Recommendation System for LMS

A PROJECT REPORT

Submitted in partial fulfillment of
the requirements for the award of the degree of

Bachelor of Technology
in
Computer Science & Engineering

Submitted by:

Krati Agarwal(19EMCCS060)

Gaurav Sharma(19EMCCS037)

Bharat Sharma(19EMCCS023)

Shudhanshu Dhariwal(19EMCCS107)

Under the guidance of

Dr. Sashi Kumar Singh

(Director Professor (R&D))

BIKANER TECHNICAL UNIVERSITY, BIKANER

MODERN INSTITUTE OF TECHNOLOGY & RESEARCH CENTER

April 2023 (2022-2023)

BIKANER TECHNICAL UNIVERSITY, BIKANER

CERTIFICATE

Certified that this project report "**Recommendation System for LMS**" is the original work of below students of B. Tech final Year VIII Semester Computer Science and Engineering "**Krati Agarwal, Gaurav Sharma, Bharat Sharma, Shudhanshu Dhariwal** who carried out the project under the **Dr. Sashi Kumar Singh** supervision.

SIGNATURE

Dr. J R Arun Kumar

HEAD OF THE DEPARTMENT

Computer Science Department

MITRC College, Alwar

SIGNATURE

Dr. Sashi Kumar Singh

(Director Professor (R&D))

(Project Guide)

ACKNOWLEDGEMENT

Firstly, we would like to express our gratitude to our advisor for the beneficial comments and remarks.

We express our sincere thanks to **Prof. S. K. Sharma (Director)** of Modern Institute of Technology & Research Center, Alwar.

We pay our deep sense of gratitude to **Dr. J.R Arun Kumar**, Head of the Computer Science & Engineering Department of Modern Institute of Technology & Research Center, Alwar for encouraging us to the highest peak and providing us the opportunity to present the project.

We pay our deep sense of gratitude to **Dr. Sashi Kumar Singh (Professor)**, Project Supervisor, Computer Science Engineering Department, Modern Institute of Technology & Research Center, Alwar (Rajasthan) for their constant support and guidance throughout the course of this work. Their sincerity, thoroughness and perseverance have been a constant source of inspiration for us.

We would like to thank my institution and my faculty members without whom this project would have been a distant reality, we also intend our heartiest thanks to my family and well-wishers.

Finally, but not least, our parents are also an important inspiration for us. With due respect we express our gratitude to them.

Krati Agarwal(19EMCCS060)
Gaurav Sharma(19EMCCS037)
Bharat Sharma(19EMCCS023)
Shudhanshu Dhariwal(19EMCCS107)

ABSTRACT

Recommendation system is a data filtering technique. They are used to provide suggestions to the users according to their interest and need. The different types of recommendation systems are used in different platforms and have become an important part of various applications. It is a decision-making process that will help the users to buy items in which they are interested. Recommendation systems are very useful and effective techniques of filtering the data. In this Project using two types of filtering collaborative and content-Based Filtering provide the user best technical courses and books according to their rating. A user firstly login in a web-app then search the technical courses and books in their sections Recommend system show the user best data according to the user interest. Recommendation systems have become extremely common in recent years. It helps the customer to discover information and settle on choices where they do not have the required learning to judge a specific item. It can be utilized as a part of different diverse approaches to encourage its customer with effective information sorting. It is a software tool and techniques that provide suggestion based on the customer's taste to discover new appropriate thing for them by filtering personalized information based on the user's preferences from a large volume of information. Users taste and preferences should be constructed accurately in order to provide most relevant suggestions. This survey paper compare's and details the various type of recommender system and popular recommendation algorithms and its uses. It is a very popular technique in recent years and used by many commercial websites and other platforms to recommend news, books, movies, shopping items, novels, music and much more that is why recommendation system has become a hot topic.

TABLE OF CONTENTS

LIST OF FIGURES

Chapter-1

INTRODUCTION

Recommendation systems can now be found in many modern applications that expose the user to a huge collection of items. Such systems typically provide the user with a list of recommended items they might prefer or predict how much they might prefer each item. These systems help users to decide on appropriate items and ease the task of finding preferred items in the collection.

For example, the DVD rental provider Netflix1 displays predicted ratings for every displayed movie in order to help the user decide which movie to rent. The online book retailer Amazon[2] provides average user ratings for displayed books, and a list of other books that are bought by users who buy a specific book. Microsoft provides many free downloads for users, such as bug fixes, products and so forth. When a user downloads some software, the system presents a list of additional items that are downloaded together. All these systems are typically categorized as recommender systems, even though they provide diverse services.

In the past decade, there has been a vast amount of research in the field of recommender systems, mostly focusing on designing new algorithms for recommendations. An application designer who wishes to add a recommendation system to her application has a large variety of algorithms at her disposal and must make a decision about the most appropriate algorithm for her goals. Typically, such decisions are based on experiments, comparing the performance of a number of candidate recommenders. The designer can then select the best performing algorithm, given structural constraints such as the type, timeliness and reliability of availability data, allowable memory and CPU footprints. Furthermore, most researchers who suggest new recommendation algorithms also compare the performance of their new algorithm to a set of existing approaches. Such evaluations are typically performed by applying some evaluation metric that provides a ranking of the candidate algorithms (usually using numeric scores).

Initially most recommenders have been evaluated and ranked on their prediction power their ability to accurately predict the user's choices. However, it is now widely agreed that accurate predictions are crucial but insufficient to deploy a good

recommendationengine. In many applications people use a recommendation system for more than an exact anticipation of their tastes. Users may also be interested in discovering new items, in rapidly exploring diverse items, in preserving their privacy, in the fast responses ofthe system, and many more properties of the interaction with the recommendation engine. We must hence identify the set of properties that may influence the success of a recommender system in the context of a specific application. Then, we can evaluate how the system preforms on these relevant properties.

A recommendation system addresses the issue perfectly. Since there is no other solution that is said to provide an accurate analysis of the student's choice and capability, a recommendation system comes handy as it is built so as to recommend the perfect course. There have been countless recommendation systems which pledge to recommend courses to students. But most of them fail to acknowledge the two main factors which decide the success rate of the student. The first factor-to recommend a course based on the student's choice. The second factor to ensure whether the student is efficient enough to do well in the course they have chosen, failing to which, the system will recommend a course not considering students ability.

The recommendation system which addresses both these factors cumulatively and recommends a course accordingly is capable of yielding maximum success in terms of the student's future. The purpose is to build a system that concentrates on only student's success history and students' present choice, unlike existing approaches – collaborative filtering and content-based filtering. Content based system recommends courses based on students interests and collaborative based system recommends courses based on the opinions of similar users.

A rule-based approach is employed. This way, it enhances the optimality of the solution. A rule-based classification can be used to refer to any classification scheme that makes use of IF-THEN rules for class prediction. We make use of a Rule Induction Algorithm to extract relevant rules from the data which can be done directly using sequential covering algorithms. Rule based systems are domain specific expert systems which makes appropriate deductions and choices. Rule Based Classification is cost efficient and accurate in terms of the end result. The speed of system can be optimized as all parts of the code are known to the user. It also provides results with high accuracy and less error.

Rate, Since the output is dependent on the rules, the output responses are stable and won't result in vague results. Rule based classification follows the same cognitive process as that of a human and hence results are handcrafted.

1.1 Problem Description

This project i.e., **Recommendation system for LMS** helps the people to provide the better educational books and courses of their recommendation which help to for their learning and developing the skills in IT and more other sector's. In this process, we ensure that only relevant studies about RS in the context of education are included for further analysis. Once the search results were available, a pre-selection criterion was applied to these papers. Researchers read paper title, abstract and keywords to apply the inclusion criteria and consider that at least papers should mention the keywords presented in the search strategy. Then, the exclusion criterion was used during the full paper reading, generating what we call the primary studies list. The Web is the most adopted platform for using RS in education; it is reported by the 52% of the reviewed papers. Web-based platforms are present in many areas of education supporting different educational scenarios. Educational RS developed on a web platform, which recommends learning programming activities in a personalized way, i.e., according to the user profile. The web-based platform is named Wise Coach and enables the user to solve problems according to their programming level.

It has been observed that most of the users do not give any ratings. So, a research problem arises that how to know whether they are satisfied with the product and how much. There are two ways of taking ratings. When is explicit like asking them after they have purchased or gone through any item. Another way which is predicting their ratings for a particular item based on their preferences on some other item. This method is known as an implicit method of collecting ratings. According to our research, explicit and implicit methods have sufficient gap which gives an opportunity for researchers to start research in this domain.

This problem arises if the recommended items are too similar to each other. One such example is that a user buys grocery items from a shopping website on the regular basis, and whenever he/she opens the website, the recommender system recommends only sugar, may be of different brands (basis on the maximum purchase), on seeing same item

again, and again and not any unique suggestion may lead user to switch the shopping website whose recommender system offers better, interesting and personalized results for the user. So, in this problem basically, The user loses interest in recommended items, one solution to this problem is recommendation diversification. In this method we list all the items that are dissimilar but it is of interest to the user.

1.2 Motivation

Recommender systems are fuelled by user feedback. As we collect information on what a user likes and dislikes, we are able generate more relevant recommendations tailored to the tastes of that user.

Purpose of recommendation system for technical learning is aims to provide users with accurate Courses and Books recommendations. Usually, basic recommendation system to make recommendations consider one of the following factors; User preference known as content based Filtering or the preference of similar users known as collaborative filtering.

1.3 Objectives

The objective of recommender systems is to provide recommendations based on recorded information on the users' preferences. These systems use information filtering techniques to process information and provide the user with potentially more relevant items.

A Recommender System refers to a system that is capable of predicting the future preference of a set of items for a user and recommend the top items. One key reason why we need a recommender system in modern society is that people have too many options to use from due to the prevalence of Internet.

The main objective is to create a Recommendation system to recommend relevant Courses and Books to users based on popularity and user interests.

The Project has the following stages:

- Develop a prototype in which the users are provided with large learning object collections and state the problem of information overload.
- Prepare, clean and process the collecting data to building a recommended engine.
- Make a comparison between the algorithm used in the prototype and other types of prototypes used in other environments, looking for the algorithm that better adjusts to

- in an educational environment.

Chapter-2

LITERATURE REVIEW

O. R. Za, in 2002 [1] a recommender system or recommendation system (RS) is a term that refers to a computer-based system designed to provide recommendations to the wearer. According to Zaiane a recommender system in education context is a system designed to provide intelligent recommendations to learners (known as item) namely: the steps / processes to be performed, study program, reading materials (courses, books, articles, website, assessment, exercises), or sources of information relating to the topic / subject based to the choices previously made by participants of learning. In providing such recommendations, an RS employed a statistical model that is trained in a supervised or unsupervised manner with a learning algorithm based on input from the previous learning participants.

Zaho, Zhi-Dan and Ming-Sheng in 2010 [2] implements user-based Collaborative Filtering algorithm on a distributed cloud computing platform that is Hadoop is used, to solve the scalability problem of Collaborative Filtering method. Merits are 1. Better for finding interests for similar items. 2. Personalized recommendation. Demerits are 1. Doesn't consider similar users' interests.

Dara, S. in 2019 [3] current recommendation method does have a great performance, according to the undertake experimentation, than numerous prior implementations, including the praised k-NN algorithm being used by suggestion especially at longer length. System of suggestion progressively that is used in fields, such as films, traveling, songs, books etc. Increasing social acts have amplified the use of recommending programs in persons and community recommending programs. Community suggestion structures also address the issue of cold starting which occurs within a person recommendation engine. This work provides a report on the latest technology relevant to several areas of community optimization algorithms. As for their accumulation and customer obvious sign designs, the scientist addressed prior structures. A certain organization is quite important to study each new domain complexities.

Ahuja, R. in 2019 [4] suggestion framework is a very common and cold e-commerce issue. Recommendation system performs in multiple ways including faculty member based on

quality, suggestion for reciprocal filtering, and suggestion for the combination technique. This article proposes a collective suggestion filtering system focused on naive Bayesian approach. The current recommendation method does have a great performance, according to the undertake experimentation, then numerous prior implementations, including the praised k-NN algorithm being used by suggestion especially at longer length.

Y. Pan, D. Wu, C. Luo, and A. Dolgui in 2019 [5] collaborative filtering is a very common technique for book recommendation, but the accuracy of this technique was 88% or 89%, which is comparatively low. However, a content-based recommendation system needs an enormous amount of training data set, which is not feasible for real-world scenarios. When Jaccard similarity was added with collaborative filtering, it achieved the highest recall. The major drawbacks of a collaborative recommender system are sparsity and cold-start issues. These issues can be removed using a kernel-based fuzzy technique that scored a 95% accuracy rate.

Y. Lee, C. Wei, P. Hu, T. Cheng, and C. Lan, in 2020 [6] content-based filtering method was used to recommend items based on the Similarity among articles. The major drawback of this method is that it ignores current users' ratings when suggesting new items. But user rating is relevant for recommending new books or journals. As the user rating information is missing in the documents, the content-based filtering has low accuracy in the current book or journal recommendation.

Esparza, Sandra Garcia, Michael P. O'Mahony, and Barry Smyth. in 2012 [7] suggested a collaborative filtering method for creating the suggestion for various items using ratings and feedback accessible on twitter. They have evaluated feedbacks given by blipper (a feedback website) for four unique products using CF method.

Chen, Li, and Feng Wang in 2013 [8] suggested a novel clustering technique built on Latent Class Regression model (LCRM), which is basically ready to consider both the general ratings and feature-level opinion values (as extracted from textual reviews) to perceive reviewers' inclination homogeneity. In the examination, they tried the proposed recommender algorithm with two true datasets. More notably, they compared it with different related methodologies, including the non-review-based technique and not-LCRM based variations.

Luo, Zhiling, Ying Li, and Jianwei Yin in 2013 [9] propose a system that considers the location as an attribute for giving the recommendation to users. Merits are 1. Better for location specific services. 2. Reduces transmission cost overhead. Demerits are 1. Not suitable where a location is not an attribute of concern. 2. Lack of similarity calculation thus not suitable for bigger datasets.

Liu, Hongyan, et al in 2013 [10] suggested recommendation technique that examines the distinction among the feedbacks of the customer to recognize the customer's predilections. These techniques consider clear ratings, an activity that can report the data sparseness issue. In view of these techniques, they also lead an experimental investigation of online-restaurant client feedbacks to make a restaurant RS and exhibit the efficiency of the suggested technique.

Chapter - 3

SOLUTION APPROACH

In this chapter considering the solution approach, Design, Methodology, Implementation and Result Analysis

3.1 Types of Recommendation System

Recommendation systems are of two types, Personalized and non-personalized. Personalized recommendation systems are those in which the group of different users is receive different suggestions where as in non-personalized recommender systems all users get same suggestions According to J. Ben Schafer, Joseph Konstan, John Non-personalized recommendation systems are automatic because in these systems recommendations are not based on customers so these systems doesn't recognize the users from one session to another and these systems requires a physical storage. Recommendation systems are grouped into these categories- Content Based Filtering, Collaborative Filtering and Hybrid Systems.

3.2 Types of Filtering

3.2.1 Collaborative Filtering

Collaborative Filtering: In 1992, "Collaborative Filtering" was invented by Goldberg el al, they conclude that for humans the process of information filtering has become very effective. The meaning of word collaboration is that people collaborate to help each other to complete a task. In collaborative filtering techniques, data and information is collected by the system (database) from different users and then based on likes and dislikes of the user the results are compared, and similar item will be suggested. In the methods of collaborative filtering the interest of one person is compared with the other user's interest and then similar items are suggested to the user.

According to G. Gupta and R. Katarya, Collaborative Filtering is a technique in recommender systems in which the recommendations are dependent on the user's neighbors and this technique use the concept of matrix factorization in which a matrix contain the users, items and the rating provide by the item with the different kind of users. These techniques are used in many kinds of e-commercial platforms and provide a better experience to suggest contents (items) than other techniques. The following diagram

easily defines the working of collaborative filtering techniques.

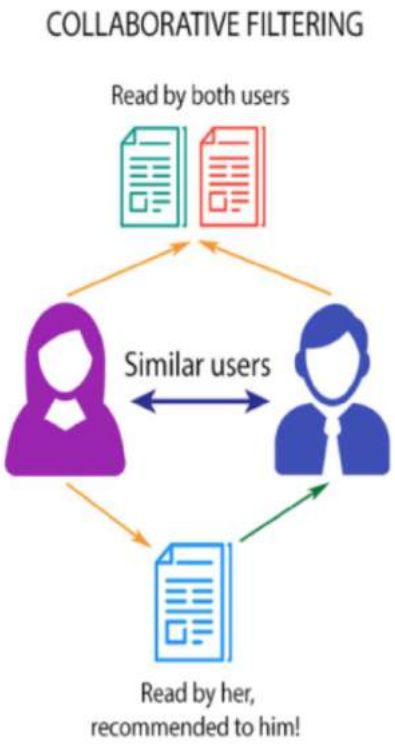

Figure 3.1: Collaborative Filtering in Recommended System

3.2.2 Content Based Filtering

Content Based Filtering: According to Po-Wah Yau and Allan Tomlinson, Firstly the quality of item is analyzed and after that the properties of product are matched, for this the present database is used. In content-based filtering techniques, the items are described with the help of keywords. Content based filtering algorithms predict the item that the user liked in the past and according to the rating of user the items are recommended. In content-based filtering the quality of the product or services are utilized for recommendation. For an active user content-based filtering techniques provide transparency. In content-based filtering the system compares the profile of the user with the content (item) and then find similar item and suggest to the user.

According to an text classification survey by Mladenic, in the technique of content based filtering the similar items are searched by the algorithm in the system, then the system construct a model based on user interest. This model generates the recommendation. The following diagram clearly shows how the content-based filtering algorithm works in the e-commercial websites.

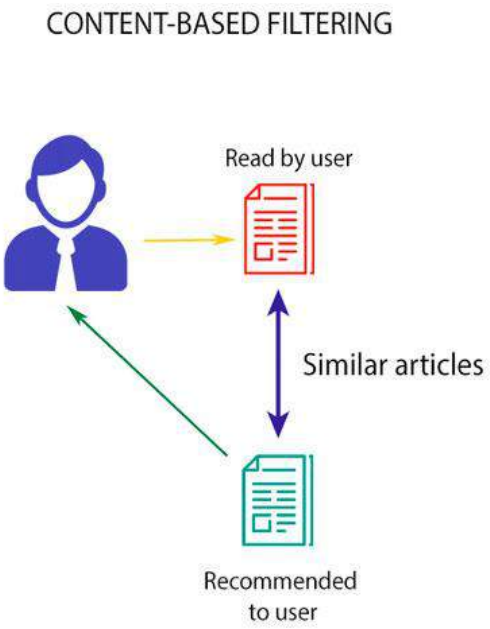

Figure 3.2: Content-Based Filtering in Recommended System

3.2.3 Hybrid Filtering

Hybrid RS involve are the combination of collaborative filtering and content-based filtering techniques. Adomavicius and Tuzhilin, suggest four different types of hybrid recommenders: Separate collaborative and content-based RS Collaborative RS with added features of content-based filtering method; Content based RS with added features of collaborative filtering method. Presents a personalized e-learning hybrid RS using the concept of "Fuzzy Tree Matching" by considering the key factors such as learning activities and learners' profiles, learning activities and pedagogical issues. proposed A Social Recommender System called as SRSH based on Hadoop parallel computing platform. SRSH system integrates content-based and collaborative filtering techniques to further improve the performance of recommendation. Presents a framework for building automatic recommendations in e-learning platforms which consists of two modules: an off-line module and an online module. The first module pre-processes the data to build user and content models. Online module uses these models dynamically to find the user's requirement and goals and predicts a list of recommendation. User preferences objects are obtained by using a "range of recommendation strategies" which are primarily based on

hybrid (content-based filtering and collaborative filtering) filtering approach.

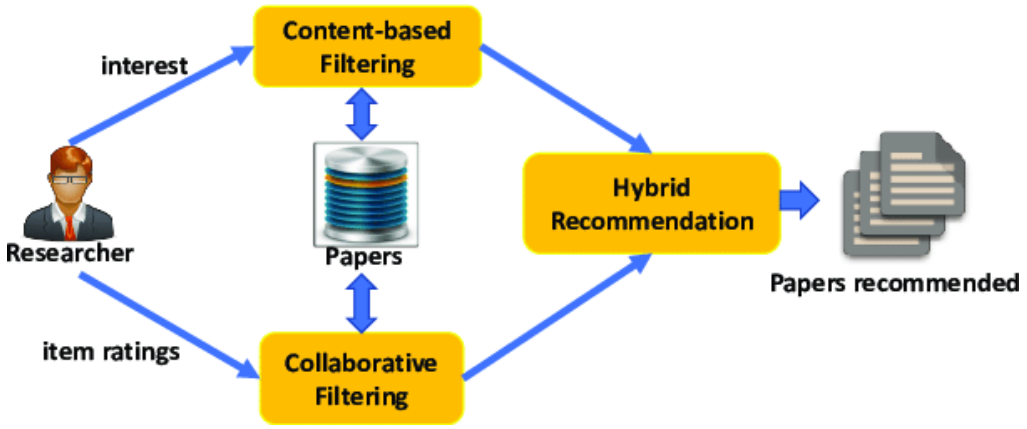

Figure 3.3: Hybrid Filtering in Recommended System

In this project use both filtering approach to build the **Recommendation System for LMS.**

Cosine Similarity defined as:

$$\text{similarity} = \cos(\theta) = \frac{\mathbf{A} \cdot \mathbf{B}}{\|\mathbf{A}\|\|\mathbf{B}\|} = \frac{\sum_{i=1}^{n} A_i B_i}{\sqrt{\sum_{i=1}^{n} A_i^2}\sqrt{\sum_{i=1}^{n} B_i^2}}$$

Correlation defined as:

$$r = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

Chapter 4

PROPOSED METHODOLOGY

4.1 Recommendation for courses

Recommendation system is an application prepared with the help of experts where the details of background of the students and their aptitudes help finding a course that is beneficial for future. A rule-based classification is being used to solve the above scenario. The term rule-based classification can be used to refer to any classification scheme that makes use of IF-THEN rules for class prediction. For this case, a set of students of a department are taken into consideration to check the working capacity of the approach. As per the academic rules and regulations, a student has to select an elective before the commencement of the third year. A student might/might not be in a position to evaluate his calibre and might end up choosing a subject in which he is not proficient yet. In other instances, a student might not be in a position to analyze the difficulty level of the elective chosen and this may end up with unnecessary consequences in the future.

As a solution to this problem, a methodological approach is devised as stated in figure:4.1 which takes into consideration - the student's performance in previous semesters as well as the student's choice of elective. This ensures a streamlined and a safe, fool proof way to recommend courses.

Before introducing the students into this CRS, all the courses available for the students throughout the graduation including electives are broadly categorized into four types of course categories namely - Programming, Conceptual, Logical and Theoretical.

Student is prompted to enroll themselves into CRS application after the completion of second year. Student is guided to the system application where in the details such as student roll number, email id is to be entered. He/she is also prompted to choose three choices from a drop-down menu of given electives. Once, the student enters his/ her preferred choice of selection, immediately, the corresponding subject tag is taken into consideration. If that case fails, the next priority subject is checked and the corresponding grade of the student in that category is chosen and so on.

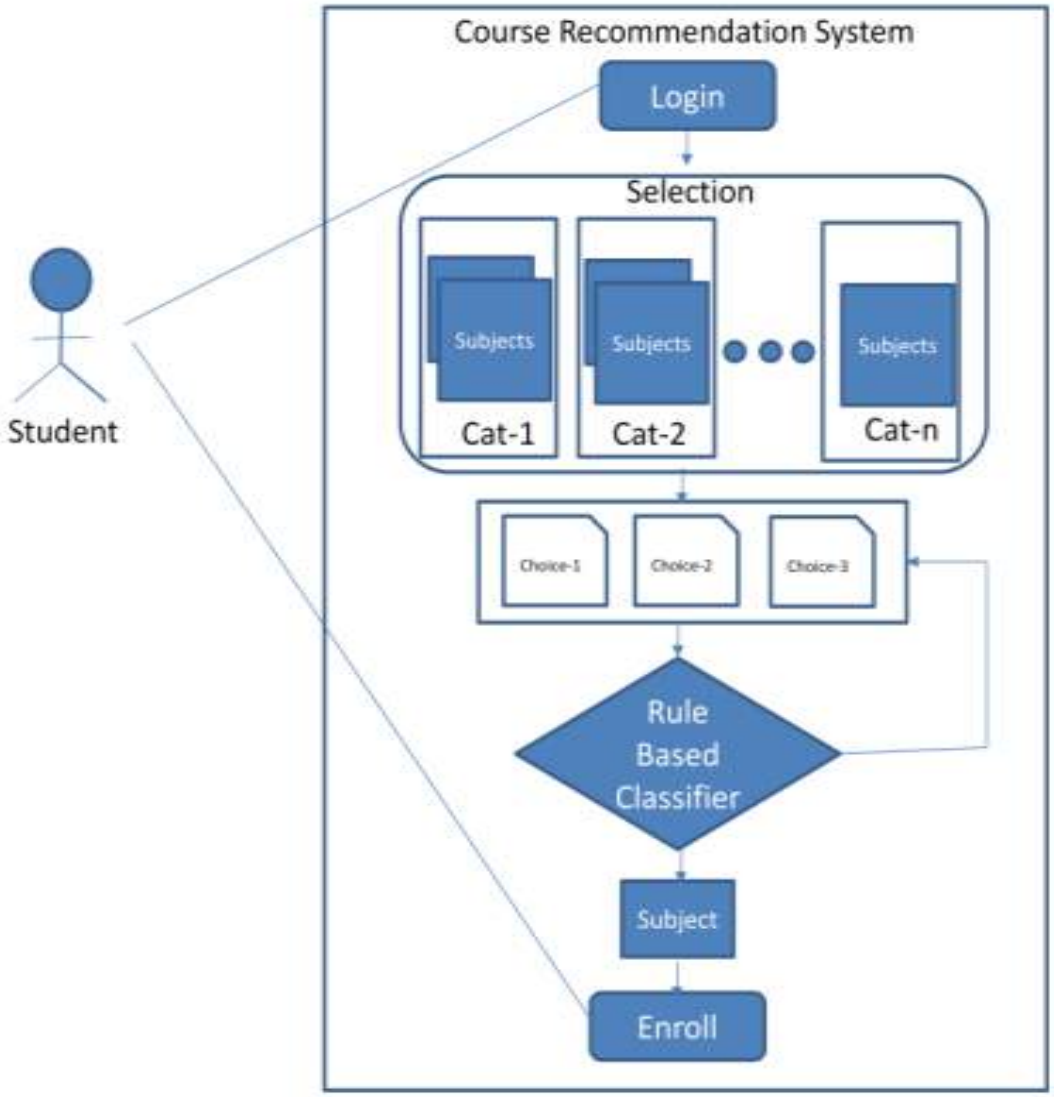

Figure 4.1: Course Recommended System

Some instances of courses and electives division into different categories in the experimental data taken are classified into courses such as C, C++, Data Structures, Python and Operating Systems into programming section. Logical Courses include Engineering Mathematics-1, Basics of Electrical Engineering, Engineering Mathematics -2, Engineering Physics, Engineering Drawing, Digital Logic Design, Discrete Mathematical Structures, Probability Statistics and Queuing Theory etc. Conceptual courses include Engineering Chemistry, Data Structures, Computer Organization, Data communication, Computer Networks, Information Systems Design.

Theoretical Subjects include Professional Ethics and Human Values, English, Soft skills lab, Workshop and Environmental Studies. Each of the elective subjects is also categorized into one of the above-mentioned categories. Software Testing and Automation (STA), Distributed Operating Systems (DOS), Client Server Technologies (CST) and Technology Management (TM) are the courses available for students to choose from. DOS is a programming course is conceptual. CST is logical and TM is theoretical. Supposedly, if a student chooses DOS as elective, then the grades of student's previous semesters consisting of programming subjects are accessed. If, the student has achieved A, A+ or O in at-least half or more subjects out of the total subjects, then the student is assigned to that course.

4.1.1 STEPS

1) Login into the Course Recommendation portal.

2) Select any three choices from the courses available according to priority. The recommendation system.

3) Takes the first choice of student into consideration.

4) With respect to the category of subject, traverse back to the grades of the student in the previous semester following the rules of rule-based classifier a. If the conditions are satisfied according to rule-based classifier go to step 5 b. Elseif goto step 4 with the second choice. c. Else goto Step 4 with third choice.

5) Displays the subject allocated to the student. The role of recommendation system ends.

6) Student has to Enroll into the course allocated and logout.

4.2 Recommendation for Books

The proposed system in Fig.4.2 used a clustering technique to develop the recommender system. Three parts are named data acquisition, preprocessing, and clustering techniques. The datasets were collected from the Kaggle in this research. Though Goodreads-books repository of Kaggle contains seven datasets, only four datasets (Books.csv, Book_tags.csv, Ratings.csv, and Max_Rating.csv) were considered for this experiment. The preprocessing technique was applied after merging all datasets where we removed the lower-rated books and developed a new dataset for analysis.

Finally, the clustering technique was applied for recommending books to those users who stay in proximity to a specific cluster. Besides, a user can then search for a book through a query interface, and results in listing recommended books.

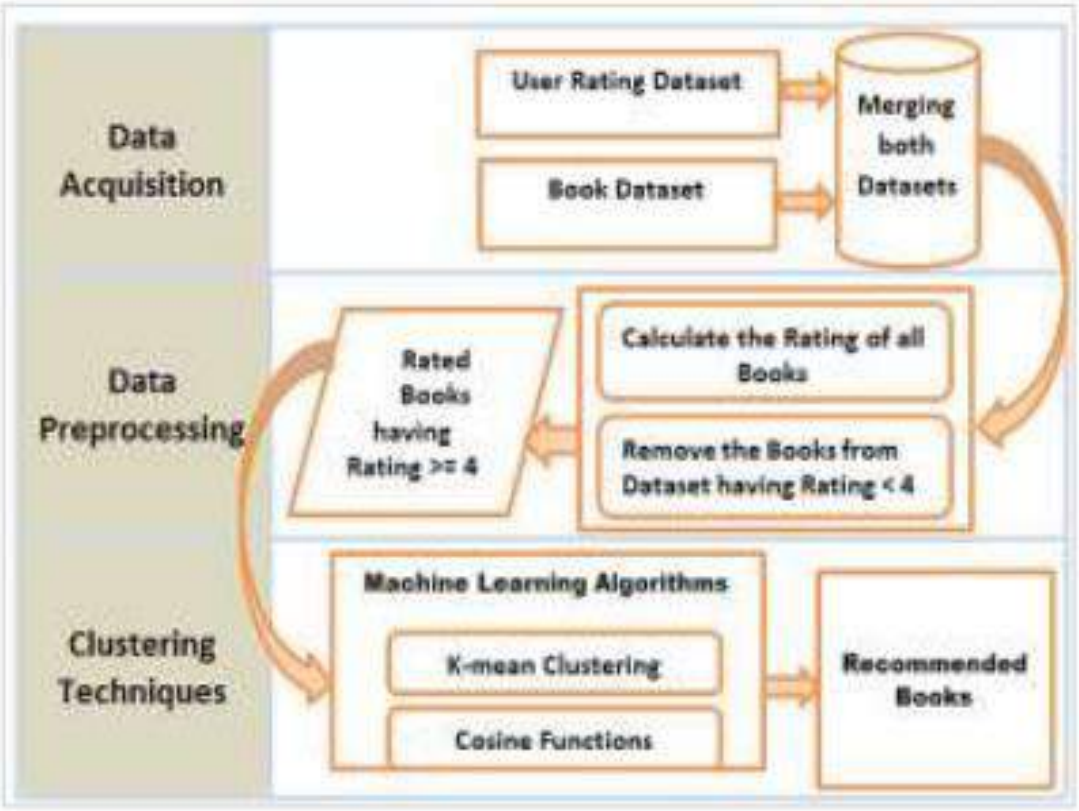

Figure 4.2: Books Recommended System

4.2.1 Data Acquisition

The dataset was collected from the Good Reads book dataset repository. It has 10,000 rated data of popular books. This data set consists of 7 tables named Books.csv, Geners.csv, Book_tags.csv, Max_rating.csv, Ratings.csv, to_read.csv, and Tags.csv, where we used Books.csv and Book_tags.csv as book dataset and Ratings.csv and Max_Rating.csv as user rating dataset.

4.2.2 Data Preprocessing

Unstructured noisy text in the data is needed to be preprocessed to make them analyzable. To do the analysis, the dataset needs to be cleaned, standardized, and noise-free. That most of the books were rated 4 or above. We want to recommend only top-rated books. So, we

remove all the rows having a rating less than 4. It shows us that 68.89% of books were rated 4 and above. Thus, our cleaned dataset becomes compact, standardized, and noise-free.

4.2.3 Clustering Techniques

K-mean algorithm is used as a cluster partition algorithm where each partition is considered as a k cluster. It is an agile algorithm applied in cluster assessment, feature discovery, and vector quantization. In this experiment, the k-mean algorithm begins with selecting the numbers of k cluster of books. Each book is assigned to the nearest cluster center and moved from the cluster center to cluster average and repeated until the algorithm reaches to convergence state.

Chapter 5

IMPLEMENTATION

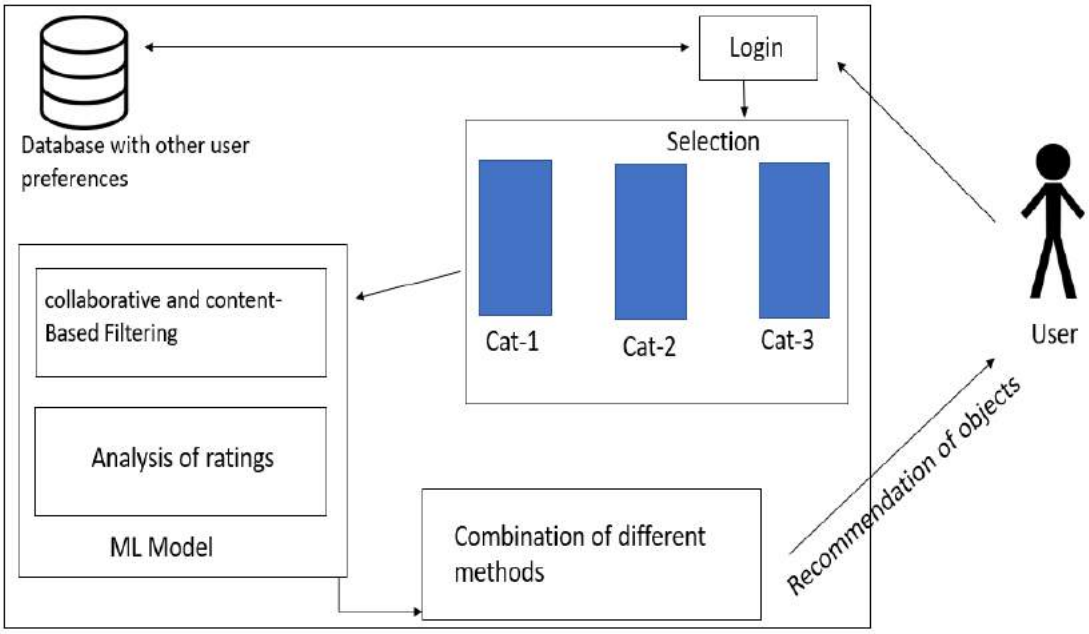

Figure 5.1: Implementation of Recommendation System

User firstly Login in web their data will store and fetch in database then a home view see a courses, books, technology section then user search her interested course, book, then recommendation system show best data according to rating.

5.1 Datasets

5.1.1 course Dataset

Data will collect by Kaggle website.

In a course section dataset have 7 columns and above 3000 values the following data is:

Course Na...	University	Difficulty L...	Course Ra...	Course URL	Course De...	Skills
Write A Feature Length Screenplay For Film Or Television	Michigan State University	Beginner	4.8	https://www.coursera.org/learn/write-a-feature-length-screenplay-for-film-or-television	Write a Full Length Feature Film Script In this course, you will write a complete, feature-length s...	Drama Comedy peering screenwritin film Documen Review dialo creative writing Writ u...
Business Strategy: Business Model Canvas Analysis with Miro	Coursera Project Network	Beginner	4.8	https://www.coursera.org/learn/canvas-analysis-miro	By the end of this guided project, you will be fluent in identifying and creating Business Model Can...	Finance business pla persona (use experience) business mod canvas Plann Business projec...
Silicon Thin Film Solar Cells	École Polytechnique	Advanced	4.1	https://www.coursera.org/learn/silicon-thin-film-solar-cells	This course consists of a general presentation of solar cells based on silicon thin films. It is t...	chemistry physics Sola Energy film lambda calcu Electrical Engineering electronics energy...
Finance for Managers	IESE Business School	Intermediate	4.8	https://www.coursera.org/learn/operational-finance	When it comes to numbers, there is always more than meets the eye. In	accounts receivable dupont analy analysis Accounting

Figure 5.2: Course Dataset

Datasets unique values of each column:

Course Name

The Name of the Course.

3416
unique values

Valid	3522	100%
Mismatched	0	0%
Missing	0	0%
Unique	3416	
Most Common	Google Clou...	0%

University

The University or Industry Partner that offers the Course.

Coursera Project Network	16%
University of Illinois at Urbana-Champaign	4%
Other (2822)	80%

Valid	3522	100%
Mismatched	0	0%
Missing	0	0%
Unique	184	
Most Common	Coursera Pr...	16%

Difficulty Level

The Difficulty Level of the Course. Has 3 values: Beginner, Intermediate, Advanced. Also has Missing Values represented by Not Calibrated.

Beginner	41%
Advanced	29%

Valid	3522	100%
Mismatched	0	0%
Missing	0	0%

Figure 5.3 (a): Unique values of course dataset columns

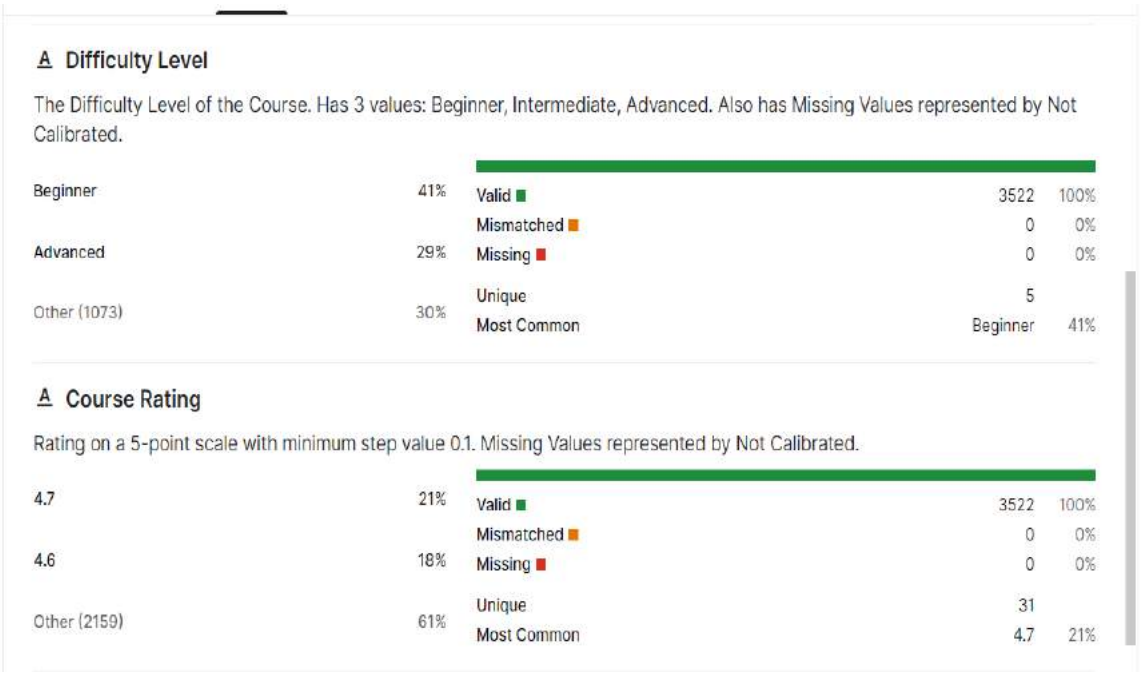

Figure 5.3 (b): Unique values of course dataset columns

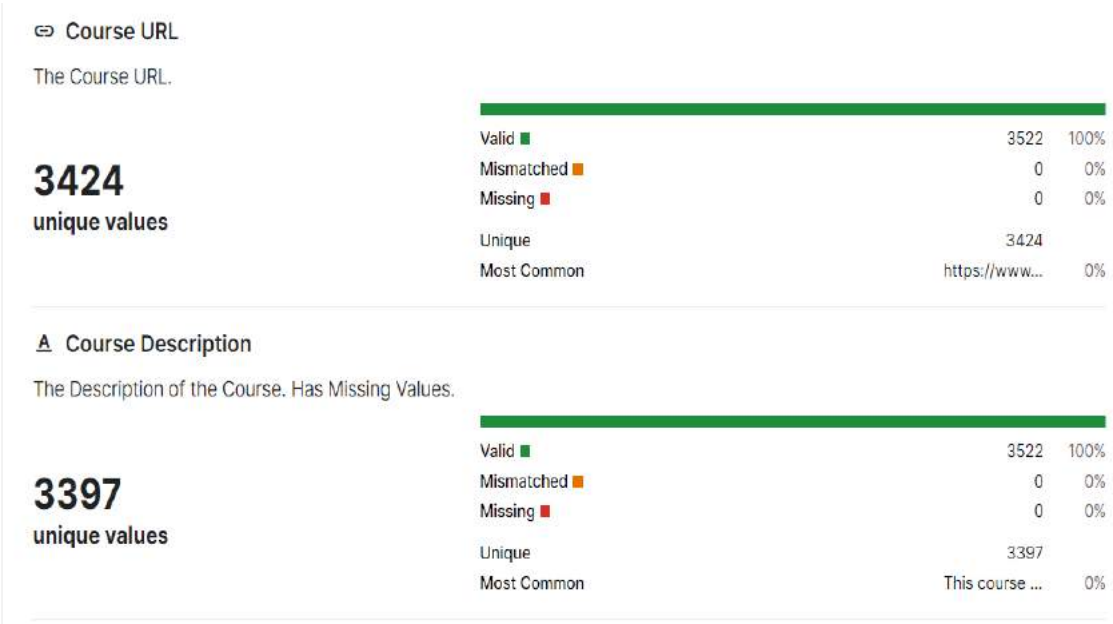

Figure 5.3 (c): Unique values of course dataset columns

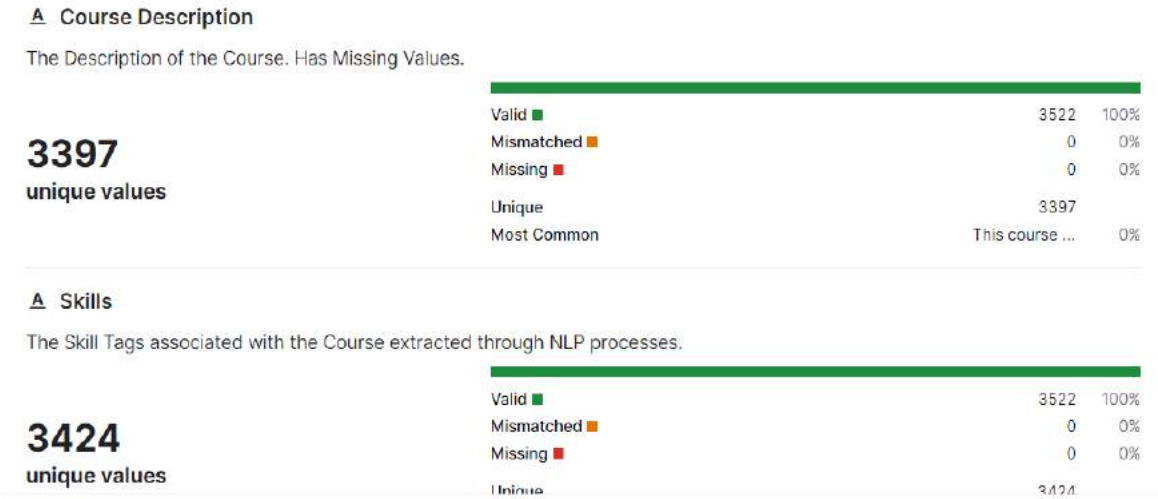

Figure 5.3(d): Unique values of course dataset columns

The following dataset is trained and tested by the ML algorithms.

Importing Dependencies

In [1]:
```
import os
import numpy as np
import pandas as pd
import matplotlib.pyplot as plt
import seaborn as sns
print('Dependencies Imported')
```

```
Dependencies Imported
```

In [2]:
```
data = pd.read_csv("../input/coursera-courses-dataset-2021/Coursera.csv")
data.head(5)
```

Out[2]:

	Course Name	University	Difficulty Level	Course Rating	Course URL	Course Description	Skills
0	Write A Feature Length Screenplay For Film Or ...	Michigan State University	Beginner	4.8	https://www.coursera.org/learn/write-a-feature...	Write a Full Length Feature Film Script In th...	Drama Comedy peering screenwriting film D...
1	Business Strategy: Business Model Canvas Analy...	Coursera Project Network	Beginner	4.8	https://www.coursera.org/learn/canvas-analysis...	By the end of this guided project, you will be...	Finance business plan persona (user experien...
2	Silicon Thin Film Solar Cells	◆cole Polytechnique	Advanced	4.1	https://www.coursera.org/learn/silicon-thin-fi...	This course consists of a general presentation...	chemistry physics Solar Energy film lambda...
3	Finance for Managers	IESE Business School	Intermediate	4.8	https://www.coursera.org/learn/operational-fin...	When it comes to numbers, there is always more...	accounts receivable dupont analysis analysis...
4	Retrieve Data using Single-Table SQL Queries	Coursera Project Network	Beginner	4.6	https://www.coursera.org/learn/single-table-sq...	In this course you◆ll learn how to effectively...	Data Analysis select (sql) database manageme...

Figure 5.4: Importing course dataset in Jupyter lab

Required Columns for System

Important columns to be used in recommendation system :

- Course Name : Names of the courses
- Course Description : Similar courses may have similar course description
- Skills : Users may want to see courses based on same skills
- Difficulty Level : Similar courses as per difficulty level

Columns not used for the recommendation system :

- Course Ratings : Numerical Column; Ratings can sometimes become a biased factor and distribution is not even
- University : Same university might offer multiple courses in different domains which the user might not want to see
- Course URL : No significance in the recommendation system

In [10]:
```
data = data[['Course Name','Difficulty Level','Course Description','Skills']]
```

In [11]:
```
data.head(5)
```

Out[11]:

	Course Name	Difficulty Level	Course Description	Skills
0	Write A Feature Length Screenplay For Film Or ...	Beginner	Write a Full Length Feature Film Script In th...	Drama Comedy peering screenwriting film D...
1	Business Strategy: Business Model Canvas Analy...	Beginner	By the end of this guided project, you will be...	Finance business plan persona (user experien...
2	Silicon Thin Film Solar Cells	Advanced	This course consists of a general presentation...	chemistry physics Solar Energy film lambda...
3	Finance for Managers	Intermediate	When it comes to numbers, there is always more...	accounts receivable dupont analysis analysis...
4	Retrieve Data using Single-Table SQL Queries	Beginner	In this course you'll learn how to effectively...	Data Analysis select (sql) database manageme...

Figure 5.5: Prepare course dataset columns.

Data Pre-Processing

An important part of the process is to pre-process the data into usable format for the recommendation system

In [12]:
```
# Removing spaces between the words (Lambda funtions can be used as well)

data['Course Name'] = data['Course Name'].str.replace(' ',',')
data['Course Name'] = data['Course Name'].str.replace(',,',',')
data['Course Name'] = data['Course Name'].str.replace(':','')
data['Course Description'] = data['Course Description'].str.replace(' ',',')
data['Course Description'] = data['Course Description'].str.replace(',,',',')
data['Course Description'] = data['Course Description'].str.replace('_','')
data['Course Description'] = data['Course Description'].str.replace(':','')
data['Course Description'] = data['Course Description'].str.replace('(','')
data['Course Description'] = data['Course Description'].str.replace(')','')

#removing paranthesis from skills columns
data['Skills'] = data['Skills'].str.replace('(','')
data['Skills'] = data['Skills'].str.replace(')','')
```

Figure 5.6: Data Pre-Processing

Tags Column

The tags column is the combination of the following columns : **Course Name** + **Difficulty Level** + **Course Description** + **Skills**

In [14]:
```
data['tags'] = data['Course Name'] + data['Difficulty Level'] + data['Course Description'] + data['Skills']
```

In [15]:
```
data.head(5)
```

Out[15]:

	Course Name	Difficulty Level	Course Description	Skills	tags
0	Write,A,Feature,Length,Screenplay,For,Film,Or,...	Beginner	Write,a,Full,Length,Feature,Film,Script,In,thi...	Drama Comedy peering screenwriting film D...	Write,A,Feature,Length,Screenplay,For,Film,Or,...
1	Business,Strategy,Business,Model,Canvas,Analys...	Beginner	By,the,end,of,this,guided,project,you,will,be,...	Finance business plan persona user experienc...	Business,Strategy,Business,Model,Canvas,Analys...
2	Silicon,Thin,Film,Solar,Cells	Advanced	This,course,consists,of,a,general,presentation...	chemistry physics Solar Energy film lambda...	Silicon,Thin,Film,Solar,CellsAdvancedThis,cour...
3	Finance,for,Managers	Intermediate	When,it,comes,to,numbers,there,is,always,more,...	accounts receivable dupont analysis analysis...	Finance,for,ManagersIntermediateWhen,it,comes,...
4	Retrieve,Data,using,Single-Table,SQL,Queries	Beginner	In,this,course,you,ll,learn,how,to,effectively...	Data Analysis select sql database management...	Retrieve,Data,using,Single-Table,SQL,QueriesBe...

In [16]:
```
data['tags'].iloc[1]
```

Figure 5.7: Tags columns

Text Vectorization

In [25]:
```
from sklearn.feature_extraction.text import CountVectorizer
```

In [26]:
```
cv = CountVectorizer(max_features=5000,stop_words='english')
```

In [27]:
```
vectors = cv.fit_transform(new_df['tags']).toarray()
```

Stemming Process

In [28]:
```
import nltk #for stemming process
```

In [29]:
```
from nltk.stem.porter import PorterStemmer
ps = PorterStemmer()
```

In [30]:
```
#defining the stemming function
def stem(text):
    y=[]

    for i in text.split():
        y.append(ps.stem(i))

    return " ".join(y)
```

In [31]:
```
new_df['tags'] = new_df['tags'].apply(stem) #applying stemming on the tags column
```

Figure 5.8: Vectorization course dataset

Similarity Measure

In [32]:
```
from sklearn.metrics.pairwise import cosine_similarity
```

In [33]:
```
similarity = cosine_similarity(vectors)
```

Recommendation Function

In [34]:
```
def recommend(course):
    course_index = new_df[new_df['course_name'] == course].index[0]
    distances = similarity[course_index]
    course_list = sorted(list(enumerate(distances)),reverse=True, key=lambda x:x[1])[1:7]

    for i in course_list:
        print(new_df.iloc[i[0]].course_name)
```

In [35]:
```
recommend('Business Strategy Business Model Canvas Analysis with Miro')
```

```
Product Development Customer Persona Development with Miro
Product and Service Development Empathy Mapping with Miro
Product Development Customer Journey Mapping with Miro
Analyzing Macro-Environmental Factors Using Creately
Business Strategy in Practice (Project-centered Course)
Innovating with the Business Model Canvas
```

So these are the 6 courses which are recommended based on our search in the recommendation function

Exporting the Model

Figure 5.9: Build Recommendation Function in course dataset.

5.1.2 Books Dataset

Data will collect by Kaggle website.

In a course section dataset have 10 columns and above 2000 values the following data is:

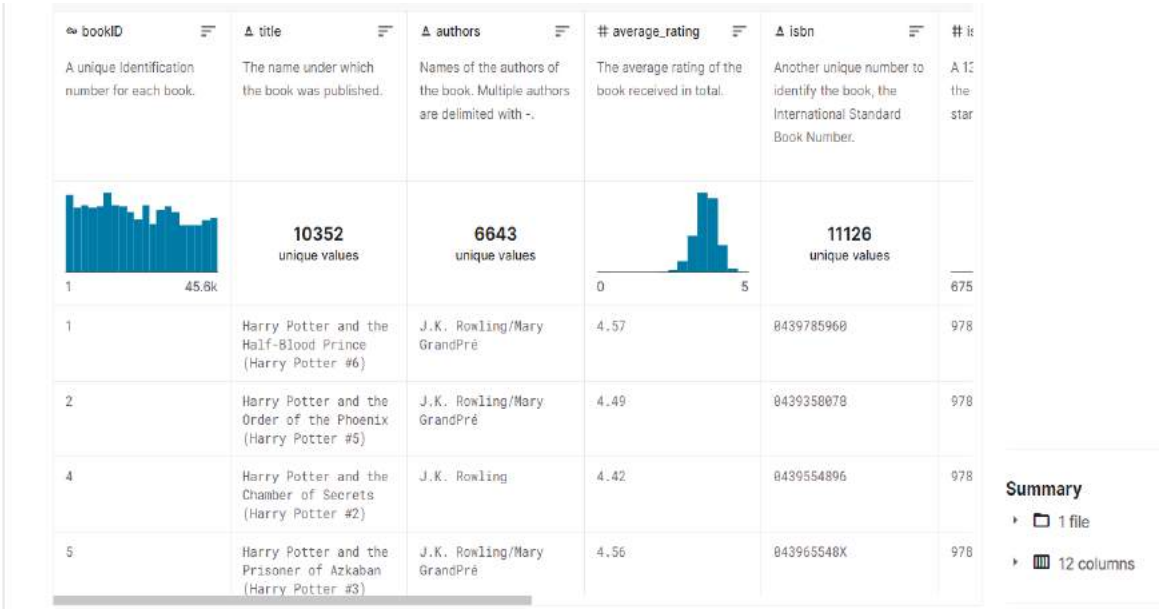

Figure 5.10: Books Dataset

Datasets unique values of each column:

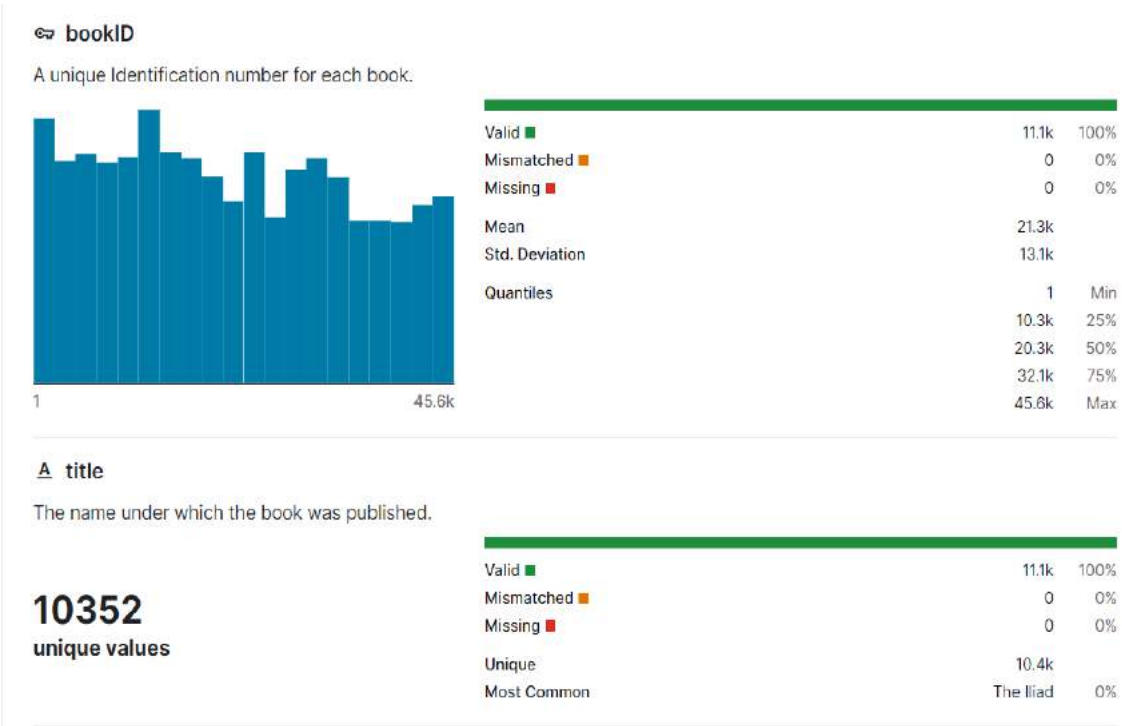

Figure 5.11 (a): Unique values of course dataset columns

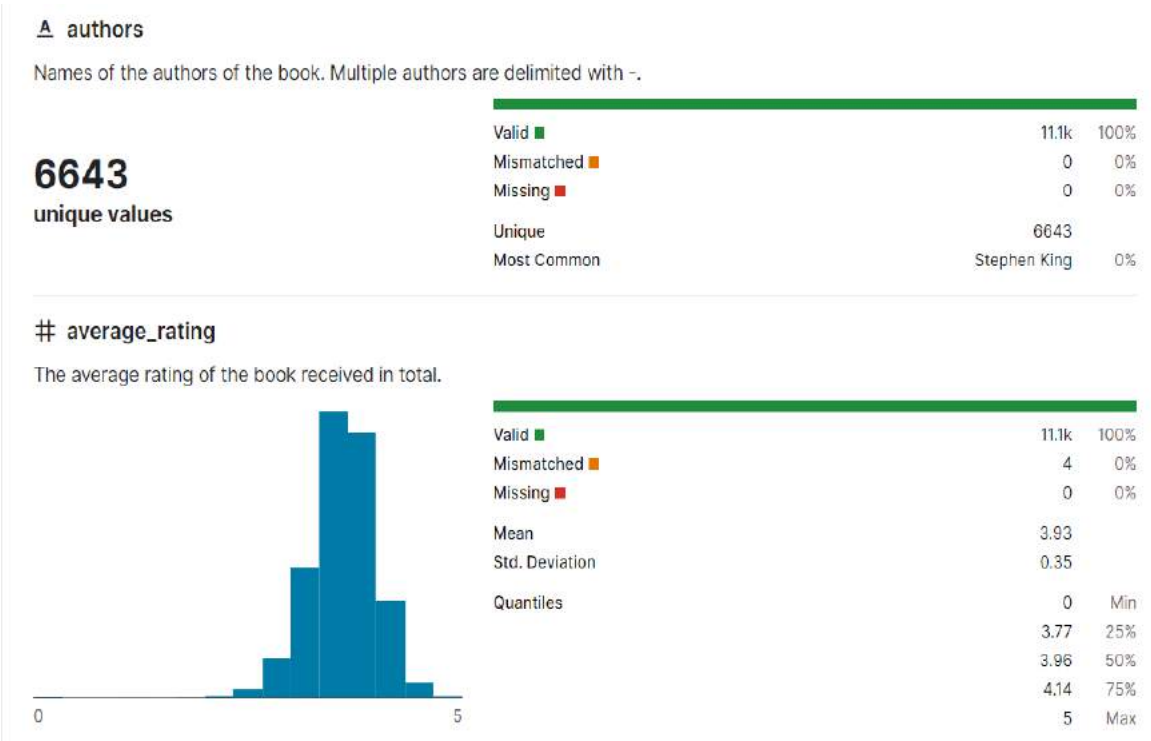

Figure 5.11 (b): Unique values of course dataset columns

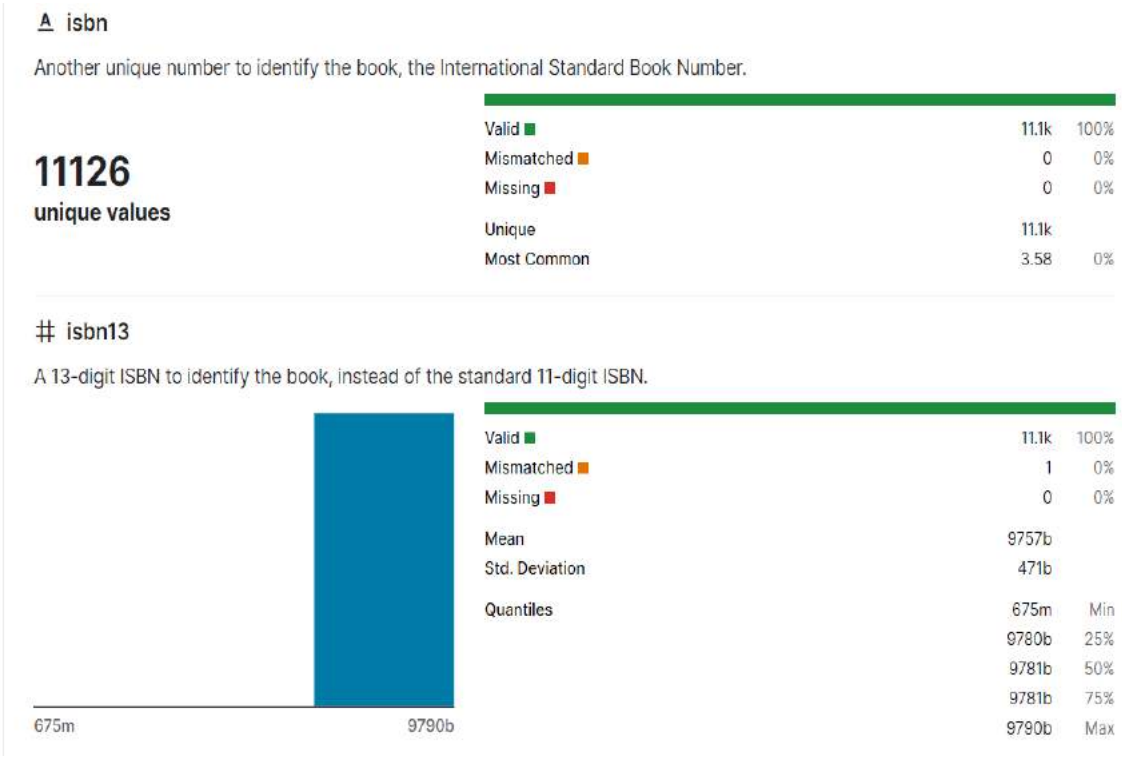

Figure 5.11 (c): Unique values of course dataset columns

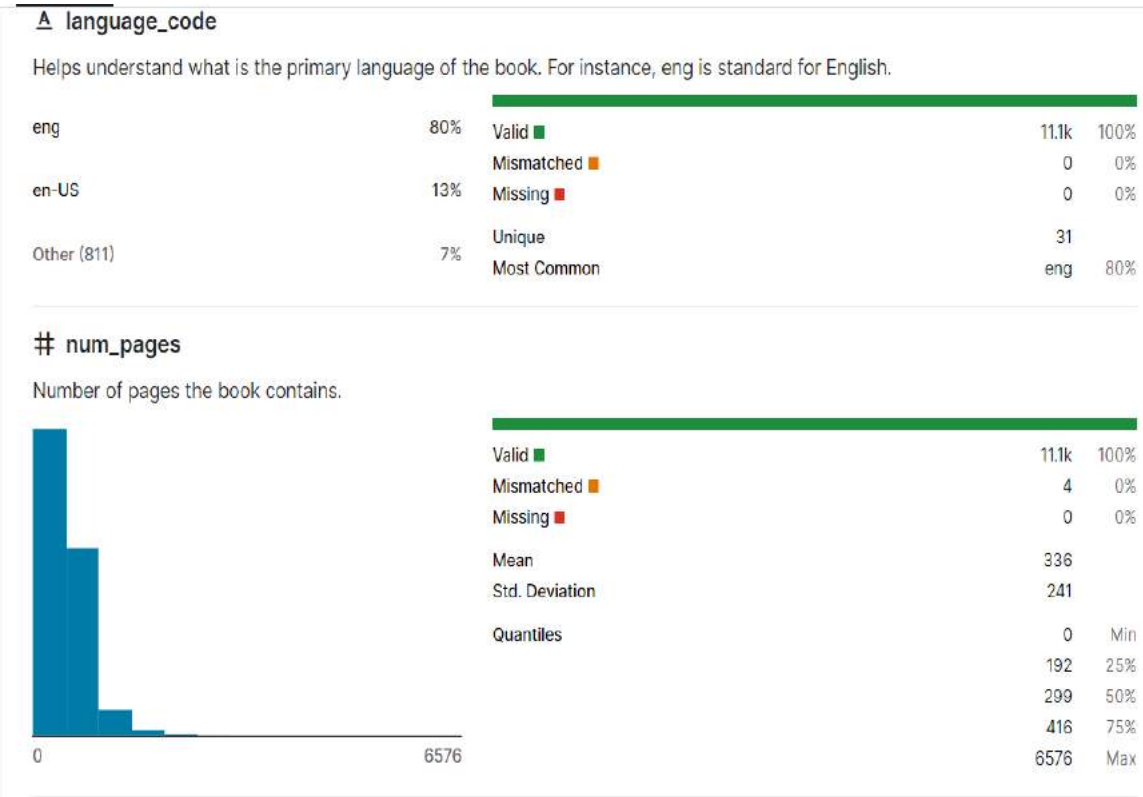

Figure 5.11 (d): Unique values of course dataset columns

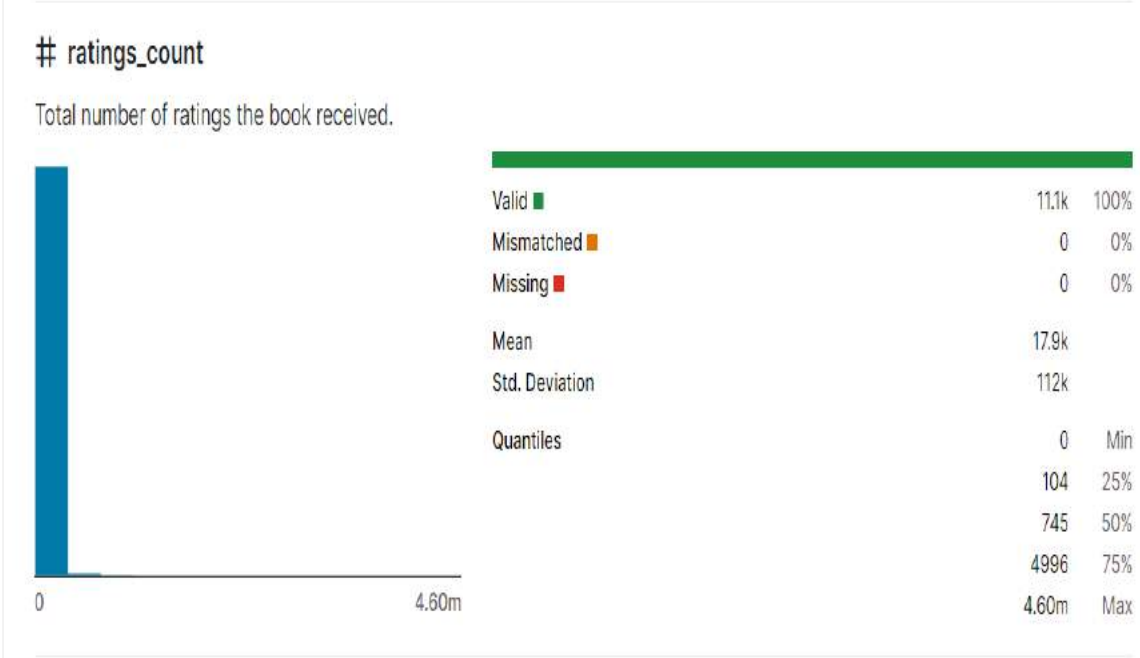

Figure 5.11 (e): Unique values of course dataset columns

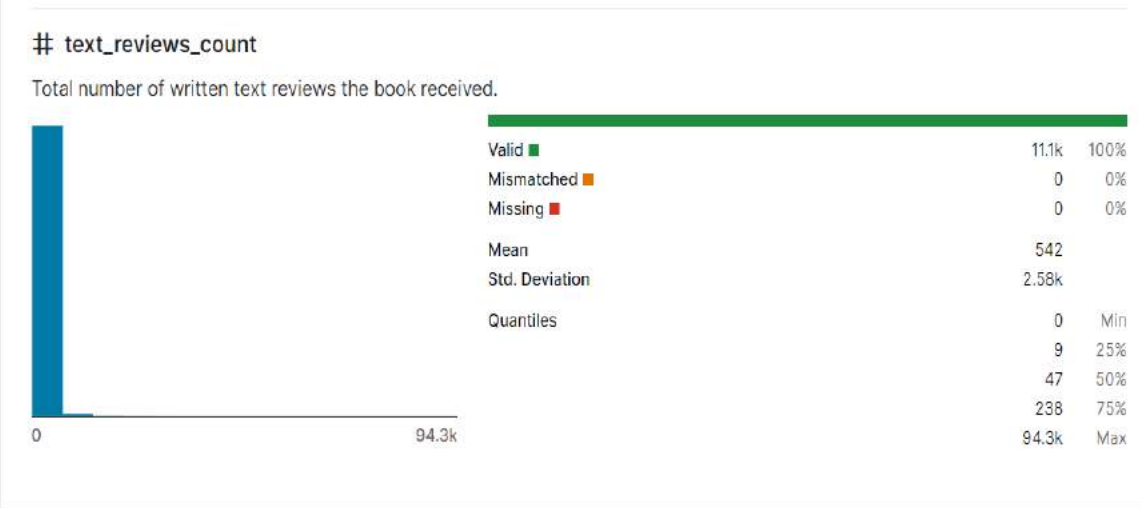

Figure 5.11 (f): Unique values of course dataset columns

The following dataset is train and test by the ML algorithms same as course dataset.

5.2 Technology Stack

In this project used following technology and their libraries

5.2.1 Python

For building Recommendation system using ML algorithm used the Python programming language.

Following Python framework and libraries used in this project:

- Framework
  - Flask for web API
- Libraries
  - Pandas
  - NumPy

- Matplotlib
- TensorFlow
- Scikit-learn
- Seaborn

5.2.2 Database and Front-end

- HTML
- CSS
- JS
- MySQL

5.2.3 Deploying ML Model

For deploying ML model used the Heroku or AWS

Chapter 6

RESULT ANALYSIS

6.1 Result analysis

The Result analysis of in this project is show in following figures: -

First XAMPP server open for admin panel and database is created in SQL for login, sign-up to Get Post method in flask app.

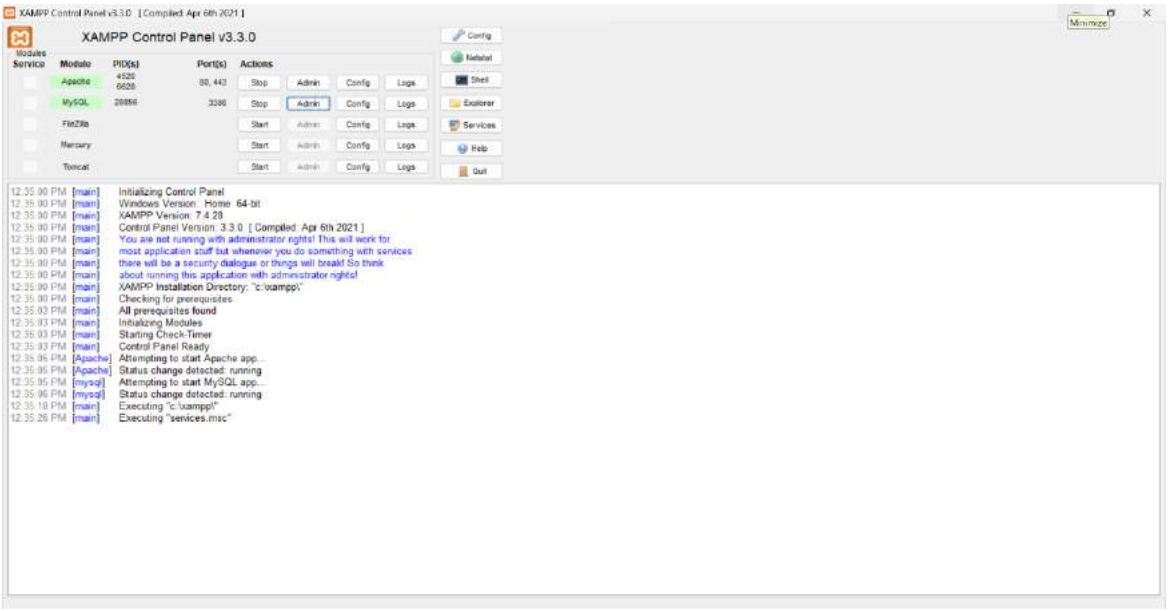

Figure 6.1: XAMPP Server start

The following code is written in vs code in flask some code is here.

```
from flask import Flask, render_template, request, redirect, session
import mysql.connector

import pickle
import numpy as np
popular_df=pickle.load(open('popular.pkl','rb'))
pt=pickle.load(open('pt.pkl','rb'))
books=pickle.load(open('books.pkl','rb'))
similarity_scores=pickle.load(open('similarity_scores.pkl','rb'))
course_pop_df=pickle.load(open('course.pkl','rb'))
import os

app = Flask(__name__, template_folder='templates', static_folder='static')
app.secret_key=os.urandom(24)

conn=mysql.connector.connect(host="localhost", user="root", database="webdb", password="")
cursor=conn.cursor()
@app.route('/')
def mainpage():
    return render_template('index.html')

@app.route('/login')
def login():
    return render_template('login.html')

@app.route('/signup')
def signup():
    return render_template('signup.html')
```

Figure 6.2: (a) flask app program

```
@app.route('/home')
def home():
        if 'user_id' in session:
            return render_template('home.html',book_name=list(popular_df['Book-Title'].values),
                                   author=list(popular_df['Book-Author'].values),
                                   image=list(popular_df['Image-URL-M'].values),
                                   votes=list(popular_df['num_ratings'].values),
                                   rating=list(popular_df['avg_rating'].values),)
        else:
            return redirect('login.html')


@app.route('/login_validation', methods=['POST'])
def login_validation():
    email=request.form.get('email')
    password=request.form.get('password')

    cursor.execute("""SELECT * FROM `users` WHERE `email` LIKE '{}' AND `password` LIKE '{}' """
                   .format(email,password))
    users=cursor.fetchall()
    if len(users)>0:
        session['user_id']=users[0][0]
        return redirect('home')
    else:
        return redirect('/')
```

Figure 6.2: (b) flask app program

```
@app.route('/add_user',methods=['POST'])
def add_user():
    name=request.form.get('aname')
    email=request.form.get('aemail')
    password=request.form.get('apassword')

    cursor.execute("""INSERT INTO `users` (`user_id`,`name`,`email`,`password` ) VALUES
    (NULL,'{}','{}','{}')""".format(name,email,password))
    conn.commit()

    cursor.execute("""SELECT * FROM `users` WHERE `email` LIKE '{}' """.format(email))
    myuser=cursor.fetchall()
    session['user_id']=myuser[0][0]
    return redirect('/login')


@app.route('/logout')
def logout():
    session.pop('user_id')
    return redirect('/')


@app.route('/home')
def index():
    return render_template('home.html',
                           book_name=list(popular_df['Book-Title'].values),
                           author=list(popular_df['Book-Author'].values),
                           image=list(popular_df['Image-URL-M'].values),
```

Figure 6.2: (c) flask app program

```python
@app.route('/recommend')
def recommend_ui():
    return render_template('recommend.html')
@app.route('/recommend_books',methods=['post'])
def recommend():
    user_input = request.form.get('user_input')

    index = np.where(pt.index == user_input)[0][0]
    similar_items = sorted(list(enumerate(similarity_scores[index])), key=lambda x: x[1], reverse=True)[1:5]

    data = []
    for i in similar_items:
        item = []
        temp_df = books[books['Book-Title'] == pt.index[i[0]]]
        item.extend(list(temp_df.drop_duplicates('Book-Title')['Book-Title'].values))
        item.extend(list(temp_df.drop_duplicates('Book-Title')['Book-Author'].values))
        item.extend(list(temp_df.drop_duplicates('Book-Title')['Image-URL-M'].values))

        data.append(item)

    print(data)

    return render_template('recommend.html',data=data)
```

Figure 6.2: (d) flask app program

```python
@app.route('/courses')
def courses():
        return render_template('courses.html',course_name=list(course_pop_df['Course Name'].values),
                               University=list(course_pop_df['University'].values),
                               link=list(course_pop_df['Course URL'].values),
                               ratings=list(popular_df['num_ratings'].values),)
```

Figure 6.2: (e) flask app program

To run this code using python./app.py command then the main page of website wayout Doubt is open.

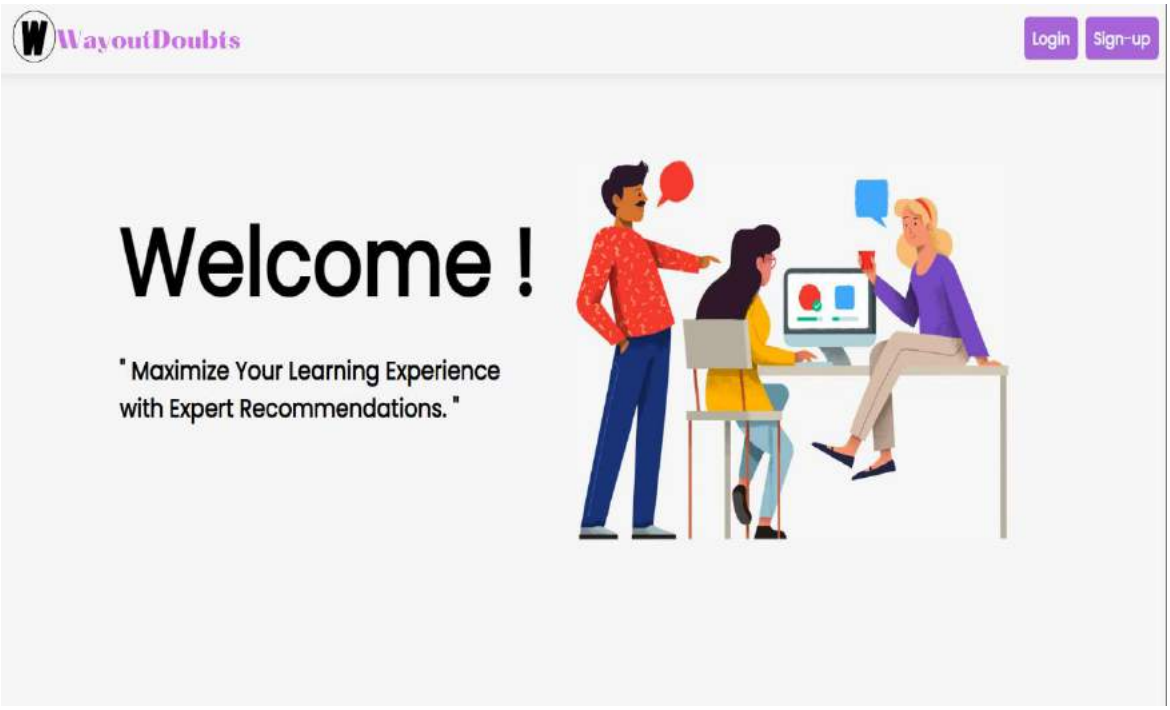

Figure 6.3: Main page of web

The user logs in and signup the page then enter in home page of the website.

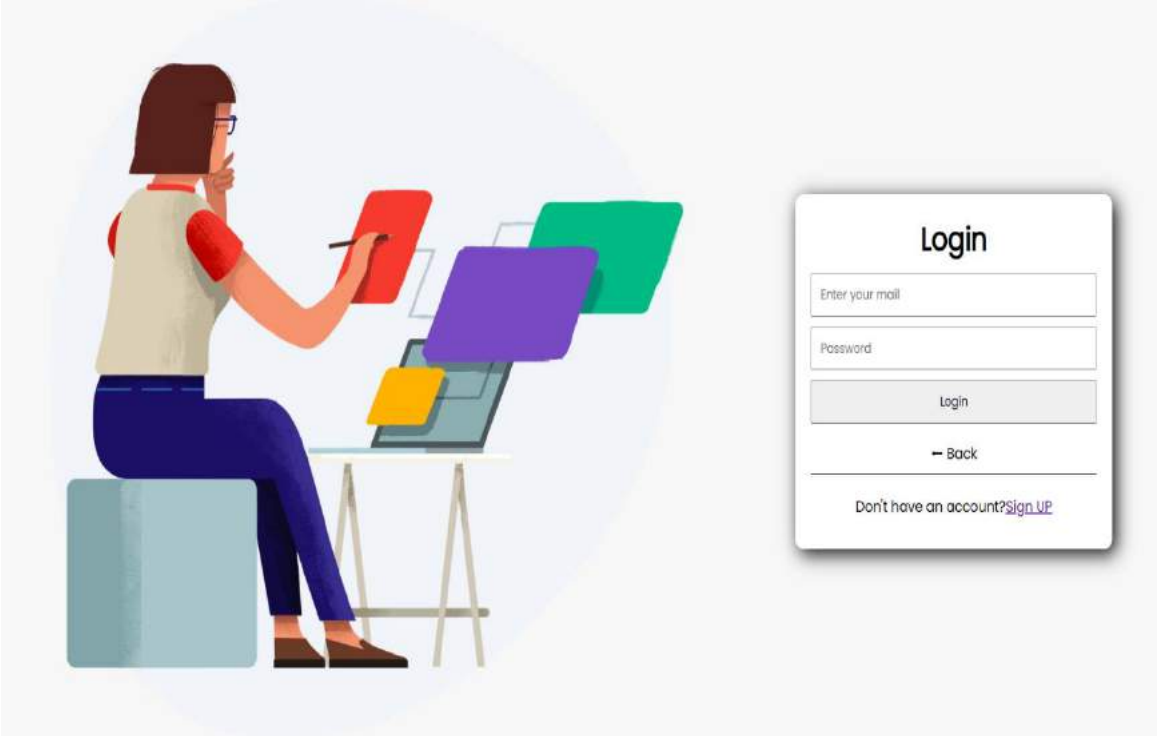

Figure 6.4: login page

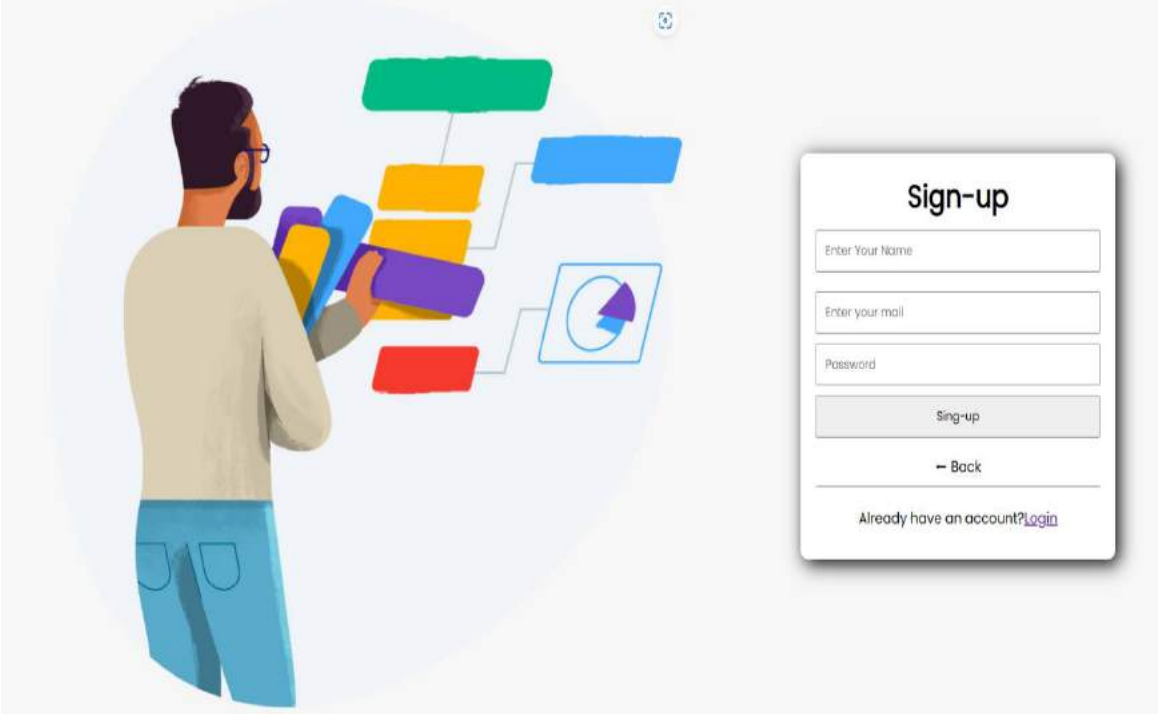

Figure 6.5: Sign-up page

Database entry is shown in following figure.

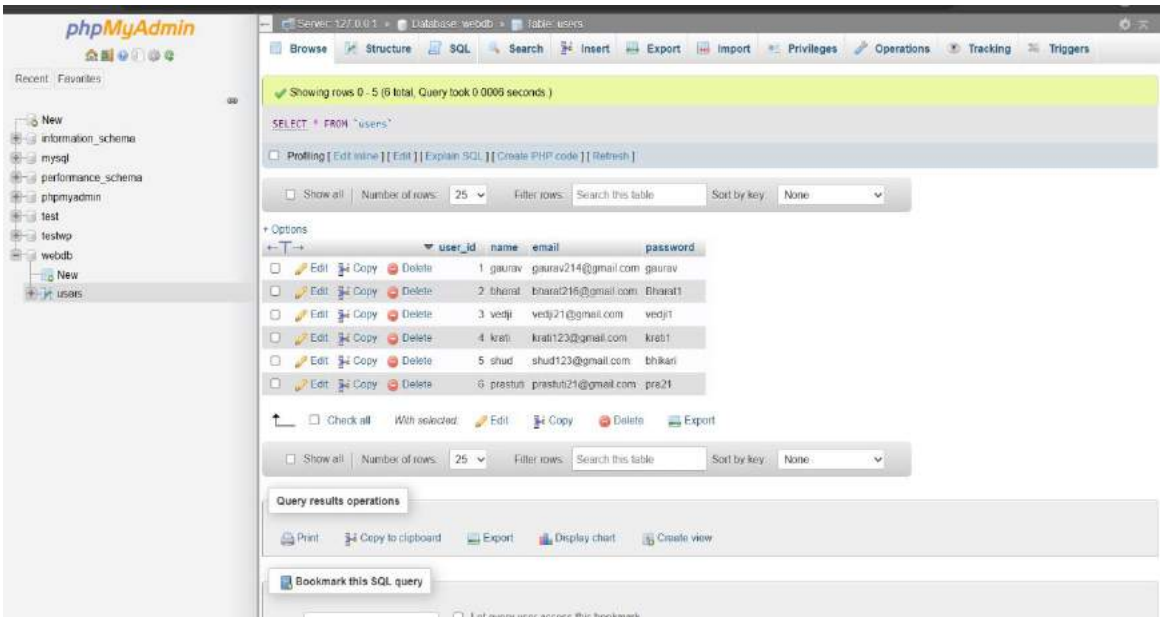

Figure 6.6: Database of login and signup users Data.

After login and signup, the user enters in home page of our website in the home page using

popularity-based filtering first show the Top 20 books data to user.

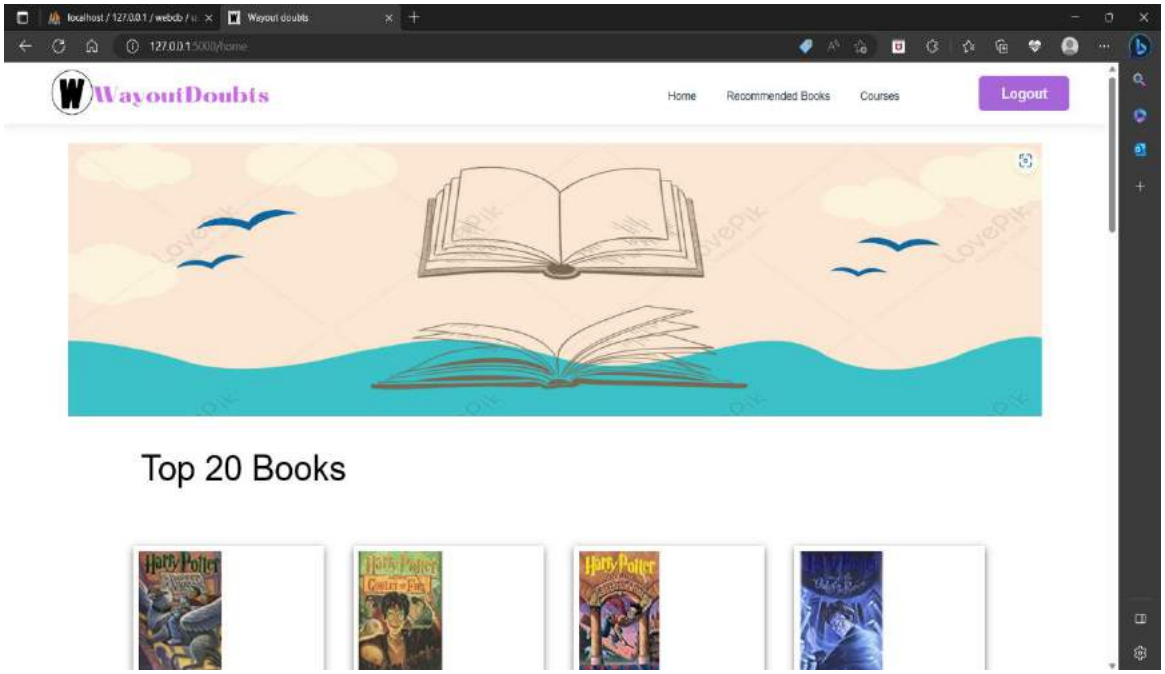

Figure 6.7: Home Page of Website

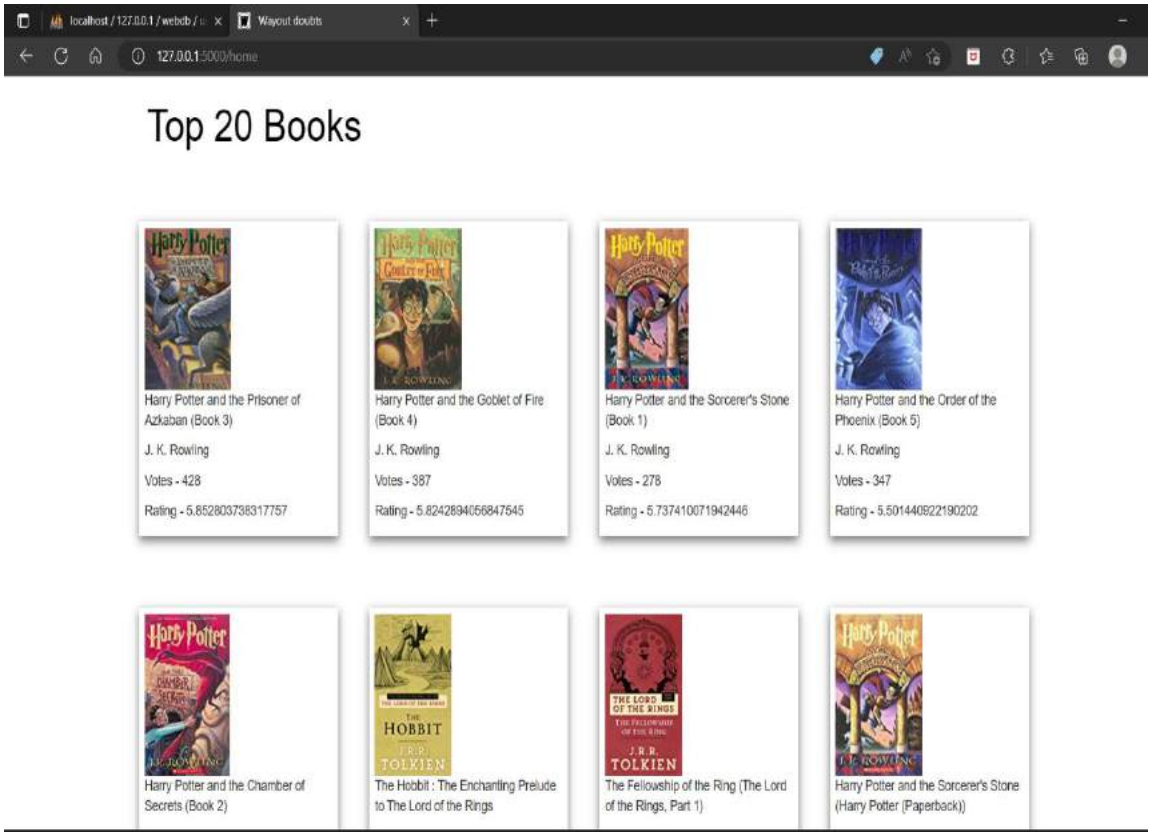

Figure 6.8:(a) Top 20 Books shows to user.

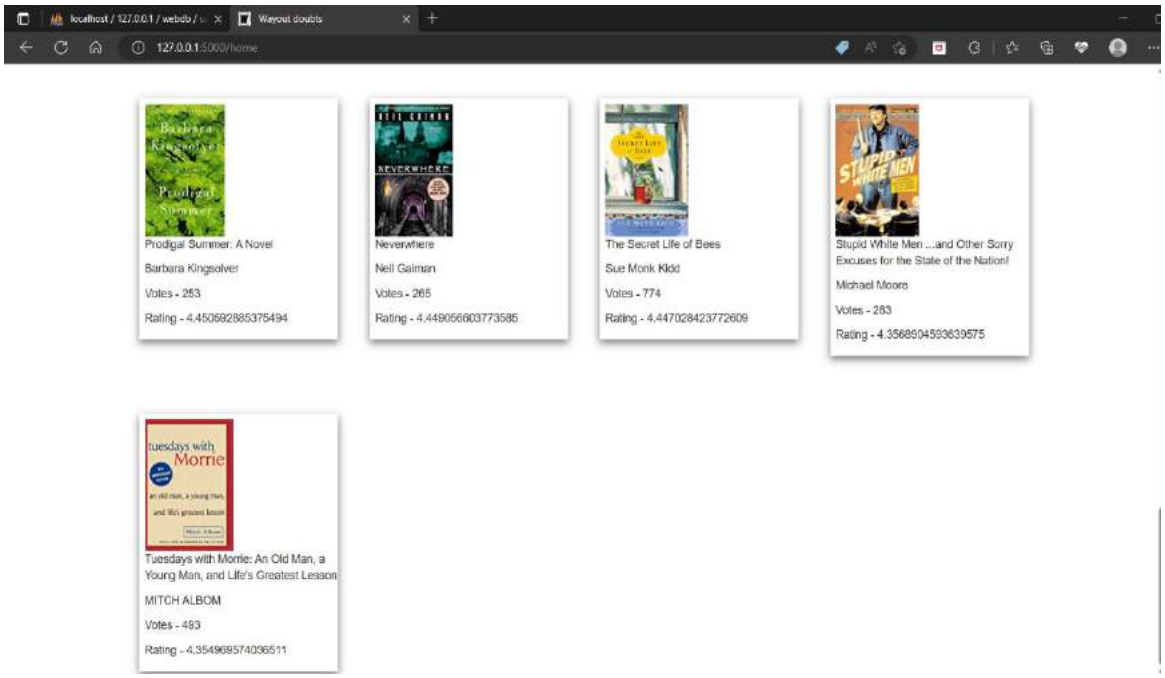

Figure 6.8:(b) Top 20 Books shows to user.

In the navbar of home page Books Section where the user enters and search the best recommended books to user using popularity based and content-based filtering.

Top 4 best rating-based books are showed by our recommendation system to the user.

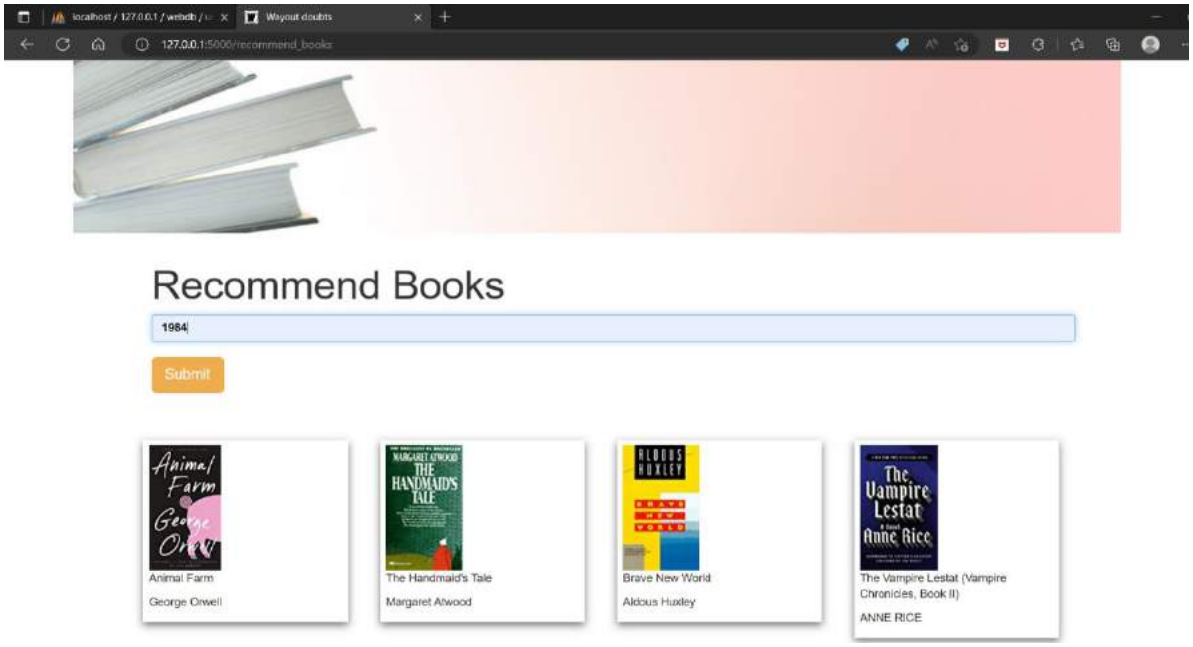

Figure 6.9:(a) Recommended Books shows to user.

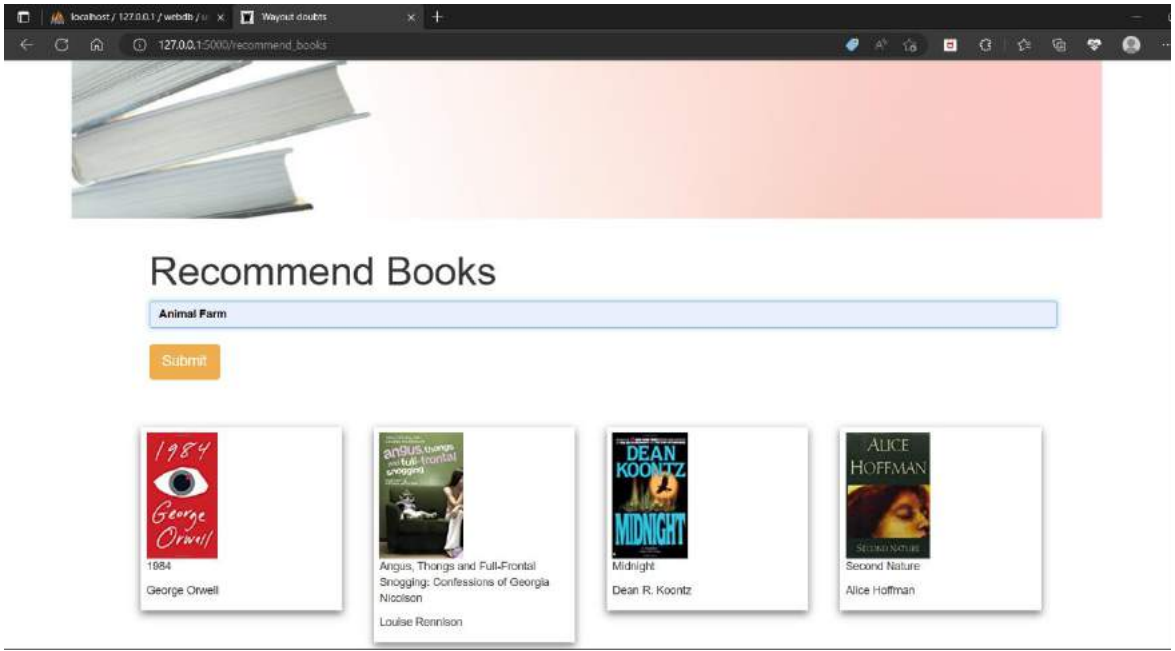

Figure 6.9:(b) Recommended Books shows to user.

In the navbar a course section is also included. When user enter in this section then top courses coursera data are show to user.

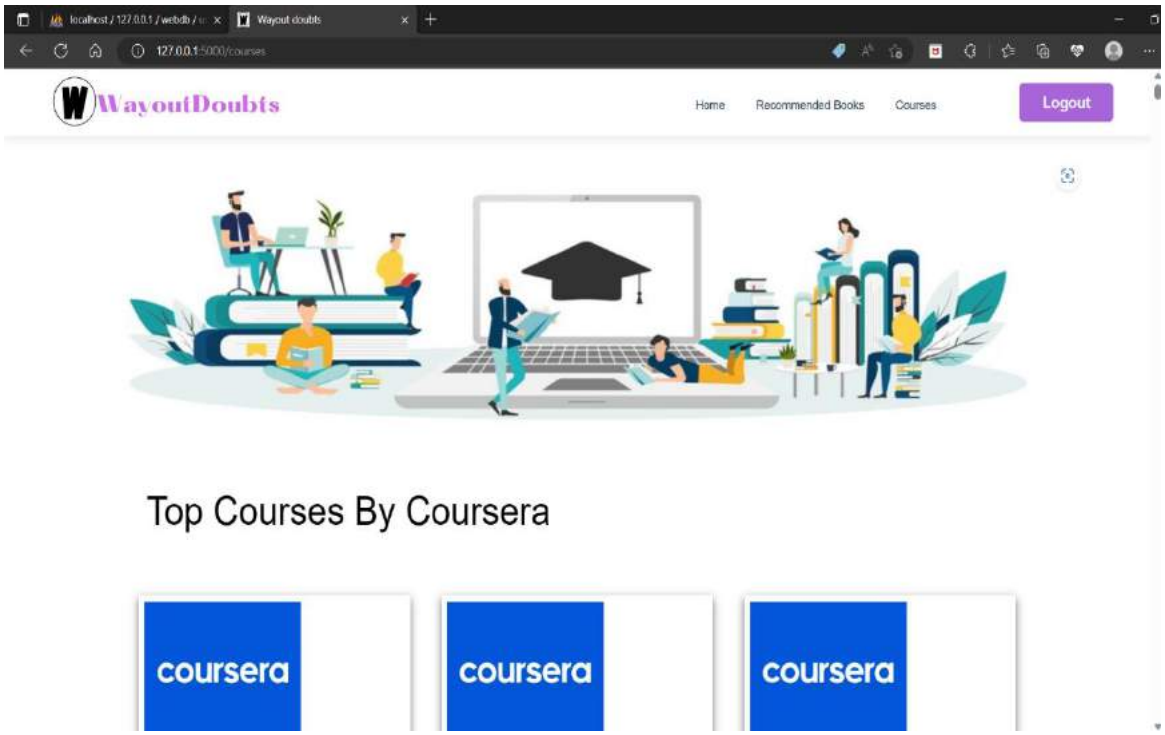

Figure 6.10:(a) Courses shows to user.

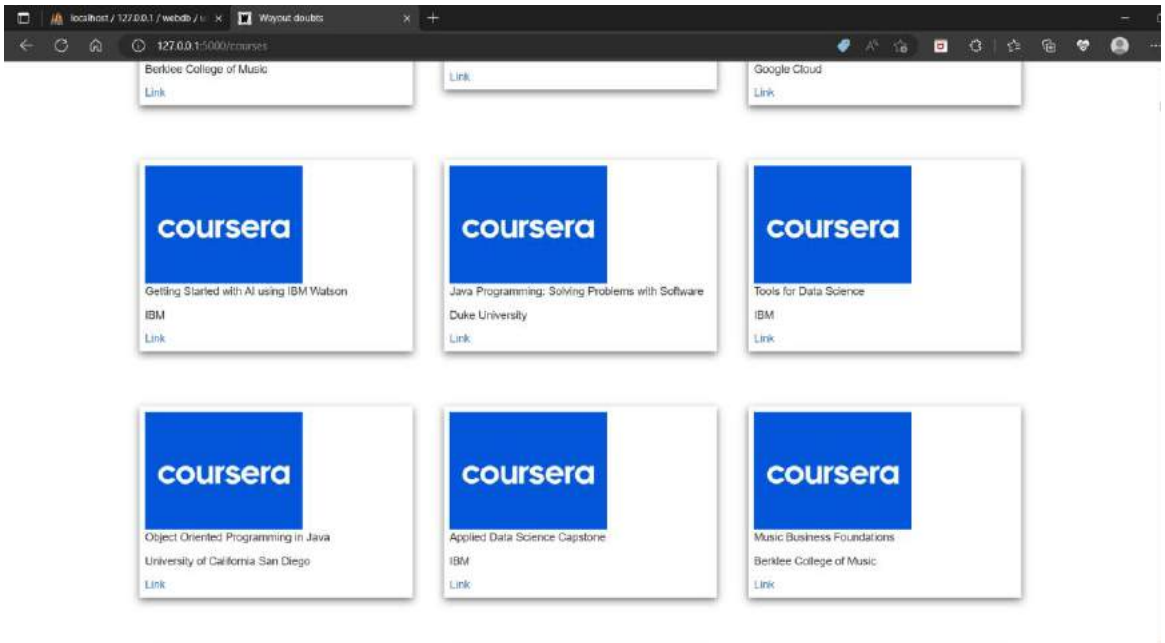

Figure 6.10:(b) Courses shows to user.

In every course card a link is also available to re-direct the coursera website to enroll the courses and get access of the courses the following example is below

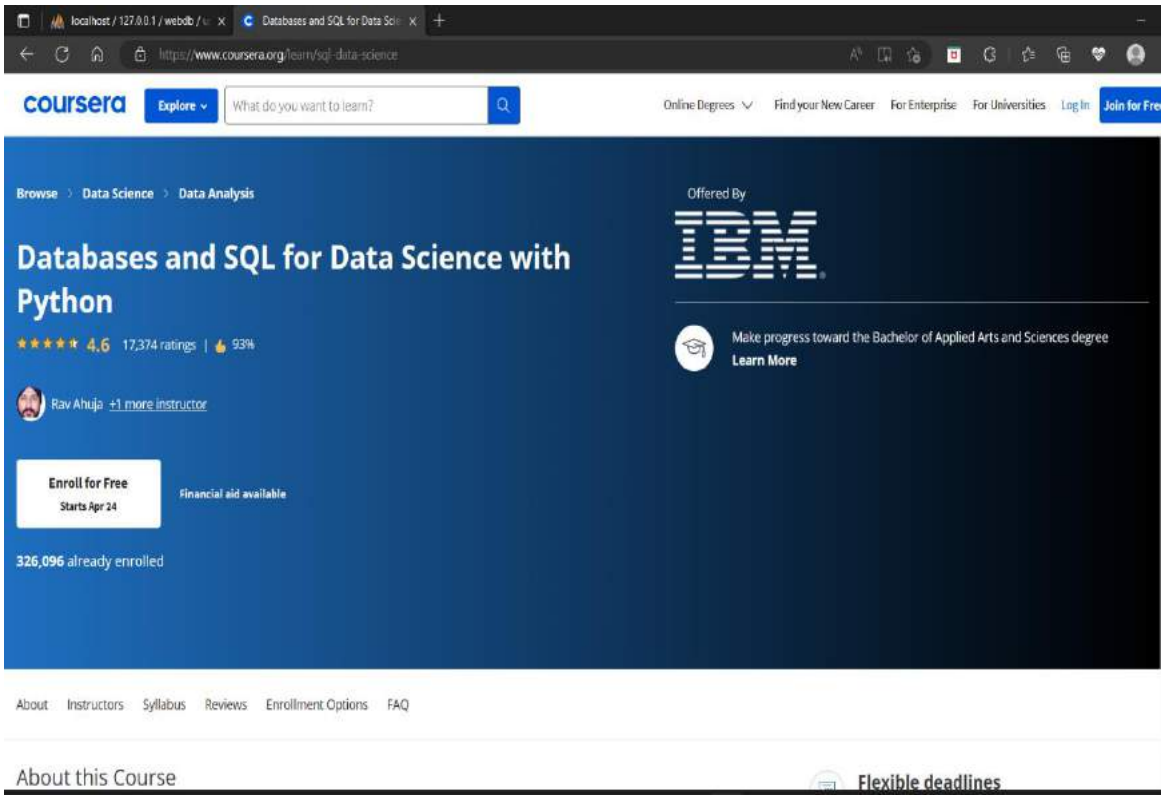

Figure 6.11: Courses links to coursera.

So, in this project the recommendation system is working quite well, expected output we get for rendering the ML we used the pickle library and render template in flask.

Recommendation system is giving the output of 4.5+ rating courses and books data using popularity-based filtering and content-based filtering and user login and sign-up is working using XAMPP server admin panel is created and using Get and Post method user data is saved in admin panel.

Chapter-7

FUTURE SCOPE AND CONCLUSION

7.1 Conclusion

Recommender System (RS) has become a revolutionary concept facing the big data era. Recommendations in the form of advice for users, is very useful to support achieving student using online learning environment. Many of the techniques provided by the recommendation system used collaborative filtering and content-based filtering. For online learning it is possible to use both techniques and also to use other techniques such as hybrid, knowledge-based and so on. This study results showed that research RS still gained research interest for various research communities. Interestingly, the SLR results pointed out some research opportunities in RS for online learning domain including.

1) Collaborative filtering methods: cold start, recommendation for item in long tail, and impact of context-awareness.

2) Demographic RS: impact of context-awareness.

3) Hybrid RS: Scalability and accuracy of recommendations.

4) Domain based RS: cold start, recommendation for item in long tail, and sparsity.

5) Knowledge based RS: Sparsity, changing dataset, and impact of context awareness.

The success of system recommendations in other fields such as e-commerce, cinema, and others, we believe will also succeed in online learning.

7.2 Future Scope

LMS, Learning Management System is a powerful tool that has become a necessity for every institute in the current scheme of things. Before the internet wave, there was the use of books and libraries. Teachers were the source of information and what they say was the only way of verifying the knowledge people gained a couple of decades back and recommendation system will help to solve this problem.

With the disruption of technology and its various avatars like the smartphone, social media, personal computers, MOOC, etc. students are not restricted to one kind of information but plenty.

This is also important especially because students have to cope with the rapidly growing and changing scientific and technological advancements in all fields.

So, what kind of solution must an institute possess to make things easier as well as highly usable with so much relevance to the present as well as futuristic goals in mind?

This blog will enhance your perception of Learning Management Systems, what is traditional learning management, its history, and what it can be to make it future-ready.

Everyone knows LMS is a Learning Management System which by definition is, “a software application for the administration, documentation, tracking, reporting, and delivery of educational courses or training programs.” Traditionally, LMS has been used only after the introduction of e-learning and distance learning programs.

REFERENCES

[1] O. R. Za, “Building a Recommender Agent for e-Learning Systems,” 2002.

[2] Zhao, Zhi-Dan, and Ming-Sheng Shang. "User-based collaborative-filtering recommendation algorithms on hadoop." Knowledge Discovery and Data Mining, 2010. WKDD'10. Third International Conference on. IEEE, 2010

[3] Dara, S. (2019). A survey on group recommender systems. Journal of Intelligent Information Systems

[4] Ahuja, R. (2019). Movie Recommender System Using K-Means Clustering AND K-Nearest Neighbor. IEEE.

[5] Y. Pan, D. Wu, C. Luo, and A. Dolgui, "User activity measurement in rating-based online-to-offline (O2O) service recommendation," Information Sciences, vol. 479, pp. 180-196, 2019.

[6] Y. Lee, C. Wei, P. Hu, T. Cheng, and C. Lan, "Small Clues Tell: a Collaborative Expansion Approach for Effective Content-Based Recommendations," Journal of Organizational Computing and Electronic Commerce, pp. 1-18, 2020

[7] Esparza, Sandra Garcia, Michael P. O’Mahony, and Barry Smyth. "Mining the real-time web: a novel approach to product recommendation." Knowledge Based Systems 29 (2012): 3-11.

[8] Chen, Li, and Feng Wang. "Preference-based clustering reviews for augmenting e-commerce recommendation." Knowledge-Based Systems 50 (2013): 44-59.

[9] Luo, Zhiling, Ying Li, and Jianwei Yin. "Location: a feature for service selection in the era of big data." Web Services (ICWS), 2013 IEEE 20th International Conference on. IEEE, 2013.

[10] Liu, Hongyan, et al. "Combining user preferences and user opinions for accurate recommendation." Electronic Commerce Research and Applications 12.1 (2013): 14-23.

[11] C. Sipio, D. Ruscio, and P. Nguyen, "Democratizing the Development of Recommender Systems by means of Low-code Platforms."

[12] S. B. Shirude and S. R. Kolhe, "Improved Hybrid Approach of Filtering Using

Classified Library Resources in Recommender System," in Intelligent Computing Paradigm: Recent Trends: Springer, 2020, pp. 1- 10.

[13] J. Das, S. Majumder, P. Gupta, and K. Mali, "Collaborative Recommendations using Hierarchical Clustering based on Kd Trees and Quadtrees," International Journal of Uncertainty, Fuzziness and Knowledge-Based Systems, vol. 27, no. 04, pp. 637-668, 2019.

[14] S. Parvatikar and B. Joshi, "Online book recommendation system by using collaborative filtering and association mining," 2015 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC), Madurai, 2015, pp. 1-4, doi: 10.1109/ICCIC.2015.7435717.

[15] S. Kanetkar, A. Nayak, S. Swamy, and G. Bhatia, "Web-based personalized hybrid book recommendation system," in 2014 International Conference on Advances in Engineering & Technology Research (ICAETR-2014), 2014: IEEE, pp. 1-5.

[16] A. Gholami, Y. Forghani, and M. Branch, "Improving Multi-class CoClustering-Based Collaborative Recommendation Using Item Tags Improving Multi-class Co-Clustering-Based Collaborative Recommendation Using Item Tags."

[17] Gao, Y. (2019). Research on Book Personalized Recommendation Method Based on Collaborative Filtering Algorithm. IEEE.

[18] Shah, K. (2019). Book Recommendation System using Item based Collaborative Filtering. International Research Journal of Engineering and Technology.

[19] Sohail, S. S. (2013). Book Recommendation System Using Opinion Mining Technique. IEEE.

[20] Tan, K. W. (2019). A New Collaborative Filtering Recommendation Approach Based on Naive Bayesian Method. Beijing: ResearchGate.

[21] S. Hossain, D. Sharma, and R. J. J. M. L. Chakma, "Machine LearningBased Phishing Attack Detection," International Journal of Advanced Computer Science and Applications (IJACSA), vol. 11, no. 9, 2020.

[22] F. Ahmed, et al., "A Combined Belief Rule based Expert System to Predict Coronary Artery Disease," in 2020 International Conference on Inventive

Computation Technologies (ICICT), 2020, pp. 252-257.

[23] W. Usino et al., "Document similarity detection using k-means and cosine distance," International Journal of Advanced Computer Science and Applications (IJACSA), vol. 10, no. 2, pp. 165-170, 2019

[24] Hofmann, Collaborative filtering via Gaussian probabilistic latent semantic analysis. In: SIGIR 03: Proc. Of the 26th Annual Int. ACM SIGIR Conf. On Research and Development in Information Retrieval, pp. 259266. ACM, New York, NY, USA (2003).

[25] Recommender Systems in E-Commerce J. Ben Schafer, Joseph Konstan, John Riedel Group Lens Research Project Department of Computer Science and Engineering University of Minnesota Minneapolis, MN 55455 1-612-625-4002