BOOK RECOMMENDATION SYSTEM

Submitted By

ALFINA I	**(21Z207)**
MADDU HEMALI SAI PRAVALLIKA	**(21Z225)**
VASUDHA R B	**(21Z267)**
VIJAYALAKSHMI P	**(21Z269)**
YOHASINI U B A	**(21Z271)**

19Z610 - MACHINE LEARNING LABORATORY

Report submitted in partial fulfillment of the requirement for the award of degree of

BACHELOR OF ENGINEERING

Branch: COMPUTER SCIENCE & ENGINEERING

of PSG College of Technology

April 2024

DEPARTMENT OF COMPUTER SCIENCE & ENGINEERING
PSG COLLEGE OF TECHNOLOGY
(Autonomous Institution)

COIMBATORE – 641 004

1. PROBLEM STATEMENT

The problem addressed in this project is the challenge of recommending relevant books to users in a large collection of books. With the vast amount of literature available, users often struggle to find books that match their interests and preferences. Traditional bookstores and online platforms may provide some recommendations based on bestsellers or popular genres, but these recommendations may not always align with the user's individual tastes.

The aim of this project is to develop a book recommendation system using various techniques such as content-based filtering, collaborative filtering, and item-based collaborative filtering. The recommendation system will help users discover new books based on their preferences, such as book title, author, and genre.

2. DATASET DESCRIPTION

Three datasets are used: [4]

i. BX-Book Ratings.csv

The dataset "BX-Book Rating.csv" contains 1,048,576 rows of data. Each row represents a user's rating for a particular book, with three columns: "User-ID," "ISBN," and "Book-Rating."

- Number of Rows: 1,048,576
- Number of Columns: 3
- Unique Users: This can be determined by counting the unique values in the "User-ID" column.
- Unique Books: This can be determined by counting the unique values in the "ISBN" column.
- Rating Distribution: Explore the distribution of ratings to understand how users have rated books on average. This can help in understanding the overall sentiment of users towards books in the dataset.

Purpose: By analyzing the ratings provided by users for various books, the system can learn patterns in user preferences and make personalized recommendations. These recommendations are based on similarities between users or books, as identified through collaborative filtering techniques.

ii. BX-Books.csv

The dataset "BX-Books.csv" contains 271,380 rows of data. Each row represents a book with several attributes, including "ISBN," "Book-Title," "Book-Author," "Year-Of-Publication," "Publisher," and URLs to images of different sizes.

- Number of Rows: 271,380
- Number of Columns: 8

- Attributes:
    - ISBN: International Standard Book Number, a unique identifier for books.
    - Book-Title: The title of the book.
    - Book-Author: The author(s) of the book.
    - Year-Of-Publication: The year when the book was published.
    - Publisher: The publisher of the book.
    - Image URLs (S, M, L): URLs to images of small, medium, and large sizes for the book cover.

Purpose: It likely acts as a reference for the book ratings dataset ("BX-Book Rating.csv"), enabling the linkage between user ratings and specific books. Additionally, it may support features such as book recommendations based on author, genre, or publication year.

iii. BX-Users.csv:

The dataset "BX-Users.csv" contains 276,272 rows of data. Each row represents a user with several attributes, including a unique identifier, location information, and possibly an age.

- Number of Rows: 276,272
- Number of Columns: 3
- Attributes:
    - User-ID: A unique identifier for each user.
    - Location: The location of the user, typically including city, state/province, and country.
    - Age: The age of the user (if available).

Purpose: The dataset likely serves as demographic data for users participating in book ratings or interactions. It can be used to analyze user demographics, preferences, and geographical distribution. Additionally, it might be utilized for building recommendation systems tailored to different user demographics.

3. METHODOLOGY

Hybrid Recommender System: Utilizing both content-based and collaborative filtering techniques to enhance recommendation accuracy and coverage.[1]

a. Content-Based Filtering: Analyzing book attributes such as genre, author, and synopsis to recommend similar books based on their content features. [5]

b. Collaborative Filtering: Leveraging user interactions and preferences to suggest books similar to those liked or rated positively by users with similar tastes.[2]

4. MODELS USED

TF-IDF Vectorization:

The TF-IDF (Term Frequency-Inverse Document Frequency) Vectorizer from scikit-learn is used in content-based filtering to transform textual data into numerical feature vectors. This process involves calculating the TF-IDF scores for each term in the combined textual features of the books. [10] TF-IDF reflects the importance of a term in a document relative to a collection of documents, emphasizing terms that are frequent in a specific document but rare across the entire collection. [6]

Cosine Similarity:

Cosine similarity is used in content-based filtering to measure the similarity between the input preferences provided by the user and the textual representations of the books in the dataset, cosine similarity is employed. This metric calculates the cosine of the angle between two vectors, providing a similarity score ranging from -1 to 1. A higher score indicates a greater similarity between the input preferences and the textual features of the books. [7]

Nearest Neighbors Model:

The Nearest Neighbors model is utilized for collaborative filtering to generate book recommendations based on users' interaction patterns. Specifically used for training the model on the book ratings data to identify similar users or items by measuring the distance between them in a high-dimensional space. By analyzing the rating patterns of users across different books, the model learns to recommend books to a target user based on the preferences of similar users. The NearestNeighbors algorithm, with the 'brute' method, is used for its simplicity and effectiveness in finding nearest neighbors without the need for extensive pre-processing or parameter tuning. Ultimately, this model enables the recommendation system to provide personalized suggestions by leveraging the collective wisdom of users with similar tastes. [3][8]

5. TOOLS USED

Development Tools:

VS Code: Visual Studio Code was utilized as the primary integrated development environment (IDE) for coding tasks, offering a streamlined interface and extensive plugin support.

Flask: Used Flask to seamlessly bridge our recommendation system's frontend and backend, allowing smooth communication between the user interface and the underlying functionality. By defining routes and handling requests with Flask, we ensure efficient data exchange and presentation of recommendations to the user. This integration streamlines the user experience, making it intuitive and responsive. [9]

Libraries/Packages:

Scikit-learn: It is a versatile machine learning library that offers a wide range of tools for classification, regression, and clustering tasks. It implements popular algorithms like k-nearest neighbors and provides efficient data structures for model training and evaluation.

Pandas is a powerful data manipulation library that simplifies tasks such as reading, cleaning, and preprocessing tabular data. It offers intuitive data structures like DataFrame and Series, making it easy to analyze and visualize datasets.

6. CHALLENGES FACED

1. **Data Availability and Quality:** Finding a dataset with user interactions and ratings proved challenging due to scarcity and quality issues, impacting system effectiveness.
2. **Model Selection:** Identifying optimal recommendation models from numerous options required extensive evaluation and experimentation for suitability and performance.
3. **Integration Challenges:** Managing integration with flask for a responsive webpage.

7. CONTRIBUTION OF TEAM MEMBERS

Roll No.	Name	Contribution
21Z207	ALFINA I	Model Selection,Report
21Z225	MADDU HEMALI SAI PRAVALLIKA	Model selection,Backend
21Z267	VASUDHA R B	Backend,Integration
21Z269	VIJAYALAKSHMI P	Front-end,Report
21Z271	YOHASINI U B A	Integration,Report

8. ANNEXURE I: CODE

Content Based Filtering

```
import pandas as pd
from sklearn.feature_extraction.text import TfidfVectorizer
from sklearn.metrics.pairwise import linear_kernel

# Load the dataset
df_books = pd.read_csv('updated_dataset.csv')

# Combine relevant columns to create a textual representation for each book
df_books['combined_features'] = df_books['title'].fillna('') + ' ' + df_books['authors'].fillna('') + ' ' +
df_books['genre'].fillna('')

# Initialize TfidfVectorizer
tfidf_vectorizer = TfidfVectorizer(stop_words='english')

# Drop rows with missing values in the combined_features column
df_books.dropna(subset=['combined_features'], inplace=True)

# Construct the TF-IDF matrix
tfidf_matrix = tfidf_vectorizer.fit_transform(df_books['combined_features'])

# Function to recommend books based on input preferences
def recommend_books(title, author, genre, top_n=10):
    # Create a feature vector for the input
    input_features = tfidf_vectorizer.transform([f'{title} {author} {genre}'])

    # Compute the cosine similarity between the input and all books
    cosine_similarities = linear_kernel(input_features, tfidf_matrix).flatten()

    # Get the indices of top similar books
    top_indices = cosine_similarities.argsort()[-top_n-1:-1][::-1]

    # Return the top recommended books
    recommended_books = df_books.iloc[top_indices][['title', 'authors', 'genre']]
    return recommended_books
```

```
# Input necessary details from the user
title_input = input("Enter the title: ")
author_input = input("Enter the author: ")
genre_input = input("Enter the genre: ")

recommended_books = recommend_books(title_input, author_input, genre_input)
print("Top 10 recommended books:")
print(recommended_books)
```

Collaborative Based Filtering

```
import pandas as pd
import numpy as np
from sklearn.neighbors import NearestNeighbors
from scipy.sparse import csr_matrix
from tabulate import tabulate

# Load the dataset
df_books = pd.read_csv('updated_dataset.csv')

# 'title' is the column containing book titles
book_pivot = df_books.pivot_table(index='title', columns='authors', values='average_rating').fillna(0)

# Convert to sparse matrix
book_sparse = csr_matrix(book_pivot.values)

# Initialize Nearest Neighbors model
model = NearestNeighbors(algorithm='brute')
model.fit(book_sparse)

def recommend_book():
    title = input("Enter the title of the book you are interested in: ")
    author = input("Enter the author of the book you are interested in: ")
    genre = input("Enter the genre of the book you are interested in: ")

    # Combine inputs to search for the book
    book_query = df_books[(df_books['title'].str.lower() == title.lower()) &
                          (df_books['authors'].str.lower() == author.lower()) &
                          (df_books['genre'].str.lower() == genre.lower())]
```

```
  if not book_query.empty:
    book_name = book_query.iloc[0]['title']
    book_id = np.where(book_pivot.index == book_name)[0][0]
          distance,  suggestion  =  model.kneighbors(book_pivot.iloc[book_id,  :].values.reshape(1,  -1),
n_neighbors=10)

    print(f"You searched for '{book_name}' by {author} in the genre {genre}\n")
    print("The suggested books are: \n")
    table = []
    for i in range(len(suggestion)):
      books = book_pivot.index[suggestion[i]]
      for book in books:
        if book != book_name:
                        author_genre  =  df_books[(df_books['title']  ==  book)]['authors'].values[0],
df_books[(df_books['title'] == book)]['genre'].values[0]
          table.append([book, author_genre[0], author_genre[1]])

    print(tabulate(table, headers=['Book', 'Author', 'Genre'], tablefmt='grid'))
  else:
    print("Book not found in the dataset.")

recommend_book()
```

Index.html (for input form)

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Book Recommendation System</title>
    <style>
        body {
            background-color: antiquewhite;
            background-position: center;
            display: flex;
            justify-content: center;
            align-items: flex-start; /* Align items to the top */
            min-height: 100vh; /* Use min-height instead of height */
            margin: 0;
        }

        form {
             background-color: rgba(255, 255, 255, 0.8); /* Adjust opacity
as needed */
            padding: 40px; /* Increase padding to increase size */
            border-radius: 10px;
            box-shadow: 0 0 10px rgba(0, 0, 0, 0.3);
            width: 400px; /* Optional: Set explicit width */
            margin-left: 10px; /* Optional: Center horizontally */
            margin-top: 150px;
            margin-right: 340px;
        }

        h1 {
            text-align:center;
            color: rgb(37, 34, 34); /* Change color as needed */
        }

        label {
            display: block;
            margin-bottom: 10px;
            color: #333; /* Change color as needed */
        }

        input[type="text"] {
            width: 100%;
            padding: 10px;
            margin-bottom: 20px;
            border: 1px solid #ccc; /* Change border color as needed */
            border-radius: 5px;
        }

        button[type="submit"] {
            background-color: #2ec78a; /* Change button color as needed */
            color: white;
            padding: 10px 20px;
            border: none;
            border-radius: 5px;
            cursor: pointer;
        }

        button[type="submit"]:hover {
             background-color: #2ec78a; ; /* Change button hover color as
needed */
        }
    </style>
</head>
<body>
    <h1>Book Recommendation System</h1>
    <form action="/recommend" method="post">
        <label for="title">Title:</label>
        <input type="text" id="title" name="title"><br><br>

        <label for="author">Author:</label>
        <input type="text" id="author" name="author"><br><br>

        <label for="genre">Genre:</label>
        <input type="text" id="genre" name="genre"><br><br>

        <button type="submit">Recommend</button>
    </form>
</body>
</html>
```

Recommendations.html (for displaying recommendations)

```
<!DOCTYPE html>
<html lang="en">
<head>
    <meta charset="UTF-8">
    <meta name="viewport" content="width=device-width, initial-scale=1.0">
    <title>Book Recommendations</title>
    <style>
        body {
            font-family: Arial, sans-serif; /* Change font to Arial */
            background-color: antiquewhite; /* Set background color */
            padding: 20px; /* Add some padding for better readability */
        }
        h1, h2 {
            color: rgb(37, 34, 34); /* Set color for headings */
        }

        ul {
            list-style-type: none; /* Remove default list style */
            padding: 0;
        }

        li {
            margin-bottom: 10px; /* Add spacing between list items */
        }
    </style>
</head>
<body>
    <h1>Book Recommendations</h1>

         <h2>Recommendations  based  on  your  past  book  choices  using
Content-Based Recommendations</h2>
    <ul>
        {% for book in recommendations['content_based'] %}
                   <li>{{  book['title']  }}  by  {{  book['authors']  }}  ({{
book['genre'] }})</li>
        {% endfor %}
    </ul>

       <h2>Recommendations  based  on  similar  users  using  Collaborative
Filtering Recommendations</h2>
    <ul>
        {% for book in recommendations['collaborative'] %}
                   <li>{{  book['title']  }}  by  {{  book['authors']  }}  ({{
book['genre'] }})</li>
        {% endfor %}
    </ul>
</body>
</html>
```

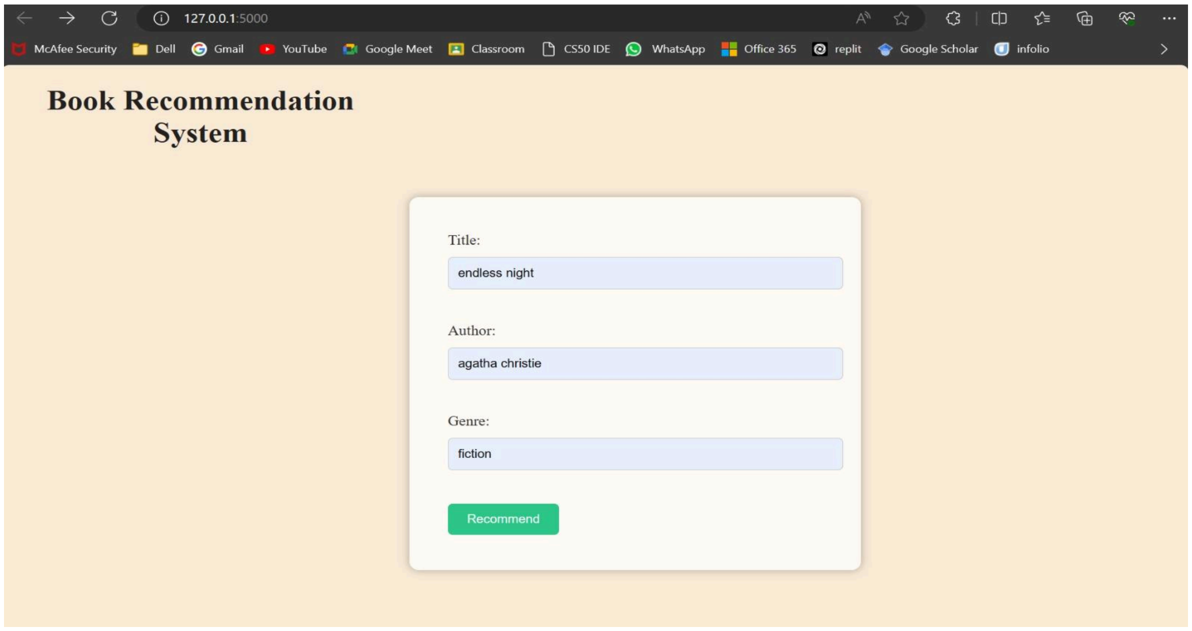
app.py (flask code for connecting front-end and back-end)

```
from flask import Flask, render_template, request
from content import recommend_books
from knn import recommend_book

app = Flask(__name__)

# Route for the main page with the form
@app.route('/')
def index():
    return render_template('index.html')

# Route to handle form submission and display recommendations
@app.route('/recommend', methods=['POST'])
@app.route('/recommend', methods=['POST'])
def recommend():
    # Get input data from the form
    title = request.form['title']
    author = request.form['author']
    genre = request.form['genre']

    # Call content-based filtering script
    content_based_results = recommend_books(title, author, genre)

    # Convert DataFrame to list of dictionaries
    recommended_books = content_based_results.to_dict(orient='records')

    # Call collaborative filtering script
    collaborative_results = recommend_book(title, author, genre)

    # Combine and format recommendations
    recommendations = {
        'content_based': recommended_books,  # Update variable name
        'collaborative': collaborative_results
    }

    # Render the template with the recommended books
            rendered_template   =   render_template('recommendations.html',
recommendations=recommendations)
```

```
    return rendered_template

if __name__ == '__main__':
    app.run(debug=True)
```

9. ANNEXURE II: SNAPSHOTS OF THE OUTPUT

Frontend:

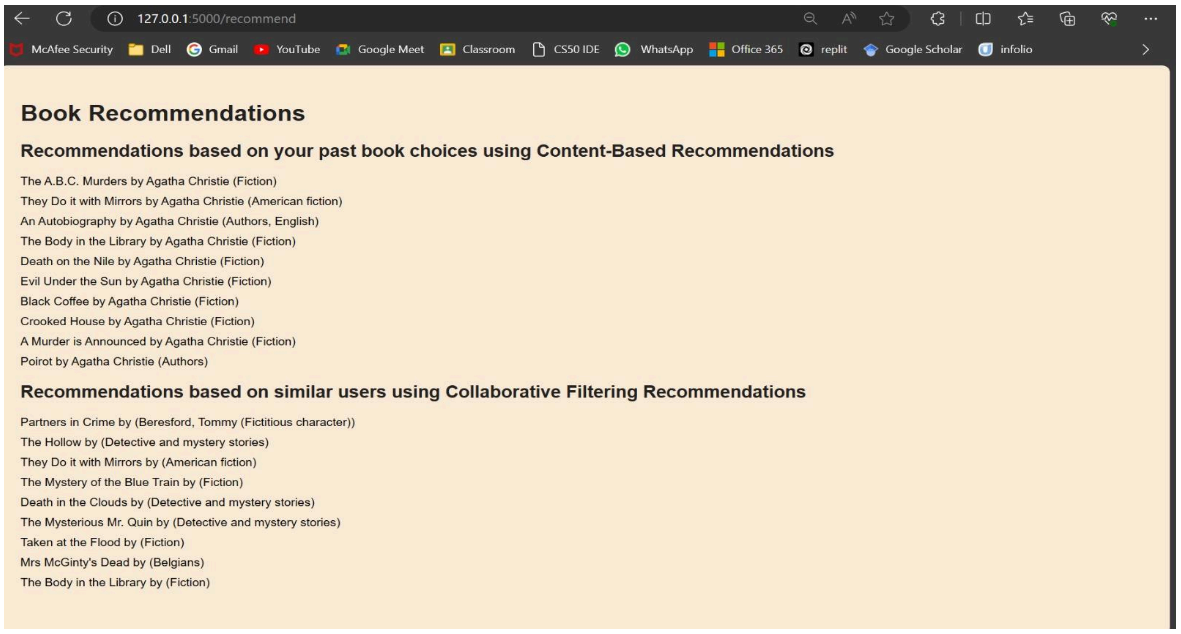


Book Recommendation System

Title: endless night

Author: agatha christie

Genre: fiction



Book Recommendations

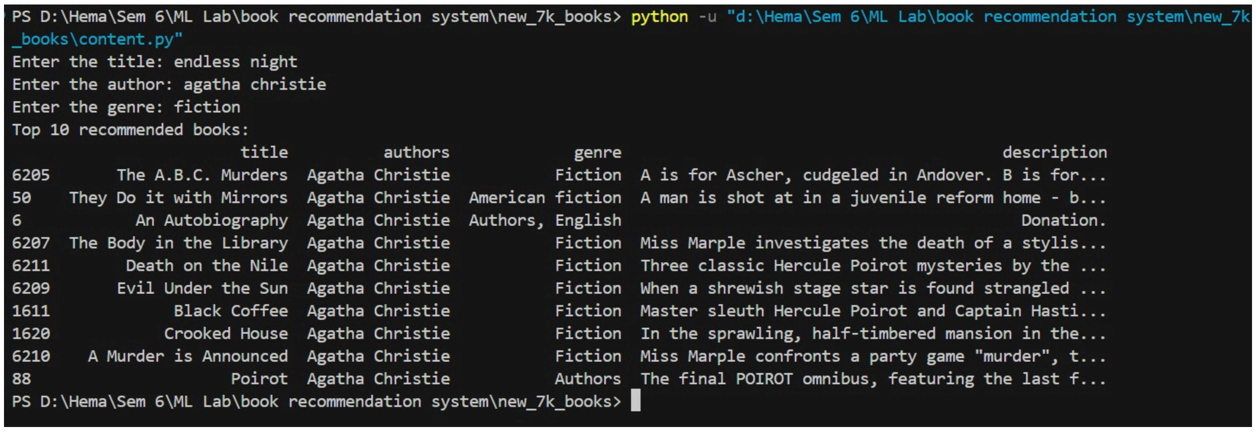
Recommendations based on your past book choices using Content-Based Recommendations

```
The A.B.C. Murders by Agatha Christie (Fiction)
They Do it with Mirrors by Agatha Christie (American fiction)
An Autobiography by Agatha Christie (Authors, English)
The Body in the Library by Agatha Christie (Fiction)
Death on the Nile by Agatha Christie (Fiction)
Evil Under the Sun by Agatha Christie (Fiction)
Black Coffee by Agatha Christie (Fiction)
Crooked House by Agatha Christie (Fiction)
A Murder is Announced by Agatha Christie (Fiction)
Poirot by Agatha Christie (Authors)
```

Recommendations based on similar users using Collaborative Filtering Recommendations

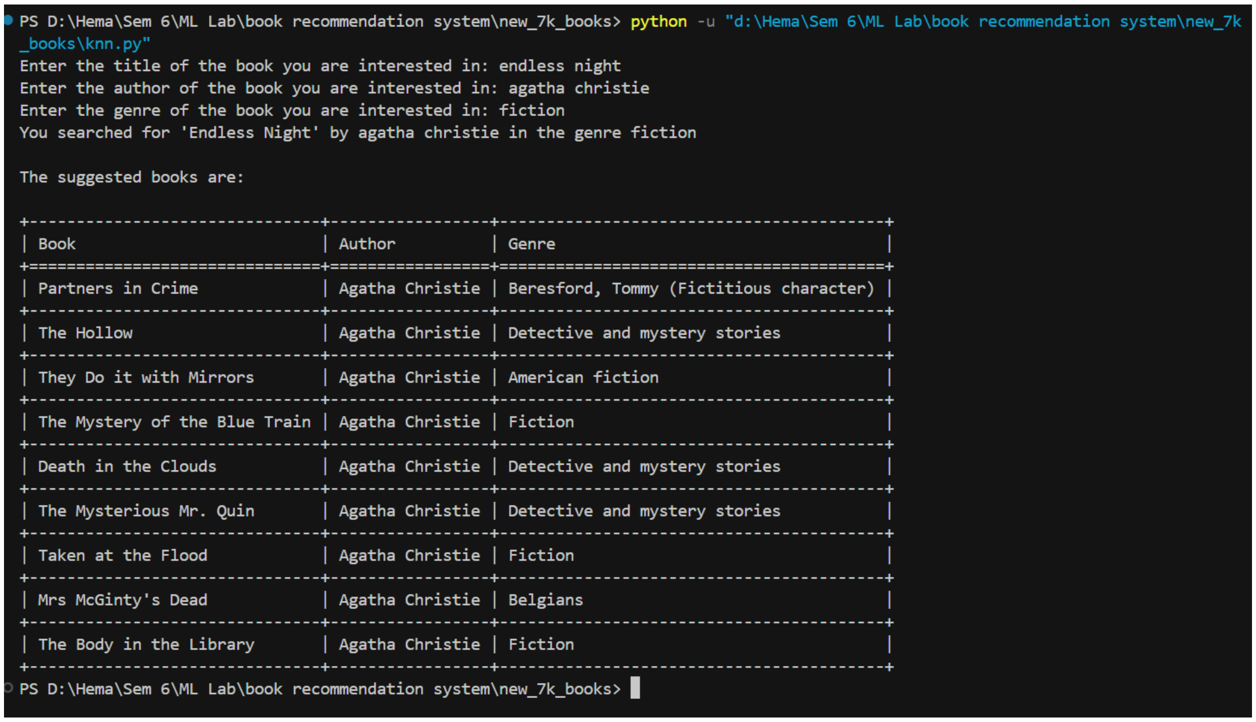
```
Partners in Crime by (Beresford, Tommy (Fictitious character))
The Hollow by (Detective and mystery stories)
They Do it with Mirrors by (American fiction)
The Mystery of the Blue Train by (Fiction)
Death in the Clouds by (Detective and mystery stories)
The Mysterious Mr. Quin by (Detective and mystery stories)
Taken at the Flood by (Fiction)
Mrs McGinty's Dead by (Belgians)
The Body in the Library by (Fiction)
```

Content-based Filtering - Backend:



```
PS D:\Hema\Sem 6\ML Lab\book recommendation system\new_7k_books> python -u "d:\Hema\Sem 6\ML Lab\book recommendation system\new_7k
_books\content.py"
Enter the title: endless night
Enter the author: agatha christie
Enter the genre: fiction
Top 10 recommended books:
                        title          authors              genre                                       description
6205       The A.B.C. Murders  Agatha Christie            Fiction  A is for Ascher, cudgeled in Andover. B is for...
50     They Do it with Mirrors  Agatha Christie  American fiction  A man is shot at in a juvenile reform home - b...
6             An Autobiography  Agatha Christie  Authors, English                                          Donation.
6207  The Body in the Library  Agatha Christie            Fiction  Miss Marple investigates the death of a stylis...
6211         Death on the Nile  Agatha Christie            Fiction  Three classic Hercule Poirot mysteries by the ...
6209        Evil Under the Sun  Agatha Christie            Fiction  When a shrewish stage star is found strangled ...
1611             Black Coffee  Agatha Christie            Fiction  Master sleuth Hercule Poirot and Captain Hasti...
1620            Crooked House  Agatha Christie            Fiction  In the sprawling, half-timbered mansion in the...
6210    A Murder is Announced  Agatha Christie            Fiction  Miss Marple confronts a party game "murder", t...
88                     Poirot  Agatha Christie            Authors  The final POIROT omnibus, featuring the last f...
PS D:\Hema\Sem 6\ML Lab\book recommendation system\new_7k_books>
```

Collaborative Filtering - Backend:



```
PS D:\Hema\Sem 6\ML Lab\book recommendation system\new_7k_books> python -u "d:\Hema\Sem 6\ML Lab\book recommendation system\new_7k
_books\knn.py"
Enter the title of the book you are interested in: endless night
Enter the author of the book you are interested in: agatha christie
Enter the genre of the book you are interested in: fiction
You searched for 'Endless Night' by agatha christie in the genre fiction

The suggested books are:

+-------------------------------+-----------------+----------------------------------------+
| Book                          | Author          | Genre                                  |
+===============================+=================+========================================+
| Partners in Crime             | Agatha Christie | Beresford, Tommy (Fictitious character) |
+-------------------------------+-----------------+----------------------------------------+
| The Hollow                    | Agatha Christie | Detective and mystery stories          |
+-------------------------------+-----------------+----------------------------------------+
| They Do it with Mirrors       | Agatha Christie | American fiction                       |
+-------------------------------+-----------------+----------------------------------------+
| The Mystery of the Blue Train | Agatha Christie | Fiction                                |
+-------------------------------+-----------------+----------------------------------------+
| Death in the Clouds           | Agatha Christie | Detective and mystery stories          |
+-------------------------------+-----------------+----------------------------------------+
| The Mysterious Mr. Quin       | Agatha Christie | Detective and mystery stories          |
+-------------------------------+-----------------+----------------------------------------+
| Taken at the Flood            | Agatha Christie | Fiction                                |
+-------------------------------+-----------------+----------------------------------------+
| Mrs McGinty's Dead            | Agatha Christie | Belgians                               |
+-------------------------------+-----------------+----------------------------------------+
| The Body in the Library       | Agatha Christie | Fiction                                |
+-------------------------------+-----------------+----------------------------------------+
PS D:\Hema\Sem 6\ML Lab\book recommendation system\new_7k_books>
```

10. REFERENCES

[1] Collaborative Filtering Recommender Systems J. Ben Schafer1 , Dan Frankowski2 , Jon Herlocker3 , and Shilad Sen2 1 Department of Computer Science University of Northern Iowa Cedar Falls, IA 50614-0507
[2] How to Build a Book Recommendation System
https://www.analyticsvidhya.com/blog/2021/06/build-book-recommendation-systEm-unsupervised-learning-project/
[3] KNN: K-NEAREST NEIGHBOURS CS456 - MACHINE LEARNING SPRING 2023 Rahul Vishwakarma, Jyothish Kumar J School of Computer Sciences, National Institute of Science Education and Research, Bhubaneshwar, Homi Bhabha National Institute
[4] Dataset Link:
https://www.kaggle.com/datasets/ra4u12/bookrecommendation?resource=download
[5] Using Content-Based Filtering for Recommendation1 1 This research has been supported by NetlinQ Robin van Meteren1 and Maarten van Someren2 1 NetlinQ Group, Gerard Brandtstraat 26-28, 1054 JK, Amsterdam, The Netherlands.
[6] TF-IDF Vectorizer: International Conference on System Modeling & Advancement in Research Trends (SMART) **-** Movie Recommendation System Using TF-IDF Vectorizer and Bag of Words Manika Manwal, Divyansh Rawat, Deekshant Rawat, Kamlesh Chandra Purohit, Tanupriya Choudhury
[7] Cosine Similarity: International Journal of Advanced Computer Science and Applications - Personalized Book Recommendation System using Machine Learning Algorithm
Dhiman Sarma, Tanni Mittra, M. Shahadat
[8] Introduction to k Nearest Neighbour Classification and Condensed Nearest Neighbour Data Reduction Oliver Sutton
[9] Flask Web Development by Miguel Grinberg Copyright © 2014 Miguel Grinberg. All rights reserved. Printed in the United States of America. Published by O'Reilly Media, Inc., 1005 Gravenstein Highway North, Sebastopol, CA 95472.
[10] TF-IDF Vectorizer: Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., ... & Vanderplas, J. (2011). Scikit-learn: Machine learning in Python. Journal of machine learning research, 12(Oct), 2825-2830.