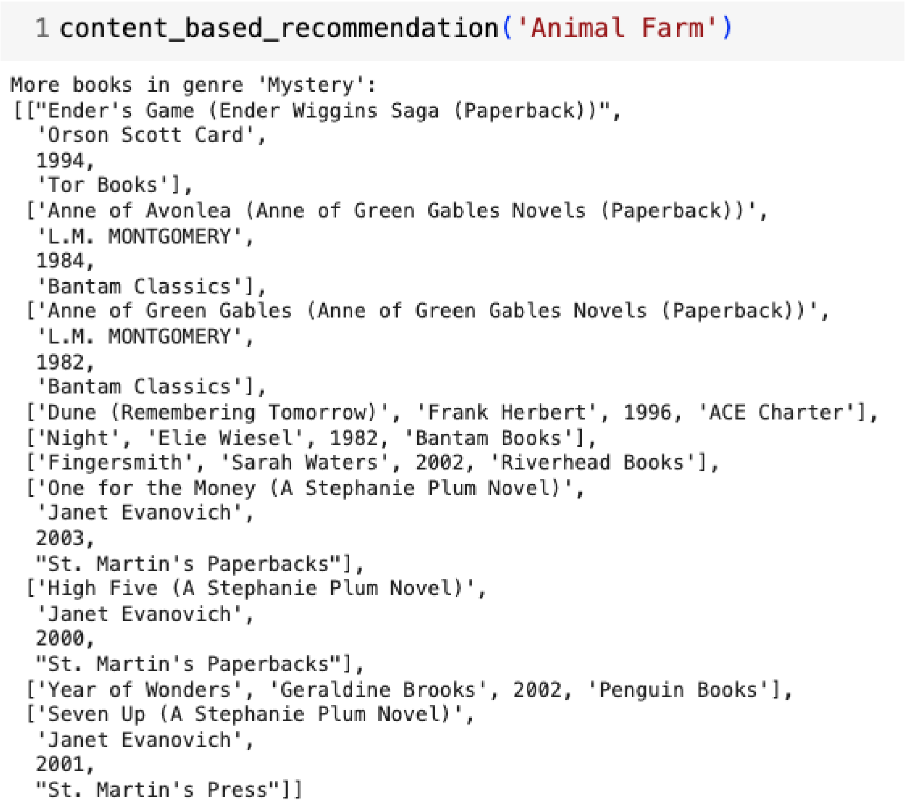
5. Recommendation function: Implement a recommendation function that recommends books within the same genre as the given book based on average user ratings. This function locates a specified book, identifies its genre, and then recommends the top 10 highly rated books within that genre.



	ISBN	Book-Title	Book-Author	Year-Of-Publication	Publisher	User-ID	Book-Rating	Location	Age	Genre
19	0786868716	The Five People You Meet in Heaven	Mitch Albom	2003	Hyperion	11400	9	ottawa, ontario, canada	49	History
20	0151008116	Life of Pi	Yann Martel	2002	Harcourt	11400	6	ottawa, ontario, canada	49	History
21	0671021001	She's Come Undone (Oprah's Book Club)	Wally Lamb	1998	Pocket	11400	8	ottawa, ontario, canada	49	Mystery
22	0312195516	The Red Tent (Bestselling Backlist)	Anita Diamant	1998	Picador USA	11400	7	ottawa, ontario, canada	49	Fiction
23	0446364193	Along Came a Spider (Alex Cross Novels)	James Patterson	1993	Warner Books	11400	8	ottawa, ontario, canada	49	Horror

Figure 3.13: DataFrame with predicted Genre on the rightmost column





```
content_based_recommendation('Animal Farm')
```

```
More books in genre 'Mystery':
[["Ender's Game (Ender Wiggins Saga (Paperback))",
  'Orson Scott Card',
  1994,
  'Tor Books'],
 ['Anne of Avonlea (Anne of Green Gables Novels (Paperback))',
  'L.M. MONTGOMERY',
  1984,
  'Bantam Classics'],
 ['Anne of Green Gables (Anne of Green Gables Novels (Paperback))',
  'L.M. MONTGOMERY',
  1982,
  'Bantam Classics'],
 ['Dune (Remembering Tomorrow)', 'Frank Herbert', 1996, 'ACE Charter'],
 ['Night', 'Elie Wiesel', 1982, 'Bantam Books'],
 ['Fingersmith', 'Sarah Waters', 2002, 'Riverhead Books'],
 ['One for the Money (A Stephanie Plum Novel)',
  'Janet Evanovich',
  2003,
  "St. Martin's Paperbacks"],
 ['High Five (A Stephanie Plum Novel)',
  'Janet Evanovich',
  2000,
  "St. Martin's Paperbacks"],
 ['Year of Wonders', 'Geraldine Brooks', 2002, 'Penguin Books'],
 ['Seven Up (A Stephanie Plum Novel)',
  'Janet Evanovich',
  2001,
  "St. Martin's Press"]]
```

Figure 3.14: Example Results of Content-Based Recommendation

4 Results and Evaluation

Comparison of Methods:

- **Collaborative Filtering (Cosine Similarity and KNN)**: These approaches show promising results due to their reliance on user interaction data. Both of the two approaches produce similar outcomes. The collaborative filtering approach is widely used and is helpful when recommending other books given a specific book.

- **Clustering**: This approach is effective in segmenting users into groups for taste-oriented recommendations. It is useful in scenarios where the user id is given, without looking into a specific book, the system can still make some recommendations based on the user's taste.

- **Content-Based Filtering using Word2Vec**: This approach uses semantic analysis to predict genres from book titles, but it faces challenges due to limited contextual data. The reliance solely on book titles constrained the model's understanding, the performance is not as good as methods that analyze user ratings. It would be helpful if some pre-given book-genres data are available as a training set. However, it is still a valuable exploratory tool for genre prediction when genre data is unavailable.

5 Conclusions

This project explored the different mechanics and applications of various recommendation system algorithms. The accuracy of recommendation systems highly depends on the available data inputs. When designing recommendation systems, we should wisely choose suitable algorithms based on the data at hand.

Given the Book Recommendation Dataset, the Content-Based method struggled with limited data, but it showcased the potential of natural language processing in identifying book genres, which could be further enhanced if we have additional data sources.

Collaborative Filtering and Clustering has strong potential for real-world applications. These approaches could also be further scaled and refined to serve larger datasets or different domains beyond book recommendations.

6 References

- Item-Based Collaborative Filtering Recommendation Algorithms, Sarwar, Karypis, Konstan, and Riedl (2001)
- Matrix Factorization Techniques for Recommender Systems, Koren, Bell, and Volinsky (2009)
- Content-Based Recommendation Systems, Pazzani and Billsus (2007)
- Toward the Next Generation of Recommender Systems: A Survey of the State-of-the-Art and Possible Extensions, Adomavicius and Tuzhilin (2005)
- Recommender Systems Handbook, Rokach, Shapira, and Kantor (2011)
- Amazon.com recommendations: item-to-item collaborative filtering, Linden, Smith, and York (2003)
- Deep Learning based Recommender System: A Survey and New Perspectives, Shuai Zhang, Lina Yao, Anxin Sun, and Yi Tay (2019)