**CHAPTER 1**

**INTRODUCTION**

**1.1 Introduction**

A recommendation system is a model used for information filtering where it tries to predict the preferences of a user and provide suggests based on these preferences. These systems have become increasingly popular nowadays and are widely used today in areas such as movies, music, books, videos, clothing, restaurants, food, places and other utilities. These systems collect information about a user's preferences, behavior and ratings, then use this information to improve their suggestions in the future.

Books are integral aspect of life, offering a diverse range of genres and serving various purposes, from entertainment to education. They encompass fiction, non-fiction, academic, and many other categories. Books can be distinguished by genres such as mystery, romance, fantasy, biography, and more. Additionally, attributes like publication year, language, author, and subject matter contribute to the differentiation of books. Book Recommendation Systems helps us to search our preferred books among all of these different types of books and hence reduce the trouble of spending a lot of time searching our favorable books. So, it requires that the books recommendation system should be very reliable and should provide us with the recommendation of books which are exactly same or most matched with our preferences.

A large number of companies are making use of recommendation systems to increase user interaction and enrich a user's shopping experience. Recommendation systems have several benefits, the most important being customer satisfaction and revenue. Book Recommendation system is very powerful and important system. But, due to the problems associated with pure collaborative approach, book recommendation systems also suffer with poor recommendation quality and scalability issues.

**1.2 Problem Statement**

In the vast landscape of literature, readers often find it challenging to discover books that suit their individual tastes and preferences. Traditional methods of book discovery may fall short in providing personalized recommendations, leading to a gap between readers and books that have the potential to capture the interest and engage the reader. To address this issue, the goal of the Book Recommendation System project is to design and implement an advanced recommendation system tailored to the unique reading habits and preferences of users.

**1.3 Objectives**

Book Recommendation System has the following objectives:

* Medium scale single page application.
* Recommendation on the basic of similarities calculated from book ratings given by users.
* Responsive site.
* Search, Filtering Functionalities.

**1.4 Scope and Limitation**

**1.4.1 Scope**

This project has many scopes. They are:

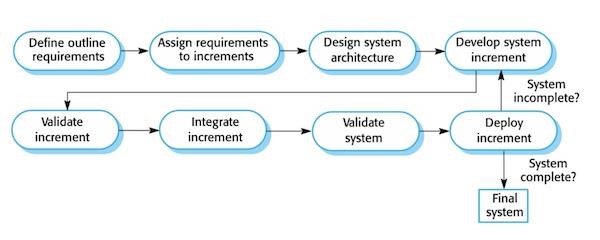
1. Personalization: A Book Recommender System should be capable of offering personalized recommendations tailored to each user's reading preferences. By analyzing user behavior, such as book ratings, reviews, and browsing history, the system should provide recommendations that align with the individual's interests and tastes.
2. Recommendation Engine: The Book Recommender System should incorporate a robust recommendation engine equipped with advanced algorithms and techniques. These algorithms should effectively analyze book attributes, user data, and interaction patterns to generate high-quality recommendations. Techniques such as collaborative filtering, content-based filtering, and hybrid approaches can be employed to enhance recommendation accuracy and relevance.
3. User Interface: The Book Recommender System should feature a user-friendly interface designed to facilitate seamless interaction and engagement. The interface should allow users to easily search for books, browse through recommendations, rate and review books, and provide feedback. Additionally, the interface should support personalized recommendations, enabling users to discover new titles that match their interests effortlessly.

**1.4.2 Limitations**

1. The system relies heavily on user behavior data. If users provide inaccurate data, the recommendations may not be accurate.
2. The system is limited by the availability and quality of data on books and user preferences. If there is insufficient data, the recommendations may not be relevant or useful.
3. The system may not be able to accurately recommend books for users with unique tastes that are not well-represented in the available data.

**1.5. Development Methodology**

The development methodology used for this project is incremental delivery, which is a kind of iterative software development process. The project is divided into a series of incremental iterations. Each function or features are being added upon the previous one to gradually deliver a full-fledged functional system. [1]



**Figure 1.1: Incremental Delivery**

Figure 1.1 explain a simple working system implementing only a few basic features is built and then that is delivered to the customer. Then thereafter many successive iterations versions are implemented and delivered to the customer until the desired system is released.

**1.6 Report Organization**

The report on “**Book Recommendation System**” is based on six chapters. Each chapter follows the constructive building of this project. **Chapter 1** gives an overview idea of our project. It anticipates and combines the main points to be described later in the chapters followingly. Similarly, **Chapter 2** usually contains the theoretical literature review. It gives an insight to distinguish the possible hypothesis, strategies and shortfalls in the current research. **Chapter 3** studies the system such that information can be analyzed, modeled, and developed. It also gives enough information to replicate the study. It addresses the problems from chapter 1 and explains the objects of each experiment. **Chapter 4** contains an insight of the system design and algorithm being used while developing the system. **Chapter 5** contains system testing it discusses the execution of a program or system with the intent of finding errors. It also includes the examination of code as well as execution of that code in various environments and conditions. **Chapter 6** describes the significance of “Book Recommender System”, moreover, discusses the future recommendations applicable to enhance the project.

# 

**CHAPTER 2**

**BACKGROUND STUDY AND LITERATURE REVIEW**

**2.1 Background Study**

In recent years, the entertainment industry has been undergoing a significant shift towards digitalization, with an increasing number of consumers opting for online streaming services to read books. The vast selection of books available online can often lead to confusion and decision paralysis, with consumers struggling to choose books that aligns with their preferences. This problem can be addressed through the development of a Book Recommender system, which can provide personalized recommendations to consumers based on their viewing history, ratings and preferences.

Book Recommender systems have been gaining popularity in recent years due to their ability to provide tailored recommendations to users, thus improving the user experience. These systems utilize machine learning algorithms and data analysis techniques to analyze user behavior and preferences, and suggest books that are most likely to be of interest to the user. These algorithms consider various factors such as book genre, book authors, ratings, and user feedback to generate personalized recommendations.

The development of a Book Recommender system requires expertise in various areas such as machine learning, data analysis, and software development. The system will need to be able to collect and analyze user data in real-time and generate personalized recommendations efficiently. Additionally, the system will need to be scalable and capable of handling large amounts of data, as well as being user-friendly and easy to navigate.

Various criteria determine how the recommender systems work. The criteria are based on machine learning or deep learning algorithms that are used in matching the similarities before the suggestions are made. The algorithms achieve different levels of accuracy and require different computational times to retrieve the suggestions. Various computational algorithms have been proposed and used to increase the efficiency of recommender systems [2]

The benefits of a Book Recommender system are numerous, both for consumers and book streaming services. Consumers will be able to easily find books that align with their preferences, leading to a more satisfying reading experience.

In conclusion, the development of a Book Recommender system has the potential to revolutionize the entertainment industry by providing personalized recommendations to consumers, thus improving the overall user experience. With the increasing demand for book reading services, the development of such a system has become a necessity for book reading services to remain competitive in the market.

**2.2 Literature Review**

Okon et.al. proposed a model that generates recommendations to buyers, through an enhanced CF algorithm, a quick sort algorithm and Object-Oriented Analysis and Design Methodology (OOADM). Scalability was ensured through the implementation of Firebase SQL. This system performed well on the evaluation metrics [3].

Kurmashov et.al. used Pearson correlation coefficient-based CF to provide internet-based recommendations to book readers and evaluated the system through an online survey [4].

Parvatikar et.al. proposed item-based collaborative filtering and association rule mining to give recommendations. Similarity between different users was computed through the Adjusted Cosine Vector Similarity function. Better recommendations were obtained as through this method data sparsity problem was removed [5].

Feng et.al. proposed a RS for movies based on a similarity model constituted of factors S1 (similarity between users), S2 (ratio of co-rated items) and S3 (user’s rating choice weight). This RS was particularly useful for sparse datasets [6].

**CHAPTER 3**

**SYSTEM ANALYSIS**

**3.1 System Analysis**

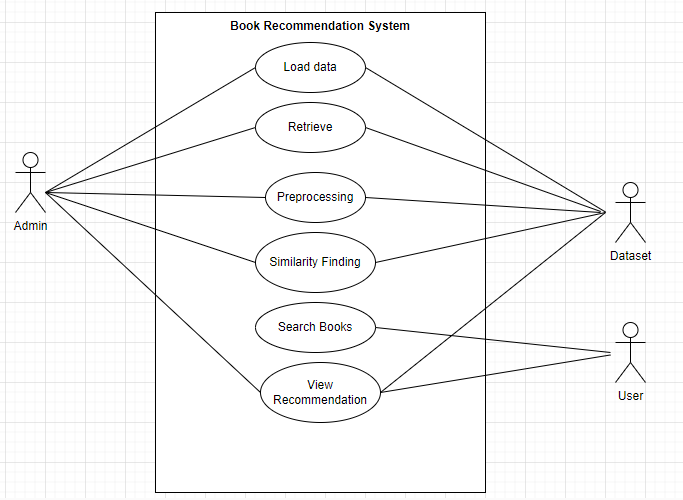
Systems analysis is a process of studying a system or organization in order to understand its components, how they interact and how they can be improved. It is a holistic approach that looks at the system as a whole and identifies the relationships between its parts. The goal of systems analysis is to identify problems and inefficiencies in the current system and to propose solutions for improvement.

**3.1.1 Requirement Analysis**

Requirement Analysis We’ve analyzed and validated the requirements, recorded and monitored the implementation throughout the project.

**3.1.1.1 Functional Requirement**

1. Book dataset: The system should maintain a dataset of books along with their ratings and users with relevant metadata such as title, author, rating, and their respective images.
2. Book search: The system should allow users to search for books.
3. Personalized recommendations: The system should provide personalized recommendations to each user based on their preferences.



**Figure 3.1 Use Case Diagram of Book Recommender**

In Figure 3.1, whenever the user searches a certain book, the system retrieves the data of the book form the database and find the similarity with other books and recommend the books to the users.

**3.1.1.2 Non-Functional Requirement**

The points below focus on the non-functional requirement of the system proposed.

* **User friendly**

User friendly generally means easy to read, use and communicate. The system is not complex and self-explanatory. Our system is well-organized, making it easy to locate different tools and options.

* **Reliability**

The system is reliable. The system takes data from various trusted sources and organization.

* **Scalability**

Scalability is crucial for handling growth in the user base and the amount of data. Our system will accommodate increasing numbers of users and growing datasets without compromising performance.

* **Easy access**

Our project is a web-based application. Considering this our platform can be accessed by anyone, anywhere where there is internet connection.

**3.1.2 Feasibility Study**

Feasibility studies aim to objectively and rationally uncover the strengths and weakness of an existing or proposed system, opportunities and threats as presented by the environment, the resources required to carry through, and ultimately the prospects for the success.

**3.1.2.1 Technical Feasibility**

The technical feasibility of the project is high, as the required hardware and software resources are widely available and accessible. Requirements of our system can be categorized as:

**Hardware Requirements:**

* A computer with a minimum of 2 GB of RAM
* Processor with 1.7-2.4 GHz speed
* Minimum of 2GB Graphic card

**Software Requirements:**

* Operating System: Windows 10, Linux, or MacOS
* Text Editor (VS-code, PyCharm)
* Anaconda distribution package
* Python libraries
* **Anaconda distribution:** Anaconda is a free and open-source distribution of the Python programming languages for scientific computing (data science, machine learning applications, large-scale data processing, predictive analytics, etc.), that aims to simplify package management system and deployment. Package versions are managed by the package management system conda. The anaconda distribution includes data-science packages suitable for Windows, Linux and MacOS.3
* **Python libraries:** For the computation and analysis certain python libraries which are used to perform analytics are required. Packages such as NumPy, pandas, Flask framework, etc. are needed.
* **NumPy:** NumPy is a general-purpose array-processing package. It provides a high-performance multidimensional array object, and tools for working with these arrays. It is the fundamental package for scientific computing with Python.
* **Pandas:** Pandas is one of the most widely used python libraries in data science. It provides high-performance, easy to use structures and data analysis tools. Unlike NumPy library which provides objects for multi-dimensional arrays, Pandas provides in-memory 2d table object called Data frame.
* **Flask:** It is a lightweight WSGI web application framework. It is designed to make getting started quick and easy, with the ability to scale up to complex application.

**3.1.2.3 Operational Feasibility**

Operational feasibility measures how well a proposed system can solve the defined problem and takes advantage of the opportunities identified during scope definition and how it satisfies the requirements identified in the requirements analysis phase. The system can be developed to be reliable, maintainable, usable, sustainable and affordable. So, this system is operationally feasible.

**3.1.2.3 Economic Feasibility**

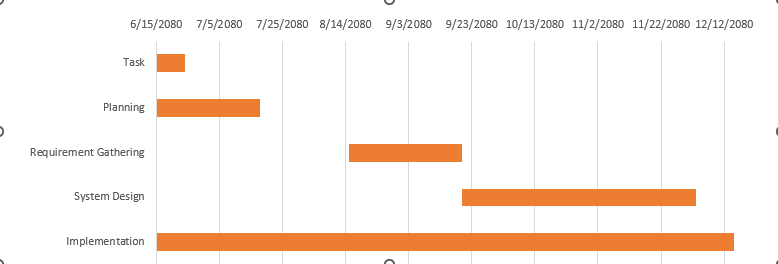
Economic feasibility analyses the project’s costs and revenue in an effort to determine whether it is possible to complete or not. There will not be any necessary equipment to be bought. However, the project will require domain, hosting and probably API which can be bought and configured with a suitable plan. Even if some features were to be added, it will be cost free as no extra equipment will be necessary. As the team already has everything needed, this system is economically feasible. Extensive databases can be maintained when the number of users of the app starts increasing.

**3.1.2.4 Schedule Feasibility**

It is the most important for the completion of the project on time. The project is to be completed within time constraints. In figure 3.2, shows the time schedule of different activities of the process.

**Table 3.1: Schedule Table**

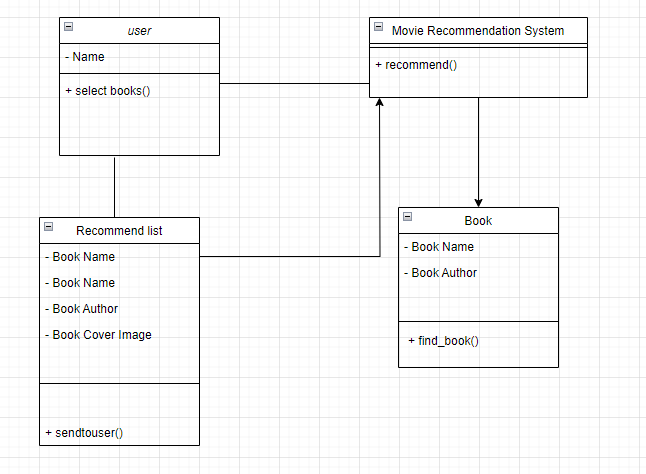
|  |  |  |  |
| --- | --- | --- | --- |
| **Task** | **Start Date** | **End Date** | **Duration** |
| Planning | 6/15/2080 | 6/24/2080 | 9 |
| Requirement Gathering | 6/15/2080 | 7/18/2080 | 33 |
| System Design | 8/15/2080 | 9/20/2080 | 36 |
| Implementation | 9/20/2080 | 12/3/2080 | 74 |
| Documentation | 6/15/2080 | 12/15/2080 | 183 |



**Figure 3.2: Gantt Chart**

**3.1.3 Analysis**

**3.1.3.1 Class Diagram**

Class diagram is a static diagram. It represents the static view of an application. Class diagram is not only used for visualizing, describing, and documenting different aspects of a system but also for constructing executable code of the software application. Class diagram describes the attributes and operations of a class and also the constraints imposed on the system. The purpose of class diagram is to model the static view of an application. Class diagrams are the only diagrams which can be directly mapped with object-oriented languages and thus widely used at the time of construction. UML diagrams like activity diagram, sequence diagram can only give the sequence flow of the application, however class diagram is a bit different. It is the most popular UML diagram in the coder community.

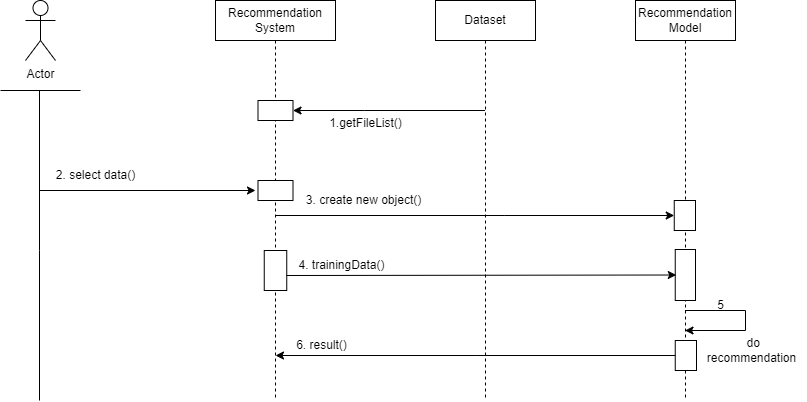
**Figure 3.3: Class Diagram of Book Recommender System**

In Figure 3.3 illustrate the blueprint of the system. It represents the types of objects residing in the system and the relationships between them.

**3.1.3.2 Sequence Diagram**

Sequence diagrams can be useful reference diagrams for businesses and other organizations. Draw a sequence diagram to:

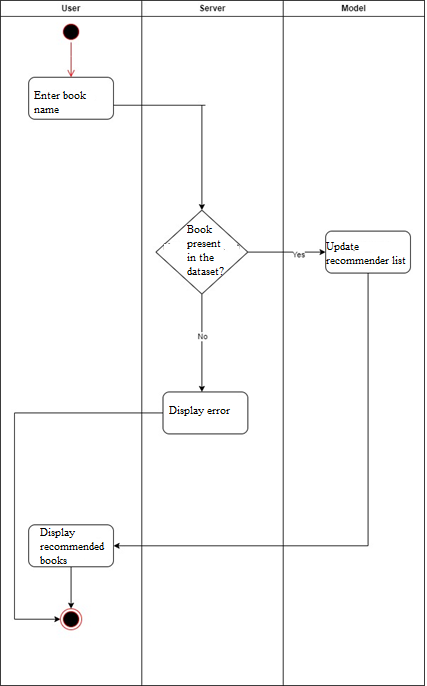
* Represent the details of a UML use case.
* Model the logic of a sophisticated procedure, function, or operation.
* See how tasks are moved between objects or components of a process.
* Plan and understand the detailed functionality of an existing or future scenario.



**Figure 3.4: Sequence diagram of book recommender system**

Figure 3.4, illustrate the interaction between user and the recommendation system. When the user selects the books, the system gets the recommended books from the database with the help of recommendation model. When the user selects a certain book in the system then the system creates a new object and transfer to the recommendation model. According to the selected book model recommend the book and provide result to the user. It shows the sequence diagram of the book recommendation system. The user enters the name of a book in the search bar and clicks the search button. The system validates the input and checks if the book is present in the dataset. If the book is found, the system retrieves the data of the book from the database and passes it to the recommendation model.

**3.1.3.3 Activity Diagram**

******Activity diagram is another important diagram in UML to describe the dynamic aspects of the system. Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent. Activity diagrams deal with all type of flow control by using different elements such as fork, join, etc. The basic purposes of activity diagrams are to capture the dynamic behavior of the system.



**Figure 3.5: Activity Diagram of Book Recommender System**

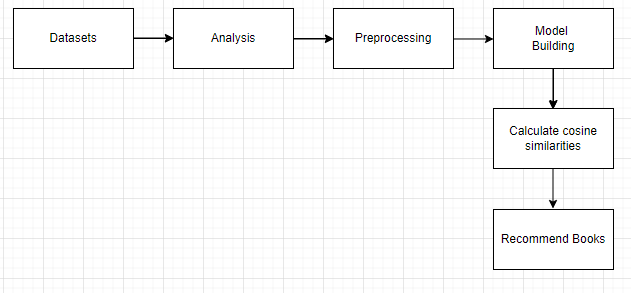
In Figure 3.5, the user searches a book in the search bar, and the server checks whether the book is present in the dataset or not. If the book is present in the dataset, then the server updates the recommendation list using the recommendation model and displays the recommended books to the user. However, if the book is not present in the dataset, then it will display an error.

**CHAPTER 4**

**SYSTEM DESIGN**

**4.1 Design**

System design is the process of representing architecture, interfaces, components that are included in the system. i.e., system design can be seen as the application of system theory to product development.



**Figure 4.1: System Design of Book Recommendation System**

The system architecture is shown in the Figure 4.1, The procedures involved are as:

* **Datasets**: We start by collecting and curating various datasets like books, ratings and users containing information about various books, books users and rating of the book.
* **Analysis**: Next, we analyze the collected datasets to gain insights into the characteristics and trends within the data. This analysis helps us understand the distribution of books, identify popular authors, and uncover patterns.
* **Preprocessing**: Before building the recommendation model, we preprocess the data to clean and transform it into a suitable format. This involves handling missing values, standardizing text data, encoding categorical variables, and scaling numerical features.
* **Model** **Building**: With the preprocessed data, we proceed to build the recommendation model. Depending on the approach, we used collaborative filtering technique. The goal is to develop a model that can effectively suggest relevant books to users based on their preferences.
* **Calculate Cosine Similarity**: In collaborative filtering-based recommendation systems, we calculate the cosine similarity between books or users to measure their similarity. This similarity metric helps us identify books that are closely related to each other, allowing us to make personalized recommendations.
* **Recommend Books**: Finally, using the calculated cosine similarity scores, we generate recommendations for users. We suggest a list of books that are likely to interest them. This recommendation process forms the core functionality of our book recommender system, helping users discover new and relevant reading material.

**4.2 Interface Design**

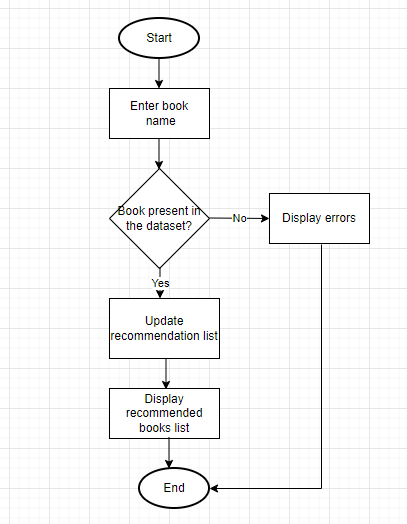
For the User Interface design of our book recommender system, we are using HTML, CSS, and Flask. HTML forms the backbone of our interface, defining the structure and layout of the pages to ensure clarity and organization.

To enhance the visual appeal and user experience, we uses the power of CSS. By styling elements and adding design elements such as colors, fonts, and layouts, CSS transforms the interface into an attractive and user-friendly platform for discovering and exploring books.

Furthermore, Flask serves as the server-side framework, seamlessly integrating with our HTML and CSS to render dynamic content, handle user requests, and manage the overall functionality of the system. With Flask, we can create interactive features, process user input, and deliver personalized recommendations, enriching the user experience and making our book recommender system a valuable resource for book enthusiasts.

**4.3 Process Design**

The simple process of how our system works is shown through the flowchart given below:



**Figure 4.2 Workflow Diagram**

**4.4 Dataset Description**

The recommendation system utilizes three datasets entitled Books.csv, Ratings.csv, Users.csv. Books are identified by their respective ISBN. Invalid ISBNs have already been removed from the dataset. Moreover, some content-based information is given (Book-Title, Book-Author, Year-Of-Publication, Publisher), obtained from Amazon Web Services. Ratings.csv contains the book rating information. Ratings (Book-Rating) are either explicit, expressed on a scale from 1-10 (higher values denoting higher appreciation), or implicit, expressed by 0. Users.csv contains information about the users. The user IDs (User-ID) have been anonymized and map to integers. Demographic data is provided (Location, Age) if available. Otherwise, these fields contain NULL-values. [7]

**4.5 Algorithm Details**

**Collaborative Filtering**

Collaborative filtering is a recommendation technique that predicts user preferences by leveraging the behavior and preferences of similar users or items. It relies on user-item interaction data to recommend items to users based on the preferences of other users with similar tastes.

To address some of the limitations of content-based filtering, collaborative filtering uses similarities between users and items simultaneously to provide recommendations. This allows for serendipitous recommendations; that is, collaborative filtering models can recommend an item to user A based on the interests of a similar user B. Furthermore, the embeddings can be learned automatically, without relying on hand-engineering of features. [8]

**4.5.1 Cosine Similarity**

After the user is prompted to enter a book, the algorithm provides 10 other books like the one used as an input by the user.

In cosine similarity, vectors are taken as the data objects in data sets, when defined in a product space, the similarity is figured out. The smaller this distance, the higher the similarity, but the larger the distance, the lower the similarity. Cosine similarity is a measure that helps to find out how similar data objects are, regardless of size. Mathematically, it is the cosine of the angle between two vectors projected in a multi-dimensional space.

𝑎⃗. 𝑏⃗⃗ ∑𝑛1 𝑎𝑖𝑏𝑖

𝐶𝑜𝑠𝜃 = = ……………………(i)

√

∑

𝑎

𝑖

2

𝑛

1

√

∑

𝑏

𝑖

2

𝑛

1

‖𝑎⃗‖‖𝑏⃗⃗‖

Where*, 𝑎⃗. 𝑏⃗⃗ = ∑𝑛1 𝑎𝑖𝑏𝑖 = 𝑎𝑖𝑏𝑖 + 𝑎2𝑏2 + ⋯ + 𝑎𝑛𝑏𝑛* is the dot product of the two vectors.[9]

The angle between two vectors determines its direction and is measured in ‘θ’. This angle can be calculated by using above equation (i).

When θ = 0 °, the `*x*` and` *y*` vectors overlap and prove to be similar.

When θ = 90 °, the `*x*` and` *y*` vectors are therefore dissimilar.

The steps involved are:

* We merged user ratings and book information tables to create a matrix with books as rows and users as columns, containing user ratings for each book.
* We mainly focused on users who have rated over 200 books and books with over 50 ratings to ensure engagement and popularity.
* This calculation of cosine similarities between users and the items ensures reliability by basing recommendations on preferences of dedicated readers and well-reviewed books.
* By filtering data, we aim to provide more robust and meaningful suggestions to users, enhancing the overall recommendation system.

**CHAPTER 5**

**IMPLEMENTATION AND TESTING**

.

**CHAPTER 6**

**CONCLUSION AND FUTURE RECOMMENDATION**

**6.1 Conclusion**

Thus, the recommender system was successfully implemented. Here we have used an approach to a book recommendation system using Cosine Similarity to recommend similar books based on the one chosen by the user. To enhance the user experience, this system performs sentiment analysis on the reviews of the movie chosen using Naïve Bayes algorithm. The Sentiment analysis done on the reviews of the movies achieves impressive accuracies of 99% and 99% respectively. Hence, we found that Content Based Collaborative Filtering was the best in our situation. For working on large dataset, it was an approach in implementing the algorithm and making it a Recommender System.

It was a challenge to implement a recommender system on this scale of huge data. Recommender systems have become ubiquitous. People use them to find books, music, news, smart phones, vacation trips, and romantic partners. Nearly every product, service, or type of information has recommenders to help people select from among the various alternatives the few they would most appreciate. Sustaining these commercial applications is a vibrant research community, with creative interaction ideas, powerful new algorithms, and careful experiments.

## 6.2 Future Recommendations

In the future, we would like to improve the suggestion process by:

1. Add registration feature.
2. We can add user registration feature to store each users’ unique preferences and recommend them books accordingly.
3. Introduce user dislike book list. The user data is always useful in recommender systems. In the future we can collect more user data and add user dislike book list. We will input dislike book list into the recommender system as well and generate scores that will be added to previous result. By this way we can improve the result of recommender system.
4. Make the recommender system as an internal service. In the future, we can make it as an internal APIs for developers to invoke. Some book lists in the website will be sorted by recommendation.

**REFERENCES**

[1] "https://www.geeksforgeeks.org/software-engineering-incremental-process-model/".

[2] H. H. M. H. Cami B.R., "A content-based movie recommender system based on temporal user preferences," Shahrood, Iran, 2017

[3] Okon, E.U., Eke, B.O. and Asagba, P.O. (2018). An improved online book recommender system using collaborative filtering algorithms. International Journal of Computer Applications (0975-8887) Volume 179-No.46, June 2018

[4] Kurmashov, N., Konstantin, L., Nussipbekov, A. (2015). Online book recommendation System. Proceedings of Twelve International Conference on Electronics Computer and Computation (ICECC)

[5] Liu et.al. (2012) [15] added the dimension of user-interest. They proposed iExpand, a 3 tier model i.e.buser, user-interest and item. This helped in overcoming the issues of overspecialization and cold-start as well as reducing computation costs.

[6] Liu, Q., Chen, E., Xiong, H., Ding, C.H.Q. and Chen, J. (2012). Enhancing Collaborative Filtering by User Interest Expansion via Personalized Ranking. IEEE Transactions on Systems, Man and Cybernetics-Part B: Cybernetics, Vol. 42, No.1, February 2012.

[7] <https://www.kaggle.com/datasets/arashnic/book-recommendation-dataset?select=Users.csv>

[8] https://developers.google.com/machine-learning/recommendation/collaborative/basics

[9] https://www.youtube.com/watch?v=e9U0QAFbfLI&t=515s