Book Recommendation System Using Collaborative Filtering

Presented by
Himasree Deka
246102008

Course name : Python Lab
Course code : DA514

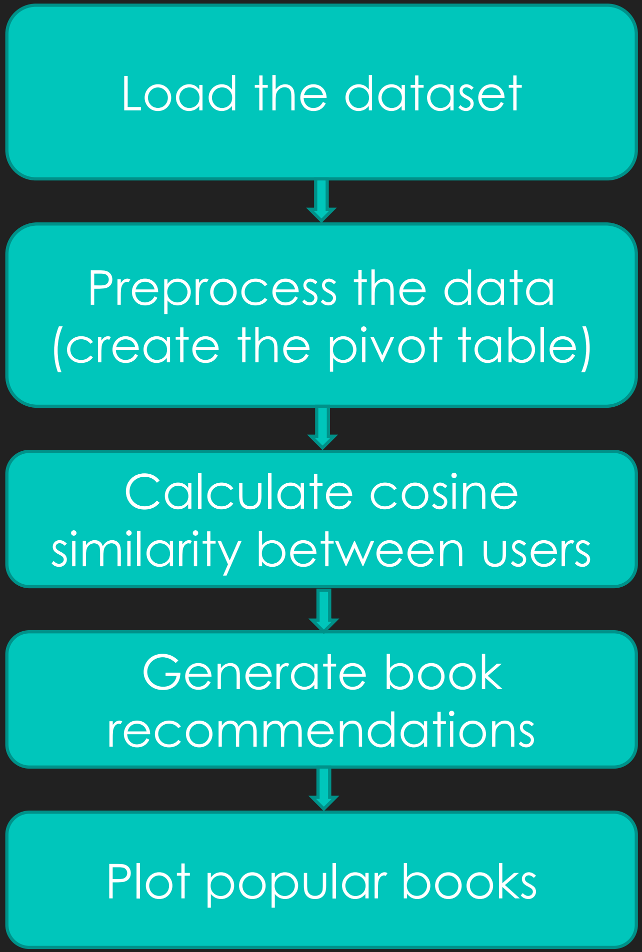
Indian Institute of Technology Guwahati
Guwahati, India 781039

Introduction/Motivation

1. **Personalized Experience**: Helps users discover books based on preferences, saving time and improving user satisfaction.
2. **Overcome Information Overload**: Recommends relevant books, reducing the overwhelm of choosing from a large catalog.
3. **Scalability**: Grows with the dataset, adapting as more users rate books, offering dynamic recommendations.
4. **Collaborative Filtering**: Uses user similarity to recommend books, based on the idea that people with similar tastes like similar books.
5. **Real-World Applications**: Similar systems used by platforms like Amazon and Goodreads, demonstrating real-world value.
6. **Challenges**: Solves problems like cold start (new users/books) and sparse data by leveraging collaborative filtering.
7. **Potential for Improvement**: Can be expanded with hybrid models, content-based filtering, and real-time feedback for better recommendations.

Approach Overview(Block Diagram)

- Steps involved in the book recommendation system:



Pseudocode

- Load book rating data from CSV.

```
def load_data(file_path):
```

- Preprocess data into user-book rating matrix.

```
def preprocess_data(data):
```

- Calculate cosine similarity between users.

```
def calculate_similarity(pivot_table):
```

- Generate book recommendations based on similar users.

```
def get_recommendations(user_id, similarity_matrix, pivot_table, num_recommendations=5):
```

- Display top recommendations.
- Visualize the most popular books

```
def plot_popular_books(data):
```

Step 1: Loading Data

- The data is loaded from a CSV file using pandas:
- data = pd.read_csv(file_path)

UserID	BookTitle	Rating
1	To Kill a Mockingbird	5
1	1984	4
1	The Great Gatsby	3
1	The Catcher in the Rye	4
2	To Kill a Mockingbird	4
2	1984	5
2	Pride and Prejudice	4
2	The Hobbit	5
3	To Kill a Mockingbird	2
3	The Great Gatsby	5
3	The Catcher in the Rye	3
3	The Hobbit	4
4	1984	3
4	The Great Gatsby	4
4	Pride and Prejudice	5
4	The Lord of the Rings	5
5	To Kill a Mockingbird	5
5	Pride and Prejudice	3
5	The Lord of the Rings	4
5	The Hobbit	3
6	1984	5
6	The Catcher in the Rye	4
6	The Great Gatsby	4
6	The Lord of the Rings	3

Step 2: Data Preprocessing

- Pivot the data to create a user-item matrix:

```
pivot_table = data.pivot_table(index='UserID', columns='BookTitle', values='Rating')
pivot_table.fillna(0, inplace=True)  # Fill NaN values with 0
```

UserID	1984	Pride and Prejudice	The Catcher in the Rye	The Great Gatsby	The Hobbit	The Lord of the Rings	To Kill a Mockingbird
1	4.0	0.0	4.0	3.0	0.0	0.0	5.0
2	5.0	4.0	0.0	0.0	5.0	0.0	4.0
3	0.0	0.0	3.0	5.0	4.0	0.0	2.0
4	3.0	5.0	0.0	4.0	0.0	5.0	0.0
5	0.0	3.0	0.0	0.0	3.0	4.0	5.0
6	5.0	0.0	4.0	4.0	0.0	3.0	0.0

Step 3: Calculating Cosine Similarity

- Cosine Similarity is calculated as:

  similarity(A, B) = (A . B) / (||A|| ||B||)
  Where A and B are the rating vectors for two users. Example calculation below:
  user1=[4,0,4,3,0,0,5]   user2=[5,4,0,0,5,0,4]

  similarity(user1,user2)=40/(8.12*9.05)=0.544

```
similarity_matrix = cosine_similarity(pivot_table)
```

UserID	1	2	3	4	5	6
1	1.000000	0.543727	0.619773	0.341121	0.400629	0.727273
2	0.543727	1.000000	0.420779	0.446304	0.675717	0.339830
3	0.619773	0.420779	1.000000	0.314270	0.389762	0.536020
4	0.341121	0.446304	0.314270	1.000000	0.526152	0.653816
5	0.400629	0.675717	0.389762	0.526152	1.000000	0.192302
6	0.727273	0.339830	0.536020	0.653816	0.192302	1.000000

Step 4: Generating Recommendations

- Recommendations are generated based on the ratings of similar users.
- For similar users, we take the weighted average of their ratings for unseen books.
- For user 1:

$$\text{Recommendation Score} = \frac{\sum(\text{Similarity of User 1 with other users} \times \text{Rating by other users})}{\sum \text{Similarities of User 1 with other users}}$$

- Recommendation scores:

1984=2.80

The Great Gatsby=2.80

Pride and Prejudice=2.45

The hobbit= 2.43

The Lord of Rings=2.08

The catcher in the Rye=1.81

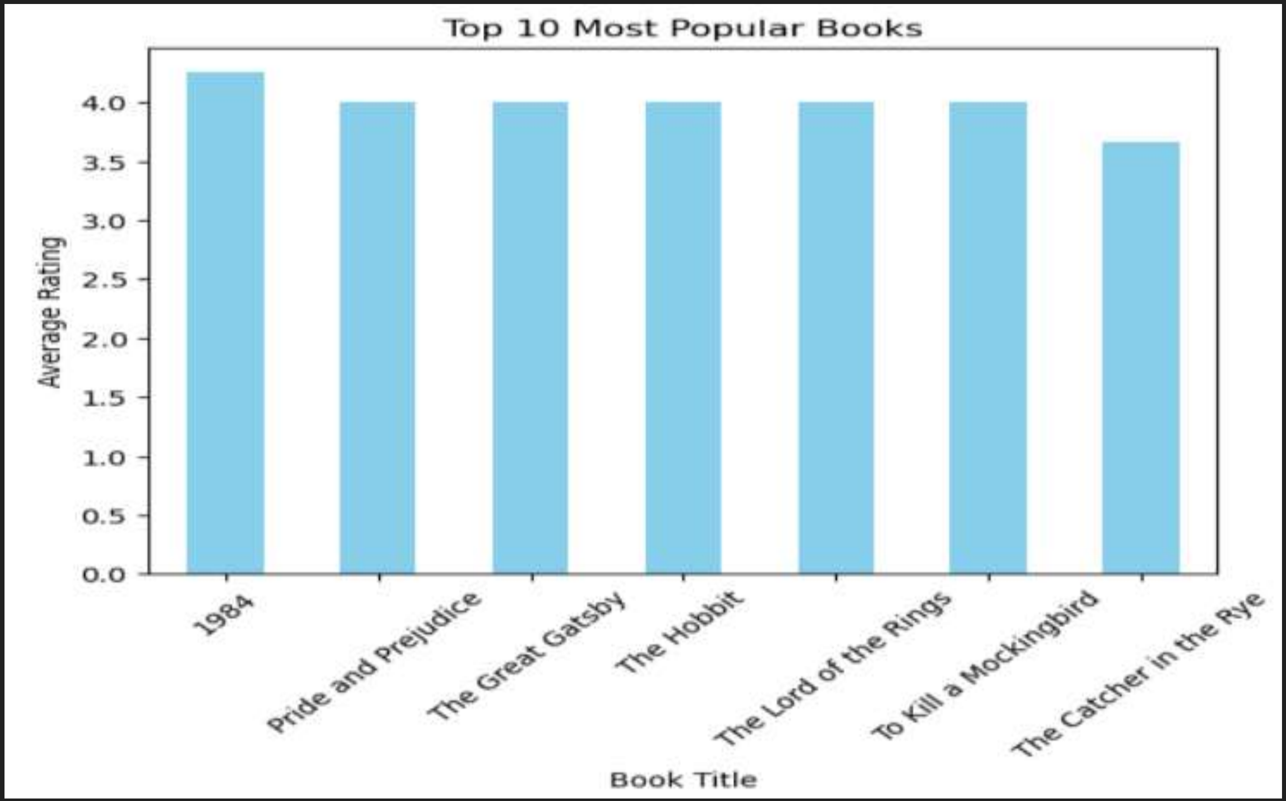
To kill a mockingbird=1.06

```
Top book recommendations for user 1:
1. 1984
2. The Great Gatsby
3. Pride and Prejudice
4. The Hobbit
5. The Lord of the Rings
```

Step 5: Plotting Popular Books

- Plot the most popular books based on average ratings of each book

```
book_popularity = data.groupby('BookTitle')['Rating'].mean().sort_values(ascending=False).head(10)
book_popularity.plot(kind='bar', color='skyblue')
```



Total Run Time=5.7574secs

Conclusion & Future Work

- **Collaborative Filtering**: Implemented user-based collaborative filtering using cosine similarity.
- **Data Preprocessing**: Cleaned and transformed data for use in recommendation algorithms.
- **Python Libraries**: Gained experience using Pandas, Numpy, and Scikit-learn.
- **Recommendation System Design**: Built a personalized recommendation system.
- **Visualization**: Created visual insights using Matplotlib (bar charts for popular books).