**Mini Project Report on**



**Recommender System**



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

**BACHELOR OF TECHNOLOGY**

**IN**

**COMPUTER SCIENCE & ENGINEERING**

**Submitted by:**

**Manas Pant 2018916**

***Under the Mentorship of***

**Mrs. Vishu Tyagi**

**Assistant Professor**



**Department of Computer Science and Engineering**

**Graphic Era (Deemed to be University)**

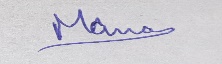
**Dehradun, Uttarakhand**

**January 2024**



**CANDIDATE’S DECLARATION**

I hereby certify that the work which is being presented in the project report entitled **“Recommender System”** in partial fulfillment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science and Engineeringof the Graphic Era (Deemed to be University), Dehradun shall be carried out by the under the mentorship of **Mrs. Vishu Tyagi, Assistant Professor**, Department of Computer Science and Engineering, Graphic Era (Deemed to be University), Dehradun.

Manas Pant 2018916 

**Table of Contents**

|  |  |  |
| --- | --- | --- |
| **Chapter No.** | **Description** | **Page No.** |
| Chapter I | Introduction | 1 |
| Chapter II | Literature Survey | 4 |
| Chapter III | Methodology | 6 |
| Chapter IV | Result and Discussion | 11 |
| Chapter V | Conclusion and Future Work | 13 |
|  | References | 14 |

**Chapter I**

**Introduction**

* 1. **Introduction**

Recommender systems, also known as recommendation systems or engines, are a type of software application that provides personalized suggestions to users. Usually, these recommendations are for goods, services, articles, or other things that the user would find interesting. Many online platforms, such as social networking, streaming services, e-commerce websites, and more, employ recommender systems extensively. While there are various ways to create recommender systems, the most popular ones include content-based filtering, collaborative filtering, and hybrid approaches.

The various types of Recommender Systems are :-

Collaborative Filtering:

The premise behind collaborative filtering is that users who have previously agreed are more likely to do so in the future. It is dependent upon user-item interaction data, including past purchases or user ratings. Two forms of collaborative filtering can be distinguished further:

* User-Centered Cooperation Filtering:

1. User-Item Matrix: Display interactions between users and items as a matrix.
2. Calculate Similarity: Determine how similar two users are to one another by seeing how they interact.
3. Prediction: Estimate a user's interest in a product based on the tastes of users who like them.

* Item-Based Collaborative Filtering:

1. User-Item Matrix: Like user-based, but compares items based on a computed similarity.
2. Calculate Similarity: Determine how similar two things are to one another based on user interactions.
3. Based on the user's interactions with related things, make a prediction about the user's interest in a particular item.

Content-Based Filtering:

Items are recommended using content-based filtering according to user preferences and feature sets. When making suggestions, it considers both the attributes of the objects and the past preferences of the users. Important actions consist of:

* Item Profile: Based on an item's features, create a profile for it.
* User Profile: Based on their preferences, create a profile for every user.
* Matching: Make suggestions for products based on the user's profile.

Popularity-Based:

Popularity-based recommendation systems offer a straightforward and natural way to make suggestions by recommending content that users like or use frequently. Individual user preferences or item attributes are not considered by popularity-based systems, in contrast to collaborative or content-based approaches. Rather, they use indicators related to overall popularity to provide suggestions.

Hybrid Recommender Systems:

Hybrid recommender systems take advantage of the advantages of both content-based and collaborative filtering. Two primary categories of hybrid systems exist:

* Weighted Hybrid: Add scores with varying weights from content-based and collaborative models.
* Hybrid Switching: Employ a single suggestion technique when appropriate, then move to a different one when necessary.

**Chapter II**

**Literature Survey**

Recommender systems have become an essential tool in various domains due to the increasing amount of online information. They help users find relevant and personalized content, thereby reducing the effort and time required for searching the internet [1].

One of the significant advancements in this field is the application of deep learning technology. Unlike traditional recommendation models, deep learning can effectively capture non-linear and non-trivial user-item relationships, enabling the codification of more complex abstractions as data representations in the higher layers [2].

Over the past few years, numerous research articles have been published on recommender systems. A systematic review of these articles reveals various applications of recommender systems, such as web, books, e-learning, tourism, movies, music, e-commerce, news, specialized research resources, television programs, etc. [1].

However, despite their effectiveness, recommender systems face several challenges, including scalability, cold-start, and sparsity. Each technique used in building recommender systems comes with its own set of features, advantages, and disadvantages, making the selection of techniques a complex task [1].

Research in this field is ongoing, with new methods and datasets being explored. Recent literature surveys have provided comprehensive overviews of the current state of research, highlighting existing gaps and challenges. These surveys are instrumental in guiding future research directions and developing more efficient recommender systems [3] [4] [5].

In conclusion, recommender systems have revolutionized the way users interact with online information. The integration of deep learning technology has further enhanced their capabilities. However, there are still many challenges to overcome, and ongoing research continues to explore new methods and techniques to improve these systems [1] [2].

**Chapter III**

**Methodology**

**Necessities:**

The basic requirements that must be present are as follows:-

* Visual Studio Code – It is a code editor by Microsoft. I chose it due to its easy to use and helpful interface and the wide variety of extensions it provides us.[6]
* Python – It is a high level and easy to understand programming language. I chose it due to the wide variety of first-party and third-party libraries it provides the support for.[7]
* Dataset - A dataset is a must to train any model such that it predicts future inputs. The one I used can be downloaded from Kaggle which has a large number of books and related information about the books. There are datasets named users and rating as well which further help in the building of the system.[8]

**1) Data Sets Used:**

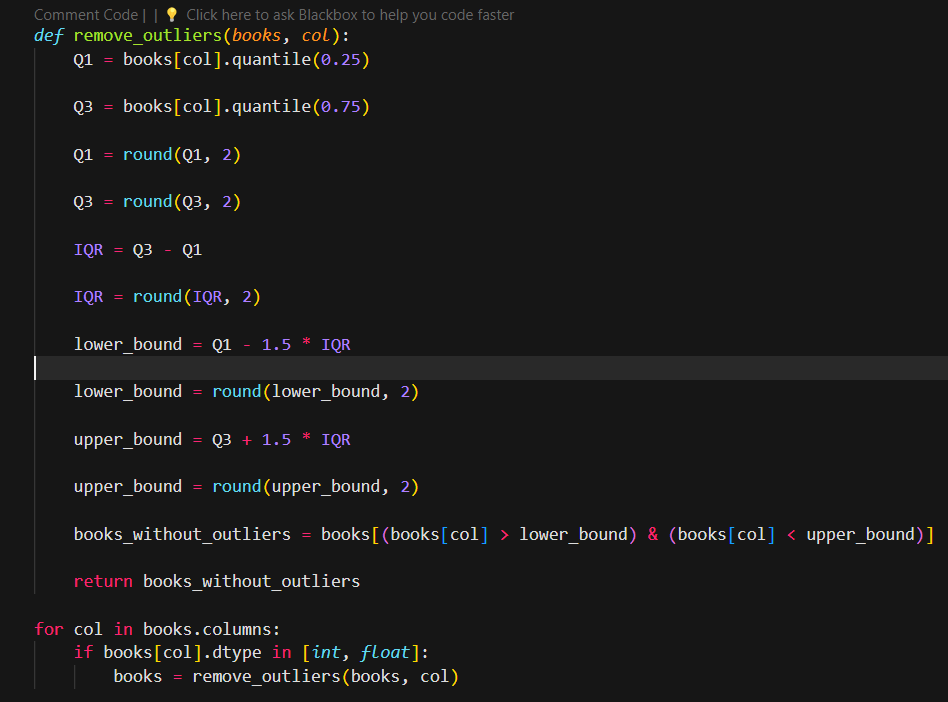
In this project we have created a Book Recommender System which recommends 5 to 6 books which are similar in genre and author to the entered book, for this project we have used 3 datasets :-

* Books : This dataset contains the information about books such as Book Name, Book ISBN, Author info, etc.
* Users : This data set contains the information about users i.e. User-ID, Ratings, etc.
* Ratings : This dataset contains the number of ratings on a book based on ISBN and Book Title.

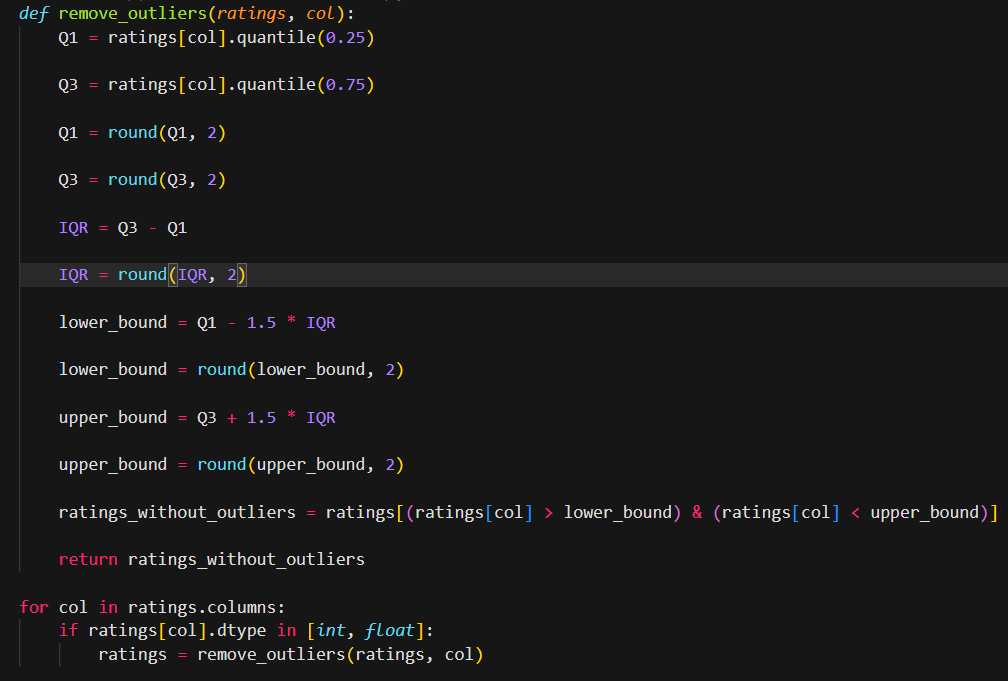
**2) Data Cleaning:**

A crucial step in the machine learning data preprocessing procedure is data cleaning. It entails locating and fixing problems such irrelevant features, inconsistent formatting, duplicate entries, outliers, and missing values. Among the methods are class imbalance correction, robustness checks, duplicate removal, unit standardization, encoding categorical variables, treating outliers using transformations, and imputing missing data. Making ensuring the dataset is reliable, consistent, and appropriate for training machine learning models is the aim to improve the models' dependability and performance. To make conclusions that are well-informed and based on the context and aims of the research, data cleansing is an iterative process that frequently requires collaboration between domain experts and data scientists.

**Fig. 3.1 Code for Data Cleaning by IQR for Books Dataset**



**Fig. 3.2 Code for Data Cleaning by IQR for Ratings Dataset**



**Fig. 3.3 Code for Data Cleaning by IQR for Users Dataset**

A screen shot of a computer

Description automatically generated

**3) Creating Popularity-Based Recommender System:**

To create a popularity-based Book-Recommender System I have decided to display top 50 rated books and the criteria to find these books is, those 50 books which have highest average rating and criteria to in popular books is a book should be rated by minimum of 250 users.

**Fig. 3.4 Top 50 Books**



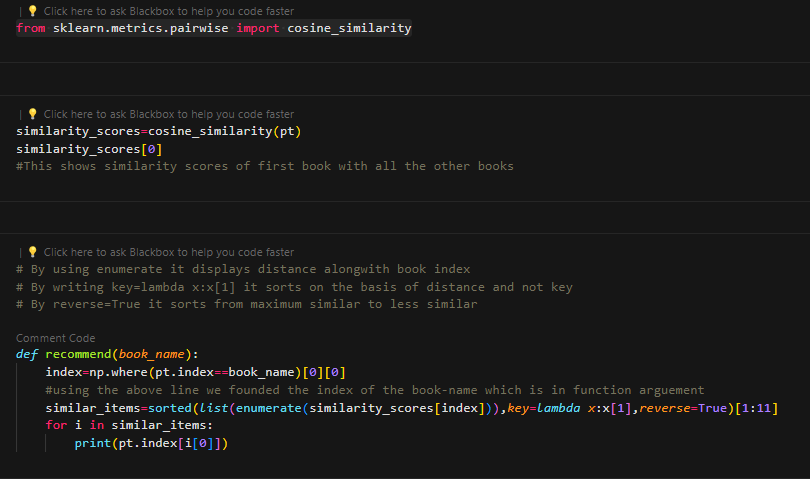
**4) Collaborative-Filtering Based Recommender System:**

To create a collaborative filtering-based book recommender system I have used a criterion i.e. finding similarity between the books. In this project I have found cosine similarity between each book, and I will recommend the books based on the similarity scores of books.

Cosine-Similarity:

By calculating the cosine of the angle that forms between two vectors in a multidimensional space, a metric known as cosine similarity can be used to quantify how similar two vectors are. Cosine similarity is frequently used in recommender systems to evaluate how similar user or item vectors—represented by their respective feature sets—are to one another. Greater similarity is indicated by higher values in this similarity metric, which goes from -1 (totally dissimilar) to 1 (identical). For scalable and computationally efficient relationship capture and item recommendation based on user preferences or content attributes, cosine similarity is especially helpful.

**Fig. 3.4 Recommendation function using Cosine-Similarity**



**Chapter IV**

**Result and Discussion**

**4.1 Result**

We are successful in building a Book Recommendation System which is primarily based on finding Cosine-Similarity between the books and then recommending the books which are of same genre.

**Fig. 4.1 Recommendation of Books**

**A screenshot of a computer

Description automatically generated**

**4.2 Discussion**

The resulting project is a system capable of generating recommendations of 2 types :-

* Popularity based recommendations.
* Collaborative filtering-based recommendations.

The project is now capable of recommending books present in the database accurately based on the genre and the user ratings present in the database. This system is now complete and can be used for generating recommendations for similar books.

**Chapter V**

**Conclusion and Future Work**

**5.1 Conclusion**

This project helps us in recommending books based on popularity or similarity accordingly and is now fully functional given the dataset and the code.

The resulting system accurately recommends books as it is evident that the recommended books are of the same genre and writing style as the book given as a reference.

**5.2 Future Work**

The project covers just the basics of recommendation systems and is not very complex of sorts, but this can surely lay a good understanding of how recommendation systems work and help in developing more complex codes.

The project also lacks a proper interface which can be implemented using python with the immense libraries and web app frameworks it provides for the purpose.

**References**

[1] Roy, D., Dutta, M. A systematic review and research perspective on recommender systems. J

Big Data 9, 59 (2022).

[2] R. Mu, "A Survey of Recommender Systems Based on Deep Learning," in IEEE Access, vol.6,

pp. 69009-69022, 2018.

[3] Beel, J., Gipp, B., Langer, S. et al. Research-paper recommender systems: a literature survey.

Int J Digit Libr 17, 305–338 (2016).

[4] Kreutz, C.K., Schenkel, R. Scientific paper recommendation systems: a literature review of

recent publications. Int J Digit Libr 23, 335–369 (2022).

[5] Zhang, Z., Patra, B.G., Yaseen, A. et al. Scholarly recommendation systems: a literature survey.

Knowl Inf Syst 65, 4433–4478 (2023).

[6] Visual Studio Code website was used to download the setup of VS Code.

<https://code.visualstudio.com/download>

[7] Python website was used to download Python 3.10.9 setup.

<https://www.python.org/downloads/release/python-3109/>

[8] Kaggle website was used to download the dataset.

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