NATIONAL INSTITUTE OF TECHNOLOGY JAMSHEDPUR

MINOR PROJECT (CS1607)

BOOK RECOMMENDER SYSTEM

A Report Submitted
in Partial Fulfillment of the requirements
for the Degree of
Bachelor of Technology (Hons.)
in
Computer Science & Engineering
(2020-2024)

Submitted By
Aryan Sinha (2020UGCS028)
Ruchi Gautam (2020UGCS074)
Hariom Kumar (2020UGCS081)

Under the Supervision of
Dr. Ashish Kr. Sahu
Assistant Professor
Department of Computer Science and Engg.

UNDERTAKING

We hereby declare that the project work presented in this report entitled **"Book Recommendation System"** submitted to the **NAITONAL INSTITUTE OF TECHNOLOGY JAMSHEDPUR**, is a record of an original work done by us under the guidance of **Dr. Ashish Kumar Sahu, Assistant Professor** in Department of Computer Science and Engineering , National Institute of Technology Jamshedpur, and this project work is submitted in the partial fulfillment of the requirements for the award of the degree of **Bachelor of Technology** in **Computer Science and Engineering**. I have not plagiarized or submitted the same work for the award of any other degree.

Aryan Sinha (2020UGCS028)

Ruchi Gautam (2020UGCS074)

Hariom Kumar (2020UGCS081)

CERTIFICATE

This is to certify that the project report entitled **Book Recommendation System** submitted by Aryan Sinha, Ruchi Gautam and Hariom Kumar to the **National Institute of Technology Jamshedpur,** in partial fulfillment for the award of the degree of **B.Tech in (Computer Science and Engineering)** is a bonafide record of project work carried out by them under my supervision. The contents of this report, in full or in parts, have not been submitted to any other Institution or University for the award of any degree.

Dr. Ashish Kumar Sahu
Assistant Professor
Computer Science and Engineering Department
National Institute of Technology Jamshedpur

ABSTRACT

Recommender systems provide users with personalized suggestions for products or services. These systems often rely on Collaborating Filtering (CF), where past transactions are analyzed in order to establish connections between users and products. The two more successful approaches to CF are User-based and Item-Based.

In this work we are focusing on building a Recommendation System based on Sentiment Analysis with the help of User-Based Collaborative Filtering and Matrix Factorization. We are applying several techniques and algorithms on the Amazon Book Reviews dataset to build a recommender system. The main focus in this work have been given to the ratings and reviews given by different users to the books as the sentiment analysis of each review can help us to get insights about the interest of the users. The system will help users discover new books and enhance their reading experience.

Acknowledgement

I want to sincerely thank everyone who has assisted us in finishing this B.Tech minor project report. First of all, I want to express my gratitude to my project mentor Dr. Ashish Kumar Sahu (Assistant Professor) for giving me the necessary direction, encouragement, and insightful advice throughout the project. Your words of support and inspiration inspired me to finish the project successfully.

Also want to express our gratitude to our family and friends for their support throughout the project. Their input, criticism, and suggestions were crucial in helping to shape this report. We want to thank you all for your contributions and say how proud I am that we were able to finish this project together.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

Chapter 1

Introduction

The purpose of this project is to create a personalized book recommendation system for consumers based on their reading inclinations. Machine learning techniques will be used by the system to assess user activity and make book suggestions that are relevant to their tastes. In our work, we are mainly focusing on the past ratings and reviews provided by the user to various books to get best books for each user.

1.1 Motivation

This idea was inspired by the fact that there are millions of books in the market, making it difficult for readers to select a book to read. Many readers have trouble finding books that interest them, so they could seek suggestions from friends, family, or book stores. These suggestions might not always match their particular choices, though.

However, conventional book recommendation algorithms sometimes ignore the tastes of the reader in favour of general choices like bestsellers or popular novels. As a result, readers frequently receive recommendations for publications that they are not interested in.

Hence, the main motivation behind this project is to create a book recommendation system that uses machine learning algorithms to offer individualized suggestions based on individuals' unique reading interests. This technique will assist readers in finding new books that are relevant to their interests, enhancing their reading experience overall.

1.2 Recommendation System

Recommendation system is a domain of AI and NLP which offers consumers tailored suggestions based on their preferences and previous actions. A recommendation system's objective is to assist users in finding new things or goods that they would find interesting, which raises user satisfaction and engagement.

1.2.1 Importance

Because they assist users in finding new and pertinent items or services based on their tastes and habits, recommendation systems are crucial and relevant today. These systems evaluate massive databases using machine learning algorithms and provide consumers with tailored recommendations. Among the advantages of recommendation systems are the following:

1. Enhanced customer engagement: By offering customers tailored recommendations based on their interests and habits, recommendation systems can enhance customer engagement. Increased user retention, loyalty, and satisfaction may result from this.

2. Increased sales and revenue: By presenting users with pertinent goods or services based on their past purchases, browsing habits, and preferences, recommendation systems can aid in boosting sales and revenue. Increased conversions, average order value, and customer lifetime value may result from this.

3. Improved user experience: By offering individualized recommendations that are catered to each user's interests and needs, recommendation systems can improve user experience. A more interesting and delightful user experience may result from this.

4. Lessened decision fatigue: By giving consumers a constrained number of pertinent options to select from, recommendation systems might lessen decision fatigue. Users may be able to make quicker and more informed decisions as a result.

1.2.2 Techniques of Recommendation System

The Recommendation System techniques can be broadly divided into four categories:

1. Content-based filtering [25] generate suggestions based on the users' previous selection and preferences. This mainly focuses on the content consumption of user and try to recommend the products or movies on the basis of users' past consumptions and interests.

2. The justification for demographic filtering [26] is the idea that people who share particular personal characteristics (such as gender, age, country, etc.) will also have similar preferences

3. Collaborative [27] In order to make recommendations to each user based on the data provided by the users we believe to have the most in common with them, we can store enough information on the system through the use of filtering, which enables users to rate a set of elements (such as videos, songs, films, etc.) in a CF-based website.

4. Hybrid filtering [28] uses Collaborative filtering and is frequently used in conjunction with demographic filtering or content-based filtering to maximize the benefits of each method.

1.3 Some Wonderful Minds

There are many experts in the field of recommendation systems. Few of them are :

1. Yehuda Koren: Koren[29] is a research scientist at Google and a professor at Technion - Israel Institute of Technology. He is well known for his work in the domain of collaborative filtering and has published various papers on the same.

2. Joseph A. Konstan: Konstan[30] is a professor of computer science at the University of Minnesota. He is known for his work on recommender systems and has contributed to the development of the GroupLens system, which is widely used for research on collaborative filtering.

3. George Karypis: Karypis[31] is a professor of computer science at the University of Minnesota. He has worked in the field of clustering and collaborative filtering and has developed multiple algorithms for recommender systems.

4. John Riedl: Riedl[32] was a professor of computer science at the University of Minnesota until his passing in 2013. He was a well-known personality in the recommendation system domain and co-founded the GroupLens System with Konstan.

5. Jure Leskovec: Leskovec[33] is a professor of computer science at Stanford University. He is known for his work in data mining and has contributed in the development of various recommendation system including the famous GraphLab system.

These are just a few notable people in the field of Recommendation system. Their contributions in this field are also used in current research works.

Chapter 2

Related Works

Recommender systems can be improved in a variety of ways. Various works have been done in the field. Few of them are –

In [1], Cach N. Dang has used deep learning techniques to perform the sentiment analysis on the users' reviews and applied Matrix Factorization to build a recommendation system and validated their model on Food reviews dataset and movie movies dataset.

In [2], social tag embedding is used in a collaborative filtering recommendation approach where user similarities based on both ratings and tag embeddings are combined to generate the recommendations.

Sentiment Analysis also enhance the recommendation system. We can find the same in the work of Preethi et al. [3], where sentiment analysis is done with the help of Recursive Neural Networks. This method improve and enhance the restaurant and movie based recommendation of a Cloud Based Recommendation system.

In the work of Amel Ziani et al. [4], they have proposed a multilingual recommendation system based on sentiment analysis to help Algerian users decide on products, restaurants, movies and other services using online product reviews. Their main goal is to combine both sentiment analysis and recommendation system in order to generate more accurate recommendation for users.

Hybrid Recommendation System was proposed by Kumar et al. [5] by combining both collaborative filtering and content-based filtering with the help of sentiment analysis of movie tweets to enhance the recommender system.

A recommendation system was designed by Rao et al. [6] which contains the user list and item list along with user reviews and with the help of these sentiment dictionaries the researchers divided the items into three categories: brand, quality and price. They used sentiment dictionaries to calculate the sentiment score of each item.

A different approach was adopted by Gurini et al. [7] to describe a user recommendation system for twitter. Their work mainly focused on enhancing the recommendation system with the help of implicit sentiment analysis. They defined a weighting function that considers the sentiment, objectivity and volume related to users' interests.

In yet another approach, Osman et al. [8] presented an electronic product recommender system based on contextual information from sentiment analysis by making use of user comments and preferences.

In another approach, Contratre et al. [9] also proposed a recommender process that includes sentiment analysis of textual data extracted from Facebook and Twitter in order to increase conversion by matching product offers and consumer preferences.

In a work, Moorthi K et al. [10] they demonstrated a recommendation model that involves Matrix Factorization as a collaborative filtering solution used for providing recommendations. They have worked on the hybrid recommendation system where they are focusing on various contents and ratings given by the user.

Chapter 3

Proposed Work

We have mainly focused on the user-based collaborative filtering recommendation system which is based on the ratings and reviews provided by the users on several books. Here, on the basis of these ratings and sentiment analysis of these reviews we are finding out the similarity between different users and with the help of that we are recommending top 5 books to the user. We have gone through the various steps which are described below.

I. Dataset Collection

For our work we have collected dataset from the Open-Source site KAGGLE [20]. The dataset contains details of Books from the Amazon i.e. Amazon Book Dataset [11].

This dataset contains two files: books.csv and ratings.csv

II. Feature Engineering and Sentiment Analysis

This is one of the important steps in any machine learning or AI project where we mainly trieds to find out the important features of the dataset on which we are going to work upon and also helps in cleaning the dataset by removing the irrelevant data.

1. Feature Engineering

   We kept only user ID, item ID (in our case we have Book ID), title, rating and textual reviews in our cleaned data and removed others.

   We have focused on few more important points which are as follows –
   1. Kept only those users which have given significant amount of ratings, so we defined the threshold to be 200 for this.
   2. Removed those books which have less than 50 ratings.

2. Sentiment Analysis and Sentiment Score Calculation

   As we can't deal with text directly, so we have decided to find out the sentiment score of each review. For this we have gone through two different methods.

   1. Using NLTK library

The Natural Language Toolkit (NLTK) [12] is a platform used for building Python programs that work with human language data for applying in statistical natural language processing (NLP). It contains text processing libraries for tokenization, parsing, classification, stemming, tagging and semantic reasoning.

So, we have used SentimentIntensityAnalyzer[13] method of NLTK library to convert the textual reviews into sentiment score of each review. This method uses VADER (Valence Aware Dictionary sEntiment Reasoning) [14] to find out the sentiment score. VADER contains list of positive and negative words and phrases. It first tokenizes the whole review into words and then assign a particular score to each word on the basis of their positivity and negativity (+1 for most positive and -1 for most negative). After assigning these scores it finally computes the sum and normalize them in the range of -1 to +1.

We also normalized these values between 0 to 1 as it is required in our case.

2. Using BERT and Hugging Face

   "Bidirectional Encoder Representations from Transformers" [15] is the abbreviation for BERT. A neural network architecture based on transformers is used to train BERT on a vast amount of text input. For different NLP tasks including text classification, question-answering, and language translation, it can be customised. BERT's ability to analyse the context of a word depending on the words that come before and after it in a sentence is one of its important characteristics. As a result, BERT can capture a richer grasp of language and generate more precise outcomes for NLP tasks.

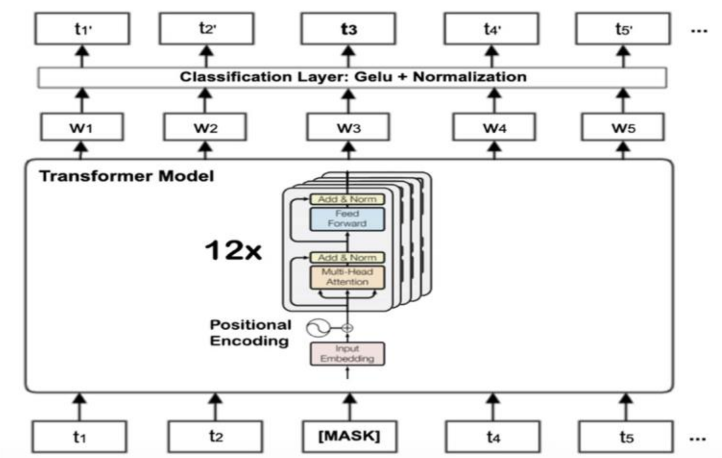

Fig. 1

We have used the pretrained BERT model available on the Hugging Face [16] platform to find out the sentiment score of each text review. These sentiment scores lied between 0-1 (0 means most negative and 1 means most positive).

After getting the Sentiment Score we did some processing where we removed those rows where rating is very high but sentiment score is very less and vice versa as it may reduce the accuracy and hence we can treat them as outliers. Then we merged both the ratings column and sentiment column by multiplying both of them to get a hybrid score.

III. Training and Testing

For the training and testing purpose we split the dataset into training and testing data in the ratio of 3:1 (i.e. 75% for training purpose and 25% for testing purpose).

For the Training purpose we have used KNN [17] and Matrix Factorization [18] algorithm of Surprise library.

A. Surprise library [19]

A machine learning library for developing and analysing recommender systems is called the Surprise [18] library in Python. It offers a straightforward and understandable API for developing and testing different recommendation systems.

The Surprise library's primary characteristics include:

1. support for a range of recommendation algorithms, including KNN [17], SVD++, and matrix factorization.
2. Common assessment metrics like RMSE, MAE, and precision/recall have built-in support.
3. Data loading and manipulation made simple by integration with pandas.
4. Cross-validation and hyperparameter tuning support.

So, using this library we converted our dataset in the format accepted by this library and then trained using KNNwithMeans [24], Cosine Similarity and Pearson Baseline.

B. KNN [17]

The popular machine learning technique known as K-Nearest Neighbours is utilised for both classification and regression problems.

The KNN algorithm determines the label or value of the input by locating the K training examples (neighbours) that are closest to the input data point. The output for classification jobs is the average of the labels of the K closest neighbours. The outcome for regression tasks is the average of the values of the K closest neighbours.

The KNNwithMeans [24] is a version of KNN which helps us to find out the nearest similar users based on the calculated hybrid score and fetched the top 5 books for each user.

To measure the similarity between users we have used Cosine similarity and Pearson baseline and found that Pearson baseline outperforms Cosine Similarity.

<u>Cosine Similarity</u>

Cosine similarity is a measure of similarity between two non-zero vectors in an inner product space. It can be calculated by dot product between two vectors and return the values ranging between 0 and 1.
Mathematically, cosine similarity between two vectors u and v is calculated as:

$$\cos(\theta) = \frac{u \cdot v}{||u|| * ||v||}$$

where dot product $(u \cdot v)$ is the sum of the products of the corresponding components of u and v, and $||u||$ and $||v||$ are the magnitudes of the vectors u and v, respectively.

<u>Pearson Baseline</u>

The Pearson baseline in this context refers to the difference between the average rating given by a user to a group of items and the average rating given to those items by all users.

The formula for the Pearson baseline is as follows:

$$b_{ui} = \mu + b_u + b_i$$

Here,

b_{ui} is the Pearson baseline for user u and item i
μ is the overall average rating across all items and all users
b_u is the user-specific bias term that captures how the user u deviates from the overall average rating

b_i is the item-specific bias term that captures how the item i deviates from the overall average rating

Here,

$$b_u = \left(\frac{1}{|I_u|}\right) \sum (\mathrm{r_{ui}} - \mu)$$

$$b_i = \left(\frac{1}{|U_i|}\right) \sum (\mathrm{r_{ui}} - \mu)$$

where:

$|I_u|$ is the number of items rated by user u

$|U_i|$ is the number of users who have rated item i

$\mathrm{r_{ui}}$ is the rating of item i by user u

C. Matrix Factorization [18]

Matrix Factorization, in recommendation system is a technique to decompose the user-item matrix into two lower dimensional matrices, one for the users and other for the items.

Matrix Factorization helps in finding out the missing value in user-item matrix by capturing the similarity between different other users and items. This also reduces the complexity of the model and helps in storing the details in less storage area.

In this work we have gone through NMF (Non-Negative Matrix Factorization) and SVD (Single-Valued Decomposition) with the help of Surprise library for Recommendation System.

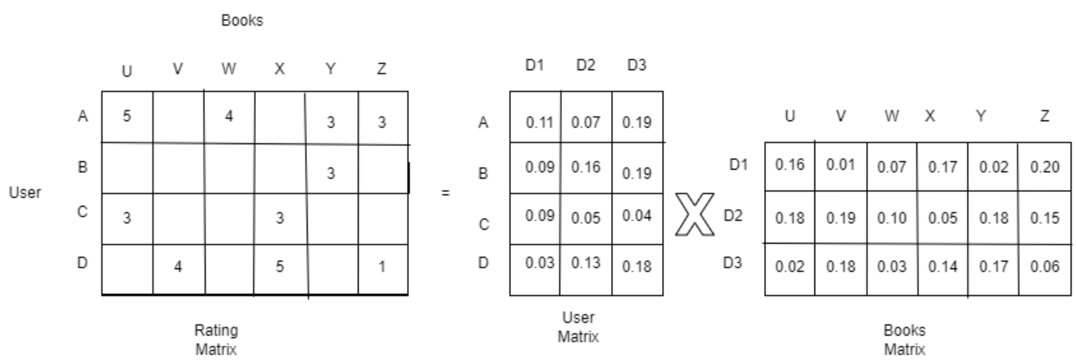

Fig. 2

IV. Predictions and Accuracy

On the basis of above mentioned methodology, recommendation systems were built and to compare them with each other we have used few metrics like RMSE, Precision and Recall.

RMSE

RMSE stands for Root Mean Square Error which is the measure of the standard deviation of the residuals i.e. how spread out the residuals are.

$$RMSE = \sqrt{\sum_{i=1}^{N} \frac{\left(Z_f - Z_o\right)^2}{N}}$$

Z_f is predicted output, Z_o is actual output and N is total number of outputs.

Precision and Recall

Precision is defined as the number of True Positives among the predicted positives.

Recall is defined as the number of the True Positives among the actual Trues in the dataset.

$$Precision = \frac{True\ Positive}{True\ Positive + False\ Positive}$$

$$Recall = \frac{True\ Positive}{True\ Positive + False\ Negative}$$

Chapter 4

Experimental Setup and Results Analysis

We have used the Kaggle [20] Notebook for training and testing purpose and accessed it's GPU T4 X2 to train the BERT model for sentiment analysis.

The following is the configuration of the GPU we have used (GPU T4 X2) –

```
Mon Apr 10 11:17:31 2023
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 470.161.03   Driver Version: 470.161.03   CUDA Version: 11.4     |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|                               |                      |               MIG M. |
|===============================+======================+======================|
|   0  Tesla T4            Off  | 00000000:00:04.0 Off |                    0 |
| N/A   38C    P8     9W /  70W |      0MiB / 15109MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+
|   1  Tesla T4            Off  | 00000000:00:05.0 Off |                    0 |
| N/A   39C    P8     9W /  70W |      0MiB / 15109MiB |      0%      Default |
|                               |                      |                  N/A |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                                  |
|  GPU   GI   CI        PID   Type   Process name                  GPU Memory |
|        ID   ID                                                   Usage      |
|=============================================================================|
|  No running processes found                                                 |
+-----------------------------------------------------------------------------+
```

Fig. 3

<u>Dataset used in our work</u> –

We are using the Amazon Book Reviews Dataset[11] which is available on Kaggle[20].
This dataset contains two files : books.csv and ratings.csv

The first file reviews file contain feedback about 3M user on 212404 unique books the data set is part of the Amazon review Dataset it contains product reviews and metadata from Amazon, including 142.8 million reviews spanning May 1996 - July 2014.

Features	Description
Id	The id of book
Title	Book title
Price	The price of book
User_id	Id of the user who rates the book
ProfileName	Name of the user who rates the book
Review/Helpfulness	Helpfulness rating of the review, eg.2/3
Review/Score	Rating from 0 to 5
Review/Time	Time of given the review
Review/Summary	The summary of a text review
Review/Text	The full text of a review

T1 - Ratings Metadata

The second file Books Details file contains details information about 212404 unique books it file is built by using google books API to get details information about books it rated in the first file.

Features	Description
Title	Book title
Descripe	Description of book
Authors	Name of the book authors
Images	Url for book cover
PreviewLink	Link to access the book on google books
Publisher	Name of the publisher
PublishedDate	The date of publish
InfoLink	Link to get more information about the book on google books
Categories	Genres of books
RatingsCount	Averaging rating for book

T2 - Books Metadata

Result and Analysis

We have gone through different techniques to build the recommendation system. Those algorithms have been illustrated in the above section. So following are the techniques on which we have worked in this project –

i. KNNwithMeans with Pearson Baseline Similarity and NLTK Sentiment Analyzer
ii. KNNwithMeans with Pearson Baseline Similarity and BERT Sentiment Analyzer
iii. Non-Negative Matrix Factorization with NLTK Sentiment Analyzer
iv. Non-Negative Matrix Factorization with BERT Sentiment Analyzer
v. Single Value Decomposition with NLTK Sentiment Analyzer.
vi. Single Value Decomposition with BERT Sentiment Analyzer.

After going through the all above compositions we have got the following results.

Algorithm	**Sentiment Analyzer**	**RMSE**	**Precision**	**Recall**
KNNwithMeans	NLTK	0.6296	0.966	0.644
KNNwithMeans	BERT	0.5162	0.836	0.548
NMF	NLTK	0.71	0.98	0.64
NMF	BERT	0.5062	0.88	0.59
SVD	NLTK	0.74	0.96	0.63
SVD	BERT	0.525	0.83	0.58

T3 – Result

As we can see in the above result table, among the sentiment analysis models, the VADER (NLTK) algorithm is performing good in each case (mainly in precision) and BERT is generating best loss value and precision is decent. The best model among NMF, SVD and KNN with Means is found to be NMF (Non-negative Matrix Factorization).
If we consider loss and precision important metrics then following combination is performing best in our work –

1. NMF with VADER sentiment analysis
2. NMF with BERT sentiment analysis

In recommendation system, both precision and recall are important metrices and their value depend on various conditions and dataset. In our case we can see the Precision-Recall trade-off where we have high precision but low recall. Precision in book recommendation system means recommended books are founded by the user relevant. By recall it means that what the system is recommending how many of them are relevant.

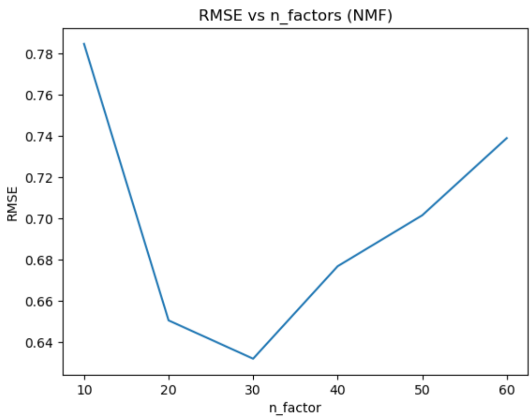

Fig. 4(i)

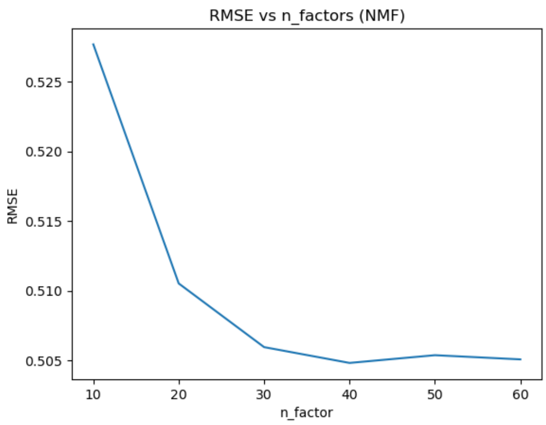

Fig. 4(ii)

Above graph is showing how RMSE varies with change of n_factors (number of factors) in NMF while using NLTK sentiment analyser Fig.3(i) and BERT sentiment analyser Fig.3(ii).

With the help of above graph, we can conclude that for NMF with NLTK, n_factor = 30 will perform best and in case of NMF with BERT, n_factor = 50 is giving better result.

Chapter 5

Conclusion and Future Work

I. Conclusion

Good Recommendation System is very important to give the people best experience and results. We have tried to built the same and tried lots of methods to recommend the books to the user including the KNN and Matrix Factorization and this is mainly focusing on Ratings and Reviews given by the users. We have also focused on the sentiment analysis of the reviews for which we have used two different techniques – VADER and BERT. The RMSE and precision are pretty good but recall is not so impressive as our result is suffering from Precision-Recall Trade-off.

II. Future Work

After trying all these techniques, we are only left out to try the Neural Network method to build the recommendation system. One popular method to go with that is Neural Collaborative Filtering [21] where MLP [22] (Multi-Layered Perceptron) will be used as Neural Networks and will help to do matrix factorization [18] and similarity calculation between users considering various aspects. So, we have decided that we will try to build more accurate and precise Recommendation engine using NCF and other Neural Network Methods like GNN (Graph Neural Network) [23].

References

1. Cach N. Dang , María N. Moreno-García and Fernando De la Prieta ; Using Deep learning and Matrix-Fatorization [CrossRef]

2. Sánchez-Moreno, D.; Moreno-García, M.N.; Mobasher, B.; Sonboli, N.; Burke, R. Using Social Tag Embedding in a Collaborative Filtering Approach for Recommender Systems. In Proceedings of the 2020 IEEE/WIC/ACM International Joint Conference on Web Intelligence and Intelligent Agent Technology, Melbourne, Australia, 14–17 December 2020; IEEE: New York, NY, USA, 2021.

3. Preethi, G.; Krishna, P.V.; Obaidat, M.S.; Saritha, V.; Yenduri, S. Application of Deep Learning to Sentiment Analysis for Recommender System on Cloud. In Proceedings of the 2017 International Conference on Computer, Information and Telecommunication Systems (CITS), Dalian, China, 21–23 July 2017; IEEE: New York, NY, USA, 2017; pp. 93–97

4. Amel ZIANI, Nabiha AZIZI, Didier SCHWAB, Monther ALDWAIRI, Nassira CHEKKAI, Djamel ZENAKHRA, Soraya CHERIGUENE; Recommendation system through sentiment analysis [CrossRef]

5. Kumar, S.; De, K.; Roy, P.P. Movie recommendation system using sentiment analysis from microblogging data. IEEE Trans. Comput. Soc. Syst. 2020, 7, 915–923. [CrossRef]

6. Rao, K.Y.; Murthy, G.; Adinarayana, S. Product recommendation system from users reviews using sentiment analysis. Int. J. Comput. Appl. 2017, 975, 8887.

7. Gurini, D.F.; Gasparetti, F.; Micarelli, A.; Sansonetti, G. A Sentiment-Based Approach to Twitter User Recommendation. RSWeb@ RecSys 2013, 1066, 1–4.

8. Osman, N.; Noah, S.; Darwich, M. Contextual sentiment based recommender system to provide recommendation in the electronic products domain. Int. J. Mach. Learn. Comput. 2019, 9, 425–431. [CrossRef]

9. Contratres, F.G.; Alves-Souza, S.N.; Filgueiras, L.V.L.; DeSouza, L.S. Sentiment Analysis of Social Network Data for Cold-Start Relief in Recommender Systems. In Proceedings of the World Conference on Information Systems and Technologies, Naples, Italy, 27–29 March 2018; Springer: Berlin/Heidelberg, Germany, 2018; pp. 122–132.

10. Mercy Milcah Y, Moorthi K. AI based Book Recommender System with Hybrid Approach [CrossRef]

11. Amazon Book Dataset; Kaggle [CrossRef]

12. Edward Loper and Steven Bird ; NLTK: The Natural Language Toolkit [CrossRef]

13. Venkateswarlu Bonta, Nandhini Kumaresh and N. Janardhan; A Comprehensive Study on Lexicon Based Approaches for Sentiment Analysis [CrossRef]

14. C.J. Hutto Eric and Gilbert Georgia; VADER: A Parsimonious Rule-based Model for Sentiment Analysis of Social Media Text [CrossRef]

15. Jacob Devlin Ming-Wei Chang Kenton Lee Kristina Toutanova Google AI Language; BERT: Pre-training of Deep Bidirectional Transformers for Language Understanding [CrossRef]

16. Thomas Wolf, Lysandre Debut, Victor Sanh, Julien Chaumond, Clement Delangue, Anthony Moi, Pierric Cistac, Tim Rault, Remi Louf, Morgan Funtowicz, Joe Davison, ´ Sam Shleifer, Patrick von Platen, Clara Ma, Yacine Jernite, Julien Plu, Canwen Xu, Teven Le Scao, Sylvain Gugger, Mariama Drame, Quentin Lhoest, Alexander M. Rush; Transformers: State-of-the-Art Natural Language Processing [CrossRef]

17. Gongde Guo, Hui Wang, David Bell, Yaxin Bi, and Kieran Greer; KNN Model-Based Approach in Classification [CrossRef]

18. Daniel D. Lee & H. Sebastian Seung; Learning the parts of objects by non-negative matrix factorization [CrossRef]

19. Surprise: A Python library for recommender systems ; Nicolas Hug [CrossRef]

20. Kaggle, A Data Science Community [CrossRef]

21. Neural Collaborative Filtering [CrossRef]

22. Dennis W. Ruck Steven K. Rogers Matthew Kabrisky; Feature Selection Using a Multilayer Perceptron [CrossRef]

23. A Comprehensive Survey on Graph Neural Networks [CrossRef]

24. Movie Recommendation Systems Based on Collaborative Filtering: A Case Study on Netflix (Based on KNN) [CrossRef]

25. Content-Based Recommendation System; Michael J. Pazzani and Daniel Billsus [CrossRef]

26. On the enhancement of collaborative filtering by demographic data [CrossRef]

27. Applying Collaborative filtering to Usenet News [CrossRef]

28. Hybrid Recommender Systems: Survey and Experiments [CrossRef]

29. Yehuda Koren; Matrix Factorization Techniques for Recommendation System [CrossRef]

30. Joseph A. Konstan; Recommender System: From Algorithms to User-experience [CrossRef]

31. George Karypis; Item-Based Collaborative Filtering Recommendation algorithm [CrossRef]

32. John Reidl; Experiences with GroupLens: Making Usenet useful again [CrossRef]

33. Jure Leskovec; Patterns of Influence in Recommendation Network [CrossRef]