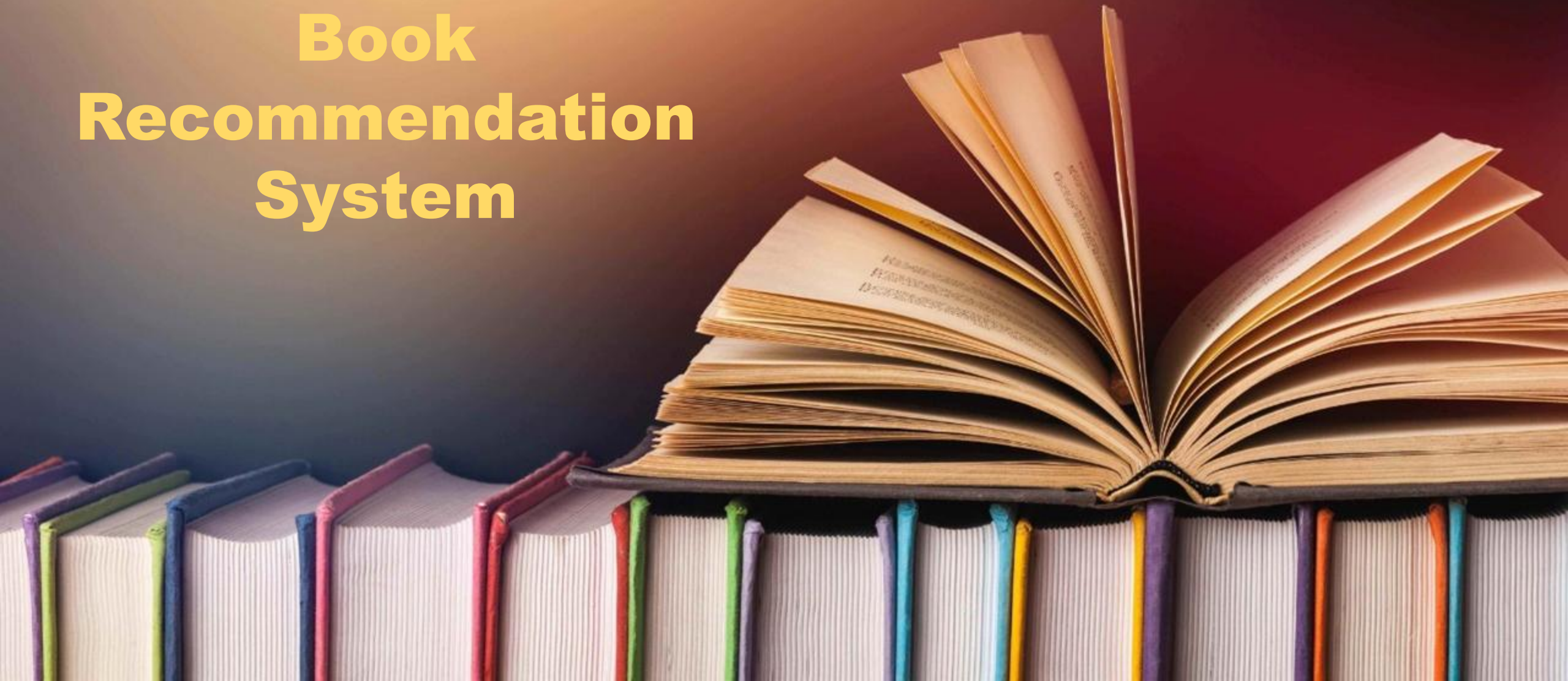
Book Recommendation System



AGENDA

An Intelligent Approach To Personalized Book Suggestions

Prepared By : Group 4

GOKILA G

SUSHANTH SHETTY

BALU SRIVIDYA

Contents

Project Architecture

Datasets	• Import Libraries • Load Datasets • Explore Data Summary
Data cleaning	• Handle Missing Values • Remove Duplicates • Fix Data Types
Data Preprocessing /EDA	• Feature Scaling • Visualize Data Patterns
Model Building	• Select Algorithms • Train the Model
Model Evaluation	• Evaluate Accuracy • Cross-Validation
Model Deployment	• Deploy Model to Production • Update Model as Needed

INTRODUCTION TO RECOMMENDATION SYSTEM

- A BOOK RECOMMENDATION SYSTEM AIMS T0 SUGGEST BOOKS TO USERS BASED ON THEIR INTERESTS, READING HISTORY, AND PREFERENCES. WITH THE VAST NUMBER OF BOOKS AVAILABLE, SUCH SYSTEMS HELP USERS DISCOVER NEW BOOKS THEY MAY ENJOY, ENHANCING THEIR READING EXPERIENCE.
- BY ANALYZING DATA SUCH AS USER RATINGS, BOOK DETAILS, AND USER INTERACTIONS, THESE SYSTEMS CREATE PERSONALIZED SUGGESTIONS THAT INCREASE USER ENGAGEMENT AND SATISFACTION. GIVE IN A POINT WISE.

Dataset Details

BOOK DATASET

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 271360 entries, 0 to 271359
Data columns (total 8 columns):
 #   Column               Non-Null Count   Dtype
---  ------               --------------   -----
 0   ISBN                 271360 non-null  object
 1   Book-Title           271360 non-null  object
 2   Book-Author          271358 non-null  object
 3   Year-Of-Publication  271360 non-null  object
 4   Publisher            271358 non-null  object
 5   Image-URL-S          271360 non-null  object
 6   Image-URL-M          271360 non-null  object
 7   Image-URL-L          271357 non-null  object
dtypes: object(8)
memory usage: 16.6+ MB
```

USER DATASET

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 278858 entries, 0 to 278857
Data columns (total 3 columns):
 #   Column    Non-Null Count   Dtype
---  ------    --------------   -----
 0   User-ID   278858 non-null  int64
 1   Location  278858 non-null  object
 2   Age       168096 non-null  float64
dtypes: float64(1), int64(1), object(1)
memory usage: 6.4+ MB
```

RATING DATASET

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 1149780 entries, 0 to 1149779
Data columns (total 3 columns):
 #   Column       Non-Null Count    Dtype
---  ------       --------------    -----
 0   User-ID      1149780 non-null  int64
 1   ISBN         1149780 non-null  object
 2   Book-Rating  1149780 non-null  int64
dtypes: int64(2), object(1)
memory usage: 26.3+ MB
```

Data Preprocessing And EDA

In Book Dataset

CLEAN DATA: Handle missing values, duplicates, and fix data types.

ANALYZE FEATURES: Examine distributions of ratings, genres, and authors.

USER-BOOK INSIGHTS: Study user activity and book popularity.

VISUALIZE TRENDS: USE CHARTS TO REVEAL KEY PATTERNS.

In User Dataset

CLEAN DATA: Handle missing values, remove duplicates, and correct data types.

ANALYZE FEATURES: Explore user activity, ratings, and demographics.

VISUALIZE TRENDS: Use charts to uncover patterns in user behavior.

In Rating Dataset

CLEAN DATA: Identify and address missing or invalid ratings and remove duplicates.

EXPLORE RATINGS: Analyze trends like average ratings, most frequent ratings, and variations.

VISUALIZE INSIGHTS: Use heatmaps or bar charts to highlight top-rated books and active users.

Details About Recommendation Techniques

1. **COSINE SIMILARITY FOR USER SIMILARITY :**
   - Measures the angle between user rating vectors to find similar users.
   - Helps recommend items preferred by users with similar preferences.
2. **COSINE SIMILARITY FOR ITEM SIMILARITY :**
   - Calculates the similarity between item vectors based on user ratings.
   - Suggests items that are most similar to those already interacted with.
3. **CONTENT-BASED RECOMMENDATIONS :**
   - Uses cosine similarity on item attributes (e.G., Genres, keywords) to recommend items with high feature overlap.

MODEL SELECTION

- **MODEL SELECTION:** CHOOSING THE BEST MODEL BASED ON PERFORMANCE METRICS.
- **EVALUATION METRICS:** USING ACCURACY, PRECISION, RECALL, ETC., TO ASSESS MODEL PERFORMANCE.
- **CROSS-VALIDATION:** Validating models with k-fold cross-validation to ensure robustness.
- **HYPERPARAMETER TUNING:** Using grid search or random search to optimize model parameters.
- **COMPARATIVE ANALYSIS:** Comparing multiple models to choose the best performer.

MODEL DEPLOYMENT

- Model deployment is the process of integrating a trained machine learning model into a production environment to provide predictions or insights to end-users or systems.
- It involves setting up the infrastructure, creating apis or user interfaces, and ensuring scalability and reliability for real-world usage.
- Deployment platforms like streamlit, flask, aws, azure, or docker are commonly used for deploying models as web applications or services.
- Monitoring and updating the model post-deployment are critical to maintaining its performance over time.

Using Streamlit We Have Deployed Our Application

```
import pickle
import streamlit as st
import pandas as pd
import numpy as np
import warnings
warnings.filterwarnings('ignore')
```

```
st.title('Book Recommendation System')  ## adding title
```

```
DeltaGenerator()
```

```
## loading the files
df_new=pickle.load(open('df_new.pkl','rb'))
df=pickle.load(open('df1.pkl','rb'))
similarity_scores=pickle.load(open('similarity_scores.pkl','rb'))
```

```
def recommend(book_name):
    # Ensure index consistency
    df_new.index = df_new.index.astype(str).str.strip()
    df['Book-Title'] = df['Book-Title'].astype(str).str.strip()

    # fetch the index of the book from the pivot table
    index = np.where(df_new.index == book_name)[0][0]

    # getting similar suggestions with greater similarity score,
    similar_items = sorted(list(enumerate(similarity_scores[index])),
                           key=lambda x: x[1],  # sort by similarity score
                           reverse=True)[1:6]  # exclude the input book

    data = []
    for i in similar_items:
        if i[0] < len(df_new.index):  # Ensure index is within range
            book_title = df_new.index[i[0]]  # Get the book title safely
            item = []
            temp_df = df[df['Book-Title'] == book_title]
            item.extend(list(temp_df.drop_duplicates('Book-Title')['Book-Title'].values))
            item.extend(list(temp_df.drop_duplicates('Book-Title')['Book-Author'].values))
            data.append(item)
        else:
            print(f"Index {i[0]} is out of bounds for df_new.index.")

    return pd.DataFrame(data, columns=['Book', 'Author'])
```

```
book_list=df_new.index.values
```

```
selected_book=st.selectbox('Type or select a book from the dropdown',book_list)
```

```
if st.button('Show Recommendation'):
    recommended_books=recommend(selected_book)
    recommended_books
```

STREAMLIT INTERFACE :

Book Recommendation System

Type or select a book from the dropdown

10 Lb. Penalty

Show Recommendation

	Book	Author
0	Devil's Waltz (Alex Delaware Novels (Paperback))	Jonathan Kellerman
1	Communion : A True Story	Whitley Strieber
2	Apollo 13 : Lost Moon	Jim Lovell
3	The Cat Who Came to Breakfast	Lilian Jackson Braun
4	Lucy Sullivan Is Getting Married	Marian Keyes

Top 10 popular books based on ratings

index	Book-Title	Book-Author	Book-Rating
1	The Poisonwood Bible: A Novel	Barbara Kingsolver	10
2	On Writing	Stephen King	10
3	Zen and the Art of Motorcycle Maintenance: An Inquiry into Values	ROBERT PIRSIG	10
4	Memnoch the Devil : The Vampire Chronicles (Vampire Chronicles)	ANNE RICE	10
5	The Power of One	Bryce Courtenay	10

Challenges in Building a Book Recommendation System

- **COLD START PROBLEM:** Difficulty in recommending to new users or books with little data.
- **DATA SPARSITY:** Not enough interaction data to make accurate recommendations.
- **SCALABILITY:** Handling large volumes of users and books efficiently.

Thank You