Anime Recommendation System

Submitted in partial fulfilment of the requirements of the degree

BACHELOR OF ENGINEERING IN **COMPUTER ENGINEERING**

By

1. **Mr. Ayser Ahmed Bijapur 312009**

Name of the Guide
Asst. Prof. Ahlam Ansari

Department of Computer Engineering

AII’s M H Saboo Siddik College of Engineering

8, Saboo Siddik Polytechnic Road, Byculla – 400 008

University of Mumbai

(AY 2023-24

CERTIFICATE

This is to certify that the Mini Project entitled “**Movie Recommendation System**” is a bonafide work of **Attarwala Murtuza Suber (312008); Khan Daniyal Mushtaque (312012); Khan Akbar Nafees (312020)** submitted to the University of Mumbai in partial fulfilment of the requirement for the award of the degree of “**Bachelor of Engineering**” in “**Computer Engineering**”.

Asst. Prof. Ahlam Ansari *(Internal Guide)*	*(External Guide)*
(Internal Examiner)	*(External Guide)*
Dr Mohammed Ahmed S. *(Head of Department)*	Dr Javed Habib *(Principal)*

Mini Project Approval

This Mini Project entitled "**Anime Recommendation System**" is a bonafide work of **Ayser Ahmed Bijapur (312009)** is approved for the degree of **Bachelor of Engineering** in **Computer Engineering**.

Examiners

1. ……………………………………

(Internal Examiner Name & Sign)

2. ……………………………………

(External Examiner Name & Sign)

Supervisors

1. ……………………………………

(Internal Supervisor Name & Sign)

2. ……………………………………

(External Supervisor Name & Sign)

Chairman

……………………………………

Date:

Place: **Mumbai**

DECLARATION

I declare that this written submission represents my ideas in my own words and where others' ideas or words have been included, I have adequately cited and referenced the original sources. I also declare that I have adhered to all principles of academic honesty and integrity and have not misrepresented fabricated or falsified any idea/data/fact/source in my submission. I understand that any violation of the above will be cause for disciplinary action by the Institute and can also evoke penal action from the sources that have thus not been properly cited or from whom proper permission has not been taken when needed.

Mr. Ayser Ahmed Bijapur -312009

Abstract

The online recommendation system has become a trend. Now a days rather than going out and buying items for themselves, reason being, online recommendation provides an easier and quicker way to buy items and transactions are also quick when it is done online. Recommended systems are powerful new technology and it helps users to find items which they want to buy. A recommendation system is broadly used to recommend products to the end users that are most appropriate. This paper presents Anime Recommendation System (BRS) based on combined features of content based filtering (CBF), collaborative filtering (CF) and association rule mining to produce efficient and effective recommendation. For this we are proposing a hybrid algorithm in which we combine two or more algorithms, so it helps the recommendation system to recommend the Anime based on the viewer's interest.

Anime recommendation system has been developed rapidly due to the Web technology and library modernization, which provide a new way for the viewers to acquire the anime demands. However, existing recommendation systems can't supply enough information for readers to decide whether to recommend a book or not, and they don't analyse the recommendation information.

Some systems also lack of a feedback mechanism for viewers, which would hurt their enthusiasm. In order to solve these problems, we designed a anime recommendation system. Viewers will be redirected to the recommendation pages when they can't find the required anime through the library bibliographic retrieval system. The recommendation pages contain all the essential and expanding anime information for readers to refer to. Viewers can recommend an anime on these pages, and the recommendation data will be analysed by the recommendation system to make scientific purchasing decision. We proposed two formulas to compute the book value and copy number respectively based on the recommendation data. The application of the recommendation system shows that both the recommended book utilization and readers' satisfaction were greatly increased.

List Of Figures

Table Of Content

Chapter 1

1. Introduction

An anime recommendation system is a technology or platform designed to assist readers in discovering books that align with their interests, preferences, and reading habits. It leverages algorithms, user data, and various content-based and collaborative filtering techniques to suggest relevant books to individuals. These systems are widely used in online bookstores, libraries, and digital reading platforms to enhance user engagement, increase book sales, and improve the overall reading experience.

This data set contains information on user preference data from 73,516 users on 12,294 anime. Each user is able to add anime to their completed list and give it a rating and this data set is a compilation of those ratings. The data was scraped thanks to myanimelist.net API.

The goal of data mining is to analyse process and extract knowledge from data in the context of large databases by using different data mining methods and techniques. A large number of data are available in the Information industry, all these data are no use until unless converted into meaningful or useful information.

So, it is necessary for analyse this large amount of data and extract useful information from it. It allows users to utilize data from many different dimensions. Recommendation systems are the subordinate class of information filtering system which helps to recommend a particular item to the user. Recommendation applied in a variety of applications like movies, music, news, books and products, etc...

In general recommendation systems produce a list of recommendations in different ways - through various technologies like content-based filtering and collaborative filtering. In the case of collaborative filtering approach, build the model from various aspects like users past behavior, which includes items purchased by the user previously as well as the rating given by the users for a particular item. This approach helps to predict the interesting item that the user may have. Content based filtering works based on the description or content of a particular item

Purpose: Book recommendation systems are developed to address the growing challenge of information overload in the digital age. With an abundance of books available in various genres and styles, readers often find it overwhelming to choose what to read next. Recommendation systems aim to simplify this process by providing personalized suggestions based on a user's unique tastes and preferences.

Types of Recommendation Systems:

- Content-Based Filtering: This method recommends books by analyzing the content and attributes of books, such as genre, author, keywords, and other textual data. It suggests books similar to those the user has shown an interest in.
- Collaborative Filtering: Collaborative filtering recommends books based on user behavior and interactions. It identifies users with similar reading patterns and suggests books that people with similar tastes have enjoyed.
- Hybrid Systems: Many modern recommendation systems combine both content-based and collaborative filtering approaches to provide more accurate and diverse recommendations.

Data Sources:

- User Data: This includes data on a user's past reading history, ratings, and reviews. It helps the system understand the individual's reading preferences.
- Book Data: Information about books, such as genre, author, publication date, and other metadata, is used to categorize and recommend books.
- Interactions: Data on user-book interactions, such as clicks, views, purchases, and ratings, is vital for collaborative filtering.

Algorithm Complexity: Recommendation systems use complex algorithms, including machine learning models, neural networks, and natural language processing, to make predictions. These algorithms continuously learn and adapt as more user data becomes available.

Challenges:

- Cold Start Problem: Recommending books to new users who have little to no interaction history can be challenging.
- Data Sparsity: In many cases, user-item interaction data is sparse, making it difficult to generate accurate recommendations.
- Scalability: For large-scale platforms, scalability and real-time recommendation are crucial.

Importance of Book Recommendation System: -

Book recommendation systems hold significant importance in various aspects of the book industry and for readers. Their value extends to both consumers and businesses, as they provide numerous benefits, as outlined below:

- Enhancing User Experience: Book recommendation systems improve the user experience by making it easier for readers to discover new books that match their interests and preferences. This personalization enhances user engagement and satisfaction with book-related platforms.

- Reducing Information Overload: In an era of information overload, recommendation systems help users navigate the vast sea of available books. They filter out irrelevant or less appealing options, saving users time and effort in their search for the next great read.

- Increasing Book Sales: For book retailers, recommendation systems have a direct impact on sales. By suggesting books that users are more likely to buy, these systems boost revenue and encourage repeat business.

- Cross-Selling and Upselling: Recommendation systems can promote related books, authors, or genres, encouraging users to explore a wider range of content. This cross-selling and upselling strategy can lead to increased purchases and diversified reading habits.

- Improved Content Discovery: They help readers discover books they might not have found on their own. This leads to a more diverse reading experience and exposes users to new authors, genres, and perspectives.

- User Engagement: Personalized book recommendations keep users engaged with book-related platforms, encouraging them to spend more time on these websites or apps. This can lead to increased user loyalty and higher retention rates.

Chapter 2

Aim & Objective

1.1 Aim

The aim of our project, the Book Recommendation System, is to provide users with a personalized and seamless experience in discovering and enjoying books tailored to their individual preferences. By suggesting relevant books, the system aims to keep users engaged with book-related platforms, encouraging them to explore and discover more content. By leveraging advanced algorithms and user behavior analysis, our system will offer insightful recommendations based on factors such as genre preferences, author and user ratings. Users will have the convenience of accessing a wide range of books options, ensuring they find content that aligns with their tastes and interests. This system will enhance the overall good experience, making it easy for users to discover new books while ensuring they continue to enjoy their favorite genres.

1.2 Objective

The objectives of our Book Recommendation System project are as follows:

a) The primary objective is to provide readers with a personalized and enjoyable experience by recommending books that align with their individual tastes and preferences.

b) Enhance user convenience by enabling them to discover and select books from the comfort of their own homes.

c) Offer a diverse and extensive selection of books across various genres and catogory to cater to a wide range of tastes and preferences.

Chapter 3

Problem Statement

In this project, we aim to develop a Book Recommendation System, a technology-driven tool designed to provide personalized Books suggestions to users based on their interest, reading habit and preferences. The system will operate through a three-step process: data collection, user profiling, and recommendation generation. Initially, data on books and user interactions will be gathered. Subsequently, user profiles will be established, summarizing their preferences and behaviours. Finally, recommendations will be generated by comparing a user's profile with others and suggesting books that align with their tastes.

The recommendation system will employ various techniques, including collaborative filtering, content-based filtering, and hybrid approaches, to enhance its accuracy and effectiveness. Similar to popular e-commerce platforms like flipcart and Amazon , our system aims to enhance user engagement and satisfaction by delivering relevant books. By leveraging data and algorithms, the Book Recommendation System aims to streamline the process of finding books to read, providing users with tailored suggestions for a more convenient and enjoyable reading experience.

Chapter 4

Scope

The Book Recommendation System project aims to revolutionize the way users discover books. This technology-driven tool will employ sophisticated algorithms to analyze user data, including book reviews, genres, and reading habits, in order to generate personalized book recommendations.

The project will consist of three main components: data collection, user profiling, and recommendation generation. Initially, data on books and user interactions will be gathered. Subsequently, user profiles will be created to encapsulate their preferences and behaviours. Finally, recommendations will be generated by comparing a user's profile with others and suggesting books that align with their tastes. By incorporating various recommendation techniques such as collaborative filtering, content-based filtering, and hybrid approaches, the system will deliver tailored books suggestions. Users will benefit from a seamless experience, making the process of finding books to read more convenient than ever before.

Chapter 5

Proposed System

Proposed system helps to increase the accuracy, scalability and also provide efficient book recommendation to the user. To achieving these features through the proposed algorithm and methodology, first we need to identify whole books available in the dataset by scanning the dataset. From the scanned dataset perform data pre-processing, which includes the extraction of data that are needed for mining.

From the extracted data perform filtering of transactions here we categorize the books based on category and subcategories. For example if we consider the book category as computer science, which contain various subcategory like programming languages(c, c++, java etc...) and different subjects (operating system, data mining, software engineering etc...).Each subcategories have different kind of books written by different authors. From the identified book perform content based filtering and collaborative filtering which leads to the final recommendation of the book to the users.

Steps involved in the algorithm,

- Step 1: Scan the Books Dataset In this step scan the entire storage server and simultaneously perform the data cleaning, which include removing the irrelevant data and keeping the relevant data for mining.

- Step 2: Data Preprocessing According to our application, it includes the extraction of data that are needed for mining, which means extraction of only book categories and subcategories.

- Step 3: Filtering Transactions For filtering the transactions categorize the book based on category and subcategories.

- Step 4: Perform Content based Filtering In this step we need to perform content based filtering of books according to user preferences. for example, User1 clicked on book B1,Assume that we have some related books B2, B3 and B4 in the dataset. Assume B2 is of different type, but B3 and B4 is of same type of book B1. Now we check the content of the books B3 and B4, if the contents match with book kB1, then the system will recommend books B3 and B4 for the user. If user clicks on book B1, then the user will get books B3 and B4 as the recommended.

- Step 5: Perform Collaborative Filtering Here we consider the quality of the book content. In our example, recommending the books B3 and B4. This will perform based on the registered user's opinions and rating.

- Step 6: Final Recommendations In the final recommendation, the order of the book is consider based on the book which holds highest rating compare to remaining books and arrange in descending order

Chapter 6

Project Requirements

4.1 Hardware Requirements

a) Laptop/Desktop
b) Processor (Intel Core i3 or AMD Ryzen 3)
c) RAM: A minimum of 8GB RAM is recommended.
d) Storage: A solid-state drive (SSD) is preferable for faster data access and model training.

4.2 Software Requirements

a) Python Libraries
b) Machine learning Models
c) Jupyter Notebooks
d) Operating system such as Windows 10, 11 (32-bit and 64-bit), Ubuntu

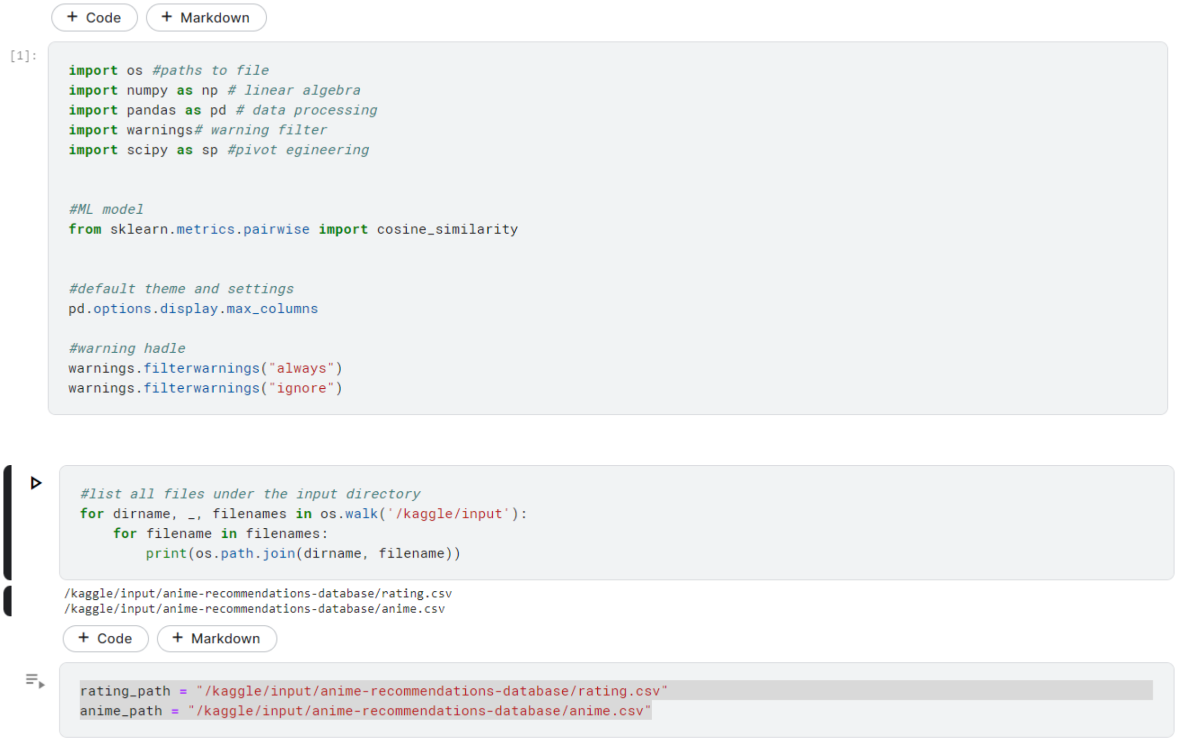
Chapter 7

Code

```
import os #paths to file
import numpy as np # linear algebra
import pandas as pd # data processing
import warnings# warning filter
import scipy as sp #pivot egineering
from sklearn.metrics.pairwise import cosine_similarity

pd.options.display.max_columns
warnings.filterwarnings("always")
warnings.filterwarnings("ignore")

for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
rating_path = "/kaggle/input/anime-recommendations-database/rating.csv"
anime_path = "/kaggle/input/anime-recommendations-database/anime.csv"
rating_df = pd.read_csv(rating_path)
rating_df.head()
anime_df = pd.read_csv(anime_path)
anime_df.head()
print(f"anime set (row, col): {anime_df.shape}\n\nrating set (row, col): {rating_df.shape}")
print("Anime:\n")
print(anime_df.info())
print("\n","*"*50,"\nRating:\n")
print(rating_df.info())
print("Anime missing values (%):\n")
print(round(anime_df.isnull().sum().sort_values(ascending=False)/len(anime_df.index),4)*10
0)
print("\n","*"*50,"\n\nRating missing values (%):\n")
print(round(rating_df.isnull().sum().sort_values(ascending=False)/len(rating_df.index),4)*100)
print(anime_df['type'].mode())
print(anime_df['genre'].mode())
# deleting anime with 0 rating
anime_df=anime_df[~np.isnan(anime_df["rating"])]
# filling mode value for genre and type
anime_df['genre'] = anime_df['genre'].fillna(
anime_df['genre'].dropna().mode().values[0])
anime_df['type'] = anime_df['type'].fillna(
anime_df['type'].dropna().mode().values[0])
#checking if all null values are filled
anime_df.isnull().sum()
rating_df['rating'] = rating_df['rating'].apply(lambda x: np.nan if x==-1 else x)
rating_df.head(20)
anime_df = anime_df[anime_df['type']=='TV']
rated_anime = rating_df.merge(anime_df, left_on = 'anime_id', right_on = 'anime_id', suffixes= ['_user', "])
rated_anime =rated_anime[['user_id', 'name', 'rating']]
rated_anime_7500= rated_anime[rated_anime.user_id <= 7500]
rated_anime_7500.head()
pivot = rated_anime_7500.pivot_table(index=['user_id'], columns=['name'], values='rating')
pivot.head()
pivot_n = pivot.apply(lambda x: (x-np.mean(x))/(np.max(x)-np.min(x)), axis=1)
pivot_n.fillna(0, inplace=True)
pivot_n = pivot_n.T
pivot_n = pivot_n.loc[:, (pivot_n != 0).any(axis=0)]
piv_sparse = sp.sparse.csr_matrix(pivot_n.values)
anime_similarity = cosine_similarity(piv_sparse)
#Df of anime similarities
ani_sim_df = pd.DataFrame(anime_similarity, index = pivot_n.index, columns = pivot_n.index)
def anime_recommendation(ani_name):
    number = 1
    print('Recommended because you watched {}:\n'.format(ani_name))
    for anime in ani_sim_df.sort_values(by = ani_name, ascending = False).index[1:6]:
        print(f'#{number}: {anime}, {round(ani_sim_df[anime][ani_name]*100,2)}% match')
        number +=1
anime_recommendation(Death note)
```

Chapter 8

Output



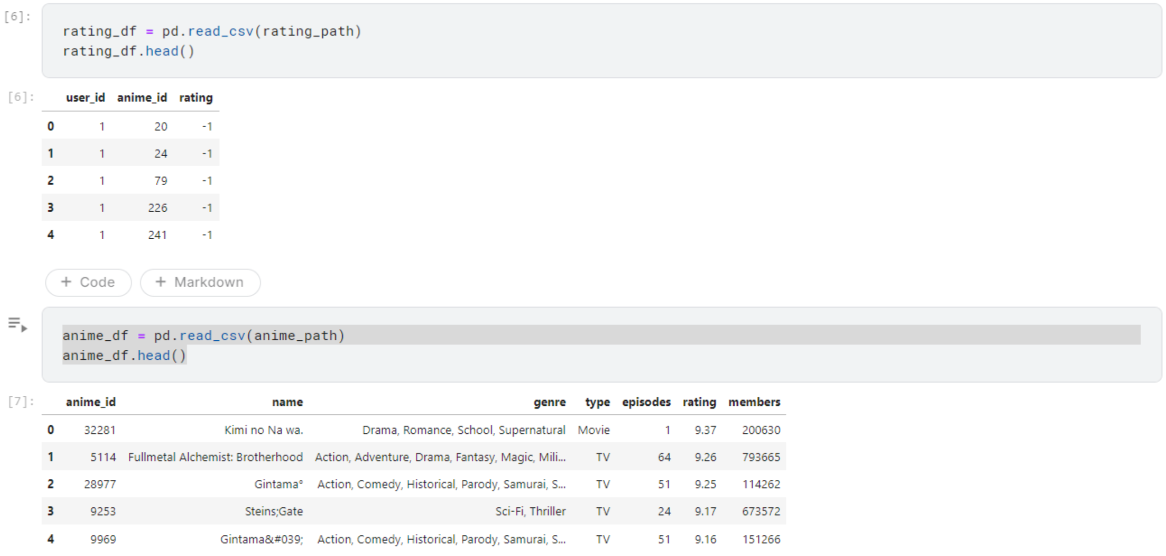
```
[1]:
import os #paths to file
import numpy as np # linear algebra
import pandas as pd # data processing
import warnings# warning filter
import scipy as sp #pivot egineering


#ML model
from sklearn.metrics.pairwise import cosine_similarity


#default theme and settings
pd.options.display.max_columns

#warning hadle
warnings.filterwarnings("always")
warnings.filterwarnings("ignore")
```

```
#list all files under the input directory
for dirname, _, filenames in os.walk('/kaggle/input'):
    for filename in filenames:
        print(os.path.join(dirname, filename))
```

```
/kaggle/input/anime-recommendations-database/rating.csv
/kaggle/input/anime-recommendations-database/anime.csv
```

```
rating_path = "/kaggle/input/anime-recommendations-database/rating.csv"
anime_path = "/kaggle/input/anime-recommendations-database/anime.csv"
```

Figure 7.1 Output screen 1



```
[6]:
rating_df = pd.read_csv(rating_path)
rating_df.head()
```

[6]:

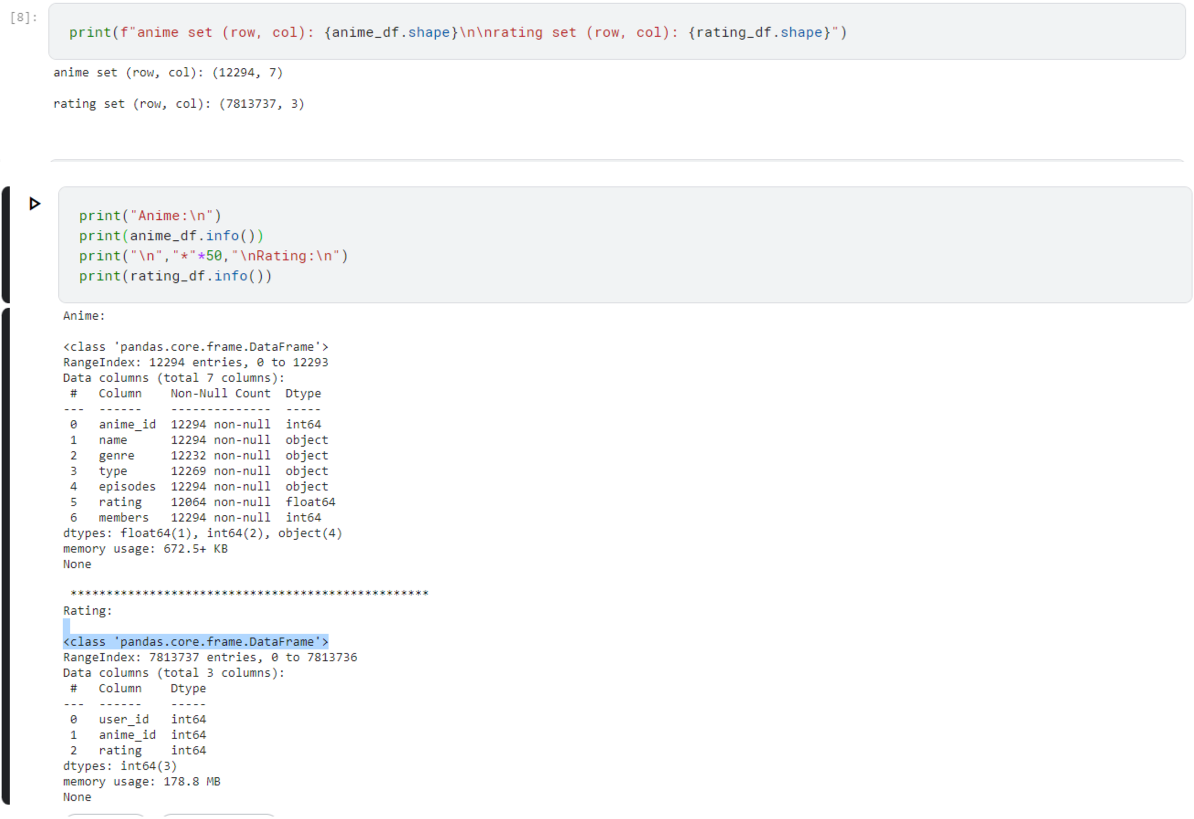
	user_id	anime_id	rating
0	1	20	-1
1	1	24	-1
2	1	79	-1
3	1	226	-1
4	1	241	-1

```
anime_df = pd.read_csv(anime_path)
anime_df.head()
```

[7]:

	anime_id	name	genre	type	episodes	rating	members
0	32281	Kimi no Na wa.	Drama, Romance, School, Supernatural	Movie	1	9.37	200630
1	5114	Fullmetal Alchemist: Brotherhood	Action, Adventure, Drama, Fantasy, Magic, Mili...	TV	64	9.26	793665
2	28977	Gintama°	Action, Comedy, Historical, Parody, Samurai, S...	TV	51	9.25	114262
3	9253	Steins;Gate	Sci-Fi, Thriller	TV	24	9.17	673572
4	9969	Gintama&#039;	Action, Comedy, Historical, Parody, Samurai, S...	TV	51	9.16	151266

Figure 7.2 Output screen 2



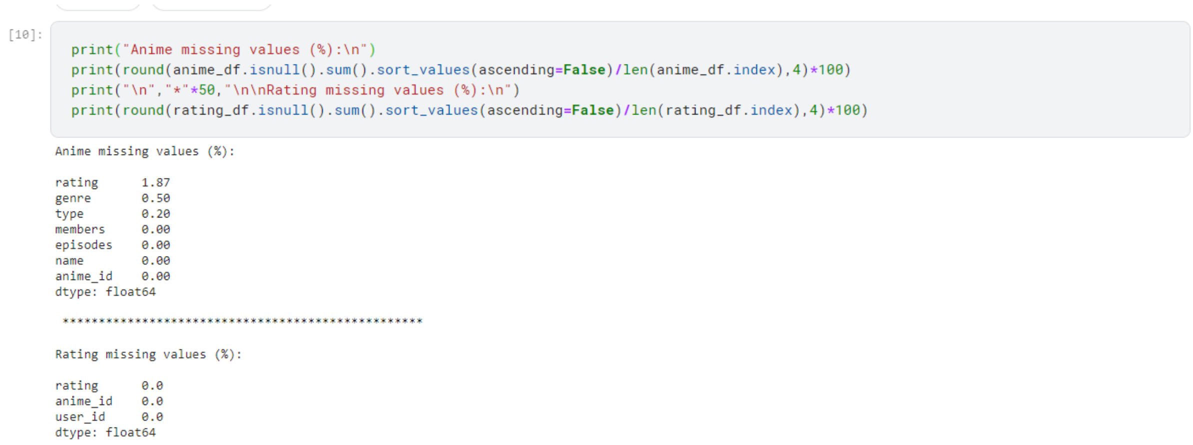
```
[8]: print(f"anime set (row, col): {anime_df.shape}\n\nrating set (row, col): {rating_df.shape}")
```

```
anime set (row, col): (12294, 7)

rating set (row, col): (7813737, 3)
```

```
print("Anime:\n")
print(anime_df.info())
print("\n","*"*50,"\nRating:\n")
print(rating_df.info())
```

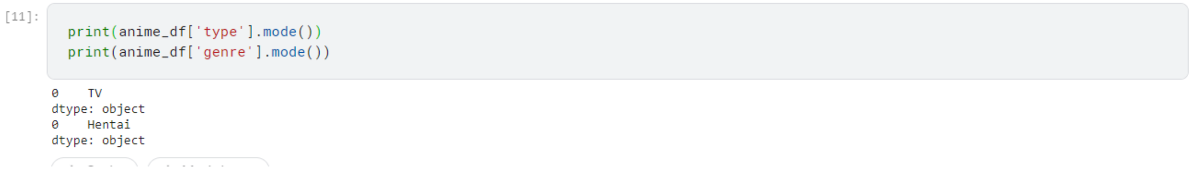
```
Anime:

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 12294 entries, 0 to 12293
Data columns (total 7 columns):
 #   Column    Non-Null Count  Dtype  
---  ------    --------------  -----  
 0   anime_id  12294 non-null  int64  
 1   name      12294 non-null  object 
 2   genre     12232 non-null  object 
 3   type      12269 non-null  object 
 4   episodes  12294 non-null  object 
 5   rating    12064 non-null  float64
 6   members   12294 non-null  int64  
dtypes: float64(1), int64(2), object(4)
memory usage: 672.5+ KB
None

 **************************************************
Rating:

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 7813737 entries, 0 to 7813736
Data columns (total 3 columns):
 #   Column    Dtype
---  ------    -----
 0   user_id   int64
 1   anime_id  int64
 2   rating    int64
dtypes: int64(3)
memory usage: 178.8 MB
None
```

Figure 7.3 Output screen 3



```
[10]: print("Anime missing values (%):\n")
      print(round(anime_df.isnull().sum().sort_values(ascending=False)/len(anime_df.index),4)*100)
      print("\n","*"*50,"\n\nRating missing values (%):\n")
      print(round(rating_df.isnull().sum().sort_values(ascending=False)/len(rating_df.index),4)*100)
```

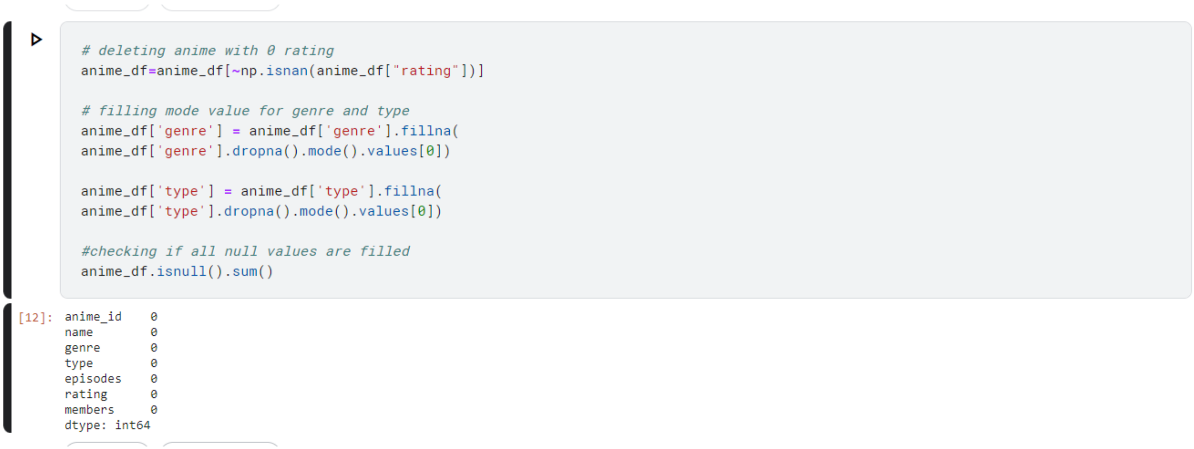
```
Anime missing values (%):

rating      1.87
genre       0.50
type        0.20
members     0.00
episodes    0.00
name        0.00
anime_id    0.00
dtype: float64

 **************************************************

Rating missing values (%):

rating      0.0
anime_id    0.0
user_id     0.0
dtype: float64
```



```
[11]: print(anime_df['type'].mode())
      print(anime_df['genre'].mode())
```

```
0    TV
dtype: object
0    Hentai
dtype: object
```

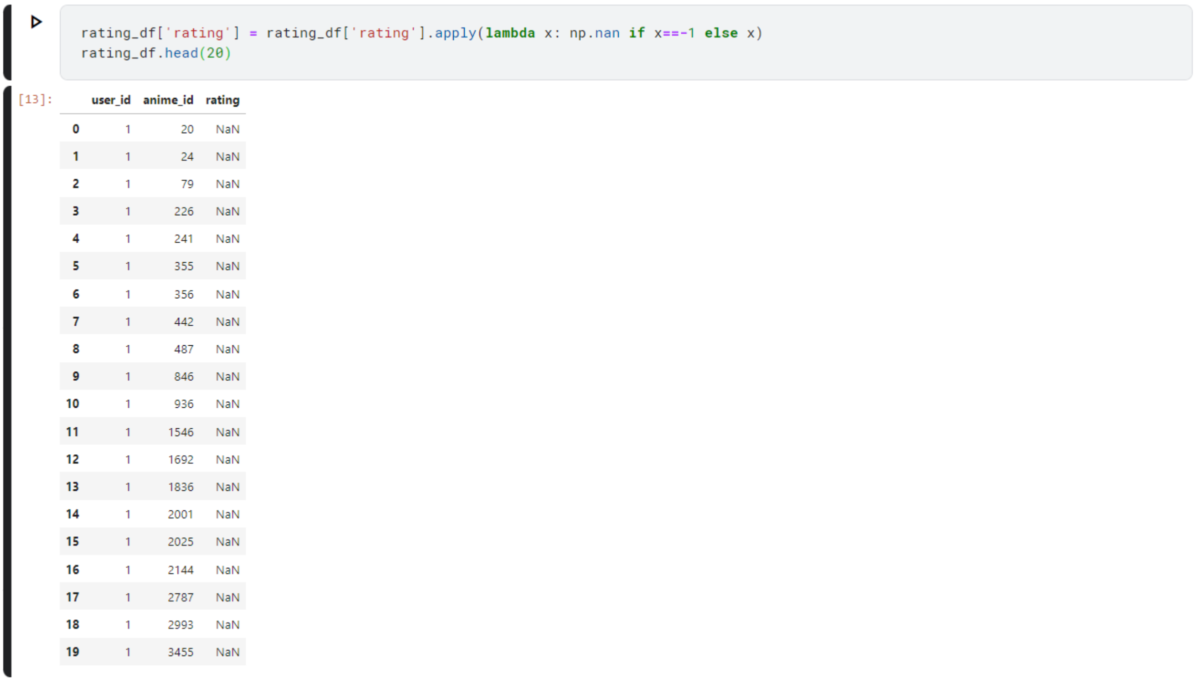
Figure 7.4 Output screen 4



```
# deleting anime with 0 rating
anime_df=anime_df[~np.isnan(anime_df["rating"])]

# filling mode value for genre and type
anime_df['genre'] = anime_df['genre'].fillna(
anime_df['genre'].dropna().mode().values[0])

anime_df['type'] = anime_df['type'].fillna(
anime_df['type'].dropna().mode().values[0])

#checking if all null values are filled
anime_df.isnull().sum()
```

```
[12]: anime_id    0
      name        0
      genre       0
      type        0
      episodes    0
      rating      0
      members     0
      dtype: int64
```

Figure 7.5 Output screen 5

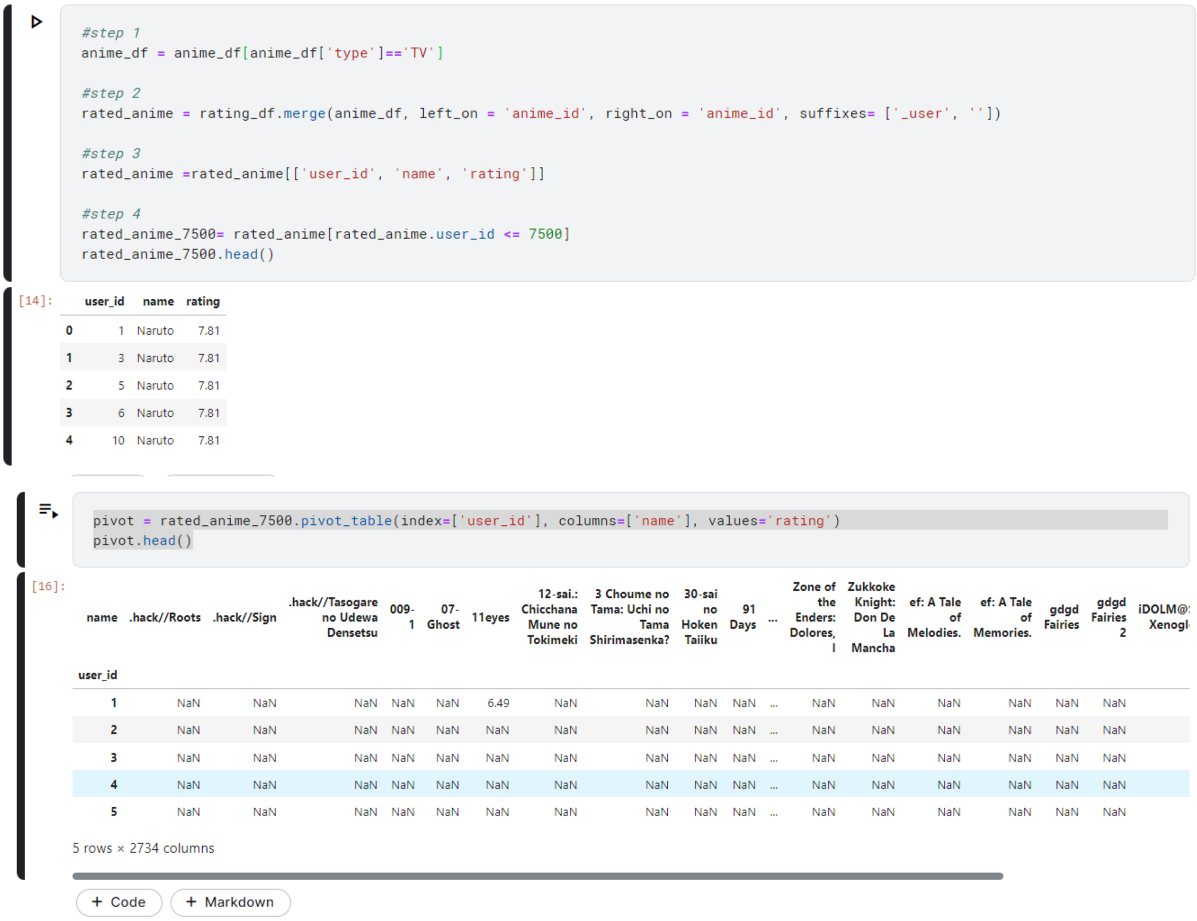


```
rating_df['rating'] = rating_df['rating'].apply(lambda x: np.nan if x==-1 else x)
rating_df.head(20)
```

[13]:

	user_id	anime_id	rating
0	1	20	NaN
1	1	24	NaN
2	1	79	NaN
3	1	226	NaN
4	1	241	NaN
5	1	355	NaN
6	1	356	NaN
7	1	442	NaN
8	1	487	NaN
9	1	846	NaN
10	1	936	NaN
11	1	1546	NaN
12	1	1692	NaN
13	1	1836	NaN
14	1	2001	NaN
15	1	2025	NaN
16	1	2144	NaN
17	1	2787	NaN
18	1	2993	NaN
19	1	3455	NaN

Figure 7.6 Output screen 6



```
#step 1
anime_df = anime_df[anime_df['type']=='TV']

#step 2
rated_anime = rating_df.merge(anime_df, left_on = 'anime_id', right_on = 'anime_id', suffixes= ['_user', ''])

#step 3
rated_anime =rated_anime[['user_id', 'name', 'rating']]

#step 4
rated_anime_7500= rated_anime[rated_anime.user_id <= 7500]
rated_anime_7500.head()
```

[14]:

	user_id	name	rating
0	1	Naruto	7.81
1	3	Naruto	7.81
2	5	Naruto	7.81
3	6	Naruto	7.81
4	10	Naruto	7.81

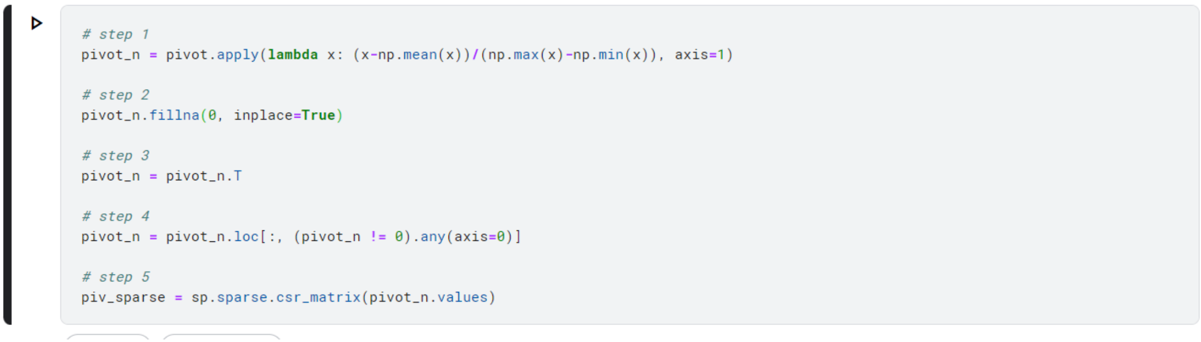
```
pivot = rated_anime_7500.pivot_table(index=['user_id'], columns=['name'], values='rating')
pivot.head()
```

[16]:

name	.hack//Roots	.hack//Sign	.hack//Tasogare no Udewa Densetsu	009-1	07-Ghost	11eyes	12-sai.: Chicchana Mune no Tokimeki	3 Choume no Tama: Uchi no Tama Shirimasenka?	30-sai no Hoken Taiiku	91 Days	...	Zone of the Enders: Dolores, I	Zukkoke Knight: Don De La Mancha	ef: A Tale of Melodies.	ef: A Tale of Memories.	gdgd Fairies	gdgd Fairies 2	iDOLM@ Xenogl
user_id																		
1	NaN	NaN	NaN	NaN	NaN	6.49	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	
2	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	
3	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	
4	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	
5	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	

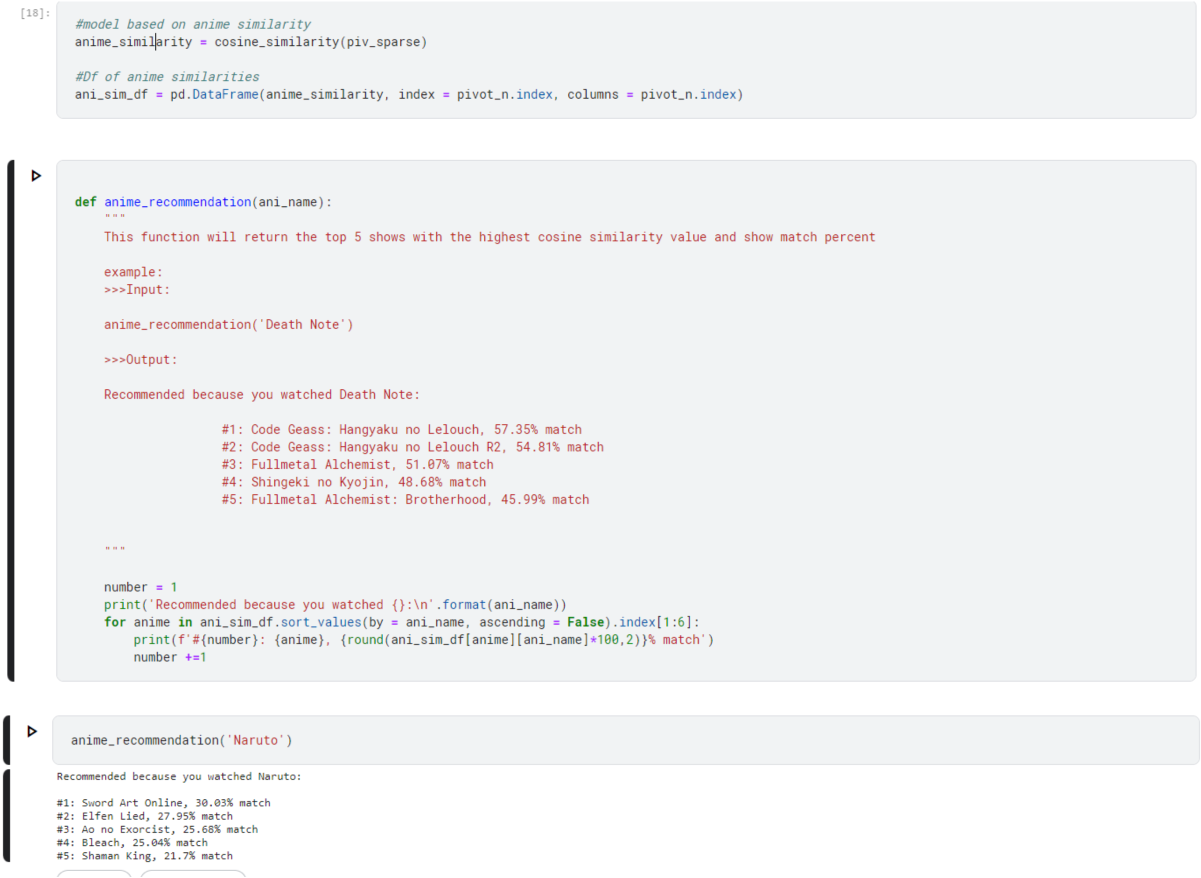
5 rows × 2734 columns

Figure 7.7 Output screen 7



```
# step 1
pivot_n = pivot.apply(lambda x: (x-np.mean(x))/(np.max(x)-np.min(x)), axis=1)

# step 2
pivot_n.fillna(0, inplace=True)

# step 3
pivot_n = pivot_n.T

# step 4
pivot_n = pivot_n.loc[:, (pivot_n != 0).any(axis=0)]

# step 5
piv_sparse = sp.sparse.csr_matrix(pivot_n.values)
```

Figure 7.8 Output screen 8



```python
#model based on anime similarity
anime_similarity = cosine_similarity(piv_sparse)

#Df of anime similarities
ani_sim_df = pd.DataFrame(anime_similarity, index = pivot_n.index, columns = pivot_n.index)
```

```python
def anime_recommendation(ani_name):
    """
    This function will return the top 5 shows with the highest cosine similarity value and show match percent

    example:
    >>>Input:

    anime_recommendation('Death Note')

    >>>Output:

    Recommended because you watched Death Note:

                    #1: Code Geass: Hangyaku no Lelouch, 57.35% match
                    #2: Code Geass: Hangyaku no Lelouch R2, 54.81% match
                    #3: Fullmetal Alchemist, 51.07% match
                    #4: Shingeki no Kyojin, 48.68% match
                    #5: Fullmetal Alchemist: Brotherhood, 45.99% match


    """

    number = 1
    print('Recommended because you watched {}:\n'.format(ani_name))
    for anime in ani_sim_df.sort_values(by = ani_name, ascending = False).index[1:6]:
        print(f'#{number}: {anime}, {round(ani_sim_df[anime][ani_name]*100,2)}% match')
        number +=1
```

```python
anime_recommendation('Naruto')
```

```
Recommended because you watched Naruto:

#1: Sword Art Online, 30.03% match
#2: Elfen Lied, 27.95% match
#3: Ao no Exorcist, 25.68% match
#4: Bleach, 25.04% match
#5: Shaman King, 21.7% match
```

Figure 7.9 Output screen 9

Chapter 9

Future Scope

The future scope of book recommendation systems is poised for significant growth and innovation. As technology continues to advance, these systems are expected to become even more sophisticated and capable of providing highly personalized and accurate book suggestions. Key areas of development include the integration of emerging technologies like natural language processing, sentiment analysis, and deep learning to enhance the understanding of book content and user preferences.

Furthermore, the expansion of recommendation systems into various niche genres, educational sectors, and global markets presents substantial opportunities. Ethical considerations, such as bias mitigation and user privacy, will likely become even more critical, leading to the development of responsible recommendation algorithms. With the ever-increasing volume of digital content and the continuous evolution of reader preferences, book recommendation systems are well-positioned to play an integral role in the book industry, aiding readers in their literary journeys and supporting publishers and retailers in connecting their books with the right audience.

Along with the recommendation we can find many other important challenges in the field of book recommendation system, like payment of books through online, customer order tracking, confirmation of order and the replacement or cancellations of order. These features can be incorporate for better recommendation in the future work of book recommendation.

Chapter 10

Conclusion

In conclusion, book recommendation systems have revolutionized the way readers discover and engage with literature in the digital age. These systems, driven by advanced algorithms and data analytics, not only simplify the overwhelming process of book selection but also hold immense potential in shaping the future of the book industry. By providing personalized recommendations tailored to individual tastes and preferences, they enhance the reading experience, promote content diversity, and drive book sales for retailers and publishers. Moreover, the continual advancements in machine learning, natural language processing, and ethical considerations will further refine and expand the scope of book recommendation systems.

As readers increasingly rely on these systems to navigate the vast world of books, the future promises more accurate, responsible, and user-centric recommendations, ensuring that readers find their next great read while contributing to a vibrant and inclusive literary landscape. In this evolving landscape, book recommendation systems stand as a testament to the power of technology in enhancing one of the most timeless and cherished human experiences—reading.

In summary, book recommendation systems represent a significant and transformative aspect of the modern book industry and reading culture. These systems have evolved to address the challenges of information overload, providing readers with personalized and relevant book suggestions, thereby enhancing the overall reading experience. By leveraging data analytics, machine learning, and natural language processing, recommendation systems offer valuable insights into user preferences and content trends, enabling retailers, publishers, and authors to connect with their target audiences more effectively. As technology continues to advance, the future scope of book recommendation systems holds great promise, with potential applications in educational settings, niche genres, and global markets. However, it is crucial to address ethical concerns such as bias and user privacy in these systems. As the literary landscape continues to evolve, book recommendation systems will remain essential tools in helping readers discover, explore, and enjoy the world of books.

References

[1] *Wikipedia contributors, "Recommender system," Wikipedia, Oct. 30, 2023. https://en.wikipedia.org/wiki/Recommender_system*

[2] R. "SVD-Based Collaborative Filtering for Anime Recommendation" by Koren, Y., Bell, R., & Volinsky, C. (2009): https://datajobs.com/data-science-repo/Collaborative-Filtering-[Koren-and-Bell].pdf

[3] "Matrix Factorization Techniques for Anime Recommendations" by Shani, G., & Gunawardana, A. (2011): https://datajobs.com/data-science-repo/Recommender-Systems-[Netflix].pdf

Acknowledgement

We would like to express our gratitude and appreciation to our parents for motivating and encouraging us throughout our careers.

We wish to express our sincere thanks to our principal **Dr. Javed Habib**, M.H. Saboo Siddik College of Engineering, Vice Principal, **Dr. Zahir Khan**, M. H. Saboo Siddik College of Engineering and the Head of Computer Engineering Department, **Dr. Mohammed Ahmed Shaikh** for providing us with all the facilities, support, and wonderful environment to meet our project requirements.

We are highly thankful to our internal project guide **Asst. Prof. Ahlam Ansari** whose valuable guidance helped us understand the project better, their constant guidance and willingness to share their vast knowledge made us understand this project and its manifestations in great depths and helped us to complete the project successfully.

We would also like to acknowledge with much appreciation the role of the staff of the Computer Department, especially the Laboratory staff, who gave us the permission to use the labs when needed and the necessary material to complete the project.

We would like to express our gratitude and appreciate the guidance given by other supervisors and project guides, their comments and tips helped us in improving our presentation skills. Although there may be many who remain unacknowledged in this humble note of appreciation there are none who remain unappreciated.