Collaborative And Popularity Based Book Recommender System

Abhishek Samar Singh[1], Progyajyoti Mukherjee[1], Syed Aamir Bokhari[1], Suhail Shaik[1],
Prof. Sonia Maria D'Souza[2]

[1]Department of Artificial Intelligence and Machine Learning, New Horizon College of Engineering, Bangalore, India

[2]Sr. Asst Professor, Department of Artificial Intelligence and Machine Learning , New Horizon College of Engineering, Bangalore, India

ABSTRACT

Utilization of online websites to shop for a range of products has been frequent in our day to day lives. As a result, consumer demand is becoming more diverse, making it difficult for a general store to deliver the proper products based on the tastes of its customers. To deliver a favorable buying experience for the consumer, these E-commerce websites use various recommendation system algorithms. Recommendation systems are a tool for dealing with this problem; they allow you to meet consumer's demands and expectations while also attracting new ones. A product recommendation system is essentially a filtering system that suggests particular things to customers depending on their interests. Recommendation systems have exploded in popularity in recent years with applications in music, news, movies search queries and others. The bulk of today's E- commerce sites such as Amazon, Flipkart ,Myntra, make use of their own recommendation algorithms to better offer their customers with products they are likely to like .Recommendation engines are data filtering technologies that use algorithms and data to suggest the most relevant items to the user.

Keywords: RTBP, CRTBP, Three Body Problem.

I. INTRODUCTION

Today, a lot of things are recommended to users based on their interests via recommendation algorithms. One of the most effective systems for boosting profits by retaining more users in a fiercely competitive market is a recommendation system. Numerous companies, including online commerce, music streaming services, and movie streaming services, all use recommendation systems. For instance, collaborative filtering algorithms that suggest products to customers were heavily influenced by the online store Amazon. To find music that matches their consumers' preferences, music services like Pandora identify up to 450 uniquely identifying features of songs. Other music streaming services, like

Spotify, heavily rely on the musical preferences of comparable users when creating tailored radio stations and weekly song recommendations. [1]These methods are used by Netflix, a well-known streaming service for television.

We can observe how recommendation systems have a significant impact on the content users interact with during their daily lives. A system for recommending books is one in which we must suggest related works to the reader in accordance with his interests. Online retailers of eBooks like Google Play Books, Open Library, Good Reads etc., use the book recommendation algorithm. In this project, we'll create a book recommendation system using the Popularity and collaborative based filtering technique using python.

II. METHODOLOGY

A recommendation engine is a class of machine learning which offers relevant suggestions to the customer. Before the recommendation system, the major tendency to buy was to take a suggestion from friends. But Now Google knows what news you will read, YouTube knows what type of videos you will watch based on your search history, watch history, or purchase history. A recommendation system helps an organization to create loyal customers and build trust by them desired products and services for which they came on your site.[2] The recommendation system today are so powerful that they can handle the new customer too who has visited the site for the first time. They recommend the products which are currently trending or highly rated and they can also recommend the products which bring maximum profit to the company. A recommendation system is usually built using 3 techniques which are content-based filtering, collaborative filtering, and a combination of both.

1) **Content-Based Filtering**: The algorithm recommends a product that is similar to those which used as watched. In simple words, In this algorithm, we try to find finding item look alike. For example, a person likes to watch Sachin Tendulkar shots, so he may like watching Ricky Ponting shots too because the two videos have similar tags and similar categories. Only it looks similar between the content and does not focus more on the person who is watching this. Only it recommends the product which has the highest score based on past preferences.

2) **Collaborative-Based Filtering**: Collaborative based filtering recommender systems are based on past interactions of users and target items. In simple words here, we try to search for the look-alike customers and offer products based on what his or her lookalike has chosen. Let us understand with an example. X and Y are two similar users and X user has watched A, B, and C movie. And Y user has watched B, C, and D movie then we will recommend A movie to Y user and D movie to X user. YouTube has shifted its recommendation system from content-based to Collaborative based filtering technique. If you have experienced sometimes there are also videos which not at all related to your history but then also it recommends it because the other person like you has watched it.

3) **Hybrid-Based Filtering**: It is basically a combination of both the above methods. It is a too complex model which recommends product based on your history as well based on similar users like you. There are some organizations that use this method like Facebook which shows news that is important for you and others also in your network.

Recommender systems are widely used in several different domains for the recommendation of articles, music, movies, and even people. Portals such as Amazon and Submarino use recommender systems to suggest products to their customers.

- The first objective of this project is to recommend top 50 books to the user based on popularity of the books.

- The second main objective is to recommend any 5 books based on the user's given book of interest.

III. EXISITING SYSTEM

Here are a few of the book suggestion tools now in use by the best book shopping websites. The engines that are now in use traditional algorithms for suggestions. In a content-based recommendation engine, recommendations are generated from a source using information about the user and features associated with the products.[3] Content-based recommenders approach recommendations as a user-specific classification issue and learn a classifier for the user's preferences based on the characteristics of the products. Suggestions are generated by collaborative recommendation algorithms based on the ratings provided by a group of users. It searches for peer users who have ratings histories similar to the one of the present user and produces recommendations for them. The system needs extra information on the context of item consumption, such as time, mood, and behavioral factors, in context-based recommendation engines. When compared to what could be done without the usage of this additional source of information, these facts could be used to improve the recommendation.

Following are some of the existing book recommendation engines used by the top rated book purchasing websites.

The existing engines make use of conventional algorithms for recommendations In Content based Recommendation Engine, system generates recommendations from source based on the features associated with products and the user's information.

Content-based recommenders treat recommendation as a user-specific classification problem and learn a classifier for the user's likes and dislikes based on product features.

In Collaborative recommendation engines, suggestions are generated based on ratings given by group of people. It locates peer users with a rating history like the current user and generates recommendations for the user.

In Context based Recommendation Engine, system requires the additional data about the context of item consumption like time, mood and behavioral aspects. These data may be used to improve the recommendation compared to what could be performed without this additional source of information.

IV. PROPOSED SYSTEM

In reality, a recommendation engine is a system that can operate in a clustered or non-clustered environment, taking user online footprint as one of its input sets and generating a probable footprint for the user, giving its users a prediction that is more accurate.

The technology of collaborative recommendation is perhaps the most well-known, commonly used, and developed one.[4] Collaborative recommender systems combine user ratings for object suggestions, identify ratings-based similarities amongst users, and produce fresh recommendations based on user comparisons. In a collaborative system, a typical user profile is made up of a vector of things and their ratings, which is regularly updated as the user uses the system over time. To determine how similar two movies are, we'll use the Cosine Similarity metric from Sklearn. statistic used to assess how similar two items are is called cosine similarity. It calculates the cosine of the angle formed by two vectors that are projected onto a multidimensional space.

We have also used a type of recommendation system which works on the principle of popularity and or anything which is in trend. These systems check about the product or movie which are in trend or are most popular among the users and directly

recommend those. In this it recommends the top 50 most popular books.

The suggested system's architecture is depicted in **Figure 1**. The Recommender system is the system's primary module. The user has access to books in a variety of categories. The user has the option to rate books based on their appeal. Each person's rating and book-searching history is recorded in the database. Three strategies are primarily employed in the recommender system module for recommendations. On the data existing in the user's history, collaborative filtering and content filtering approaches are used. Demographic recommender is employed if these strategies produce null results. The list of suggested books is formed by combining the findings from all recommender techniques
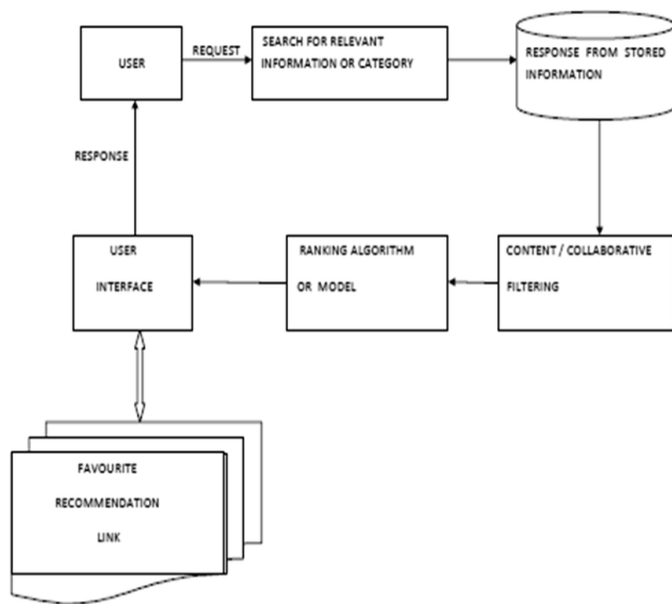

Figure 1. Proposed System

We have used Book Recommendation Dataset available in Kaggle which contains all the information about the user and books. With the help of NumPy, pandas, Sklearn and matplotlib library the data is preprocessed and converted into a structured form. Then the books are converted in the form of vectors and the cosine similarity is found between all the books. With the help of cosine similarity we can find out which of the books are similar to each other. Then the given output are the 5 books predicted.

Cosine similarity measures the similarity between two vectors of an inner product space. It is measured by the cosine of the angle between two vectors and determines whether two vectors are pointing in roughly the same direction.[5] It is often used to measure document similarity in text analysis. A document can be represented by thousands of attributes, each recording the frequency of a particular word (such as a keyword) or phrase in the document. Thus, each document is an object represented by what is called a term-frequency vector.

Equation 1

$$cosine(x, y) = \frac{\sum_{i=1}^{n} x_i y_i}{\sqrt{\sum_{i=1}^{n} x_i^2} \sqrt{\sum_{i=1}^{n} y_i^2}}$$

Figure 2. Cosine Formula

The dataset we have used is Book Recommendation Dataset from Kaggle. It contains 5000 dataset containing information about the user and books. With the help of python libraries like NumPy and pandas we created data frames and we processed and cleaned our data. The data consisted of some missing values and useless input features which we were able to handle.
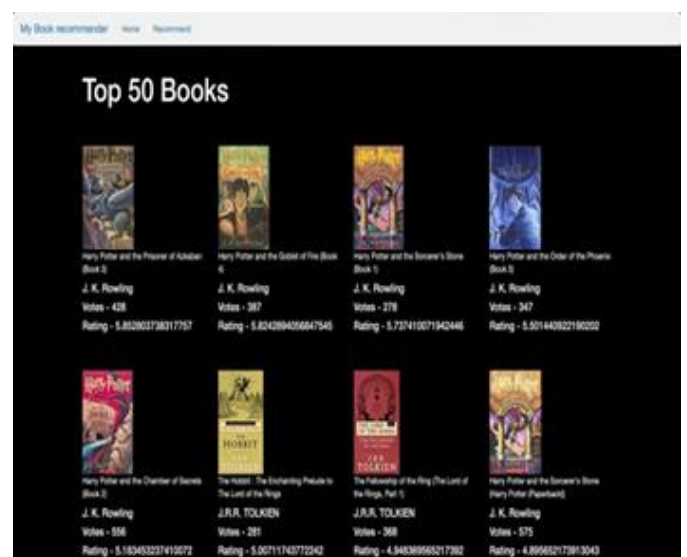
To get the top 50 books based on popularity we found out the average rating for each book and the number of ratings. A certain condition was given

which was the number ratings for the book should be greater than 250. so, the books which contain number of ratings above 250 will only get listed in top 50 most popular books.

For collaborative based filtering. when a user gives a book so the system can recommend him similar books, the book title enters the given function and with the help of cosine similarity it will give 5 books having very close cosine score and recommend it to the user.

Flask is a web framework, [6] it's a Python module that lets you develop web applications easily. It's having a small and easy-to-extend core: it's a micro framework that doesn't include an ORM (Object Relational Manager) or such features. It does have many cool features like URL routing, template engine. It is a WSGI web app framework. We have connected our python program with our site.
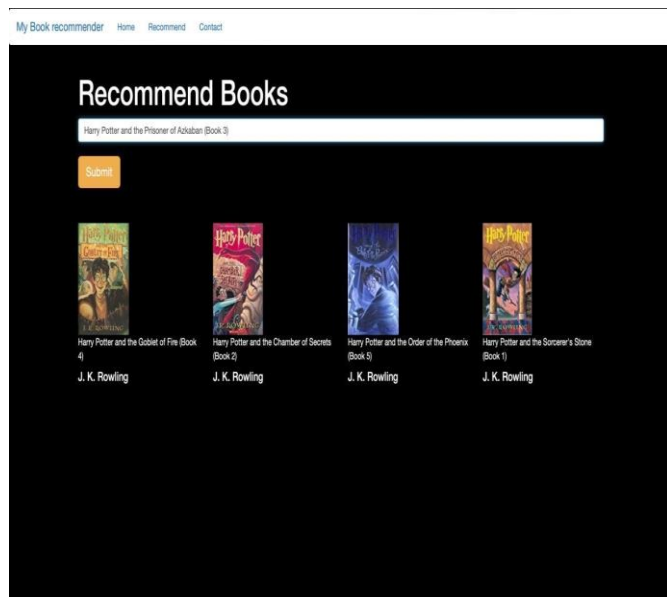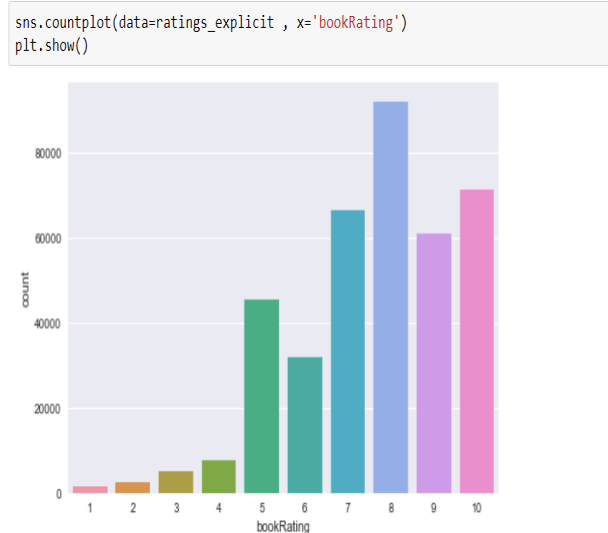

Figure 3. Result

```
sns.countplot(data=ratings_explicit , x='bookRating')
plt.show()
```


Graph 1. Book rating

V. CONCLUSION AND FUTURE ENHANCEMENT

A recommendation system is an information service that connects users and projects. It does this by assisting both users and project providers in finding and delivering projects to people who are interested in them. The suggestion system is an effective tool that can benefit the organization or business. It will continue to be studied and modified in the future to give users a better experience. Over the past ten years, when different Content, Collaborative, and Hybrid techniques were put out and numerous algorithms were created, recommender systems have advanced significantly. Nevertheless, despite all these developments, the current generation of recommender systems presented in this document still must be improved in order to increase the effectiveness of recommendation methods across a wider range of applications. In this article, we examined several the present recommendation systems' drawbacks and talked about several potential extensions that can enhance recommendation capabilities. In the hope that the challenges raised in this work would advance the conversation in the recommender systems community about the next generation of recommendation technologies, we further propose to build an algorithm that addresses the limits faced by previously developed recommender systems.

VI. REFERENCES

[1]. Suresh K. Gorakala and Michele Usuelli, Building a Recommendation System with R, Packt Publishing, 2015.

[2]. Michael D. Ekstrand, John T. Riedl and Joseph A. Konstan, "Collaborative Filtering Recommender Systems," in Foundations and Trends in HumanComputer Interaction, ACM, vol. 4, no. 2, pp. 81-173,2011

[3]. Sneha Khatwani and M.B. Chandak, " Building Personalized and Non-Personalized recommendation systems," in Proceedings of IEEE International Conference on Automatic Control and Dynamic Optimization Techniques, 2016, pp. 623-628

[4]. Jugendra Dongre, Gend Lai Prajapati and S. V. Tokekar, "The role of Apriori algorithm for finding the association rules in Data mining," in Proceedings of IEEE International Conference on Issues and Challenges in Intelligent Computing Techniques, 2014, pp. 657-660

[5]. Karthiya Banu.R, Dr. Ravanan.R and Gopal.J, "Analysis and Implementation of Association Rule Mining," in Proceedings of IEEE International Conference on Issues and Challenges in Intelligent Computing Techniques, 2014, pp. 657-660

[6]. Najdt Mustafa, Ashraf Osman Ibrahim, Ali Ahmed and Afnizanfaizal Abdullah, "Collaborative filtering: Techniques and applications," in Proceedings of IEEE International Conference on Communication, Control, Computing and Electronics Engineering, 2017

Cite this article as :