Name of Project
Cognitive Computing Project
(UCS420)

Made By: -

Name	Roll Number
Mishika Sardana	102317282
Daksh Jain	102497020
Aryaman Gudwani	102317279

Submitted To: -

Mr. Sukhpal Singh

Department of Computer Science and Engineering,

THAPAR INSTITUTE OF ENGINEERING AND TECHNOLOGY,
(A DEEMED TO BE UNIVERSITY),
PATIALA, PUNJAB, INDIA

Jan-May, 2025

An Advanced Book Recommendation Model:

1. Introduction & Problem Statement

In today's digital world, many choices can be overwhelming, particularly for book lovers who filter through vast catalogues, unsure of what to read next. We have made an intelligent book recommendation system that uses machine learning and collaborative filtering to address this challenge. This model streamlines the process of finding books, making it easier for users to find titles that align with their interests.

Why is it Important?

It effortlessly helps users discover books that match their preferences. This reduces the cognitive burden of sifting through thousands of options and enhances user experience and engagement on digital reading platforms.

Dataset Description

We used the Book Recommendation Dataset from Kaggle.It has three main components:

- Books Data: Contains key metadata about books.
- Users Data: Provides demographic details of users.
- Ratings Data: Represents user interactions with books through ratings.

Dataset Link: https://www.kaggle.com/datasets/arashnic/book-recommendation-dataset?resource=download&select=Ratings.csv

2. Data Exploration & Pre-processing

Dataset Details:

We used the Book Recommendation Dataset from Kaggle.

- Books Data:
  No of rows - 271361
  No of Columns - 8
    - ISBN, Book-Title
    - Book-Author
    - year of publication
    - publisher
    - Image-URL-S
    - Image-URL-M
    - Image-URL-L
- Users Data:
  No of rows – 278860
  No of columns - 3
    - User IDs
    - Locations
    - Age
- Ratings Data:
  No of rows – 1149781
  No of columns – 3
    - User-ID

- ISBN
- Book-Rating

This dataset serves as the foundation for training and evaluating our recommendation model.

Pre-processing Steps

To ensure data quality, we performed the following:

- Handling Missing Data: Filled 0 in missing values for book titles, authors, and user demographics where possible.
- Data Cleaning: Standardized publication years and removed inconsistencies.
- Feature Engineering: Mapped ISBNs to book details and filtered out books with minimal interactions.

3. Model Implementation & Evaluation

Technologies Used

To build our recommendation system, we have used:

- NumPy: For numerical computations.
- Pandas: For data manipulation and analysis.
- Matplotlib & Seaborn: This is used to visualize data trends
- Scikit-learn: For implementing machine learning algorithms.
- K-Nearest Neighbors **(**KNN**)**: For similarity-based recommendations.
- Surprise Library: For collaborative filtering techniques.

Model Approach

We implemented collaborative filtering in two ways:

- User-Based Filtering: Finds users those who have read more than 200 books.
- Item-Based Filtering: Recommends books having more than 50 reviews.

Performance Metrics

To evaluate our model, we used:

- Cosine Similarity: Measures the closeness between users or books.
- KNN Algorithm: Identifies the most relevant neighbours to generate recommendations.
- Root Mean Square Error (RMSE): Assesses prediction accuracy (achieved RMSE: **0.8054**).
- Precision-Recall Analysis: Evaluates recommendation relevance.

4. Results & Insights

Observations & Visualizations

- Popularity Trends: Recommended books with more than 50 reviews.
- User Analysis: Recommended user that have read more than 200 books.

Sample Recommendations

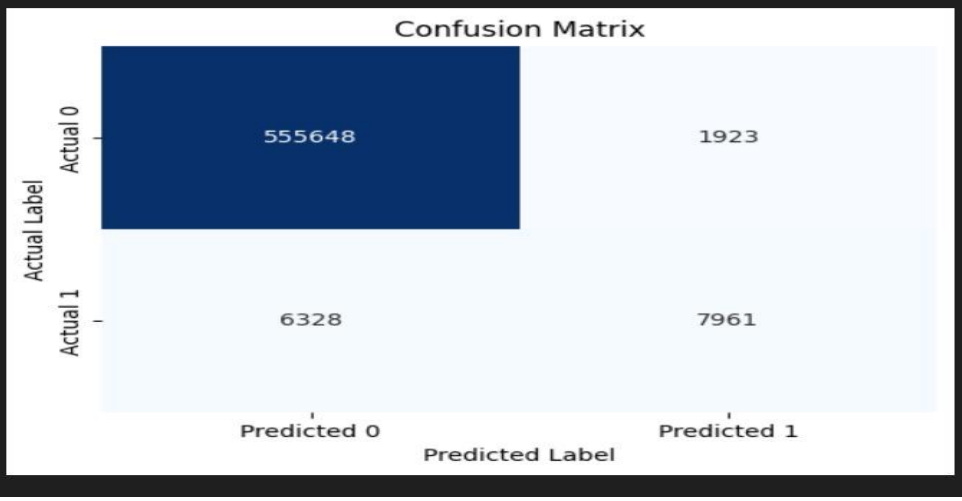
For instance, if a user likes *The Notebook*, our model suggests book id and author with it:

- 226 , Nicolas Sparks
- 4173 , Dean R. Koontz
- 995 , Robin Cook

```
Best RMSE: 3.2497935602680403
Best MAE: 2.6221566729872614
Best Parameters: {'n_epochs': 20, 'lr_all': 0.005, 'reg_all': 0.4}

<surprise.prediction_algorithms.matrix_factorization.SVD at 0x7ec3d1657490>
```

```
Precision@14: 0.8054
Recall@14: 0.6718
Accuracy@14: 0.8054
F1-Score: 0.7325772142168451
```



5. Challenges & Future Enhancements

Issues Encountered

- Cold Start Problem: New books and users lack sufficient historical data.
- Sparse Rating Matrix: Some books had too few ratings, affecting similarity calculations.
- Computational Complexity: KNN-based similarity measures were resource-intensive on large datasets.

Potential Improvements

- Hybrid Model: Combine content-based and collaborative filtering for better recommendations.

6. Conclusion & Key Takeaways

This project demonstrates how collaborative filtering can improve book recommendations, reducing decision fatigue. Our user-based and item-based approaches yielded promising results. Looking ahead, implementing hybrid models and real-time recommendations will enhance personalization and scalability.

7. References

- Surprise Library Documentation
- Scikit-learn Documentation