A Project Report On

Book Recommendation System

Submitted in partial fulfillment of the requirement for the award of the degree

MASTER OF COMPUTER APPLICATIONS

from

Marwadi University

Academic Year 2022 – 23

Shakti Makawana (92200584132)
Raj Dodiya (92200584127)

Internal Guide

Prof. Dr. Dimple Thakar
Assistant Professor

Marwadi University

Marwadi Education Foundation's Group of Institutions (MEFGI)

Rajkot-Morbi Road, At & PO: Gauridad, Rajkot 360 003, Gujarat.

Faculty of Computer Applications (FoCA)

Certificate

This is to certify that the project work entitled

The Book Recommendation System

Submitted in partial fulfillment of the requirement for the award of the degree of

Master of Computer Applications

of the

Marwadi University

is a result of the bonafide work carried out by

Shakti Makawana (92200584132),

Raj Dodiya (92200584127)

during the academic year 2023-2024

Faculty Guide **HoD**

External Viva

Name of the Examiners **Signature with Date**

DECLARATION

We hereby declare that this project work entitled **A Book Recommendation System** is a record done by us.

We also declare that the matter embodied in this project is genuine work done by us and has not been submitted whether to this University or to any other University / Institute for the fulfillment of any course of study.

Place: Rajkot

Date:

Shakti Makawana (92200584132) **Signature: _____**
Raj Dodiya (92200584127) **Signature: _____**

ACKNOWLEDGEMENT

It is indeed a great pleasure to express our thanks and gratitude to all those who helped us. No serious and lasting achievement one can ever achieve without the help of friendly guidance and co-operation of so many people involved in the work.

We're very thankful to our faculty guide **Prof. Dr. Dimple Thakar,** the person who makes our follow the right steps during project work. We express our deep sense of gratitude to his guidance, suggestions and expertise at every stage. Apart from that his valuable and expertise suggestion during documentation of our report indeed helped us a lot.

Thanks to our friends who have been a source of inspiration and motivation that helped us during our project work. We're heartily thankful to our external guide _____ for providing us an opportunity to work over this report and for their endless and great support. We are also thankful to our Head **Dr. Aswin R. Dobariya** and Dean **Dr. R. Sridaran** for their support as and when required. We would like to extend our gratitude towards all who directly or indirectly supported and helped us to accomplish my project work.

Shakti Makawana (92200584132) **Signature: _____**
Raj Dodiya (92200584127) **Signature: _____**

Table of Contents

Chapter 1: Synopsis 1

Chapter 2: Preamble 2

2.1 General introduction 2

2.2 Statement of problem 3

2.3 Objective of the study 3

2.4 Scope of the study 4

2.5 Feasibility study 6

Chapter 3: Review of literature 10

Chapter 4: Technical description 15

4.1 Hardware Requirement 15

4.2 Software Requirement 15

4.3 Technology Used 15

Chapter 5: System design and development 16

5.1 Data Flow (Level-0 & Level-1) diagram 16

5.2 Use case diagram 18

5.3 Activity diagram 19

5.4 Sequence diagram 21

5.5 Class diagram 22

5.6 System Architecture 23

5.7 Consideration of Libraries for Machine Learning 24

5.8 Datasets 25

Chapter 6: System UI\UX 27

Chapter 7: Conclusion 30

Chapter 8: Learning during project work 31

Bibliography 32

a. Online references 32

b. Offline references 32

1. SYNOPSIS

There are millions of books in the entire world wide and people need some instructions to find the appropriate book. Making decision and scrolling for the right book from millions of books can be hard and a complete waste of time. To compete and keep up in the market, some viewers may just rate the book in unbiased form.

The use of such overstated descriptions is misleading the viewers to unpromising products or books. Resulting in inaccuracy of recommendation of books to the user. In this project, this problem domain has been addressed and has been tried to solve with a better approach using a collaborative filtering method.

This project offers a quick and intuitive Book Recommendation System that enables readers to navigate and find the appropriate book to read next. The end goal of this project is to use collaborative filtering method to providea better recommendation.

2. PREAMBLE

2.1 General Introduction

In a rapidly evolving digital age, information is abundant, and book choices are seemingly limitless. However, this abundance often makes it challenging for readers to discover books that genuinely resonate with their preferences. This is where recommender systems come into play. Recommender systems leverage user data and machine learning techniques to provide personalized recommendations, thereby enhancing the overall user experience.

A Book Recommendation System is a type of recommendation system where we have to recommend similar books to the reader based on his interest. The books recommendation system is used by online websites which provide E-books like Google play books, open library, good Read's, etc.

It tries to recommend items to the customers according to their needs and taste. The system analyses the books that were liked by the customer with the unrated books potentially looking for best suited content to recommend them.

Expected Outcomes:

Upon completion, the project is expected to deliver a functional collaborative-based Book Recommender System that can make personalized book recommendations to users based on their historical preferences and interactions. The system will be accessible through a user-friendly web interface, making it practical and enjoyable for users to discover books that align with their tastes.

This project not only demonstrates the application of machine learning and data science in creating valuable recommendation systems but also showcases the capabilities of popular Python libraries and tools in developing interactive applications. It serves as a bridge between technology and the avid reading community, promoting engagement and fostering a culture of lifelong learning.

2.2 Statement of Problem

During the last few decades, with the rise of YouTube, Amazon, Netflix, and many other such web services, recommender systems have become much more important in our lives in terms of providing highly personalized and relevant content.

The main objective is to create a recommendation system to recommend relevant books to users based on popularity and user interests.

2.3 Objective and Scope of the Study

The objective of this study is to develop and deploy an accurate and user-friendly Book Recommendation System using machine learning, with a focus on increasing book sales and user engagement through personalized recommendations. Some of other objectives can be described as below:

- Create a Book Recommendation System using collaborative filtering method.
- Get the required dataset for the system.
- Build a design/prototype.
- Evaluate and run tests on prototype to ensure the usability and functionality.
- To build an interactive interface allowing users to interact with the system.

Scope of this study would be:
This study will encompass the entire project lifecycle, from data acquisition and model development to system integration, deployment, and ongoing maintenance. It will consider both technical and economic feasibility, as well as legal and ethical compliance. The project aims to optimize user experiences and revenue generation while ensuring the long-term sustainability of the Book Recommendation System.

2.4 Module Description with functionality:

A Book Recommendation System is a software module designed to suggest books to users based on their preferences reading history and other relevant data.

A Book Recommendation System leverages user data and sophisticated algorithms to suggest books tailored to individual preferences. It aims to enhance the user's reading experience by helping them discover books they are likely to enjoy, ultimately promoting love for reading and book discovery.

Steps to build recommendation system

1. Collection of Data
2. Analysis of this Data
3. Pre-processing of Data
4. Data Model (Cluster Algorithm or Unsupervised ML)
5. Web Application
6. Deploy it to cloud

Data Collection Module:

We have used dataset of about 11 lac books out of the internet which is publicly available to use. Here, we also have used different real users and it’s ratings to different books for giving best recommendations with the use of this system.

Data Preprocessing Module:

Pandas are a versatile Python library for data preprocessing that provides powerful data structures, such as Data Frames, and a wide range of functions to manipulate and clean data. It simplifies tasks like handling missing values, filtering, transforming, and aggregating data, making it an essential tool for data preparation and analysis in various domains, including data science, machine learning, and research.

Pandas were used in the project for data collection, preprocessing, and managing user interaction data. It helped clean and structure the dataset, enabling accurate user-item interaction analysis and personalized book recommendations.

Data Visualization:

Matplotlib is a popular Python library for data visualization, offering a wide range of tools for creating high-quality charts, graphs, and plots. It provides flexibility and customization options to tailor visualizations to specific data analysis needs. With Matplotlib, you can create various types of visual representations, including line charts, bar plots, scatter plots, histograms, and more.

Its integration with Jupyter Notebook makes it a preferred choice for data scientists and analysts to effectively communicate insights and trends from their data. Matplotlib is a foundational tool for exploratory data analysis, presentation of results, and generating clear, informative visualizations.

Collaborative Filtering Module:

Collaborative filtering was implemented using user-based and item-based approaches. It identified similar users or items based on historical interactions and made personalized book recommendations for the project.

User Interface Module (Streamlit):

The User Interface module was implemented using Streamlit, enabling the creation of a user-friendly web interface for the Collaborative-Based Book Recommender System. This interface allowed users to input their preferences, such as favorite genres or authors, and then communicated with the collaborative filtering recommendation system.

The recommended books were displayed within the interface, with details like titles, authors, and cover images. Users could interact with the recommendations, explore additional details, and provide feedback. Streamlit's user-friendly design and ease of use enhanced the overall user experience and facilitated the discovery of personalized book recommendations within the project.

Documentation and Reporting Module (Jupyter Notebook):

Jupyter Notebook was instrumental in the project. It enabled data exploration, preprocessing, model development, and documentation, facilitating transparency and collaboration among team members. Its interactive environment streamlined the entire project's workflow.

2.5 Feasibility Study

2.5.1 Technical Feasibility

We need to check some technical feasibility like:

Data Availability: Access to high-quality book-related data, encompassing user preferences, book metadata, and reviews, is crucial for effective machine learning model training and evaluation. A comprehensive and reliable dataset is essential for personalized recommendations.

Machine Learning Algorithms: Evaluate the most suitable machine learning algorithms for building your recommender system, whether it's collaborative filtering, content-based filtering, or hybrid models. The choice depends on your data and use case.

Computational Resources: Ensure you have the necessary computational resources, including CPUs/GPUs, memory, and storage, to efficiently train and deploy machine learning models. Adequate resources contribute to faster model training and responsive recommendations.

Scalability: Consider your system's ability to handle a growing user base and an expanding book catalogue. A scalable infrastructure ensures that performance remains robust as user numbers increase.

Integration: Determine how your recommender system will seamlessly integrate with your existing infrastructure, whether it's a website, mobile app, or standalone service. Integration is critical for a cohesive user experience.

Model Training and Maintenance: Establish processes for regular model training and maintenance to keep recommendations up-to-date with evolving user preferences and the book catalogue.

Evaluation Metrics: Define and use appropriate evaluation metrics, such as precision, recall, and mean average precision, to measure the quality of your recommendations.

Privacy and Security: Address data privacy and security concerns, especially when handling user data. Implement safeguards to protect user information and comply with relevant regulations.

User Experience: Consider the user experience when implementing recommendations. Ensure that recommendations are presented in a user-friendly and intuitive manner to enhance user engagement.

Budget and Costs: Estimate the costs associated with data acquisition, infrastructure, development, and on-going maintenance. Ensuring the project is financially feasible is essential for long-term sustainability.

2.5.2 Economic Feasibility

We need to check some economic feasibility like:

1. Maintaining the Recommender System: Maintaining the recommender system is an on-going process that involves managing expenses for various aspects, including data acquisition, hardware and software maintenance, development, and staffing. It's crucial to budget for these recurring costs to ensure the continued functionality and effectiveness of the system.

2. Revenue Generation: Determine how the recommender system will generate revenue or create value. This could be through increased book sales, advertising, subscription fees, or other monetization strategies. Clearly defining your revenue model is essential for the sustainability of the project.

3. Return on Investment (ROI): Assess the potential ROI by comparing the expected benefits, such as increased sales or improved user engagement, to the costs incurred in building and maintaining the recommender system. Determine whether the expected return justifies the initial and on-going investment.

4. Cost-Benefit Analysis: Perform a comprehensive cost-benefit analysis to evaluate the advantages of implementing the recommender system against the associated costs. Consider both short-term and long-term benefits, such as improved user satisfaction and increased sales.

5. Market Demand: Analyse the demand for book recommendations within your target market. Understand whether users are likely to adopt and engage with the recommender system, leading to increased sales or other desired outcomes. A strong market demand can enhance the economic viability of your project.

6. Competition: Research existing Book Recommendation Systems and competitors in the market. Assess whether your system offers a unique value proposition that can attract users and drive revenue. Understanding your competitive advantage is vital for success.

2.5.3 Operational Feasibility

1. User Acceptance: Determine if users (both customers and internal staff) are likely to accept and use the recommender system. User feedback and engagement are critical for the system's success.
2. Compatibility: Assess whether the system can seamlessly integrate with your existing platforms and technologies, such as e-commerce websites, mobile apps, or databases. Compatibility issues can impact operational efficiency.
3. Data Availability and Quality: Ensure that the required data for the recommender system is available, reliable, and accessible. Poor data quality can hinder system performance and recommendations.
4. Training and Skill Sets: Evaluate whether your organization has or can acquire the necessary skills and expertise to develop, operate, and maintain the system. Training and up skilling may be required.
5. Change Management: Plan for the organizational changes needed to accommodate the system. This includes training employees, creating new workflows, and addressing potential resistance to change.
6. Security: Implement robust security measures to protect user data, system integrity, and intellectual property. Security breaches can disrupt operations and damage reputation.
7. Cost of Operations: Estimate the on-going operational costs, including server hosting, data storage, personnel, and support. Ensure that the system remains cost-effective over time.

3. REVIEW OF LITERATURE

This chapter gives summary on a detailed background of the field that is being covered on this project. It aims to provide a brief knowledge and understanding of the technical aspects of the project and the outcomes of the research that was conducted. Comparison, contrast and evaluation on this filed is conducted.

Introduction to Jupyter Notebook:

Jupyter Notebook is an open-source web application that allows users to create and share documents that contain live code, equations, visualizations, and narrative text. It was originally developed as an evolution of the IPython project and has become an integral tool in various fields. The review begins with an introduction to Jupyter Notebook and its fundamental features.

It originated in 2014 when the IPython project decided to separate the interactive computing environment from the notebook interface. Since then, Jupyter has grown in popularity and has become a pivotal tool in data science, machine learning, and scientific research.

Key Features and Functionality (Jupyter Notebook):

One of the primary advantages of Jupyter Notebook is its ability to support various programming languages, such as Python, R, and Julia. The review covers how Jupyter allows users to write and execute code in cells, integrate text and rich media, and generate interactive visualizations, making it a versatile platform for research and data analysis.

Applications in Data Science and Research (Jupyter Notebook):

The literature review highlights Jupyter Notebook's extensive use in data science, where it simplifies data exploration, visualization, and modeling. Researchers and analysts use it for reproducible research, sharing insights, and collaborating on projects. It has gained traction in academia and industry for its role in promoting transparency and collaboration.

Community and Ecosystem (Jupyter Notebook):

Jupyter Notebook has a vibrant and active community, which contributes to its growth and development. The review discusses how this community has created a wide range of extensions and plugins, expanding its capabilities and making it adaptable to various domains.

Integration with Data Science Tools (Jupyter Notebook):

The review explores how Jupyter Notebook seamlessly integrates with popular data science libraries like NumPy, Pandas, Matplotlib, and scikit-learn. It is praised for its role in supporting the entire data analysis workflow, from data cleaning to model building and deployment.

Challenges and Limitations (Jupyter Notebook):

While Jupyter Notebook offers numerous benefits, it is essential to acknowledge its limitations. The literature review addresses challenges such as version control issues, difficulties in managing large notebooks, and the potential for non-reproducibility when not used with caution.

Jupyter Notebook has revolutionized the way data scientists, researchers, and educators work with code, data, and documentation. Its interactive and collaborative nature has made it an indispensable tool in various domains, with a promising future ahead.

What is recommendation system?

To recommend products or content based on user's behavior or choices
Ex., Spotify app or YouTube Algorithm

A basic collaborative recommender system using machine learning is a powerful tool for providing personalized recommendations to users based on their past interactions with items. There are two primary approaches to collaborative filtering: user-based and item-based. In user-based collaborative filtering, the system identifies users with similar preferences to the target user and recommends items that those similar users have liked but the target user has not seen. On the other hand, item-based collaborative filtering suggests items to users based on the similarity between the items they have previously shown an interest in and other items in the catalog.

Collaborative filtering systems have found widespread applications in industries like e-commerce, online streaming platforms, and online marketplaces. For example, Netflix utilizes collaborative filtering to recommend movies and TV shows to its subscribers based on their viewing history and the preferences of similar users. Similarly, Amazon uses collaborative filtering to suggest products to customers based on their browsing and purchasing behaviour. These systems enhance user engagement, drive sales, and improve the overall user experience by providing tailored recommendations.

Simply, it shows us if what kind of product or content user might like based on their current choices.

Types of recommender system

1) Content based

It recommends content based on content similarity.

2) Collaborative filtering

If user1 likes content X, then it is possible that user2 will also like it... It is called collaborative filtering.

(Kind of similar in interest based on their behavior)

3) Hybrid based

Mixed of both collaborative filtering and content based recommender systems.

Introduction to Collaborative Filtering:

Collaborative Filtering is the process by which items are filtered or evaluated using the views or opinions of other users. The term collaborative filtering (CF) has been around since just a little over a decade, CF is rooted in what people do over centuries to share opinions with others. For many years, people have discussed the books they have read, the restaurants they have tried, music they have listened, and about the films they have seen – then use these discussions to form opinions. For instance, if enough colleagues of john say that they loved the new book they have read, john might decide that he should also read that book.

Similarly, john might decide to read another book if many of them found it disaster. Better yet, if john could observe Sam9s recommendation of the type of books he finds pleasant, Arya's recommendation has a history of books that john despises, and Sansa would recommend it all. Over time, john learns the opinions of whom should he listen to and applies those opinions to help her determine the quality of an item (J. Ben Schafer, 2018).

The internet enables us to consider the opinions of thousands rather than limit ourselves to twenty9s or hundreds of people. The speed of the computers enables us to process these opinions in real time and not only determine what a larger community thinks about an item, but also to provide appropriate opinions for a particular user or group of users through a genuinely personalized view of the item.

Collaborative based filtering recommender systems are based on past interactions of users and target items. In simple words here, we try to search for the look-alike customers and offer products based on what his or her lookalike has chosen. Let us understand with an example. X and Y are two similar users and X user has watched A, B, and C movie. And Y user has watched B, C, and D movie then we will recommend A movie to Y user and D movie to X user. YouTube has shifted its recommendation system from content-based to Collaborative based filtering technique. If you have experienced sometimes there are also videos which not at all related to your history but then also it recommends it because the other person similar to you has watched it.

User-Based Collaborative Filtering Algorithm:

Here, (as seen in figure A) in this method user who prefer similar items are searched and the products are recommend to the user according to their product preferences. Figure B shows that both the user X and Y has read the same article A1 and A2 which shows similar alike between these two users. As the user X has also read article A3 as shown in figure, so the article A3 is recommended to the user Y.

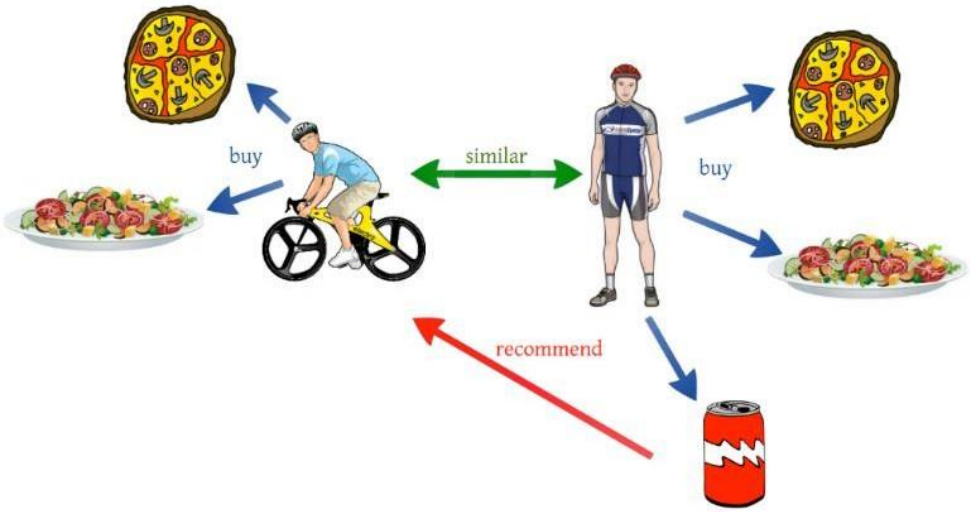


Figure A

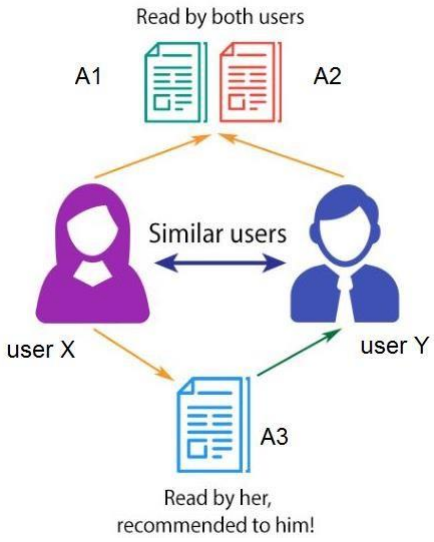


Figure B

4. TECHNICAL DESCRIPTION

4.1 Hardware Requirement

Preprocessor	Intel Pentium dual-core or AMD Radeon
Hard Disk	256 GB or Above (Recommended)
RAM	8 GB or Above (Recommended)

4.2 Software Requirement

Jupyter Notebook	Version 5.0 and above
Python	Version 3.0 and above
Anaconda (Optional)	Version 2.0 and above

4.3 Technology Used:

Python Libraries like, Numpy, Pandas, SKLearn,

Pickle, Matplotlib, and scikit-learn etc.

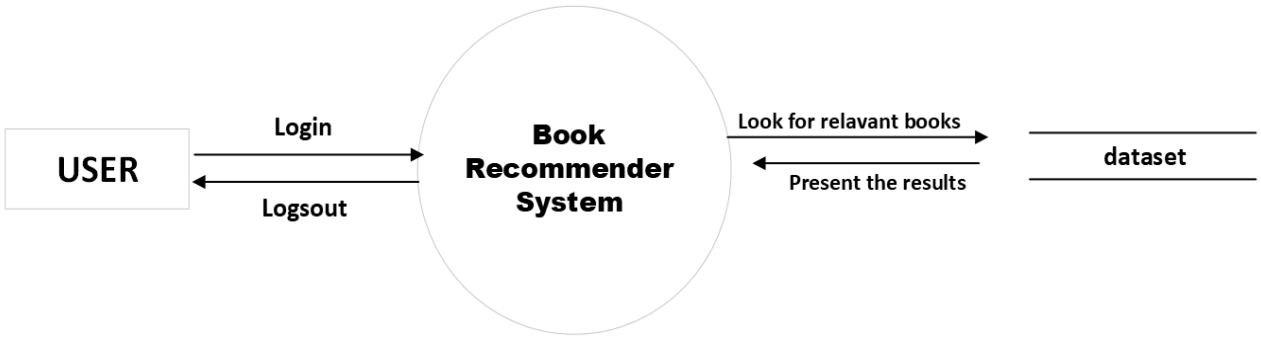
Stremlit for Localhost

Jupyter Notebook (98%)

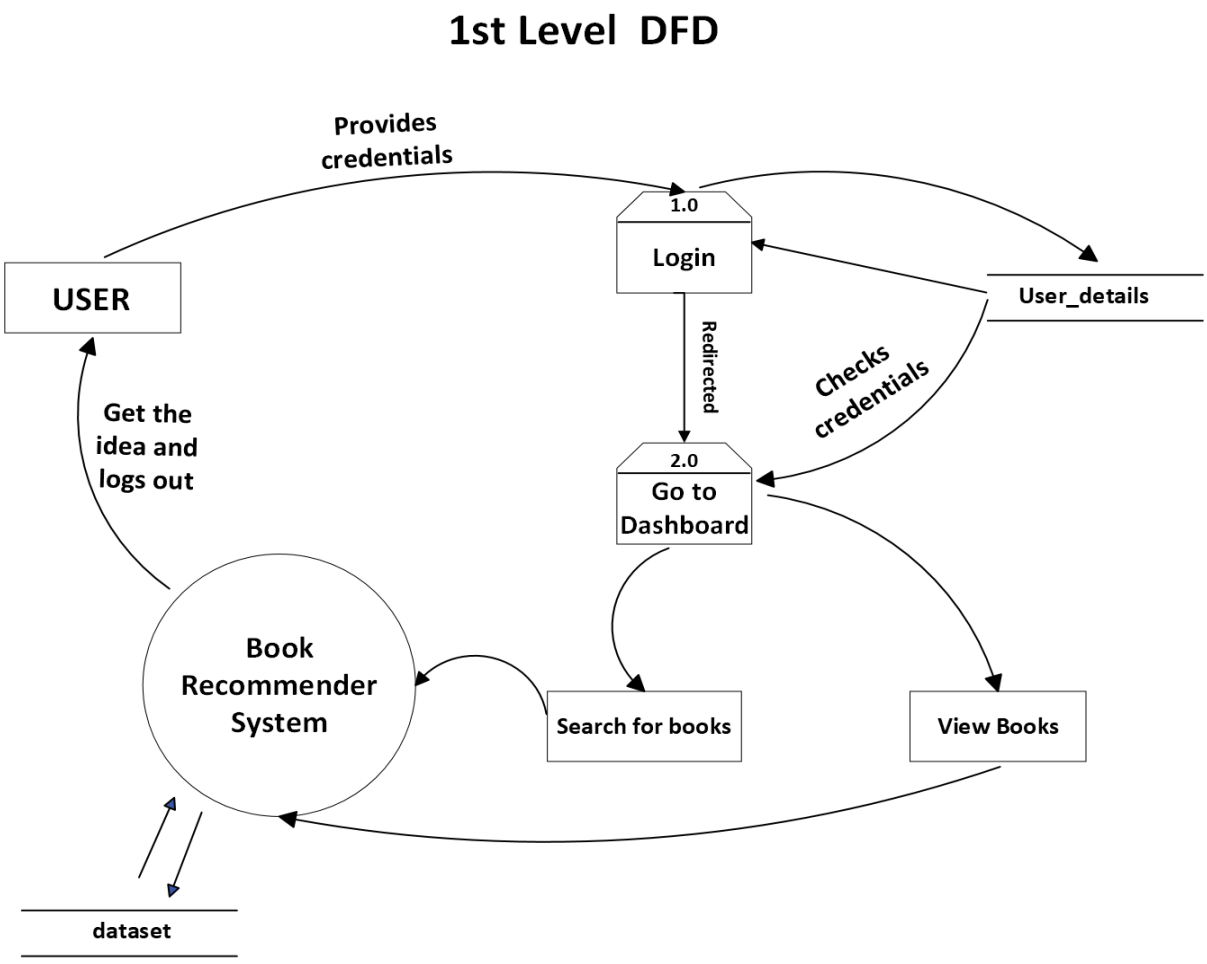
5. SYSTEM DESIGN AND DEVELOPMENT

5.1 Data Flow Diagram (Real-Life)

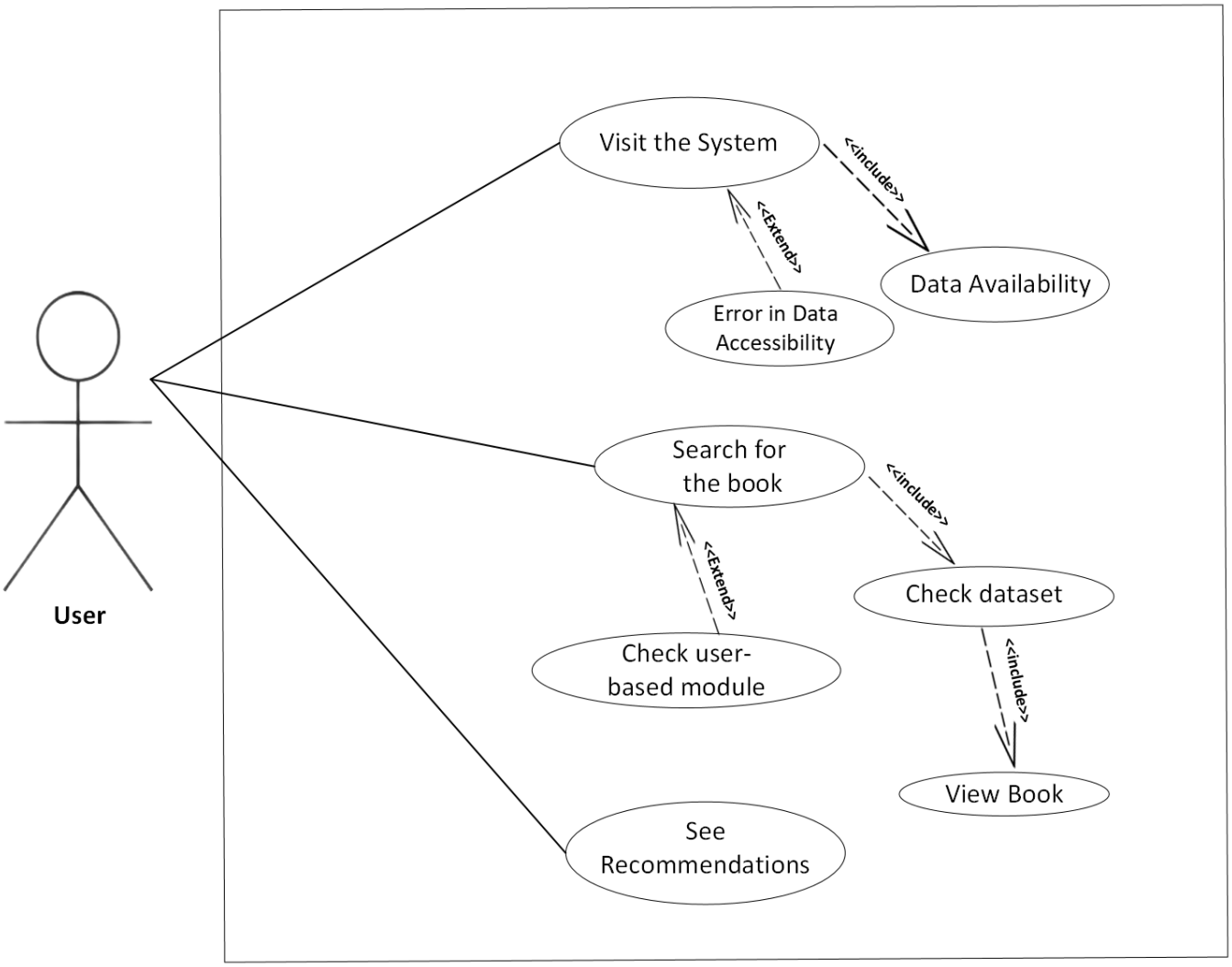
5.1.1 Data Flow Diagram Level 0



5.1.2 Data Flow Diagram Level 1



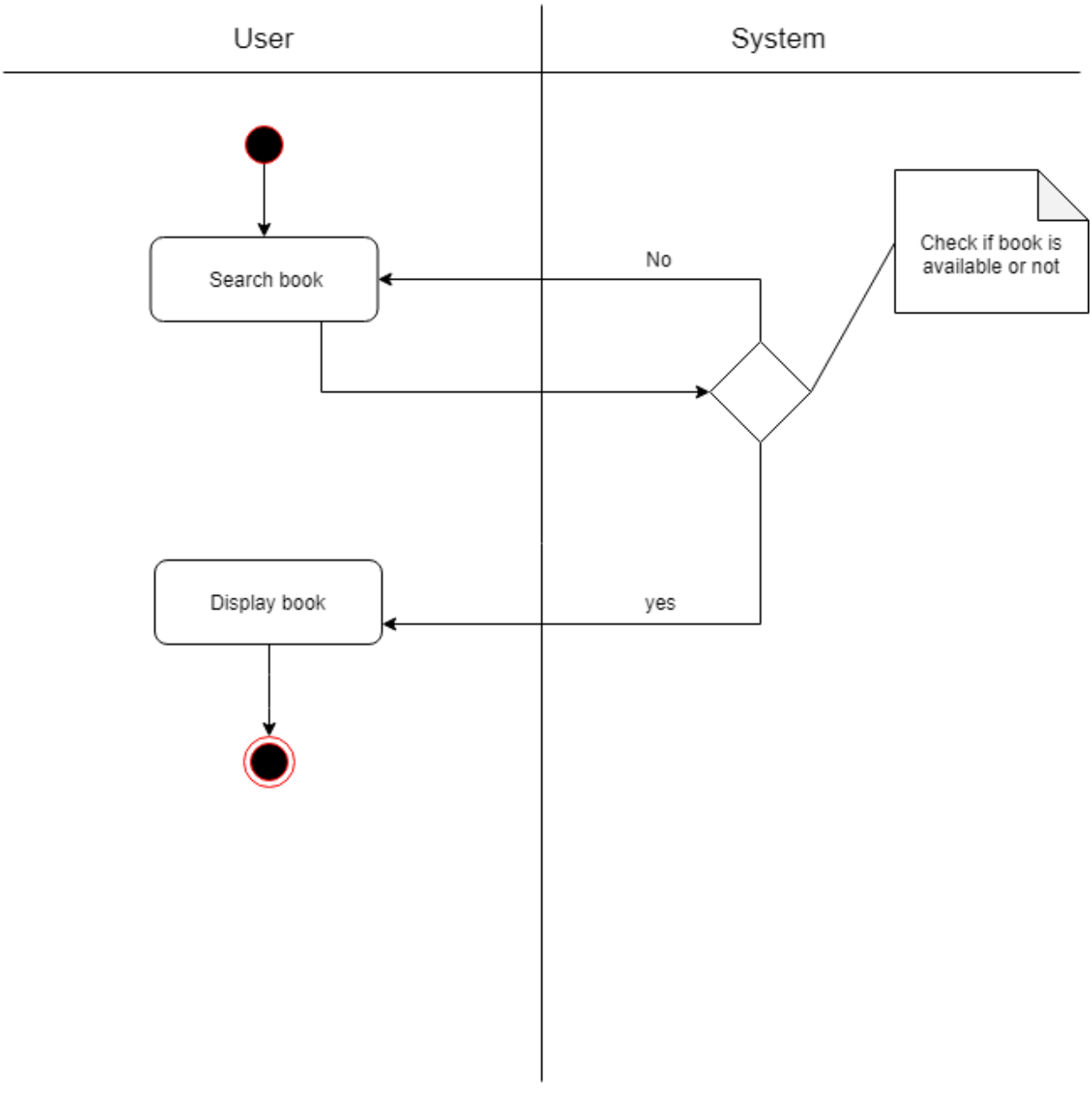
5.2 Use Case Diagram



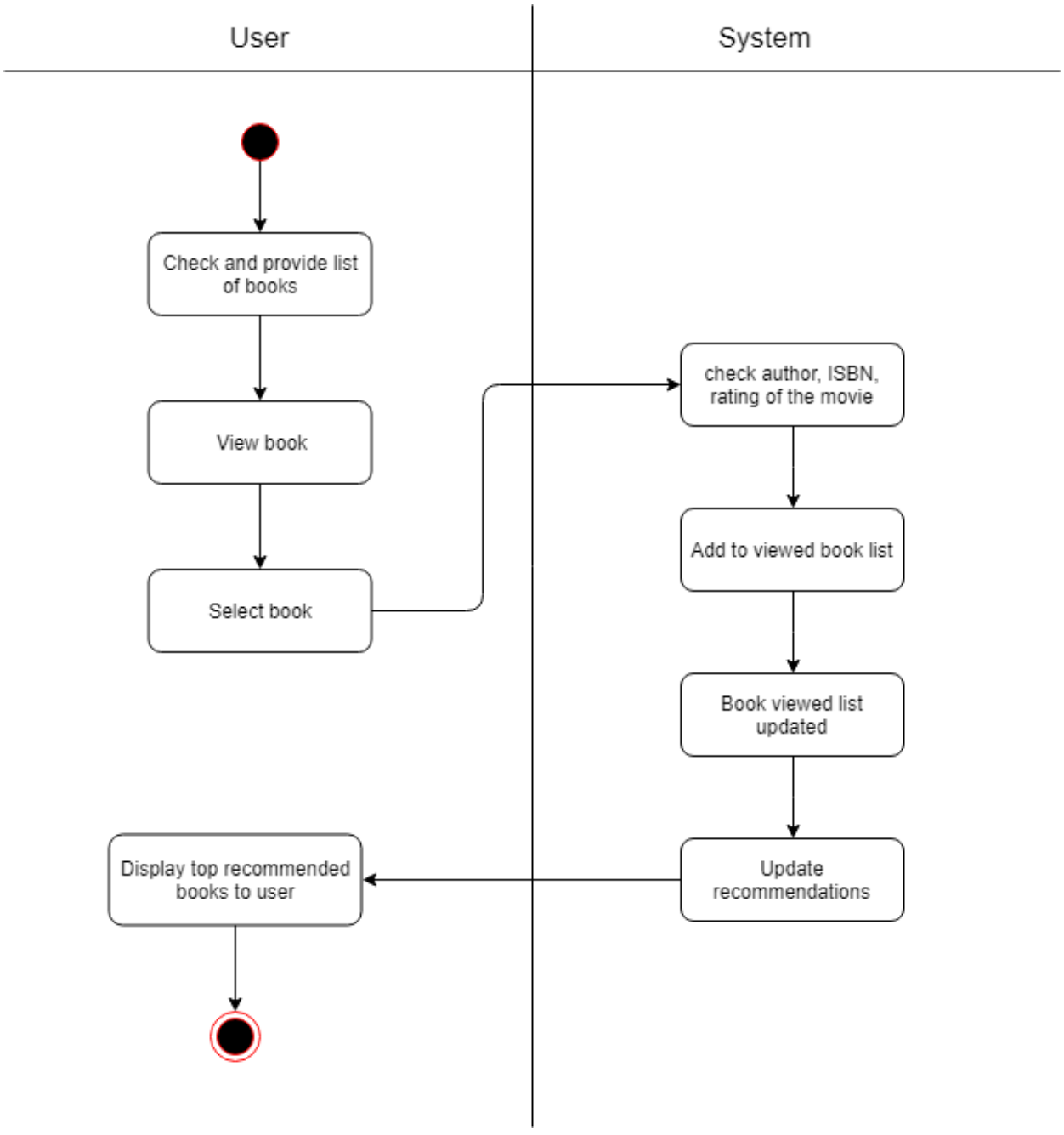
S.No.	Use Case	Description
1	Search book	To search the required book
2	View Recommendations	To view the top recommended books to the user

5.3 Activity Diagram

5.3.1 Search for the book

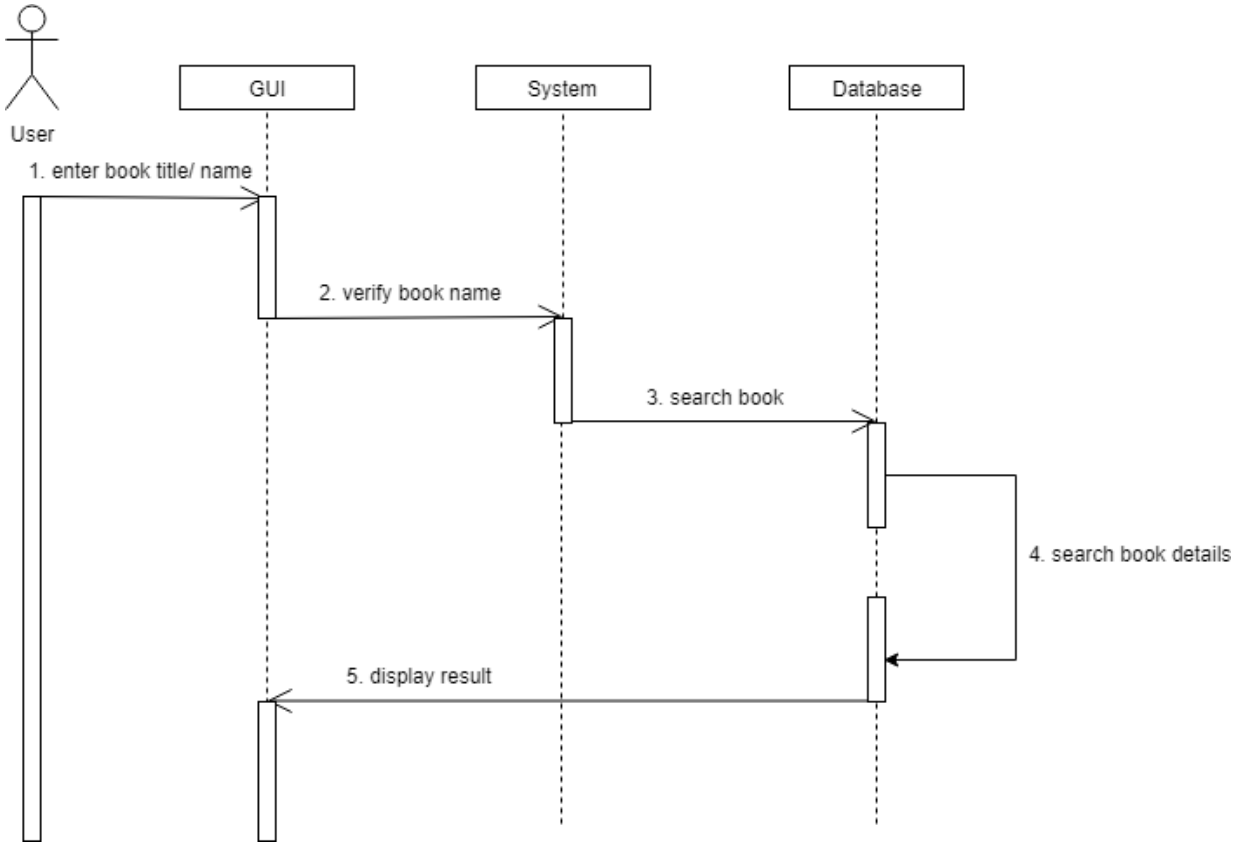


5.3.2 Book Recommendation Activity

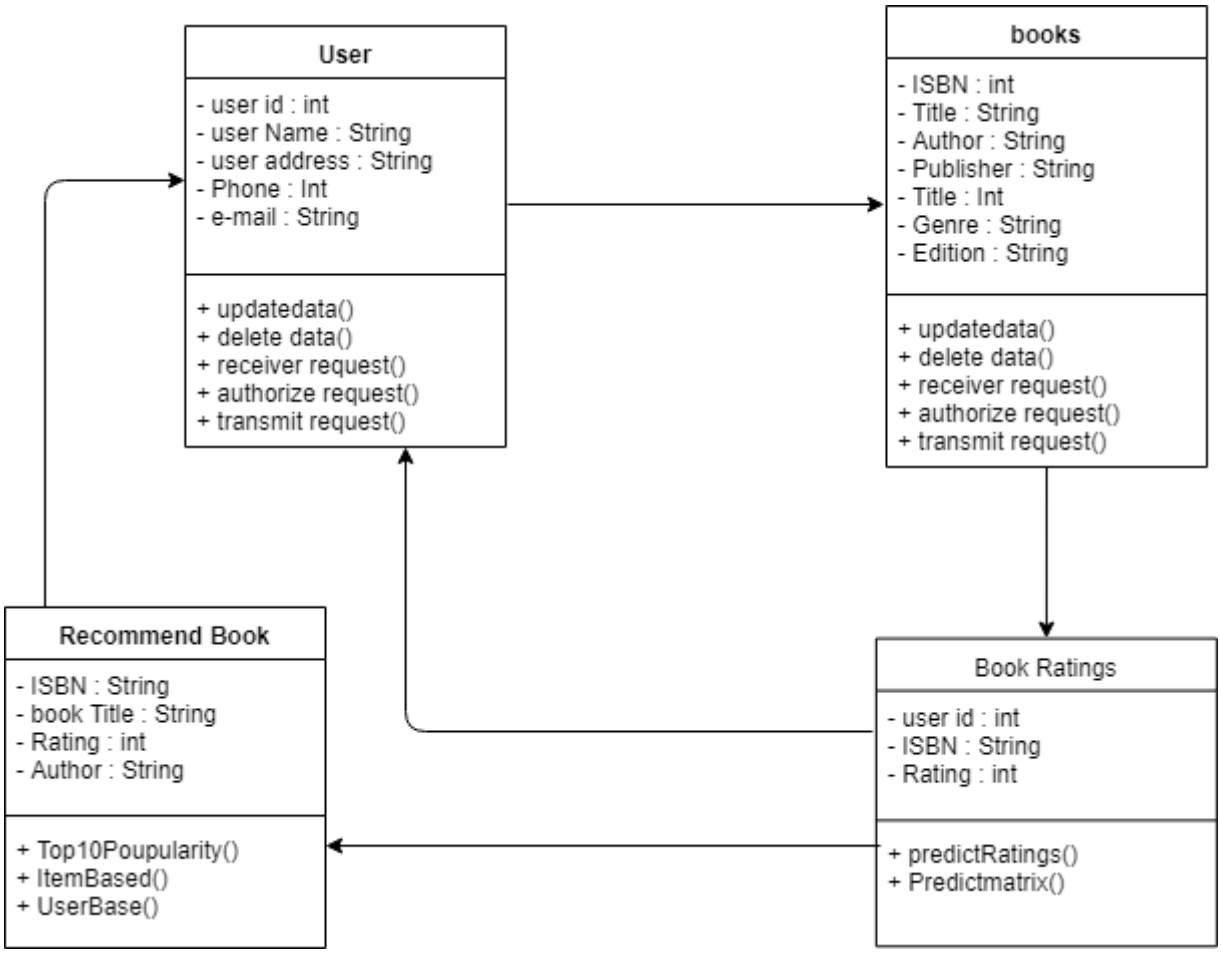


5.4 Sequence Diagram

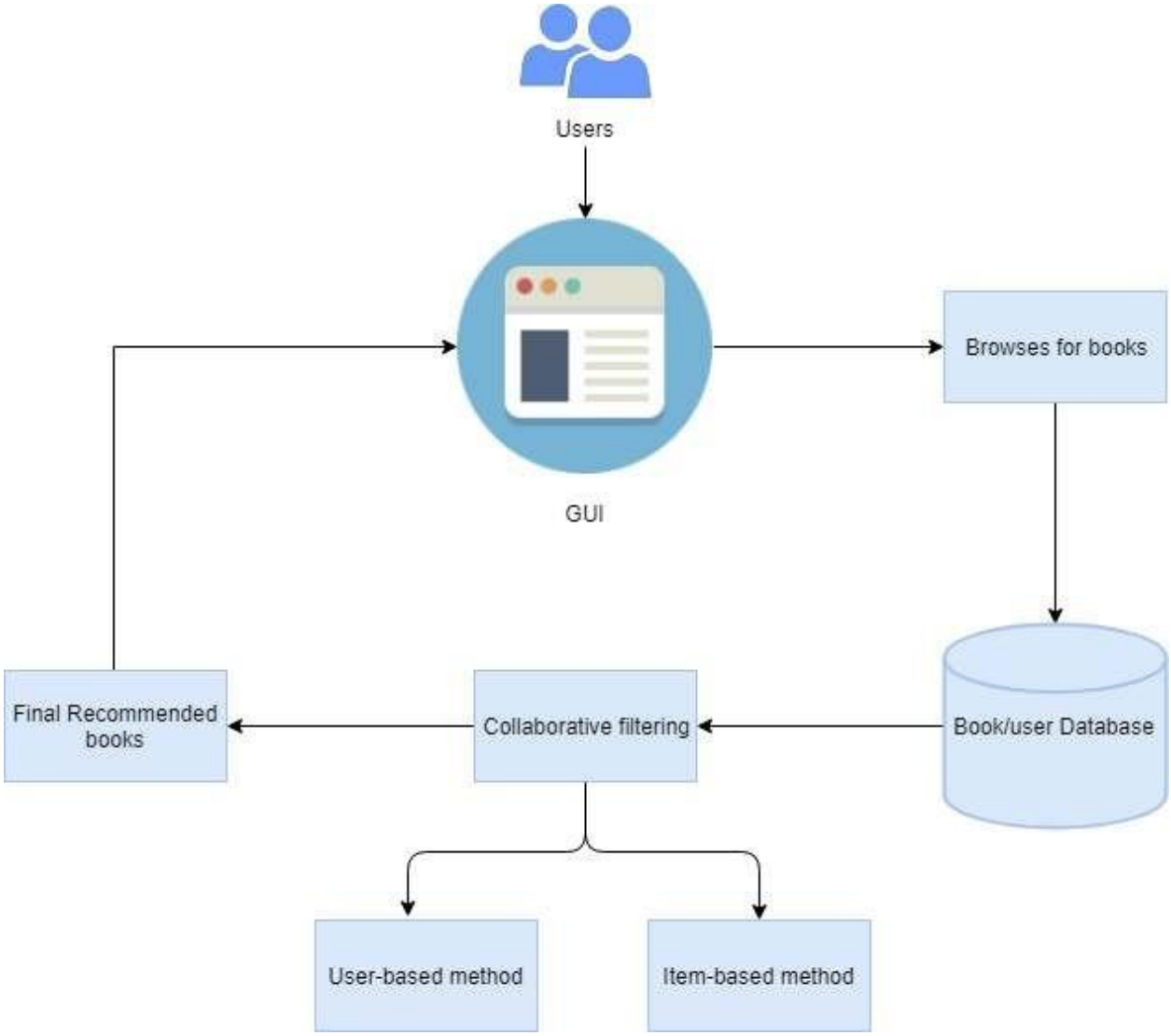
View Book



5.5 Class Diagram



5.6 System Architecture

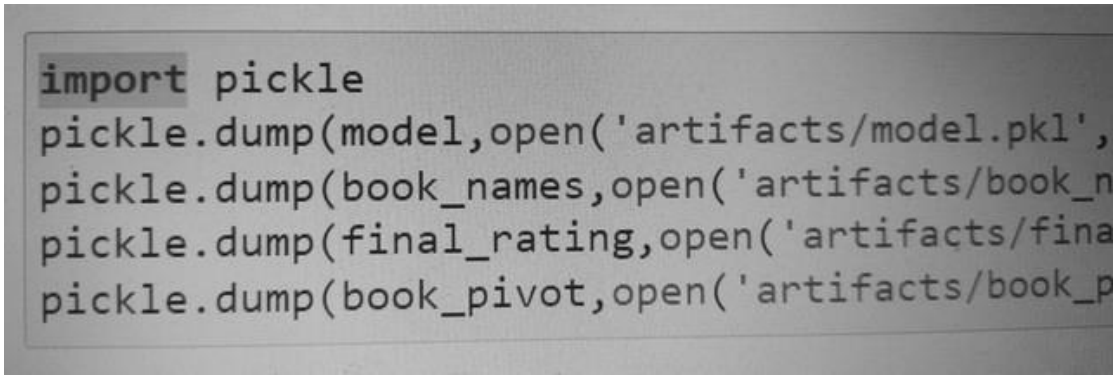


5.7 Consideration of Libraries for Machine Learning

For this project Scikit-learn library was chosen. Reasons are shown below:

- Easy and effective data mining and data analytic tool.
- Built on matplotlib, NumPy and SciPy.
- Compared to TensorFlow, the it is shallow learning curve.
- Provides various range of supervised and unsupervised learning algorithms.

```
#Imported library
import pandas as pd
import matplotlib.pyplot as plt
import sklearn.metrics as metrics
import numpy as np
from sklearn.neighbors import NearestNeighbors
from scipy.spatial.distance import correlation
from sklearn.metrics.pairwise import pairwise_distances
import ipywidgets as widgets
from IPython.display import display, clear_output
from contextlib import contextmanager
import warnings
warnings.filterwarnings('ignore')
import numpy as np
import os, sys
import re
import seaborn as sns
```



5.8 Dataset

It is important to understand what the data are and what are we trying to achieve before building any model for machine learning. Datasets are essential for training any machine learning model system. For this project https://www.kaggle.com/ra4u12/bookrecommendation is selected to predict and train the model. In the rating table, explicit ratings are expressed on a scale form 1-10 (higher values denote higher appreciation) and implicit rating is expressed by 0. Here, the data are in CSV format consisting of three table's users, books and ratings. The following figure shows how the data looks like, its column, its row and its data types. The data were loaded in Jupyter Notebook.

```
users.head(5)
```

	userID	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

```
books.head(5)
```

	ISBN	bookTitle	bookAuthor	yearOfPublication	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 & Company

```
ratings.head(5)
```

	userID	ISBN	bookRating
0	276725	034545104X	0
1	276726	0155061224	5
2	276727	0446520802	0
3	276729	052165615X	3
4	276729	0521795028	6

Above figures shows the overlook of the users, book and book ratings datasets respectively. The below figure shows the total number of the rows and columns of all the tables. Shape function is used for counting the total number of rows and columns.

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

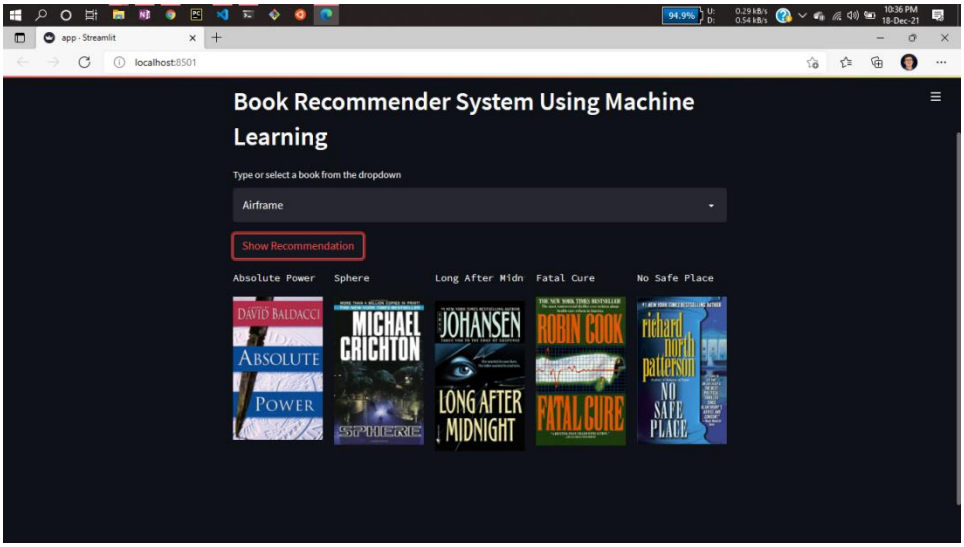
```
(278858, 3)
(1149780, 3)
(271360, 8)
```

6. SYSTEM UI\UX

The UI design of a Book Recommendation System should prioritize simplicity and user-friendly navigation. A clean and minimalistic approach with ample white space, clear typography, and a visually appealing color scheme creates a modern and inviting interface.

Book covers should be prominently displayed, using high-quality images, to make the UI visually engaging. Intuitive navigation with clear labels allows users to easily explore different sections, such as recommended books, user profiles, or search results.

Responsive design ensures the UI adapts seamlessly to different devices, maintaining usability and readability. Clear call-to-action buttons guide users through the application, enabling efficient interactions.

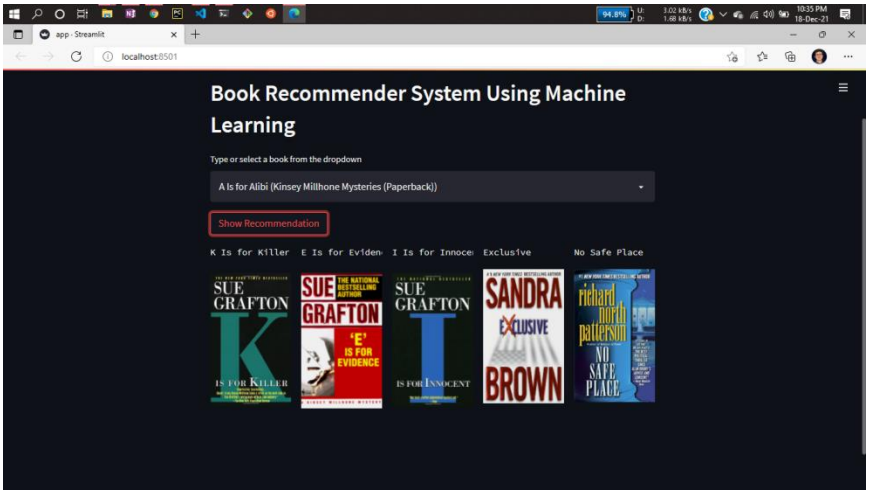


Feedback and confirmation messages provide immediate responses to user actions, keeping them informed about the status of their interactions. A prominent search bar with auto-suggestions helps users find specific books or authors quickly.

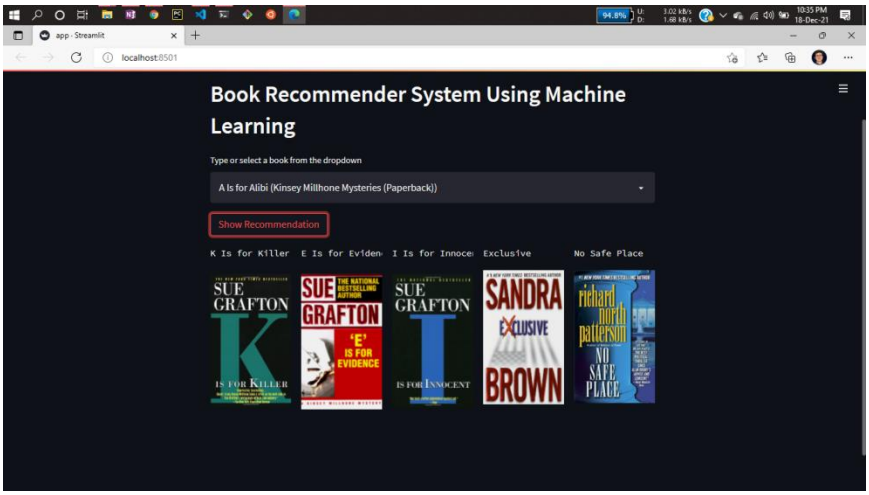
Personalization features, including user profiles, allow users to view and modify preferences, bookmarked books, and previously rated books. Displaying personalized recommendations based on user preferences enhances the tailored experience.

Error handling mechanisms with clear error messages assist users in resolving issues and provide alternatives if possible.

By focusing on simplicity, intuitive navigation, visual appeal, and personalized features, the UI design of the Book Recommendation System creates a delightful user experience.



The UX design of a Book Recommendation System should prioritize user satisfaction and efficiency. It should offer a seamless onboarding experience and provide clear instructions. The system's content and features should be organized logically and intuitively, with a well-structured navigation system.



Recommendations should be generated quickly and be relevant and diverse. The system should be transparent about its recommendation algorithms and offer user control and customization options. Feedback mechanisms should be in place to gather user input. Error handling should be graceful, and continuous improvement should be a focus.

Continuous Improvement: Regularly evaluate and improve the system's performance and user experience. Monitor user feedback, conduct usability tests, and analyze usage patterns to identify areas for enhancement. Continuously iterate on the UX design to address user needs and preferences effectively.

7. CONCLUSION

7.1 Project Conclusion

There are millions of books in the entire world wide and people need some instructions to find the appropriate book. Making decision and scrolling for the right book from millions of books can be hard and a complete waste of time. To compete and keep up in the market, some viewers may just rate the book in unbiased form. The use of such overstated descriptions is misleading the viewers to unpromising products or books.

8. LEARNING FROM PROJECT

This project involves various learning throughout the working on it. Here we have mentioned top 5 lessons we learnt below:

1. Data Integration: We learnt how to merge or join multiple datasets based on common attributes. In this case, we need to combine the books, users, and ratings datasets to create a unified dataset for analysis.

2. Data Representation: We can explore different ways of representing the data to facilitate collaborative filtering. For example, we may need to convert categorical variables like book genres or user preferences into numerical representations.

3. Collaborative Filtering Algorithms: Despite having limited data, we can still implement collaborative filtering algorithms. We learnt how to calculate user-user or item-item similarities based on the available ratings data and use these similarities to generate recommendations.

4. Handling Missing Data: In real-world scenarios, datasets may have missing values. We learnt techniques for handling missing data, such as imputation or using alternative approaches like matrix factorization that can handle missing values more effectively.

5. Recommendation Generation: We learnt how to generate personalized recommendations for users based on their past ratings and preferences. This involves selecting the top-rated items or predicting ratings for unrated items to suggest them to users.

BIBLIOGRAPHY

a. Online References

https://www.geeksforgeeks.org/collaborative-filtering-ml/

https://towardsdatascience.com/item-based-collaborative-filtering-in-python-91f747200fab

https://towardsai.net/p/machine-learning/recommendation-system-in-depth-tutorial-with-python-for-netflix-using-collaborative-filtering-533ff8a0e444

https://www.datacamp.com/tutorial/recommender-systems-python

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

b. Offline References

Collaborative Filtering Recommender Systems • 2011
By Michael D. Ekstrand, John T. Riedl, Joseph A. Konstan

Machine Learning: The Art and Science of Algorithms that Make Sense of Data • 2012
By Peter Flach

Introduction to Machine Learning • 2017
By Kubat, Miroslav

Learning Jupyter • 2016
By Dan Toomey

Recommender Systems Handbook • 2014
By Ricci, Francesco | Rokach, Lior | Shapira, Bracha