**E-BOOKS RECOMMENDATION SYSTEM USING REVIEWER RATINGS AND FEEDBACK**

**PROJECT REPORT**

***Submitted by***

# K.SASI KIRAN

**(2019202049)**

***submitted to the faculty of***

**INFORMATION AND COMMUNICATION ENGINEERING**

***In partial fulfillment of the requirement for the degree***

***of***

**MASTER OF COMPUTER APPLICATIONS**



**DEPARTMENT OF INFORMATION SCIENCE AND TECHNOLOGY COLLEGE OF ENGINEERING, GUINDY**

**ANNA UNIVERSITY**

**CHENNAI 600025**

**May 2022**

# ANNA UNIVERSITY, CHENNAI

## BONAFIDE CERTIFICATE

Certified that this Report titled “**EBooks recommendation system using reviewer ratings and feedback**” is the bonafide work of **K.Sasi Kiran (2019202049)** who carried out the work under my supervision. Certified further that to the best of my knowledge the work reported herein does not form part of any other thesis or dissertation on the basis of which a degree or award was conferred on an earlier occasion on this or any other candidate.

**PLACE**:CHENNAI KANIMOZHI.S

**DATE**: 01/06/2022 TEACHING FELLOW

PROJECT GUIDE

DEPARTMENT OF IST,CEG

ANNA UNIVERSITY

CHENNAI 600025

**COUNTERSIGNED**

**Dr. S. SRIDHAR**

**HEAD OF THE DEPARTMENT**

**DEPARTMENT OF INFORMATION SCIENCE AND TECHNOLOGY**

**COLLEGE OF ENGINEERING, GUINDY**

**ANNA UNIVERSITY**

**CHENNAI 600025**

**ACKNOWLEDGEMENT**

Its my privilege to express my sincere thanks to my project guide **Ms.S.KANIMOZHI**, TEACHING FELLOW, Department of Information Science and Technology, College of Engineering, Guindy, Anna University, Chennai for her keen interest, inspiring guidance, constant encouragement and support with my work during all the stages, to bring this thesis into fruition.

I deeply express my sincere thanks to **Dr. S. SRIDHAR**, Professor and Head of the Department, Department of Information Science and Technology, College of Engineering, Guindy, Anna University, Chennai for extending support.

I would like to express my sincere thanks to the project committee members, PROFESSOR **Dr.SASWATI MUKHERJEE**, **Dr.M.VIJAYALAKSHMI**, ASSOCIATE PROFESSOR, **Dr.E.UMA**, ASSISTANT PROFESSOR, **MS.P.S.APIRAJITHA**, TEACHING FELLOW, **MS.C.M.SOWMYA**, TEACHING FELLOW, Department of Information Science and Technology, Anna University, Chennai for giving their valuable suggestions, encouragement and constant motivation throughout the duration of my project.

K.SASI KIRAN

## TABLE OF CONTENTS

|  |  |  |
| --- | --- | --- |
| **CHAPTER NO.**  1 | **TITLE**  **ABSTRACT**  **INTRODUCTION** | **PAGE NO.**  7  8 |
|  |  |  |
|  | 1.1 Problem Statement | 9 |
|  | 1.2 Objective  1.3 Proposed System  1.4 Organization of thesis | 9  10  10 |
| 2 | **LITERATURE REVIEW** |  |
|  | 2.1 Data Preprocessing | 11 |
|  | 2.2 Sentiment Analysis  2.3 Search by words  2.4 Recommendation System | 11  13  13 |
|  |
| 3 | **SYSTEM DESIGN**  3.1 Architecture design | 14 |
|  | 3.2 Architecture Explanation | 15 |
| 4  5 | **DETAILED DESIGN**  4.1 Data Preprocess Module  4.2 Sentiment Analysis Module  4.3 Search by keyword  4.4 Search by set of words  4.5 Description of recommendation  system  **IMPLEMENTATION**  5.1 Overview of the system  5.2 Sentiment analysis  5.2.1 Dataset  5.2.2 Analysis to calculate sentiment  5.3 Search by keyword | 18  18  20  21  23  24  27      27  27  28  30 |
| 6  7  8 | **RESULTS**  **CONCLUSION**  **REFERENCES** | 33  35  36 |

# LIST OF FIGURES

[3.1 E-Book Recommendation System Architecture](#_bookmark17) 14

4.1 Data Preprocess Module 18

4.2 Sentiment Analysis Module 20

4.3 Search by keyword 21

4.4 Search by set of words 23

4.5 Item based collaborative filtering 26

5.1 Homepage of the system 27

6.1 Input screen 34

6.2 Output screen after recommendation 34

6.3 Input screen 35

6.4 Output screen after recommendation 35

# LIST OF ABBREVIATIONS

SO Subjectivity – Objectivity Polarity

PN Positivity – Negativity Polarity

# ABSTRACT

The e-books application is built for recommending different books, based on user interest. The dataset is collected from kaggle and it is preprocessed so as all the null values and duplicated values as well as any unwanted fields are removed. The books are separated into categories such as science fiction, fantasy, childrens literature, etc. The dataset is studied using python jupyter and using manual analysis for recommendation, few categories of books are selected. Genre count is identified and it is needed for the analysis. The books are recommended majorly in two ways, by taking the ratings which range from 0 to 10 into consideration and by taking the feedback in the form of text into consideration. The user gives direct books name as the input, then the user will be getting the recommendations based on the ratings of the recommended books (high to low). The user gives set of words as the input, those words are extracted and a keyword is found which matches with the genre. Then the keyword is present in the genre, then another set of books are recommended and they are shown based on the sentiment analysis. This type of recommendation is to be proposed in this project to bring out the user’s feedback into consideration in the time of search for the book itself.

***Keywords:*** Preprocessing data, reviewer ratings, feedback (positive/negative), Sentiment analysis, Recommendation system

# CHAPTER 1

# INTRODUCTION

* Recommender systems aim to predict users' interests and recommend product items that quite likely are interesting for them.
* They are among the most powerful machine learning systems that online retailers implement in order to drive sales.
* The impact is very huge which helps in marketing the products worldwide.
* Recommender systems are beneficial to both service providers and users.

The dataset is pre-processed and stored, also the sentiment analysis is made for getting the feedback of the user into consideration. Sentiment analysis is a technique in natural language processing that aims to retrieve the "sentiment" of a piece of text—positive, negative, or neutral. This is an easy way of summarizing the contents of a piece of text, and one that is easily understood.

Note, however, that sentiment analysis is a difficult problem. Humans agree on the sentiment of sentences only 80% of the time, and the best classifiers can get around that level of accuracy, but we're going to just use a built-in analyser in the NLTK (Natural Language Toolkit) python library.

***Technologies****:*

Programming Language : Java, Python

Front-end Technologies: HTML, CSS and JS

Algorithms: Sentiment Analysis

DataBase : MySQL

IDE : NetBeans

Framework: Python jupyter notebook

**1.1 Problem Statement**

The shortcomings of content based and collaborative based recommendation system are degrading the online recommendation systems. The user satisfaction is very mere when these algorithms are being used. More number of products are not been recommended, also they’re not efficient and faster. Sentiment analysis also will be of great help in order to categorize the review as positive, neutral or negative review and get the probability values for each columns and make analysis based on those values. In sentiment analysis, there are positive, negative and neutral values which are calculated with the help of python packages alone, but the use of it makes the values to be more mere and not accurate.

* 1. **Objective**

The objective of recommender systems is to provide recommendations based on recorded information on the previous users’ reviews and ratings. This system is proposed for two cases. The motive to take the two cases is to differentiate the way of getting the recommended books (for ratings and for feedback). The case 1 has user input as the direct book name itself. In this case the recommender system will process and brings out the recommending books with ratings as the categorization. The case 2 has user input as set of words (sentence), here the keyword is extracted and matched, if the keyword is found in the genre array, then the recommender system will recommend books based on genre and feedback as the categorization. This is the overall objective of this system.

* 1. **Proposed System**

The recommendation system which were proposed until now will have only the user ratings based systems. The ratings will be either star or they may be numerical values. These provides only the basic understanding of the user’s behavior across the datasets. Also if provided with the single keyword, then the recommendation system will give all the information according to the keyword. But the system which is proposed here is mainly concentrating on the fact that the keywords which are entered will be present in the summary of the book’s. So only those relevant books after going to the recommendation system algorithm will be provided. The item based collaborative filtering will be done in this single keyword search, which is not proposed to any system until now.

**1.4 Organization of Thesis**

The project report is organized as follows.

Chapter 2 discusses the existing systems and various methods required for the proposed system.

Chapter 3 discusses the overall system architecture along with the modules description and its flow diagrams.

Chapter 4 discusses the algorithms and the logic for the proposed system.

Chapter 5 discusses the implementation and the results of the system.

**CHAPTER 2**

**LITERATURE REVIEW**

This Chapter explains about the literature survey made on the existing system, analyzing the problem statements and issues with the existing system and proposed objectives for the new system.

**2.1 Data Preprocessing**

Namrata Chaudhary et all proposed the general Sources of Errors and Data Cleaning Strategies in which python jupyter is used in order to perform data cleaning using pandas library. Data should be preprocessed before storing them onto a database to reduce complexity and noise of raw data. After the dataset collection, it is cleaned and processed for analysis. Also, it is processed in such a way that the dataset contains only those rows which are needed and eliminating the null values etc.

**2.2 Sentiment Analysis**

Srishti Vashishtha et all, explains in her paper about the sentiment analysis where, keywords are words that come to you naturally, or that may be part of a specific discipline vocabulary (e.g. terms used only by midwives), or that you brainstorm when planning your search. Sentiment or Opinion Mining aims to determine the polarity of people’s opinions, feeling towards any product, service, event or any individual. One of the most popular technique applied in sentiment analysis of textual content is natural language processing. Sentiment can be evaluated using numerous methodologies like machine learning algorithms and statistical tools.

**2.3 Search by words**

Alok Kumar Gupta et all, in the journal proposed Abstract Keyword – based search is the most popular paradigm opted by traditional search engines where a user can specify a string of keywords and expect to retrieve relevant documents, possibly ranked by their relevance to the query. While searching within relational database systems, user needs to learn SQL and to know the schema of the underlying data to pose simple searches. This is a substantial barrier for casual users, such as users of Web-based information systems. Therefore, there is a need of a system that can eliminate this requirement is needed. To accomplish this task, this paper proposes a novel technique that is based on conventional keyword-based search and this technique is applied over relational databases. User has to give his/her query in the form of keywords like Google to get the desired results from database. The implementation of this query processing scheme gives efficient result.

**2.4 Recommendation System**

Rui Duan et all, proposed a recommendation algorithm to improving the recommendation model and introducing side information are two main research approaches to address the problem. Combine these two approaches and propose the Review-Based [Matrix Factorization](https://www.sciencedirect.com/topics/computer-science/matrix-factorization) method in this paper. The method consists of two phases. The first phase is review-based collaborative filtering, where an item-topic rating matrix is constructed by the feature-level [opinion mining](https://www.sciencedirect.com/topics/computer-science/sentiment-analysis) of online review text. This rating matrix is used to derive item similarities, which can be used to infer unknown users' ratings of the items. Only the first phase of this paper is used the second phase is not required as per our project requirement.

**CHAPTER 3**

# 3.1 EBook Recommendation System Architecture



**Figure 3.1 : E-Book Recommendation system Architecture diagram**

# 3.1.2 DATASET:

# The dataset for books should be collected from kaggle, it contains books.csv, users.csv and ratings.csv files. The size of dataset is large enough to make our implementations. Each csv file consists of more than 2 lakh records which consists of all the information regarding the books, users and their ratings.

# 3.1.3 PROCESSED DATA:

# After the dataset collection, it is cleaned and processed for the analysis. Also it is processed in such a way that the dataset contains only those rows which are needed and eliminating the null values etc. After the analysis and filtering out the unwanted data, we are left with only forty thousand rows of information. In Fig1, it clearly explains that the combined with the users and ratings regarding to the book “isbn” and the “user-id”.

# 3.1.4 SENTIMENT ANALYSIS:

# The purpose of sentiment analysis is based on the two sectors:

# Classifying Documents

# Classifying documents or any passages according to sentiment orientation such as positive vs. negative.

# Gathering Information

# Extracting information of opinions which contains information of particular aspects of interest and the corresponding sentiment orientation in a structured form from a set of unstructured data. The tasks of classifying documents of the sentiment analysis can be divided into three sub-tasks:

# Identifying SO polarity:

# Whether the comment or post is referring a situation or event without disclosing the subjectivity (positive or negative opinion) on it or expressing opinion on its subject matter. Briefly, it means that identify the subjective or objective polarity of a post or comment.

# Identifying PN-polarity:

# Whether a subjective post or comment is expressing positive or negative.

# 

# Identifying the Degree of PN-polarity:

# This step gives the impression of the degree of positivity or negativity on that opinion. Positivity could be weakly positive, mildly positive or strongly positive and same could be for the negative opinion.

# "I think this is going to be one of the most important datasets of this era, because we are looking at what people are talking about in real time at the scale of an entire society," says Mislove, an assistant professor of computer science at Northeastern University. So, sentiment analysis is doing the right thing as he told

# FLOW OF WORK EXPLANATION:

The User will be able to search the books by two ways:

1. *Search the books direct by giving the book name* : In this case, the user will be allowed to enter any keyword of his own, it is then checked if present in the database. If the book is present in the database, then that particular keyword is parsed into the recommendation system, where the name of the book is taken and related books are recommended for the user.
2. *Search the book indirect by giving set of words*: In this case, the user will be entering a set of words or like a sentence, that sentence is parsed into the keyword extraction algorithm and the keyword is extracted which is present in the genre array. The genre is considered as array with unique values. This keyword, after extraction, if matches in the array, it then gets that particular genre and passes into the recommendation system. As mentioned earlier, the book genre is taken and related genre books are recommended for the user.

In Fig1, for case 1, the user will get books displayed in hierarchical order of ratings (high to low). And also in Fig1, for case 2, the user will be getting the books displayed in the order of positive, neutral and negative feedback which is parsed and taken in the sentiment analysis.

**CHAPTER 4**

# DETAILED DESIGN

* Dataset Preprocessing.
* Sentiment Analysis.
* Search by name.
* Search by set of words.
* Recommendation System.

# LIST OF MODULES

* 1. **Data Pre processing :**

***Input*** *:* All three (Books, Users, Ratings) CSV files.

Process the data, manipulate it, clean it using Pandas, Numpy in jupyter platform.

***Output*** *:* Cleaned and processed Data which is loaded into another CSV file

**DESCRIPTION OF DATA PREPROCESS MODULE:**

1. **Data Cleaning**

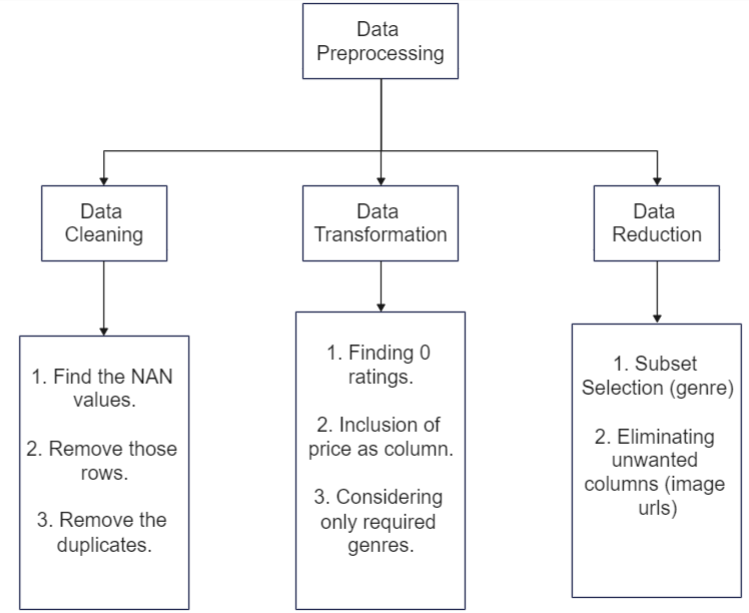
The CSV files will have more null values, so they must be eliminated. Remove the duplicates (rows and columns), this is what explained in Fig2.

1. **Data Transformation**

Find the zero ratings data and include a separate column as price. Genre column has multi label genres, so they must be eliminated which comes under data reduction part in Figure 4.1.

1. **Data Reduction**

For any data study to make, the unwanted rows/columns must be deleted. Also the subset values must be eliminated.



**Figure 4.1 : Data pre process module**

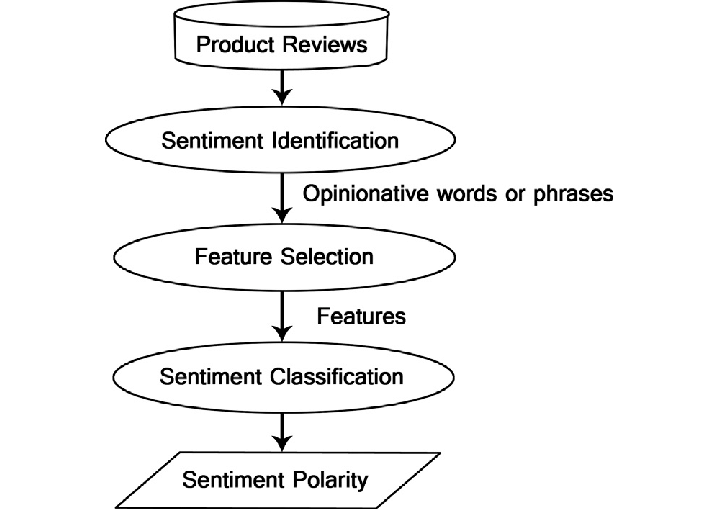
* 1. **Sentiment Analysis :**

***Input :*** The preprocessed csv file.

Import nltk sentiment vader library and loop the column for which the polarity needs to be given to. The two functions, one for stop words to detect the language and the other function for splitting up of the words. The polarity is found for positivity, neutral and negativity for the feedback column and combined with the csv file with respect to the User-ID.

***Output :***  The polarities for each column is found and they’re included in the dataset, i.e. compound, positive, neutral and negative columns.

**DESCRIPTION OF SENTIMENT ANALYSIS MODULE :**



**Figure 4.2 : Sentiment Analysis module**

Sentiment Analysis is the computational study of people’s opinions, attitudes and emotions toward an entity. In Figure 4.2, Sentiment Analysis identifies the sentiment expressed in a text then analyzes it. Therefore, the target of SA is to find opinions, identify the sentiments they express, and then classify their polarity. The data sets used in SA are an important issue in this field. The main sources of data are from the product reviews. These reviews are important to the business holders as they can take business decisions according to the analysis results of users’ opinions about their products. The reviews sources are mainly review sites. SA is not only applied on product reviews but can also be applied on stock markets, news articles,  political debates. They are also used as data sources in the SA process.

In the preprocessed data, we need to identify the reviews of each book, whether they’re positive or neutral or negative. This is done using the sentiment analysis, it makes our task easy by just applying the algorithm to the dataset and getting the required results.

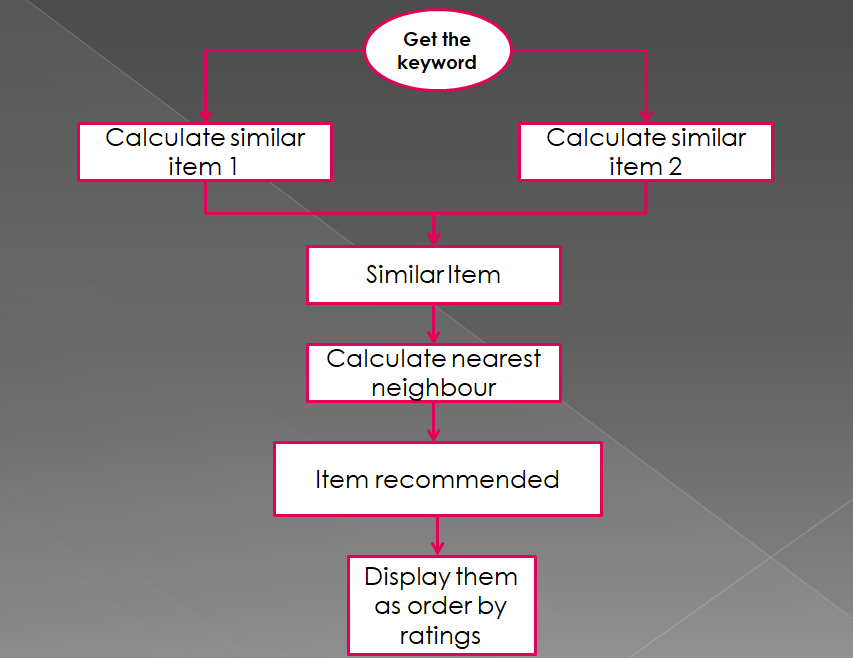
* 1. **Search by keyword :**

***Input :*** The user gives any keyword as input

The keyword is parsed to the recommendation system and similar type of books are recommended based on ratings.

***Output* :** The recommended books are displayed in the order of ratings**.**

**DESCRIPTION OF SEARCH BY KEYWORD MODULE :**



**Figure 4.3 : Search by keyword module**

In this module, the user enters a keyword for example : good, true, story, best etc. These keywords are made sure they are present in the features/description of the books. They’re matched and parsed into the recommendation system. The similar kind of books are displayed based on the order of ratings. In Figure 4.3, it is clearly explained that the input which comes from the user will directly go to the recommendation system and then similar items are searched and recommended by using the calculation of the nearest neighbour. Then the items are taken into recommendation and then they are displayed based on the ratings order by.

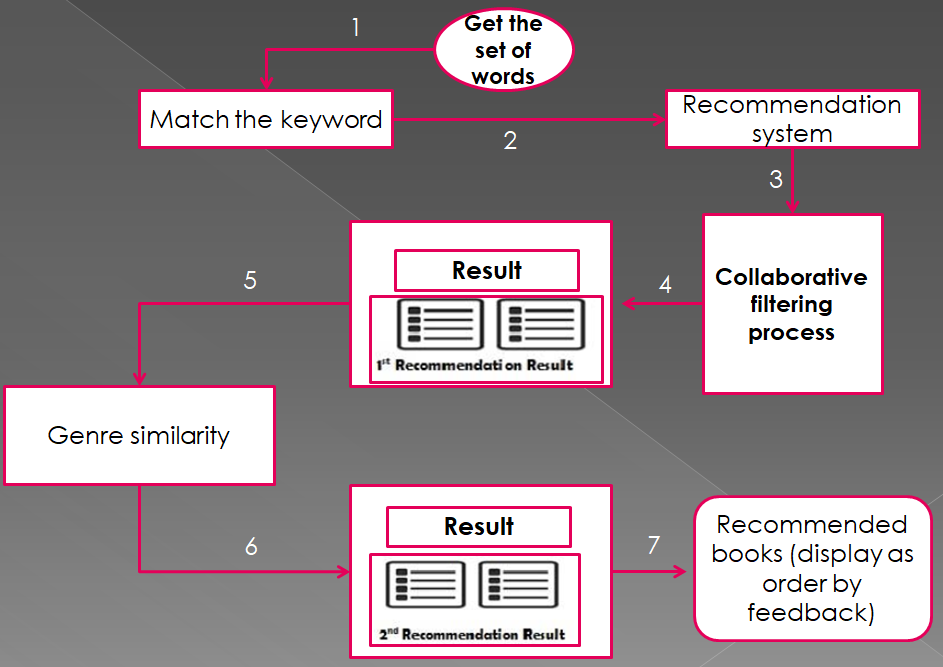
* 1. **Search by set of words :**

***Input :*** The user gives set of words as input.

The set of words is parsed, relevant word is taken and if matches with the genre, it is then parsed to the recommendation system.

***Output* :** The recommended books are displayed in the order of positivity rate of the feedback.

**DESCRIPTION OF SEARCH BY SET OF WORDS MODULE:**



**Figure 4.4 : Search by set of words module**

In this module, at first the user will enter a sentence for example : the best selling new romance books. Then the sentence is parsed and when a word in the sentence matches with the genre, it then goes for recommendation. The recommendation is then given based on the feedbacks i.e. the highest positive rate books are displayed first. As specified in the Figure 4.4, the search starts from the set of words after the words are being extracted and the keyword is found which matches with the genre. The genre of similar books are then passed into the collaborative recommendation system process which will give us the output of recommendation phases wise. In the first phase the system will give you few set of similar books, in the second phase the final set of books recommended will be displayed. The algorithm will specify only the final set of books to recommend. The those are displayed ordered by the feedback which are taken from the sentiment analysis done before.

* 1. **Recommendation system :**

***Input :*** Keywords entered by the user.

The item based filtering suggests future products by calculating item to item similarity score. The cosine similarity formula is similar to the user based score.

***Output* :** The recommended books are displayed to the user and they are categorised based on either ratings or feedback depending upon the basis of input given by the user.

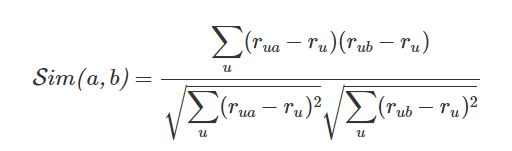
**DESCRIPTION OF RECOMMENDATION SYSTEM**

**Finding the cosine similarity score :**

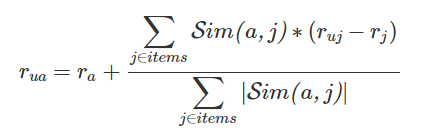
There are a few scores to measure user to user or item to item similarity. Pearson similarity and cosine similarity are the two most commonly used metrics for collaborative filtering algorithm. It is the simplest algorithm to find the similarity of two vectors. The user score vector (row) or the item score vector (column) can be extracted from the user-item interaction matrix shown earlier. Generally a very sparse matrix and a score of 0 will be assigned when there isn’t a rating in the matrix. The cosine similarity score of -1 implies that the users or the items are polar opposites while a score of 1 shows that the users or the items are twins. A score of 0 indicates that the users or the items are completely unrelated.

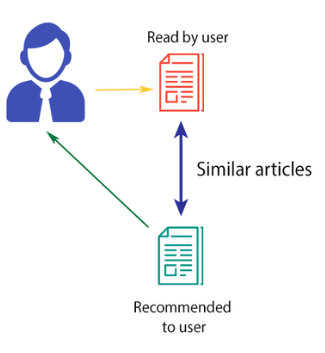
**Item based collaborative filtering :**

The item based filtering suggests future products by calculating item to item similarity score. The cosine similarity formula is similar to the user based score:



u in the above formula stands for all of the common users for item a and b. Again to remove the underlying user bias, the average rating for each common user is subtracted from the item score. The equation for the item based prediction score is also very similar except that the average item score is used to estimate the predicted rating. This can cause the rating to go slightly above the max score of 10 but this approach allows us to remove any underlying movie bias.

******



**Figure 4.5 : Item based collaborative filtering**

In the Figure 4.5, the explanation of this algorithm is clearly mentioned. The cosine similarity is taken, and then the user will be able to get the recommended books after they enter a keyword or a book name (here) in this system. The similar type of books are then recommended by getting the similar items in the dataset and then parsing them for getting the cosine similarity and then into the item based collaborative filtering algorithm.

**CHAPTER 5**

**IMPLEMENTATION**

**5.1 Overview of the system *:***

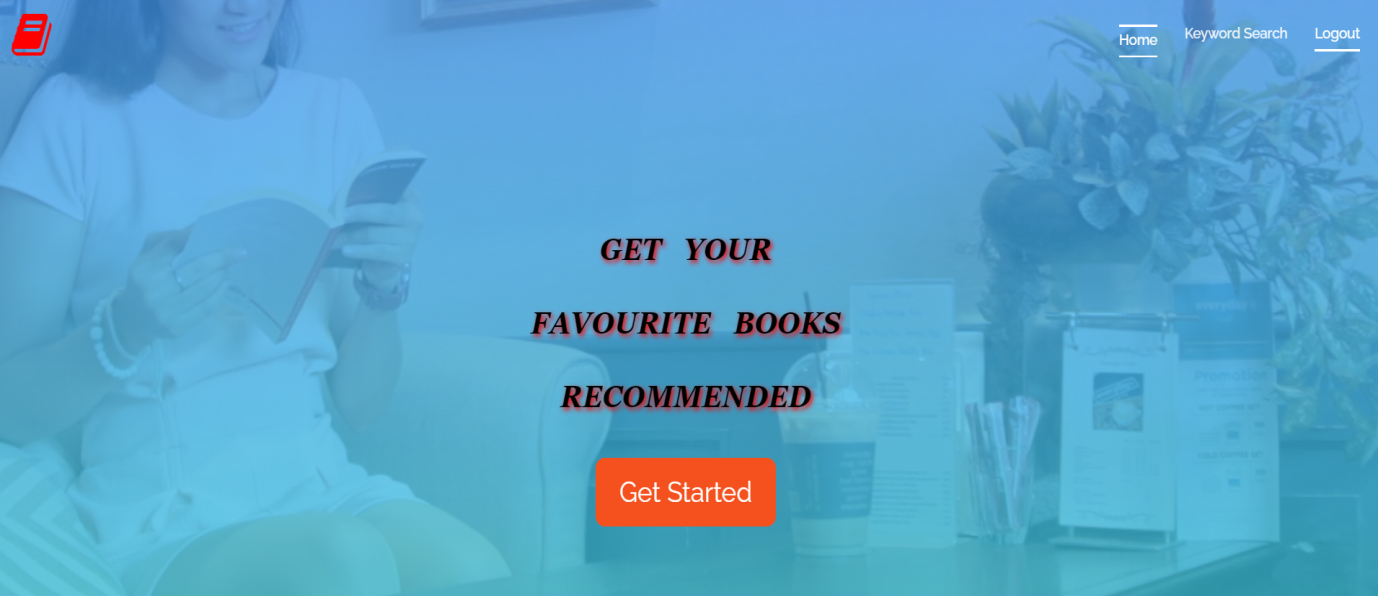
******

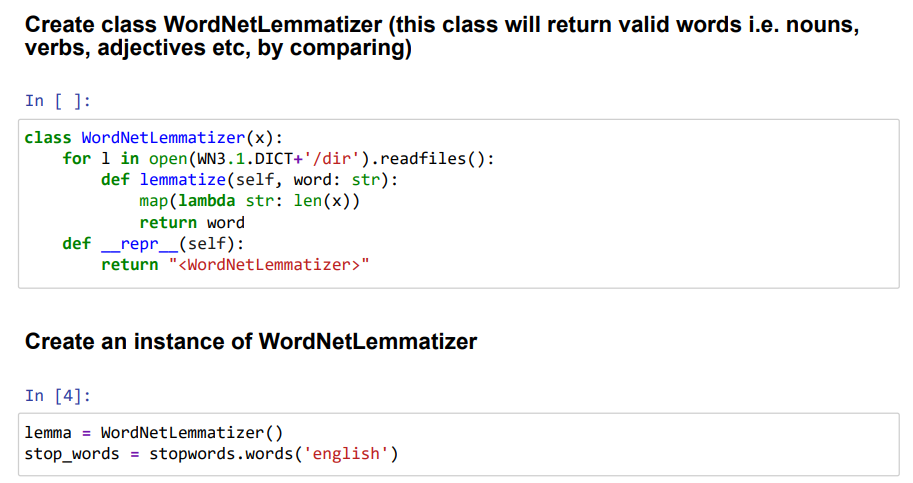
Figure 5.1 : Homepage of the system

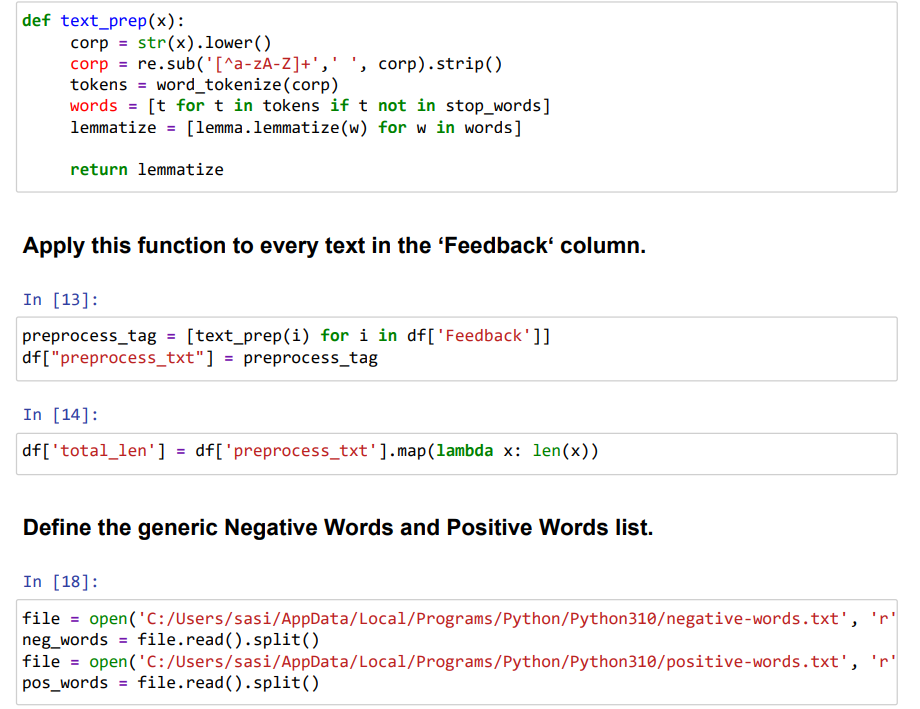
**5.2 Sentiment analysis**

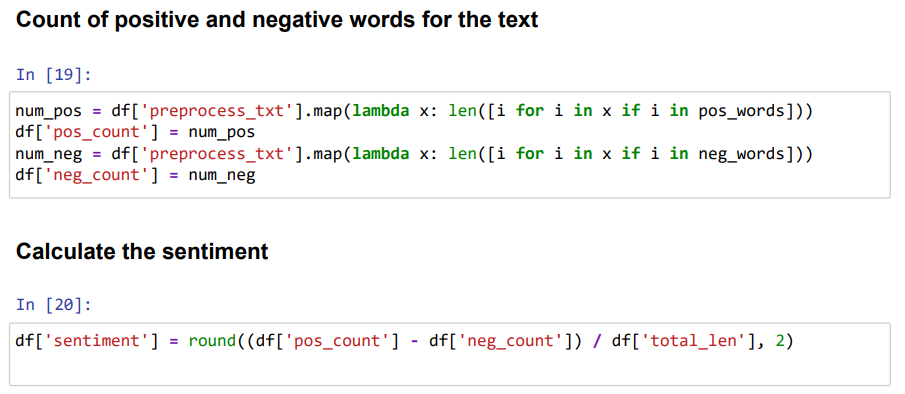
***5.2.1 Dataset***

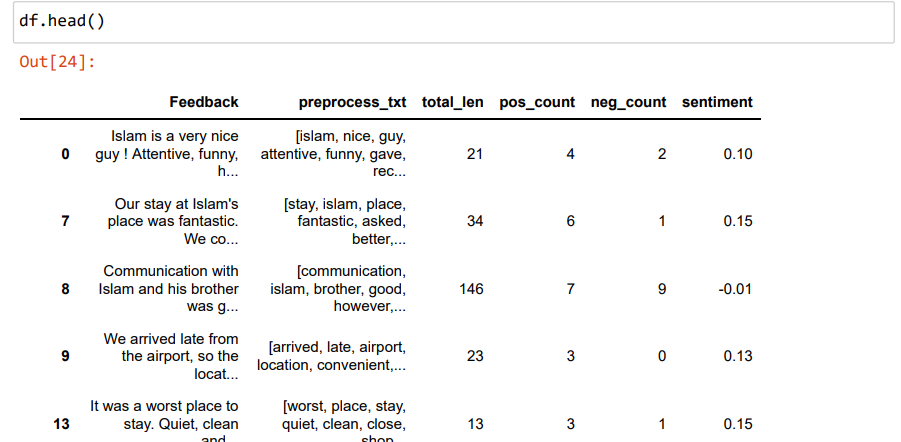
****

***5.2.2 Analysis to calculate sentiment:***









**5.3 SEARCH BY KEYWORD**

**Pseudo Code :**

BEGIN

User Input ( keyword )

Count Number of words ( Keyword )

IF count==1 THEN

Case 1 ( Recommend books by keyword )

Call Recommendation.java

Recommendation(keyword)

Algorithm for Item based collaborative filtering

**Step 1: Users input a collection name**

**Step 2**: Calculate the similarities

**Step 3**: The similar items are passed on as arguments in

the recommendation function which considers the items

liked and viewed and predicts the recommendation using

the inputs from Step 2.

**Step 4**: A weighted average of all these

recommendations is calculated and are displayed to the

user.

**Step 5**: The final recommendation is displayed to the

user based on their weighted average.

The recommendations are based on items preferred by

the user and similar items and all related items. Below is a

pseudo code for the same.

For each Item i liked

For every User U who liked Item i

For each Item J liked by User U

Record Item (i,j)

Compute the similarity between Item i and

Item j

Item to Item filtering is a technique where users

are not compared with other users but their rated

interest in items is calculated. In this approach,

similarities between i and j are computed offline

by a simple formula:



Where n-number of samples; x and y are the two inputs in

our

case the 2 ratings by different users, Now, the rating for

item

for user a can be predicted using a simple weighted average,

as in: Weighted Average = [(n1 x N1) + (n2 x N2)] / (n1 +

n2)

where n1,n2… are relative weights and N1,N2… are values

IF count>1 THEN

CASE 2 (Recommend books by extracting keyword)

Keyword extraction from the given sentence

Step 1 : Convert the given string to lowercase

Step 2 : Extract individual words

Step 3 : Check each word with the Genre column

Step 4 : IF present THEN matching = keyword break

Render SQL queries :

select \* from datas where Genres like '" + matchfound + "' order by positivity DESC

Step 5 : Render the recommended books

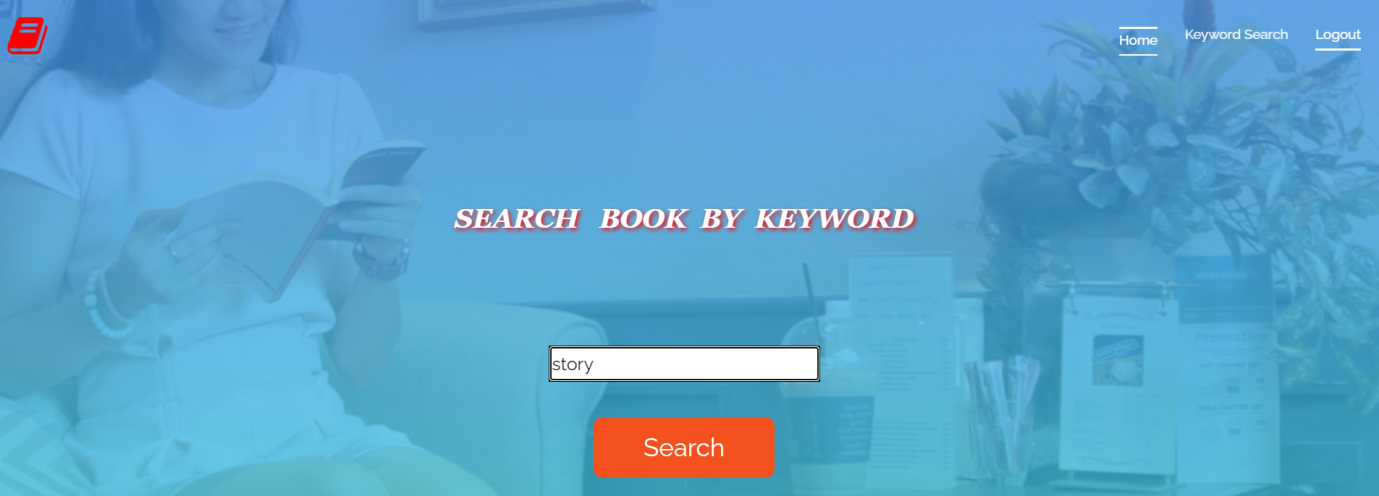
END

**CHAPTER 6**

**RESULTS**

**Search by keyword :**

**User Input**

****

**Figure 6.1 : Input screen**

**System Output**

****

**Figure 6.2 : Output screen after recommendation**

**Search by set of words :**

**User Input**

****

**Figure 6.3 : Input screen**

**System output**

****

**Figure 6.4 : Output screen after recommendation**

**CHAPTER 7**

**CONCLUSION**

**Conclusion :**

The dataset after cleaning and getting the information which is necessary for the algorithm and recommendation is stored separately into a csv file and it is then used to get the most recommended system. The ratings count are also used for analysis also. The genres count is also taken . The sentiment analysis is made and the data is stored separately in csv file and this data is loaded into the database and the variation in the keyword which is entered is processed. If the user enters only a single keyword, then the first case where, the keyword is parsed into the item based collaborative recommendation system and then similar set of books are recommended. If the user enters a set of words, the relevant keyword is matched with the genre and then the recommendation is based on the customer feedback which was given previously and it is displayed in such a way that the most positively given feedback book is given in the top and progressively neutral to the negative book.

**Future Work :**

This work can be enhanced in the future by adding the feedback neutrality in the output and recommending a new set of books based on the extending of sentiment analysis. The user interface can be modified and made up to the level of the existing ecommerce websites by using advanced css concepts. The further steps will be to make the word which the user enters to classify them as positive or negative or neutral word and display the books accordingly.

**CHAPTER 8**

# REFERENCES

* Alok Kumar Gupta , Sumit Kumar Mishra, 2018, A Robust Scheme for Keyword Based Search over Relational Databases, INTERNATIONAL JOURNAL OF ENGINEERING RESEARCH & TECHNOLOGY (IJERT) Volume 07, Issue 04 (April 2018),
* Vashishtha, S., & Susan, S. (2020, July). Fuzzy interpretation of word polarity scores for unsupervised sentiment analysis. In *2020 11th international conference on computing, communication and networking technologies (ICCCNT)* (pp. 1-6). IEEE.
* Chaudhary, N., & Roy Chowdhury, D. (2019). Data preprocessing for evaluation of recommendation models in E-commerce. *Data*, *4*(1), 23.
* Duan, R., Jiang, C., & Jain, H. K. (2022). Combining review-based collaborative filtering and matrix factorization: A solution to rating's sparsity problem. *Decision Support Systems*, *156*, 113748.
* <https://data-science-blog.com/en/blog/2018/11/04/sentiment-analysis-using-python/>
* <https://arxiv.org/ftp/arxiv/papers/1403/1403.3185.pdf>
* <https://link.springer.com/chapter/10.1007/978-3-540-72079-9_10>
* <https://ieeexplore.ieee.org/abstract/document/6965073?casa_token=smrJx4qFy1wAAAAA:H4oCy03htAEqXuLqlV3lumBMTYGQjBjC4OHi5MVq44jo1ozKGklwmk63lQ9AAvLtuNxfWZvggg>
* <https://libguides.library.cqu.edu.au/c.php?g=760913&p=5456502>
* <https://www.kaggle.com/datasets/arashnic/book-recommendation-dataset>
* <https://www.kaggle.com/code/sercanyesiloz/book-recommendation-system#Preprocessing>
* <https://journalofbigdata.springeropen.com/articles/10.1186/s40537-015-0015-2>
* <https://www.mdpi.com/2076-3417/11/9/3986>
* <https://www.researchgate.net/profile/Haque-Nawaz-Lashari-2/publication/332731286_Student_Feedback_Sentiment_Analysis_Model_Using_Various_Machine_Learning_Schemes_A_Review/links/5d961d2c299bf1c363f5759a/Student-Feedback-Sentiment-Analysis-Model-Using-Various-Machine-Learning-Schemes-A-Review.pdf>