International Journal of Scientific Research in Engineering and Management (IJSREM)
Volume: 05 Issue: 04 | April - 2021 ISSN: 2582-3930

Book analysis and recommendation system

Yash Brid[1], Prem Chawla[2], Kapil Bhavnani[3] , Vaishnavi Bidoo[4], Meena Talele[5]

[1]CO student, Dept. of Computer Engineering from VESP Mumbai, Maharashtra, India.
[2]CO student, Dept. of Computer Engineering from VESP Mumbai, Maharashtra, India.
[3]CO student, Dept. of Computer Engineering from VESP Mumbai, Maharashtra, India.
[4]CO student, Dept. of Computer Engineering from VESP Mumbai, Maharashtra, India.
[5]CO professor, Dept. of Computer Engineering from VESP Mumbai, Maharashtra, India.

Abstract - With each passing year, more number of books are published. We have always considered the magical persona, and the impact books seem to hold, and in this project we analyze what kind of books really interests people into reading. The sole purpose of this project is to provide an in-depth analysis on data related to books and recommend new books to the users based on their choices. For collaborative recommendation system we will be using KNN algorithm and for content-based recommendation, TF-IDF algorithm is used. So by developing this recommendation system, we group together the books that are similar in some aspect and provide new recommendations to the users.

The basic idea of this project is to analyze different attributes of a book and find the relation between these attributes and recommend books to the users. It analyzes how these attributes affect the book's rating or book's popularity, what features have more importance and so on. It also analyzes and visualize the sentiments of user reviews. In order to do so, an in-depth EDA process is performed on the data to get efficient results. For UI purpose, Django framework has been used to present the visuals in an interactive manner.

Key Words: collaborative recommendation, content-based recommendation, KNN algorithm, TF-IDF algorithm, EDA process.

1. INTRODUCTION

This project gives us a deep analysis on books dataset. The datasets has been read into pandas data frames and has been combined together, data cleaning has been done on the combined datasets.

After the cleaning process, EDA process has been performed to uncover the patterns, insights and anomalies within the data. Through the EDA process we get the answers to, which are the top most rated/popular books over the years, top most rated/popular authors, the rating and language distribution of the books, authors overall performance, etc. and also, how does each attribute affect the average rating of the books. After data visualization, feature engineering has been performed to predict the average rating of the books, and convert the categorical attributes into numerical attributes. After predicting the average ratings for the books, a recommendation system has been built. It provides three types of recommendation, first, the author based recommendation using TF-IDF, second, content based recommendation using multiple matrices, and third, collaborative recommendation, based on how similar are your preferences to the other users, and then recommend the books they have liked or rated. Lastly, sentiment analysis has been performed to interpret and classify the emotions related to the project topic.

2. LITERATURE REVIEW

Without a doubt we can state the fact that, reading is good for everyone, as it improves our vocabulary, helps us to learn new words, and helps us gain control over our language. And what makes this experience more fun and true pleasure is, when we find a book that is well designed and suits our choices and preferences best. According to Statista's worldwide e-book statistics, there were 938.5 million users till the year 2019, and is expected to reach 1,313.3 million by year 2025. The year 2019, had 53% male users and 47% female users, users from the age group 25-34 years had a greater share, and 40% of the users were from low income group, 32% from the medium income group and remaining 28% from the high income group, and most revenue is generated in the U.S.

A book recommendation is basically a system that predicts the future preferences of a list of books to the user and recommends top books. So, why do we need this recommendation system? It's because, in today's world, due to the growing and vast commonness of the Internet, people have way too many options to choose from. Whereas, in the past, people used to physically buy books from the book store, where the options were limited, because the number of books a store can place, depends on the size of the store.

In contrast to this, Internet nowadays provides plentiful resources online for people to access. For example, Goodreads is an American website, which has a vast collection of books, reviews, quotes, etc. But even though the size or the range of the available resources or information is increased, a new problem has emerged where people have a hard time choosing the books that they actually want to read. So, this is where the role of the recommendation system comes in, and it turns out to be helpful for the users, as it saves a lot of their time, that would otherwise have been spent on deciding which book to read next.

Recommender systems have become an important research field since the emergence of the first paper on collaborative filtering in the mid-1990s. Although academic research on recommender systems has increased significantly over the past 10 years, there are deficiencies in the comprehensive literature review and classification of that research [1].

Sentiment analysis refers to the automatic determination of subjectivity (whether a text is objective or subjective), polarity (positive or negative) and strength (strongly or weakly positive/negative). It is a growing field of research, especially given the gains to be obtained from mining opinions available online. Approaches to sentiment analysis have tackled the problem from two different angles: a word-based or semantic approach, or a machine learning (ML)

approach [2]. The word-based approach uses dictionaries of words tagged with their semantic orientation (SO), and calculates sentiment by aggregating the values of those present in a text or sentence [3]. The ML approach uses collections of texts that are known to express a favorable or unfavorable opinion as training data, and learns to recognize sentiment based on those examples [4].

3. SCOPE OF STUDY

Here, in this project, we have covered the parts like, Data Collection, Data Cleaning, Data Visualization and Recommendation System. All these processes are performed on a dataset, which is initially read and stored in the dataframe. We explore the book attributes, user attributes and also the user reviews.

4. TYPES OF RECOMMENDER SYSTEM

There are three very important types of recommender systems:

- Collaborative Recommendation.
- Content-Based Recommendation.

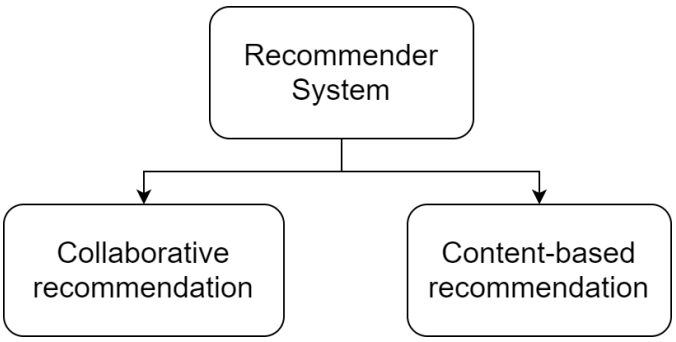

Fig 1. Types of recommender systems

4.1 COLLABORATIVE RECOMMENDATION

It is the most common technique used when it comes to building intelligent recommender systems that can learn to give better recommendations as more information about users is collected.

It is a technique that can filter out items that a user might like on the basis of reactions by similar users. It works by searching a large group of people and finding a smaller set of users. It then looks at the item they like and combines them to create a ranked list of suggestions. So to make Collaborative recommendation model we will use **KNN algorithm**.

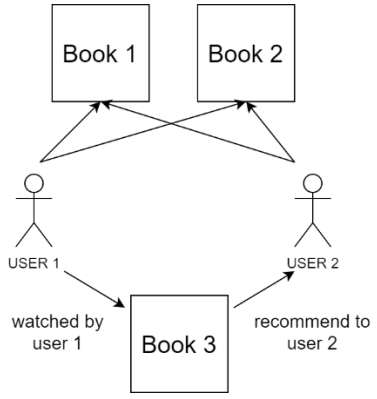

Fig 2. Collaborative Recommendation

4.2 CONTENT-BASED RECOMMENDATION

A Content-Based Recommender works by the data that we take from the user, either explicitly (rating) or implicitly (clicking on a link). By the data we create a user profile, which is then used to suggest to the user, as the user provides more input or take more actions on the recommendation, the engine becomes more accurate [2].

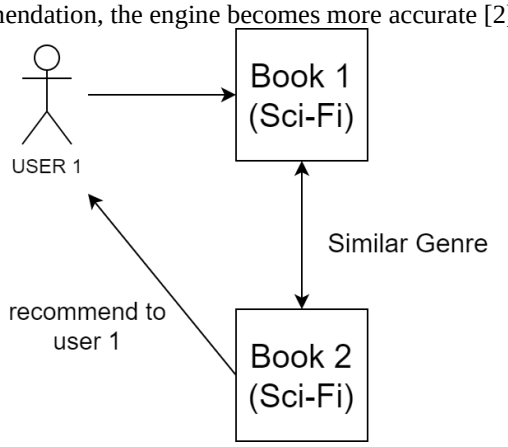

Fig 3. Content-Based Recommendation

This Recommendation System works on the principle of similar content. If a user is reading a book, the system will check about other books of similar content or genre that the user is reading. So, the idea is to tag products using keywords, understand what the users like, look up those keywords in the data and recommend different products with the same attributes. To check the similarity between the products, the system computes distance between them. For numeric data, **Euclidean distance** is used, for textual data **cosine similarity** or **TF-IDF** is used and for categorical data **Jaccard similarity** is used. So, in this project we have used **TF-IDF algorithm** to calculate the similarities between the books.

5. EXPERIMENTAL SETUP

For this project python programming language has been used, and in doing so, different frameworks and libraries provided by the python were also used and are listed below:

- **Jupyter Notebook**: Jupyter notebook is an open source web-application that supports interactive data science and scientific computing. It is used for Data Collection, Data Cleaning, Data visualization, Statistical models, Machine Learning and much more. It supports a variety of programming languages including python, R, Scala, etc.
- **Django**: Django is an open source python library, that allows to quickly build highly interactive and customizable machine learning and data science web applications around the data.
- **HTML Components**: For better UI purpose, to display HTML code or render a chart from a python visualization library.
- **Tailwind CSS**: Tailwind CSS is a utility-first CSS framework for rapidly building custom user interfaces and designs. It is a great way to write inline styling and achieve an awesome interface without writing a single line of your own CSS.
- **Pandas**: Pandas is a software library for python programming language, for data manipulation and analysis. It provides a fast and efficient way to

manage and explore data, handling missing data, cleaning up data, merging and joining datasets, etc. It provides essential data structures like series, dataframes etc.

- **NumPy**: NumPy is a python library, adding support for large, multi-dimensional arrays and matrices, along with large collection of high-level mathematical functions to operate on arrays.
- **Chart.js**: Chart.js is an interactive and open-source plotting library for JavaScript.
- **Scikit-Learn**: Scikit-Learn is a free software machine learning library for the python programming language, featuring various classification, regression and clustering algorithms including support vector machines, random forests, etc. and is designed to interoperate with python numerical and scientific libraries.
- **Tweepy**: Tweepy is an open source python package that allows to conveniently access the Twitter API with python, it includes a set of classes and methods that represent Twitter's models and API endpoints. It helps in extracting the tweets from Twitter
- **TextBlob**: TextBlob is a python library for pre-processing textual data and provides a simple API for diving into common NLP tasks such as, part-of-speech tagging, noun phrase extraction, sentiment analysis, classification and much more.

6. DATASET

This project gives us a deep analysis on books dataset obtained from https://www.kaggle.com/bahramjannesarr/goodreads-book-datasets-10m. These are multiple datasets containing information related to books and user rating such as book title, authors, language, average rating, publishing year, book description, etc. The datasets have been read into pandas data frames and have been combined together, data cleaning has been done on the combined datasets. In Fig 4. you can see the snippets of the Dataset used for analysis, recommendation.

book_id	books_count	isbn13	authors	original_publication_year	original_title	language_code	average_rating	ratings_count	ratings_1	ratings_2	ratin
2767052	272	9.780439e+12	Suzanne Collins	2008	The Hunger Games	eng	4.34	4780653	66715	127936	56
3	491	9.780440e+12	J.K. Rowling	1997	Harry Potter and the Philosopher's Stone	eng	4.44	4602479	75504	101676	45
2657	487	9.780061e+12	Harper Lee	1960	To Kill a Mockingbird	eng	4.25	3198671	60427	117415	44
4671	1356	9.780743e+12	F. Scott Fitzgerald	1925	The Great Gatsby	eng	3.89	2683664	86236	197621	60
11870085	226	9.780525e+12	John Green	2012	The Fault in Our Stars	eng	4.26	2346404	47994	92723	32

r	original_title	language_code	average_rating	ratings_count	ratings_1	ratings_2	ratings_3	ratings_4	ratings_5	image_url	Ratings_Dist
3	The Hunger Games	eng	4.34	4780653	66715	127936	560092	1481305	2706317	https://images.gr-assets.com/books/1447903603m...	Between 4-5
7	Harry Potter and the Philosopher's Stone	eng	4.44	4602479	75504	101676	455024	1156318	3011543	https://images.gr-assets.com/books/1474154022m...	Between 4-5
1	To Kill a Mockingbird	eng	4.25	3198671	60427	117415	446835	1001952	1714267	https://images.gr-assets.com/books/1361975680m...	Between 4-5
5	The Great Gatsby	eng	3.89	2683664	86236	197621	606158	936012	947718	https://images.gr-assets.com/books/1490628560m...	Between 3-4
2	The Fault in Our Stars	eng	4.26	2346404	47994	92723	327550	698471	1311871	https://images.gr-assets.com/books/1360206420m...	Between 4-5

Fig. 4: Dataset snippet

7. PROJECT PROCEDURE AND FLOW

- We start by using Jupyter Notebook, where we use Pandas and NumPy libraries for collecting the data, handling missing data, cleaning the data and for performing various manipulation techniques.
- After preparing the data, we use Plotly python library for data analysis and visualization. Where we represent the results in the form of various interactive charts and graphs, while making it easy to understand the trends and patterns in data.
- Now, we make use of the Scikit-Learn library for the recommendation system, by using the **KNN** algorithm for **collaborative recommendation** and **TF-IDF** algorithm for **content based recommendation** provided by the library.
- Next for **sentiment analysis**, we make use of Tweepy library to access Twitter API and extract tweets from Twitter and TextBlob library for preprocessing the extracted tweets. Also, make use of different methods functions provided by both the libraries for analyzing the sentiments of the tweets.
- After processing and analyzing the data in Jupyter Notebook, the next step is to integrate the code modules in the Django framework.
- For making UI more pleasing we make use of Tailwind CSS in HTML Components, without having to do any CSS coding.
- After everything is done, we integrate both the backend and frontend together and set our application running and working.

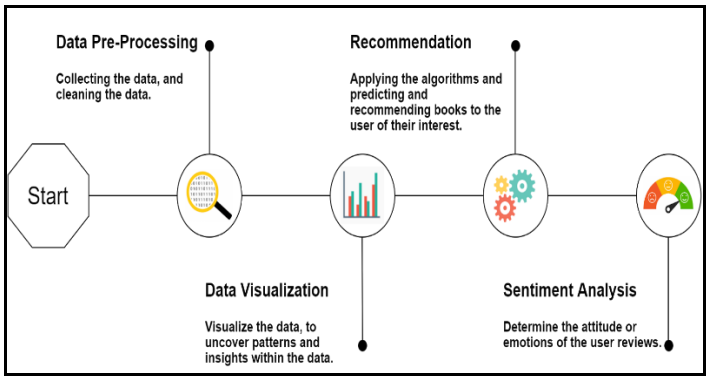

Fig. 5: Procedure and Flow

8. TERM FREQUENCY – INVERSE DOCUMENT FREQUENCY (TF-IDF)

It is a statistical measure that evaluates how relevant a word is to a document in a collection of documents.

It is calculated by multiplying two metrics : term frequency (TF) , that is how many times a word appears in a document and the inverse document frequency (IDF) of the word across a set of documents.

So, the Term Frequency (TF) of a word in a document can be calculated by calculating a row count of instances of a word appearing in a document. The Inverse Document Frequency of a word across a set of documents which means how rare or common a word is in the entire document set. It is calculated by taking total numbers of documents, dividing it by the number of documents that contain that word. The closer it is to 0, the more common the word is. So if the word is very common and appears in many documents, this number will approach 0 otherwise 1.

Now, multiplying these two metrics results in a TF-IDF score of a word in a document. The higher the score is , the more relevant that word is in that particular document.

In Fig. 6 you can see the flow of TF-IDF algorithm.

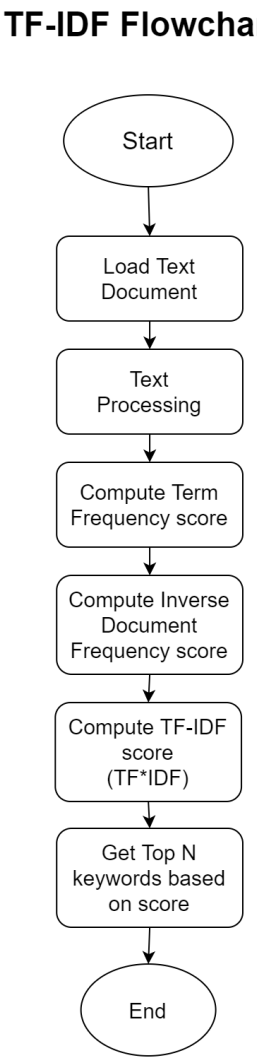

Fig. 6: TF-IDF Flowchart

9. K NEAREST NEIGHBOR (KNN)

The K-nearest neighbor is a simple, easy to implement supervised ML algorithm that can be used to solve both regression and classification problems. It assumes that similar things exist in close proximity. In other words similar things are near to each other.

It captures the idea of similarity sometimes called distance, proximity or closeness with some mathematics of calculating the distance between points on a graph. Straight line distance, also called Euclidean distance is a popular and familiar choice.

The algorithm works as follows:

- Load the data
- Initialize K to chosen no. of neighbors
- For each example in data calculate the distance between them.
- Sort the ordered collections of distance from smallest to largest by distances
- Pick the first K entries from the sorted collection.

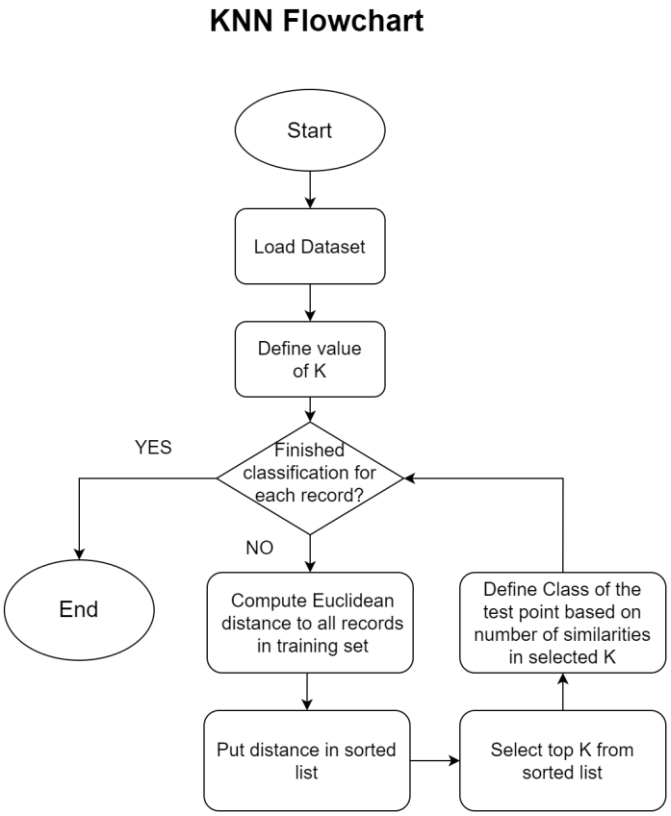

Fig. 7: KNN Flowchart

Note: to select the right K for our data, we need to run the algorithm with different values of K and choose the K that reduces the no. of errors and make accurate predictions.

10. SENTIMENT ANALYSIS

Sentiment analysis (or opinion mining) is a natural language processing technique used to determine whether data is positive, negative or neutral. Sentiment analysis is often performed on textual data to help businesses monitor brand and product sentiment in customer feedback, and understand customer needs.

For sentiment analysis, a text reviews dataset is used. Cleaning process is performed by using the re library. Polarity and Subjectivity are calculated and stored in new columns of the dataframe, you can see in Fig. 8. The percentage of the positive, negative and neutral reviews is calculated based on the polarity score, that is, if the polarity is greater than 0, the review has positive sentiment, if it is less than 0, then the review has negative sentiment, otherwise the review is neutral.

	ReviewContent	polarity	subjectivity	analysis
0	Good. It IS a page turner. You can read this b...	0.156818	0.503030	Positive
1	There are no words for how much I loathed this...	0.056667	0.549815	Positive
2	I think I would ordinarily cut this book more ...	0.089566	0.467542	Positive
3	Three disjointed characters for whom it's hard...	-0.060417	0.360417	Negative
4	Was snookered into this novel as it was compar...	0.191667	0.616667	Positive

Fig. 8: new Dataframe

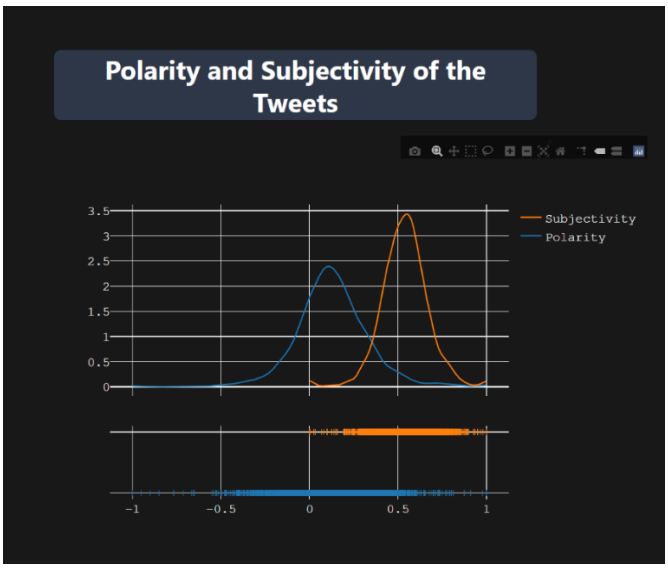

Fig. 9: Polarity and Subjectivity of Tweets

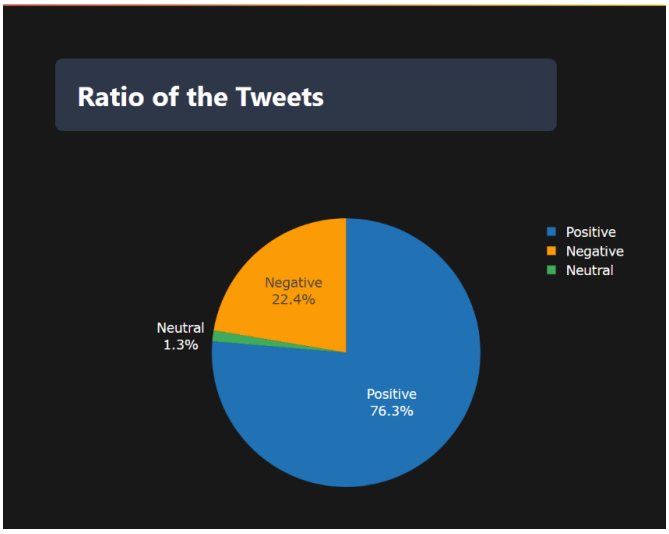

Fig. 10: Ratio of Tweets

9. LIMITATIONS

This project does not work on user login basis, so that the user can log in and directly get the suggestions based on the user's reading, and review history without having to type the book name to get recommendations.

10. SIGNIFICANCE OF RESEARCH

The whole purpose of this research is to make reading books easier for the readers, and recommend books with just a click. To present visualizations regarding reviews, books, authors, such as trending books and authors.

12. CONCLUSION

As from the research we understood that, people are moving more towards reading books online, as it saves time of going and buying the books physically.

From the datasets used for this project, we made some observations. First of all the dataset contains books from a wide range of years, with a wide range of authors and also the users of different age groups. We can conclude that most of the books receive ratings between 3-5, and most of them are the books published in English language. We can also see the books published over the years, with 2011 being the one with the highest number of books published.

For users we can say that most of the active users are between the age group of 31-50, almost 49.5%. Also teenagers are not into reading books as they have the lowest percentage in the user dataset. We can also conclude that most of the active users are from different states and cities of Canada with almost 78.3%. Lastly, from the sentiment analysis we can conclude that most of the users are having positive reviews regarding the books as the polarity of most the reviews is more between 0-1 and we can also see that almost all reviews are subjective.

11. FUTURE RESEARCH

- Future Research includes making the UI based on user login, so that the user can directly view the recommendations based on their reading history and the ratings given by the user for particular books, also the prediction of the unread books.
- Presenting sentiment analysis for a specific book by live scraping the tweets related to that book, so that the user can see and compare between two or more books.

REFERENCES

1. Deuk Hee Park, Hyea Kyeong Kim, Il Young Choi, Jae Kyeong Kim, "A literature review and classification of recommender systems research", Volume 39, Issue 11, 2012, Pages 10059-10072, ISSN 0957-4174
2. Brooke, Julian, Milan Tofiloski, and Maite Taboada. "Cross-linguistic sentiment analysis: From English to Spanish." Proceedings of the international conference RANLP-2009. 2009.
3. P. Turney. Thumbs up or thumbs down? Semantic orientation applied to unsupervised classification of reviews. Proc. of ACL, 2002.
4. B. Pang, L. Lee and S. Vaithyanathan. Thumbs up? Sentiment classification using Machine Learning techniques. Proc. of EMNLP, 2002.
5. Yugantshekhar, "ML- Content Based Recommender System", geeksforgeeks.org