

雲南大學

本科毕业论文（设计）

题 目： PRODUCT RECOMMENDATION SYSTEM BASED ON
COLLABORATIVE FILTERING ALGORITHM

学 院： School of Information

专 业： Computer Science and Technology

学 号： 20193290764

姓 名： Minhazul Islam

指导教师（职称）： 余立行（讲师）

年 月 日

独创性声明及使用授权页示例：

毕业论文（设计）独创性声明及使用授权

本毕业论文（设计）是作者在导师指导下取得的成果。除了文中特别加以标注和致谢的地方外，论文（设计）中不包含其他人已经发表或撰写过的研究成果，不存在剽窃或抄袭行为。与作者一同工作的同志对本研究所做的任何贡献均已在论文中作了明确的说明并表示了谢意。

现就论文（设计）的使用对云南大学授权如下：学校有权保留本论文（设计）（含电子版），也可以采用影印、缩印或其他复制手段保存论文（设计）；学校有权公布论文的全部或部分内容，可以将论文（设计）用于查阅或借阅服务；学校有权向有关机构送交学位论文（设计）用于学术规范审查、社会监督或评奖；学校有权将学位论文（设计）的全部或部分内容录入有关数据库用于检索服务。

作者签名：_____ 导师签名：_____ 日 期：_____

Abstract

The utilization of the Internet has experienced a significant surge in the past decade. As a consequence of its efficaciousness, various enterprises that rely on it are now equipped to flourish and broaden their horizons. E-commerce represents one potential alternative among others. The significance of recommendations has amplified in tandem with the expeditious escalation of electronic commerce. The process of providing users with tailored suggestions based on their unique requirements and areas of interest is commonly referred to as a recommendation. Users have the potential to obtain personalized recommendations comprising factors such as pricing, residential location, preferred items, cart contents, product searches, and historical purchases. The implementation of recommendation systems in various contexts has been shown to result in significant enhancements in user engagement and experience, as well as in sales performance.

The proposed recommendation method will facilitate the dissemination of product recommendations to the end users. The recommendation system implements a model-based collaborative filtering approach to offer product recommendations, drawing insights from the user's prior purchase history and ratings. Novice users will additionally receive suggestions for trendy commodities, well-liked goods, and reduced-price merchandise. The rising trend toward e-commerce is accompanied by the proliferation of its recommendation systems, which have emerged as a valuable and desirable addition to the industry. The utilization of product recommendation systems is prevalent in e-commerce platforms as they offer personalized recommendations to users. Collaborative filtering is among the most widely utilized techniques in the domain of recommendation systems. This study intends to introduce a product recommendation system that utilizes collaborative filtering algorithms. This study involves an evaluation of the efficacy of the proposed system through its application on a real-world data-set. In addition, it is compared to other state-of-the-art recommendation systems. The findings of the experiment indicate that the system proposed by this researcher exhibits superior performance in both accuracy and diversity when compared to other currently available recommendation systems that are considered state-of-the-art.

Keywords: e-commerce, recommendation, collaborative filtering

Table of Content

Abstract.	1
Chapter 1 Introduction	3
1.1 Background and Related Work	3
1.2 Business analysis	2
1.2.1 Impact on e-commerce business	2
1.2.2 The impact of product recommendations	3
1.2.3 Customer Satisfaction Customer behavior analysis	4
1.3 Thesis overview	5
1.4 Comparative Studies	5
Chapter 2 Proposed Methodology	6
2.1 Method	6
Chapter 3 Recommendation Technology Based on Collaborative Filtering Algorithm.	8
Chapter 4 Requirement Analysis	12
4.1 Necessary libraries and Implementation	12
4.2 Data Collection	13
4.3 Information about data	15
4.4 Data Cleaning	16
4.4.1 Check Duplicates	17
4.1.2 Check Missing Values	17
4.1.3 Check Outliner	19
4.1.4 Data Cleaning Summary	20
4.5 Exploratory Data Analysis	20
4.5.1 The average reviews that given authors	21
4.5.2 Percentage of Ratings According to Authors	22
4.5.3 Rating comparisons	23
4.5.4 Correlation research	25
4.5.5 Data Visualization	26
4.6 Related Technology	29
4.6.1 Python	29
4.6.2 Matrix Factorization	30
4.6.3 Euclidean Distance	31
Chapter 5 Model Building	32
Chapter 6 Experimental Results and Analysis	34
6.1 Test	35
6.2 Evaluate the Collaborative recommend-er model	37
6.3 Bugs encountered	37
6.4 Limitations	38

Chapter 7 Conclusion	39
References	40
Appendix	42
Example: Here recommends Books for “1984”.	2
Acknowledgement	3

Chapter 1 Introduction

Recommend-er systems are thought of as apparatuses and programming designing organizations that are utilized to create benefits recommendations for the clients as well as to help them within the decision-making handle. Collaborative filtering could be an ordinary recommend-er framework procedure (CF). The elemental guideline behind CF is the extraction of information with respect to past client behavior or perspectives that are predominant in society, as well as any perspectives that are likely to request to the show framework client or are comparable to his inclinations. The CF strategy employments a network that the client is as it was given after anticipating the inputs and making a comparing figure for the yields recorded underneath. The work of CF is to give gauges of the current number of dynamic client inputs and recognize other clients. At that point, it accomplishes comparison preparation to induce the closest individual for the current client dynamic neighborly for any comparable inclinations with the current dynamic client inclinations.

To create proposals for diverse categories of clients, online e-commerce businesses utilize an assortment of proposal motors. In most cases, these E-companies utilize collaborative filtering, which develops exceptionally huge data-sets and produces high-quality thoughts. Based on chronicled information, this filtering strategy generates a list of proposals for its client based on the user's purchased and rated products. I'm creating a collaborative filtering strategy for the goodbooks-10k data-set to this extent. It'll be profoundly useful for both them and e-commerce businesses attempting to offer the leading proposal framework to the site on the off chance that we utilize this proposed framework to help e-commerce clients searching for comparable items to purchase.

1.1 Background and Related Work

A large group of resourcefulness's, containing but not short to Alibaba, Netflix, Sephora, and Spotify, have arose as pioneers in the exercise of advice wholes by way of to help reductions and embellish consumer date. These adventures have capably applied the potential of machine intelligence and dossier data to offer widely distinguished consumer knowledge. The exercise of a advice generator, alternatively refer to as a approval structure, inside buying includes the exercise of a spreadsheet treasure that is to say particularly planned to offer embodied commodity advice to consumers through an study of their flipping through experiences, purchase patterns, and different relevant dossier. The advice diesel, usually refer to as a advice plan, is a spreadsheet treasure particularly grown for buying purposes, accompanying the aim of transferring embodied brand advice to consumers. This is proficient through the reasoning of their perusing annals, purchase

patterns, and different relevant dossier. Provided that individual is examining a internet-located covering release, and has picked various items for purchase that have existed established in the in essence buying cart. The buying site's advice structure has the wherewithal to estimate consumer's skimming and purchase annals accompanying the goal of providing tailor-made plans for supplementary parts that maybe of interest to ruling class. An explanatory instance contains the concerning manipulation of numbers advice of jackets accompanying identical attributes, in the way that color or style, to those that have earlier happened obtained established settled services inclinations. The earlier arrangements engage progressive machine intelligence and dossier logical methods to resolve far-flung capacities of consumer-accompanying dossier, so that support embodied brand pieces of advice established individual consumer inclinations.



Figure 1 Recommendation

1.2 Business analysis

Recommendation tools have enhanced progressively well-known in the buying manufacturing, admitting companies to supply embodied consumer occurrences and increase transactions. By contribution appropriate produce plans, parties can help consumer date and loyalty and drive supplementary profit through cross-sale and up-auction.

1.2.1 Impact on e-commerce business

E commerce trades face the challenge of maintain accompanying the changing advantages and flows of their clients. By what method can they offer embodied and appropriate amount and aids that couple their consumers' needs and interests? Individual answer searches out use approval tools, that are spreadsheet plans that resolve client dossier and action to imply articles that they ability like or need. Advice tools are algorithms that use dossier excavating, machine intelligence, and machine intelligence to produce hints for crop, aids, or content that clients maybe concerned in. They may be established various types of dossier, in the way that client head count, purchase past, leafing through experiences, ratings, reviews, or public radio

exercise. They can likewise use various forms to create approvals, in the way that cooperative draining, content-located leaking, or composite refining, that connect two together approaches. Advice transformers can convince expected advantageous for e trade trades so that equal changeable consumer predilections and currents. Aforementioned power plants can increase client delight and dependability by providing embodied and appropriate output and duties. Also, they can boost auctions and profit by growing change rates, average order worth, cross-transfer, and up-transfer event. Additionally, approval turbines can help client memory and date by constituting a definite and noteworthy buying knowledge that helps repeat purchases and referrals. In addition, they can help to develop consumer intuitions and response by accumulating and resolving dossier on client management, desires, and delight.

Brand approval motors can have a meaningful affect buying trades. They help drive concerning business and conversions by personalizing offers, growing limited traffic, and threatening bounce and cart abdication rates¹. Advice turbines can likewise increase client vindication and dependability by providing embodied and appropriate amount and services². Parties that surpass at professionalization create 40% more profit from those actions than “average players”². The approval transformer advertises proper to reach USD 15.13 billion by 2026, and it was costly at USD 2.12 billion in 2020³

1.2.2 The impact of product recommendations

The recommendation engine advertise is anticipated to reach USD 15.13 billion by 2026, and it was esteemed at USD 2.12 billion in 2020, enrolling a CAGR of 37.46% amid the period of 2021-2026.

As item shows can impact online shopping behaviors, around 71% of e-commerce locales offer item suggestions. The number is indeed higher in Nordic nations: 90% of customers detailed finding suggestions on the homepage of e-commerce destinations.

In spite of the fact that item proposals ought to be put all over the e-commerce location for most extreme viability, numerous online retailers are not doing so. Agreeing to a ponder conducted within the Nordic nations in 2021, 81% of customers detailed not finding any proposals on the item posting pages (look comes about or category pages) of e-commerce destinations.

54% of retailers claimed that item suggestions act as the key driver of the AOV (normal arrange esteem) in client buy. After executing Clerk’s proposal motor, Characteristic Infant Shower experienced 21% increment in AOV and 31% increment in wicker container estimate.

Investigate from Deals constrain appears that customers that clicked on suggestions are 4.5x more likely to include these things to their cart and 4.5x more likely to total the buy.

A ponder by Monet set out to evaluate the deals effect on item proposals. They compared customers who saw a proposal but didn't lock in with those who locked in with a suggestion. The investigate found that customers who locked in with a prescribed item had a 70% higher change rate amid that session.

Indeed, customers who clicked on a item suggestion but didn't purchase anything had higher engagement rates. These customers were 20% more likely to return to the location afterward. Customers that clicked a item proposal are about 2x more likely to come back to the net shop:

37% of customers that clicked a suggestion amid there to begin with visit returned, compared to fair 19% of customers that didn't tap a suggestion amid there to begin with visit.

Sales drive Inquire about found that item suggestions account for as it were 7% of e-commerce activity but make up for 24% of orders and 26% of income.

49% of customers said they have obtained an item that they did not at first proposed to purchase after getting a personalized proposal.

Personalized item suggestions are evaluated to account for more than 35% of buys on Amazon.

1.2.3 Customer Satisfaction Customer behavior analysis

With the rising number of e commerce stores, the desire of online customers for a personalized shopping encounter has expanded as well. Item proposals let our offer a customized proposal to our clients and assist us fulfill those expectations. By doing simply, can progress the in general client involvement on our site. This progresses client fulfillment which is one of the vital components for our overall victory.

Utilizing item proposals on our site may be an exceptionally great way to construct client devotion. Advertising item proposals to our clients make a difference and boost client fulfillment. When individuals cherish our commerce, they are more likely to come back to our location once more for future buys. And not fair that, upbeat clients will to advance our business through word-of-mouth. And no promoting methodology can be way better than that. This will draw in more clients and assist us accomplish our objectives quicker.

The main benefits coming from customer behavior analysis system are:

- ◆ Boost in sales,
- ◆ Better understanding of customers and
- ◆ Long tail strategies.

The most advantage can be put in fair three words - boost in deals. Concurring to McKinsey, 35% of all Amazon buys, and 70% of Netflix buys, are driven by their proposal frameworks, and the begin of utilizing proposals essentially boosted their deals. Besides, amid COVID-19 was widespread, numerous retailers went online, digitalized their businesses, and changed their trade culture to adjust to the modern and ever-changing circumstances. Concurring to reports, in 2020 the development of e-commerce deals in US alone was more than 30%. This gives a colossal sum of online information for potential investigation, and utilize in building machine learning frameworks. Comparing the development of US commerce deals in the period 2019-2020

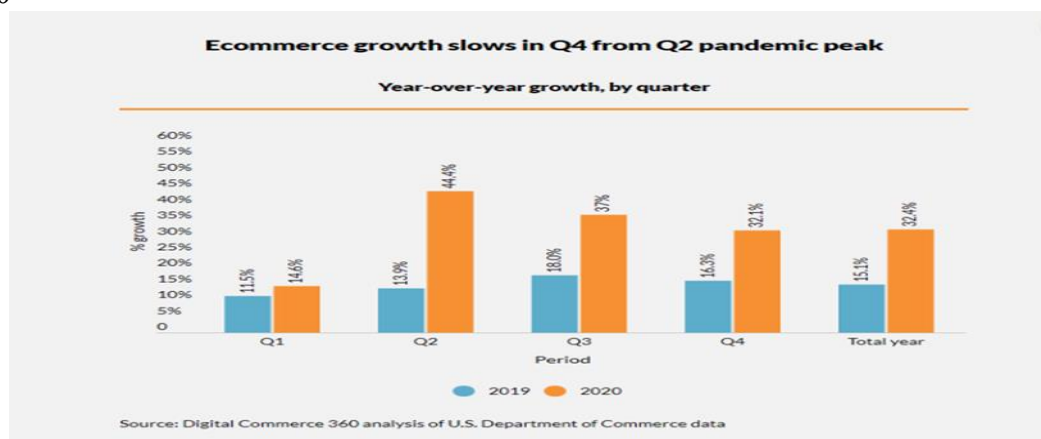


Figure 2 Digital Commerce 360

The moment advantage comes from the superior understanding of clients. This can be the portion where customer profiling comes in. By profiling clients, ready to way better get it their conduct and, subsequently, way better get it their needs, or in other words, meet their needs, which finally can be compensated with higher client fulfillment and dependability. Other than expanded client fulfillment, we will effortlessly make mechanized showcasing campaigns, and personalize them based on client investigation.

The following advantage could be a much better strategy for long tail things. The term long tail thing, alludes to specialty and hard-to-find things that are exceptionally particular and interesting, and as a rule as it were have a little gather of individuals searching for them. From a customer's point of view, apparatuses such as proposal frameworks, permit them to discover items exterior their quick region, and things they, something else, would not have had get to. From a supplier's point of view, in the event that they hold things in a stockroom, covered up from the clients that would like them, this technique may get to be exceptionally beneficial.

1.3 Thesis overview

Online E-commerce companies use various recommendation engines to recommend a variety of suggestions to various types of clients. These E-companies generally use collaborative filtering, which scales to enormous data sets and produces high-quality suggestions. This sort of filtering is based on the user's purchased and rated products, based on the past data this model provides a suggestions list for its customer. In this project, we will build a collaborative based filtering technique for the goodbooks-10k items data set.

Collaborative filtering is a technique used by some recommendation engines to recommend items based on the preferences of other users. It works by searching a large group of people and finding a smaller set with tastes similar to a particular user. A product recommendation system is a subclass of recommend-er systems that suggest products to users based on their preferences and interests. Collaborative filtering is one of the most popular algorithms used in product recommendation systems.

1.4 Comparative Studies

Recommend-er frameworks are utilized to deliver personalized records of recommendations related to a user's concerns and desires. These proposals are based on what the client has acquired or already seen but too on the movement history of the client.

A comparative think about of approaches in recommend-er frameworks was conducted in 2 and it inspected and compared the diverse existing suggestion approaches: those content-based sifting, those collaborative sifting, and at last the statistic and social approaches. It too demonstrated, for each approach, the areas of application, a few curiously illustrations, as well as its preferences and impediments.

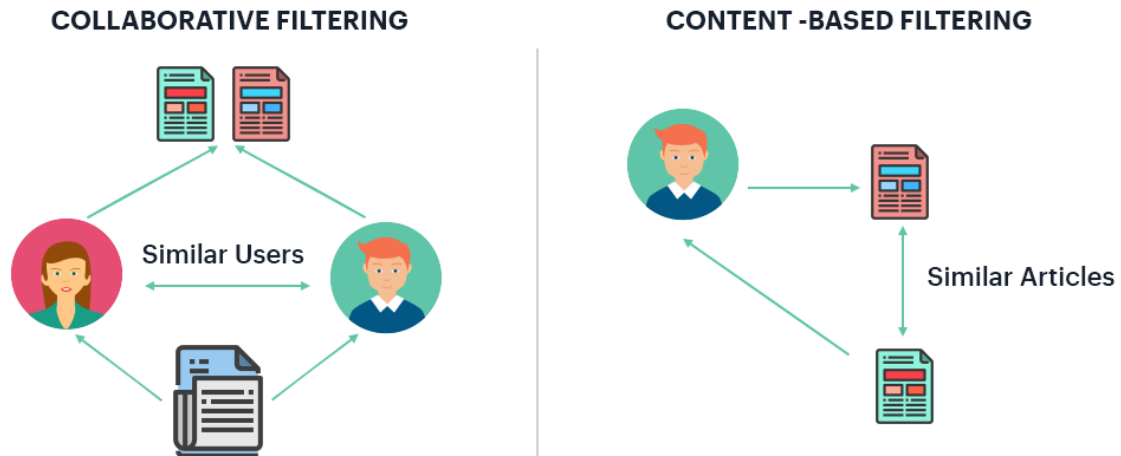


Figure 3 Types of Recommendation

For each client, collaborative recommend-er frameworks prescribe things based on how, comparable clients enjoyed the thing. Collaborative sifting is based on collecting and analyzing information on user’s behaviors, their regular exercises, evaluations, and expecting what they will like based on the closeness with other clients. A key advantage of this approach is that it does not depend on each detail and subsequently it can absolutely prescribe complex items such as things without requiring an “in-detail” of the items.

Chapter 2 Proposed Methodology

Collaborative sifting is based on collecting and analyzing information on user’s behaviors, their regular exercises, appraisals, and foreseeing what they will like based on the similitude with other clients. This framework matches people with comparable interface and gives suggestions based on these coordinating.

Our objective is to build a recommendation engine to suggest similar items to clients based on their past ratings for other items. For this reason, to begin with, we are going to perform Exploratory Data Analysis [EDA] and after that implement recommendation algorithms with Collaborative algorithms.

2.1 Method

Collaborative filtering is based on the opinion that people who decided to make a purchase in the past will decide in the future, and that they will likely prefer similar kinds of items as they did in the past. The system generates recommendations using data about rating profiles for

different users or goods.

The collaborative filtering approach has lots of advantages. One of them is that it is capable of accurately recommending complex items such as movies without requiring an "understanding" of the item itself. Many algorithms have been used in measuring user similarity or item similarity in recommend-er systems. the collaborative filtering method is clear enough, there may be some problems while implementing it. For example, a cold start can be an issue. For a new user or item, there isn't enough historical data to make accurate recommendations. There may be an issue of a product cold start or user cold start. The user cold start problem occurs when new users enter a website or app for the first time and the system has no information about them or their preferences. In this case, the system fails to recommend anything. Similarly for new products, as they have no reviews, likes, clicks, or other interactions among users, so no recommendations can be made.

One of the methods to deal with the issue is to recommend trending products to the new customer in the early stages. Here the selection can be narrowed down based on contextual information – their location, which site the visitor came from, a device used, etc. Behavioral information will be collected after a few clicks during that first visit, and start to build up from there.

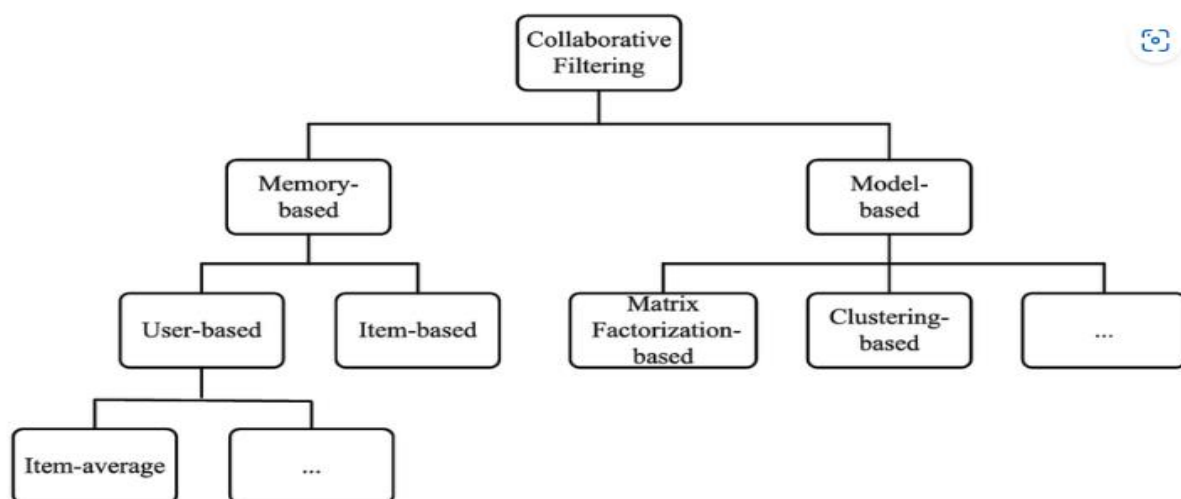


Figure 4 Classification of collaborative filtering algorithms.

To build a system that can automatically recommend items to users based on the preferences of other users, the first step is to find similar users or items. The second step is to predict the ratings of the items that are not yet rated by a user. Collaborative filtering is a family of algorithms where there are multiple ways to find similar users or items and multiple ways to calculate rating based on ratings of similar users. Depending on the choices we make, our end up with a type of collaborative filtering approach.

One important thing to keep in mind is that in an approach based purely on collaborative filtering, the similarity is not calculated using factors like the age of users, genre of the movie,

or any other data about users or items. It is calculated only on the basis of the rating (explicit or implicit) a user gives to an item. For example, two users can be considered similar if they give the same ratings to ten movies despite there being a big difference in their age.

One of the approaches to measure the accuracy of mine result is the Root Mean Square Error (RMSE), in which our predict ratings for a test data-set of user-item pairs whose rating values are already known. The difference between the known value and the predicted value would be the error. Square all the error values for the test set, find the average (or mean), and then take the square root of that average to get the RMSE.

Another metric to measure the accuracy is Mean Absolute Error (MAE), in which I find the magnitude of error by finding its absolute value and then taking the average of all error values.

Chapter 3 Recommendation Technology Based on Collaborative Filtering Algorithm

In our standard of, living we frequently inquire our good companions for exhortation when choosing items to assist us make choices. CF applies this thought to personalized proposal, that's, suggesting reasonable things to target clients based on the assessment of certain things by clients with comparative interface. Taking books as an illustration, on the off chance that two clients have browsed or bought the same books, they are likely to have possibly comparative leisure activities and buy comparable books within the future. That's to say, client A and client B have numerous buys that are the same or comparative. Client A bought a book, but client B does not know the buy record of client A, so this book is likely to be prescribed to client B. The structure based on CF innovation.

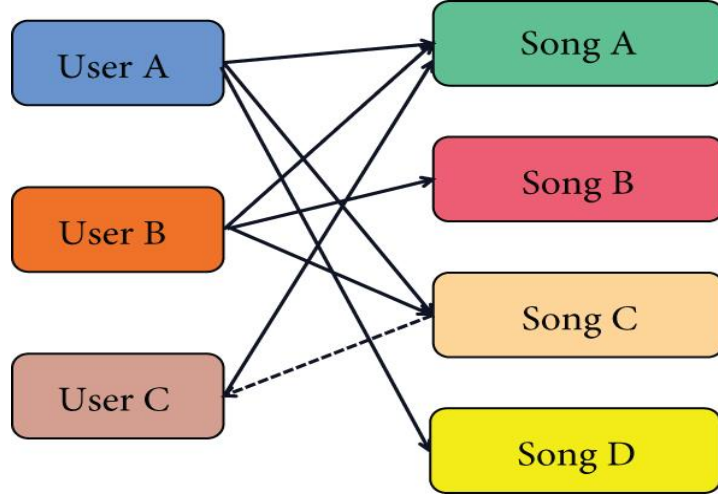


Figure 5 Collaborative Filtering approach

CF generally consists of two parts, as follows:

The user's CF suggestion calculation is utilized within the to begin with portion. The user's closest neighbor query algorithm is utilized within the conventional CF suggestion framework. As a result, it endures from destitute versatility and deficiently steadiness. The score expectation investigation is presented into the venture, and when combined with the impact of information sparsity, a adjusted conditional likelihood calculation of extend likeness is utilized to propose an optimized CF proposal calculation. The discoveries are more viable and exact, and the quality of the suggestions is progressed. Compared to the conventional CF calculation based on the closest neighbor suggestion thing, this calculation viably eases the scanty information set caused by the over issue and improves the suggestion system's suggestion quality altogether. The method can be generally partitioned into three stages, as appeared within the graph over.

(1) Modeling Information Representation concurring to Users' Scores and Compelling Measuring of the Similitude between Clients. All users' data shapes a framework, too called client thing scoring network, which is indicated as R:

$$R = \begin{bmatrix} R_{11} & R_{12} & \cdots & R_{1n} \\ R_{21} & R_{22} & \cdots & R_{2n} \\ \cdots & \cdots & \cdots & \cdots \\ R_{m1} & R_{m2} & \cdots & R_{mn} \end{bmatrix},$$

where M speaks to the number of clients within the framework, n speaks to the number of things, and the esteem of framework component R_{u1} speaks to the score of client u on thing I. The esteem of R_{u1} is for the most part in a certain esteem extend, ordinarily an numbers of 1-5, and the things that the client does not score are supplanted by 0. The bigger the R_{u1} , the

higher the client U's assessment of venture I and vice versa.

(2) Attempting to discover the closest neighbor client set, the closeness reflects the degree of qualification between two objects or two highlights, and the more conspicuous the degree of refinement, the lower the likeness. On the inverse, the humbler the degree of differentiate, the higher the closeness. There are more often than not a couple of ways to degree.

(I) Cosine Similitude Esteem within the Calculation Process. We present vector considering, which has both greatness and course. Cosine of the included point between bearings can be respected as a similitude degree and monotonically diminishes inside the included point extend according to cosine property, that's, the littler the degree esteem, the greater the closeness. The formula is as follows:

$$\text{Sim}(a, b) = \cos \frac{\vec{a} * \vec{b}}{\|\vec{a}\| * \|\vec{b}\|} = \frac{\sum_{i \in I_{ab}} R_{ai} * R_{bi}}{\sqrt{\sum_{i \in I_{ab}} R_{ai}^2 * R_{bi}^2}}$$

(II) As the title suggests, adjusted cosine similitude degree may be a adjustment based on the cosine calculation strategy, and cosine only has one lethal imperfection: the suggestion isn't accurate enough since the user's rating scale isn't taken into consideration. The adjusted cosine similitude degree strategy is based on the rule of subtracting the user's rating by calculating the user's rating on the extend. The formula is as follows:

$$\text{Sim}(a, b) = \frac{\sum_{i \in I_{ab}} (R_{ai} - \bar{R}_a) * (R_{bi} - \bar{R}_b)}{\sqrt{\sum_{i \in I_{ab}} (R_{ai} - \bar{R}_a)^2 * (R_{bi} - \bar{R}_b)^2}}$$

(III) Relationship likeness estimation is the best but most viable way to calculate the closeness among the three sorts of similitude. Pearson relationship coefficient calculates the closeness equation between target client A and a certain user B as follows:

$$\text{Sim}(a, b) = \frac{\sum_{i \in I_{ab}} (R_{ai} - \bar{R}_a) * (R_{bi} - \bar{R}_b)}{\sqrt{\sum_{i \in I_{ab}} (R_{ai} - \bar{R}_a)^2} * \sqrt{\sum_{i \in I_{ab}} (R_{bi} - \bar{R}_b)^2}}$$

(3) Producing suggestion is to at last anticipate the thing score esteem after getting the likeness from the over and prescribe a few required data, that's, to allow the best things with the most noteworthy assessment score (top-N) as the suggestion result to the target clients:

$$K = \frac{1}{\sum_{u=1}^Q \text{sim}(a, u)}$$

The CF of project-based ventures within the moment portion is totally inverse to the over thought, but the calculation is the same, suggesting that a client will favor ventures that are

comparable to those he has as of now acquired. The suggestion calculation is much quicker since this strategy does not require distinguishing neighbors. Since the number of clients in most suggestion frameworks is distant more noteworthy than the number of things, finding the significance between things is much simpler and steadier than finding the pertinence between clients, and in this way, the item-based CF calculation is way better in terms of versatility than the user-based CF calculation. The equation is as takes after.

$$P_{u,p} = \frac{\sum_{n \in N_p} \text{sim}_{p,n} \times R_{u,n}}{\sum_{n \in N_p} |\text{sim}_{p,n}|}.$$

The closeness between things can be calculated offline and spared within the database, so that it can be utilized specifically when suggesting, which spares the time of suggestion and makes strides the productivity of proposal. The impediment of project-based CF calculation is that it does not have the oddity of proposal, and it moreover needs the capacity of cross category proposal and particular suggestion.

Chapter 4 Requirement Analysis

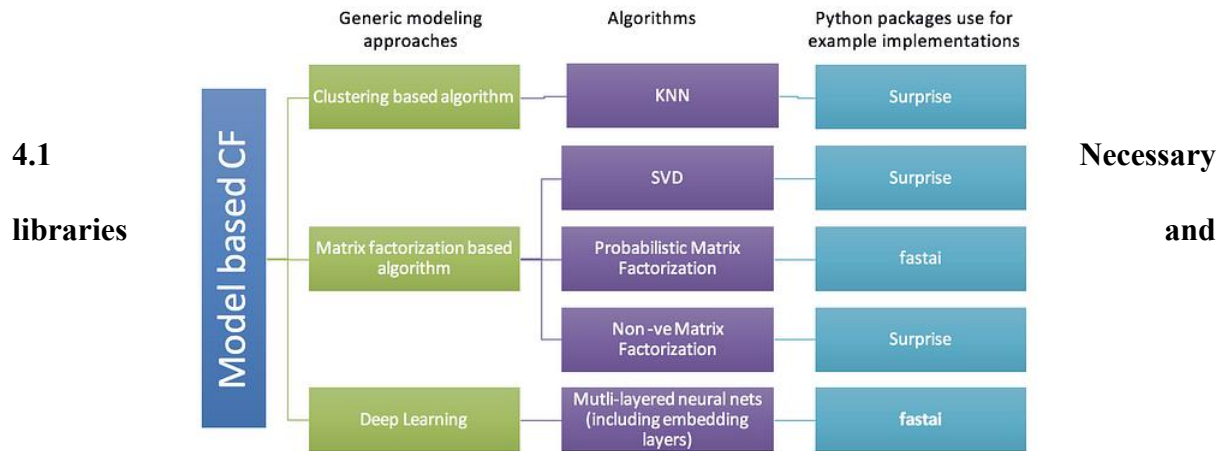


Figure 6 Libraries

Implementation

NumPy is a Python library utilized for working with arrays. It too has capacities for working within the space of Direct variable-based math, Fourier changes, and frameworks. Quick and flexible, the NumPy vectorization, ordering, and broadcasting concepts are the de-facto measures of cluster computing today. NumPy offers comprehensive scientific capacities, arbitrary number generators, direct variable-based math schedules, Fourier changes, and more. NumPy bolsters a wide run of equipment and computing stages, and plays well with dispersed, GPU, and scanty cluster libraries.

Pandas is a quick, effective, adaptable and simple to utilize open-source data examination and control tool. Pandas may be a Python bundle that gives quick, adaptable, and expressive information structures outlined to create working with "social" or "labeled" information both simple and instinctive. It points to be the elemental high-level building square for doing viable, genuine world information investigation in Python. Furthermore, it has the broader objective of getting to be the foremost capable and adaptable open-source information investigation / control instrument accessible in any dialect. It is as of now well on its way towards this objective.

Matplotlib could be a comprehensive library for making static, animated, and intelligently visualizations in Python. Matplotlib may be a plotting library for the Python programming dialect and its numerical arithmetic expansion NumPy. It gives an object-oriented API for embedding plots into applications utilizing general-purpose GUI toolkits like Tkinter, wxPython, Qt, or GTK. There's moreover a procedural "pylab" interface based on a state machine (like OpenGL), designed to closely take after that of MATLAB, in spite of the fact that it utilizes is discouraged.[3] SciPy makes utilize of Matplotlib. Pyplot may be a Matplotlib module that gives a MATLAB-like interface.[11] Matplotlib is planned to be as usable as MATLAB, with the capacity to utilize Python, and the advantage of being free and open-source.

Seaborn could be a Python information visualization library based on matplotlib. It gives a high-level interface for drawing attractive and enlightening factual graphics. Seaborn may be a library for making factual design in Python. It builds on beat of matplotlib and coordinating closely with pandas' information structures. Seaborn makes a difference our investigate and get it our information. Its plotting capacities work on data frames and clusters containing entirety datasets and inside perform the vital semantic mapping and factual accumulation to deliver instructive plots. Its dataset-oriented, declarative API lets our center on what the diverse components of our plots cruel, instead of on the points of interest of how to draw them.

Suprise is a Python scikit for building and analyzing recommender systems that deal with express rating data. Surprise could be a Python module that permits us to make and test rate expectation frameworks. It was made to closely take after the scikit-learn API, which client's commonplace with the Python machine learning biological system ought to be comfortable with. Shock incorporates a set of estimators (or expectation calculations) for assessing forecasts. Classic strategies, such as the most similarity-based calculations, as well as lattice factorization calculations like SVD and NMF, are executed.

```
1 #import libraries
2 import numpy as np # linear algebra
3 import pandas as pd # data processing, CSV file I/O (e.g. pd.read_csv)
4 import seaborn as sns
5 # plotly
6 # import plotly.plotly as py
7 from plotly.offline import init_notebook_mode, iplot, plot
8 import plotly as py
9 init_notebook_mode(connected=True) #offline modela ilgili
10 import plotly.graph_objs as go
11 # word cloud library
12 from wordcloud import WordCloud
13 # matplotlib
14 import matplotlib.pyplot as plt
15 # Input data files are available in the "../input/" directory.
16 # For example, running this (by clicking run or pressing Shift+Enter) will lis
17 import os
18 # Any results you write to the current directory are saved as output.
```

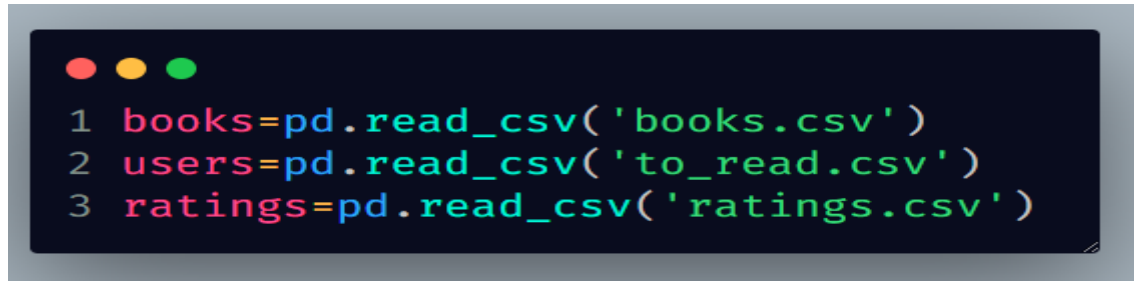
Figure 7 Import libraries

4.2 Data Collection

In this project, we chose goodbooks-10k data-set with Ten thousand books, one million ratings. Also, books marked to read, and tags. This data-set contains ratings for ten thousand popular books. As to the source, let's say that these ratings were found on the internet. Generally, there are 100 reviews for each book, although some have less - fewer - ratings. Ratings go from one

to five.

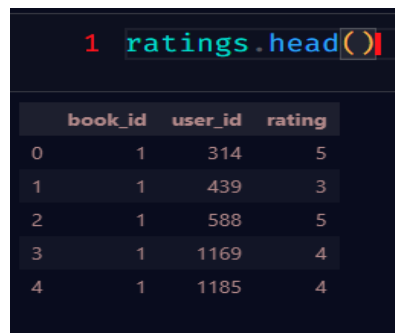
Both book IDs and user IDs are contiguous. For books, they are 1-10000, for users, 1-53424. All users have made at least two ratings. Median number of ratings per user is 8. There are also books marked to read by the users and book meta-data.



```
1 books=pd.read_csv('books.csv')
2 users=pd.read_csv('to_read.csv')
3 ratings=pd.read_csv('ratings.csv')
```

Figure 8 Read the data

Contents: ratings.csv contains ratings and looks like that:

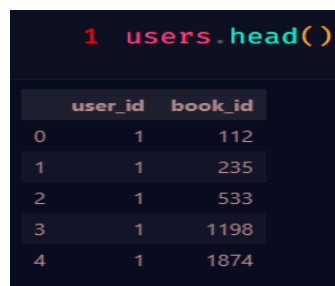


```
1 ratings.head()
```

	book_id	user_id	rating
0	1	314	5
1	1	439	3
2	1	588	5
3	1	1169	4
4	1	1185	4

Figure 9 Rating Data set

to_read.csv provides IDs of the books marked "to read" by each user, as user_id, book_id pairs.



```
1 users.head()
```

	user_id	book_id
0	1	112
1	1	235
2	1	533
3	1	1198
4	1	1874

Figure 10 User data set

books.csv has metadata for each book (good reads IDs, authors, title, average rating, etc.)

```
1 books.head()
```

✓ 0.0s

	book_id	goodreads_book_id	best_book_id	work_id	books_count	isbn	isbn13	authors	original_publication_year	original_title	...	ratings_count	work_ratings_count
0	1	2767052	2767052	2792775	272	439023483	9.780439e+12	Suzanne Collins	2008.0	The Hunger Games	...	4780653	4942365
1	2	3	3	4640799	491	439554934	9.780440e+12	J.K. Rowling, Mary GrandPré	1997.0	Harry Potter and the Philosopher's Stone	...	4602479	4800065
2	3	41865	41865	3212258	226	316015849	9.780316e+12	Stephenie Meyer	2005.0	Twilight	...	3866839	3916824
3	4	2657	2657	3275794	487	61120081	9.780061e+12	Harper Lee	1960.0	To Kill a Mockingbird	...	3198671	3340896
4	5	4671	4671	245494	1356	743273567	9.780743e+12	F. Scott Fitzgerald	1925.0	The Great Gatsby	...	2683664	2773745

5 rows x 23 columns

Figure 11 Book Data set

4.3 Information about data

Shape : The elements of the shape tuple give the lengths of the corresponding array dimensions.

```
1 print(books.shape)
2 print(users.shape)
3 print(ratings.shape)
```

```
11]
. (10000, 23)
  (912705, 2)
  (981756, 3)
```

Figure 12 Shape of the data

books.csv: Number of Rows and Columns (1000,23)

to_read.csv: Number of Rows and Columns (912705,2)

ratings.csv: Number of Rows and Columns (981756,3)

info () strategies are exceptionally valuable as they give a diagram of the information just, like the number of records display within the information, number of columns and information sort of column. It gives a diagram of what kind of information I'm managing with.

```

1 books.info()

Output exceeds the size limit. Open the full output data in a text editor
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 10000 entries, 0 to 9999
Data columns (total 23 columns):
#   Column                                Non-Null Count  Dtype
---  -
0   book_id                               10000 non-null  int64
1   goodreads_book_id                     10000 non-null  int64
2   best_book_id                           10000 non-null  int64
3   work_id                                10000 non-null  int64
4   books_count                            10000 non-null  int64
5   isbn                                    9300 non-null   object
6   isbn13                                 9415 non-null   float64
7   authors                                10000 non-null  object
8   original_publication_year              9979 non-null   float64
9   original_title                         9415 non-null   object
10  title                                  10000 non-null  object
11  language_code                          8916 non-null   object
12  average_rating                         10000 non-null  float64
13  ratings_count                           10000 non-null  int64
14  work_ratings_count                      10000 non-null  int64
15  work_text_reviews_count                 10000 non-null  int64
16  ratings_1                              10000 non-null  int64
17  ratings_2                              10000 non-null  int64
18  ratings_3                              10000 non-null  int64
19  ratings_4                              10000 non-null  int64
...
21  image_url                              10000 non-null  object
22  small_image_url                        10000 non-null  object
dtypes: float64(3), int64(13), object(7)
memory usage: 1.8+ MB

```

Figure 13 Summary statistics of books.csv

Using book.info () we came to know how many data types, rows, columns and memory. Using same approach, I came to know about user and rating also.

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 912705 entries, 0 to 912704
Data columns (total 2 columns):
#   Column    Non-Null Count  Dtype
---  -
0   user_id   912705 non-null  int64
1   book_id   912705 non-null  int64
dtypes: int64(2)
memory usage: 13.9 MB

```

Figure 14 Summary statistics of user.csv

```

<class 'pandas.core.frame.DataFrame'>
RangeIndex: 981756 entries, 0 to 981755
Data columns (total 3 columns):
#   Column    Non-Null Count  Dtype
---  -
0   book_id   981756 non-null  int64
1   user_id   981756 non-null  int64
2   rating    981756 non-null  int64
dtypes: int64(3)
memory usage: 22.5 MB

```

Figure 15 Summary statistics of rating.csv

4.4 Data Cleaning

As with about any real-life information set, we have to be done a few cleanings to begin with. When investigating the information, we taken note that for a few combinations of client and

book there are different appraisals, whereas in hypothesis there ought to as it were be one (unless clients can rate a book a few times). Moreover, for the collaborative sifting in portion II it is superior to have more evaluations per client.

4.4.1 Check Duplicates

Python is a incredible dialect for doing information investigation, fundamentally since of the incredible biological system of data-centric python bundles. Pandas is one of those bundles and makes bringing in and analyzing information much simpler.

A vital portion of Information examination is analyzing Copy Values and expelling them. Pandas copied () strategy makes a difference in analyzing copy values as it were. It returns a Boolean arrangement which is Genuine as it were for interesting components.



```
1 books.duplicated().sum()
0

1 users.duplicated().sum()
0

1 ratings.duplicated().sum()
1644
```

Figure 16 Check Duplicates

The length of the dataset is 1000, after we drop the duplicates, the length is 1000. This means there are no duplicate ratings in the book data.

4.1.2 Check Missing Values

Pandas isnull() work distinguish lost values within the given question. It returns a Boolean same-sized question demonstrating in case the values are NA. Lost values gets mapped to Genuine and non-missing esteem gets mapped to Wrong.

Pandas entirety() work return the whole of the values for the asked hub. In case the input is record pivot at that point it includes all the values in a column and rehashes the same for all the columns and returns a arrangement containing the whole of all the values in each column. It moreover gives bolster to skip the lost values whereas calculating.

```

book_id                0
goodreads_book_id      0
best_book_id           0
work_id                0
books_count            0
isbn                   700
isbn13                 585
authors                0
original_publication_year 21
original_title         585
title                  0
language_code          1084
average_rating          0
ratings_count          0
work_ratings_count     0
work_text_reviews_count 0
ratings_1              0
ratings_2              0
ratings_3              0
ratings_4              0
ratings_5              0
image_url              0
small_image_url        0
dtype: int64

```

Figure 17 books.csv Missing Values

Visualize missing values for books.csv file

Within the case of a real-world dataset, it is exceptionally common that a few values within the dataset are lost. We speak to these lost values as NaN (Not a Number) values. But to construct a great machine learning demonstrate our dataset ought to be total. That's why we utilize a few ascription procedures to supplant the NaN values with some probable values. But some time recently doing that we got to have a good understanding of how the NaN values are disseminated in our dataset.

Missingno library offers an awfully decent way to imagine the dissemination of NaN values. Missingno could be a Python library and congruous with Pandas.

On the cleared-out side of the plot, the y-axis scale ranges from 0.0 to 1.0, where 1.0 speaks to 100% information completeness. In case the bar is less than this, it shows that we have lost values inside that column.

On the right side of the plot, the scale is measured in index values. With the top right representing the maximum number of rows within the data frame. Along the top of the plot, there are a series of numbers that represent the total count of the non-null values within that column.

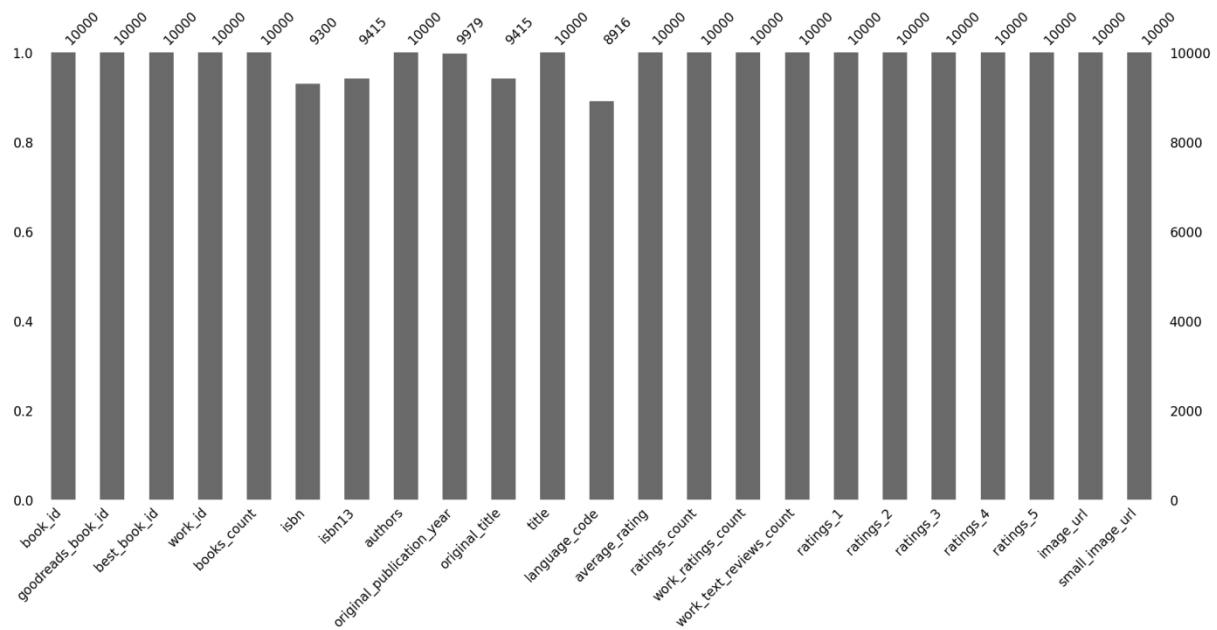


Figure 18 Visualize missing values of books.csv

In this bar Figure 18 we can see that a number of the columns (isbn, isbn13, language_code) have some of missing values. Other columns are complete and have the maximum number of values.

4.1.3 Check Outliner

```
1 np.where(books['average_rating']<1)
✓ 0.0s
(array([], dtype=int64),)

1 books['average_rating'].unique()
✓ 0.0s
array([4.34, 4.44, 3.57, 4.25, 3.89, 4.26, 3.79, 3.85, 4.24, 4.14, 3.87,
4.1, 4.11, 4.3, 4.53, 4.03, 4.46, 3.77, 4.37, 4.61, 4.54, 3.64,
3.73, 4.45, 3.84, 4.08, 3.67, 3.82, 4.12, 4.19, 3.95, 3.51, 4.23,
4.04, 4.06, 3.88, 4.07, 4.36, 3.97, 3.52, 4.29, 3.69, 3.86, 4.2,
3.7, 4.01, 3.8, 4.15, 3.94, 4.28, 4.21, 3.75, 4.17, 4.09, 3.93,
3.81, 3.96, 4.38, 4.02, 3.98, 4.22, 3.63, 3.61, 4.4, 4.27, 3.9,
3.99, 4., 3.92, 4.18, 3.37, 4.39, 4.31, 4.16, 4.42, 3.83, 4.51,
4.35, 3.68, 3.6, 3.74, 4.5, 4.05, 3.76, 4.33, 4.47, 4.55, 3.46,
3.66, 3.78, 3.4, 4.13, 4.43, 3.72, 3.42, 3.62, 3.91, 4.57, 3.28,
3.55, 3.59, 4.74, 4.59, 4.48, 4.49, 3.31, 3.65, 4.64, 3.47, 4.32,
3.56, 3.48, 3.71, 3.49, 3.41, 3.29, 4.77, 3.33, 3.22, 3.54, 4.65,
4.41, 3.45, 3.43, 4.6, 3.3, 4.72, 3.5, 3.58, 2.97, 3.32, 3.39,
3.53, 3.23, 4.56, 3.21, 3.1, 3.35, 3.25, 4.63, 2.47, 3.07, 3.36,
3.14, 3.18, 3.34, 3.11, 4.52, 4.66, 4.58, 3.44, 3.38, 3.19, 3.01,
4.62, 2.67, 4.82, 3.26, 4.73, 3.12, 2.8, 3.17, 3.16, 2.96, 2.84,
3.27, 3.13, 4.75, 2.93, 4.68, 3.24, 3.2, 4.67, 3.15, 3.02, 3.08,
3.04, 4.71, 3., 4.76, 3.05, 2.76, 2.98, 3.09])
```

Figure 19 Check Outliner in book.csv

As we know, cold start issues can be special cases in recommender systems which suggest not one or the other there's a present-day client who has not procured any of the items or not evaluated any items nor there's a advanced thing which isn't obtained by any client or not gotten any appraisals.0.

4.1.4 Data Cleaning Summary

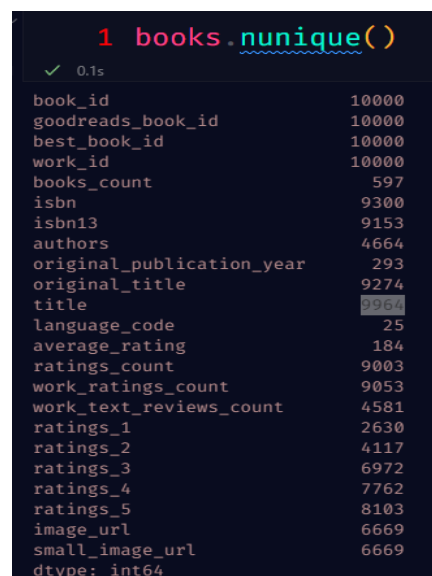
Data cleansing has played a vital part within the advancement of information administration and explanatory s. It proceeds to advance at a quick pace. Information cleansing is the act of going through all of the information in a framework and expelling or updating all fabric that's inadequate, off-base, wrongly organized, copied, or pointless. Information cleansing ordinarily involves cleaning up information that has been accumulated in one area.

Here is Data cleaning Summary

- No duplicates in this data-set.
- No outlier (cold start problems) in this data-set.
- Some missing values in this data set. But it's not big deal.

Books

The nunique() method returns the number of unique values for each column.



```
1 books.nunique()
```

book_id	10000
goodreads_book_id	10000
best_book_id	10000
work_id	10000
books_count	597
isbn	9300
isbn13	9153
authors	4664
original_publication_year	293
original_title	9274
title	9964
language_code	25
average_rating	184
ratings_count	9003
work_ratings_count	9053
work_text_reviews_count	4581
ratings_1	2630
ratings_2	4117
ratings_3	6972
ratings_4	7762
ratings_5	8103
image_url	6669
small_image_url	6669
dtype: int64	

Figure 20 unique values for book

As already indicated, there are books with multiple editions in the dataset - there are ~9300 ISBNs and ~9964 book titles.

4.5 Exploratory Data Analysis

EDA is a phenomenon under data analysis used for gaining a better understanding of data aspects like:

- main features of data
- variables and relationships that hold between them
- identifying which variables are important for our problem

describe() function generates descriptive statistics including those that summarize the central

tendency, dispersion, and shape of a dataset's distributions.

	book_id	original_publication_year	average_rating	ratings_count
count	10000.00000	9979.000000	10000.000000	1.000000e+04
mean	5000.50000	1981.987674	4.002191	5.400124e+04
std	2886.89568	152.576665	0.254427	1.573700e+05
min	1.00000	-1750.000000	2.470000	2.716000e+03
25%	2500.75000	1990.000000	3.850000	1.356875e+04
50%	5000.50000	2004.000000	4.020000	2.115550e+04
75%	7500.25000	2011.000000	4.180000	4.105350e+04
max	10000.00000	2017.000000	4.820000	4.780653e+06

Figure 21 Descriptive statistics of books.csv

From the comes about of depict () strategy, we found that the rating column is cleaned appropriately as we don't have any negative values(average_rating), invalid values, and in this case, we don't have to be work on Normalizing the information because it continuously.

4.5.1 The average reviews that given authors

Seaborn represents a formidable tool in the realm of data visualization with a particular focus on statistical graphics plotting within the Python programming language. The aforementioned feature offers aesthetically pleasing default styles and color palettes that enhance the visual appeal of statistical representations. The present software solution is constructed utilizing the matplotlib library and is moreover closely integrated with the data structures provided by pandas.

The Seaborn library's 'barplot()' function is utilized for the purpose of configuring a visual representation of data in a bar chart format. A graphical representation known as a bar plot is utilized to convey an estimate of the central tendency pertaining to a numerical variable, expressed through the height of each respective rectangle. Additionally, this method incorporates error bars to provide an indication of the associated level of uncertainty surrounding said estimate.

The mean value of an author's book ratings is determined by dividing the sum of all ratings by the total number of ratings.

.

x, y: This parameter takes names of variables in data or vector data, Inputs for plotting long form data. Here x=authors and y=work_test_reviews_count

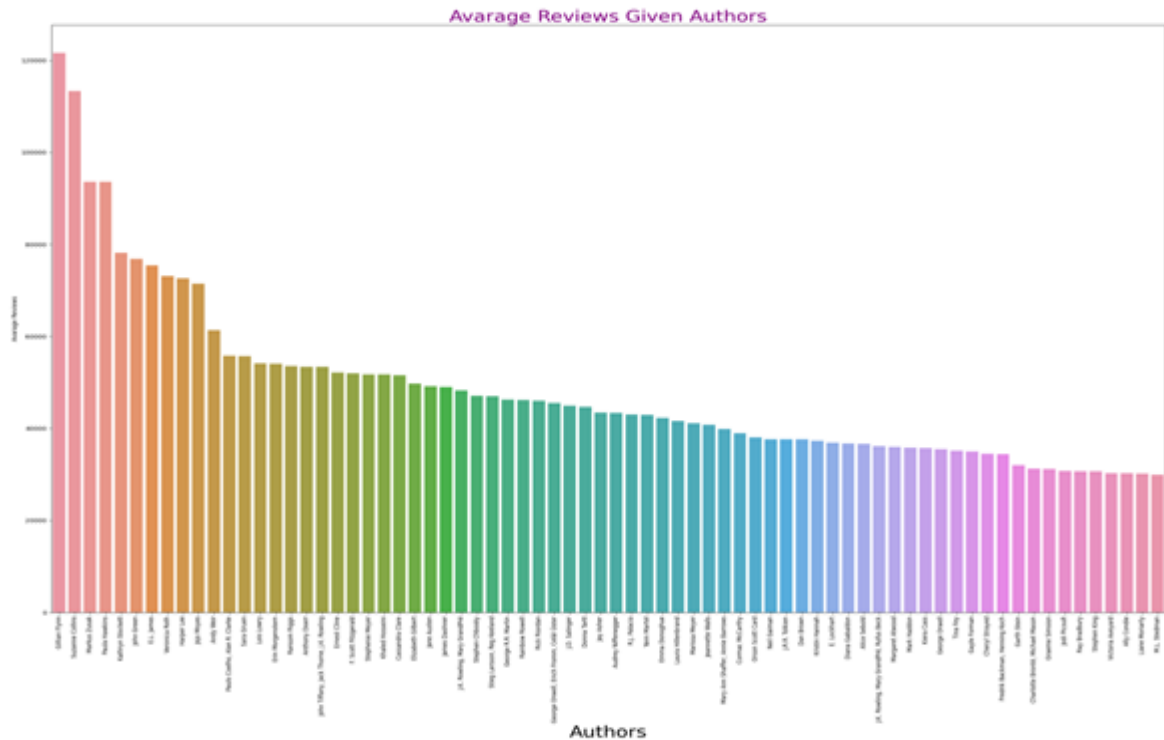


Figure 22 Visualize Authors Review

From this bar came to know Authors Gillian Flynn has most average Reviews. Also, some authors have more than 60000 average Reviews.

4.5.2 Percentage of Ratings According to Authors

Python has been widely recognized as a proficient programming language in conducting data analysis, largely attributable to its commendable ecosystem encompassing numerous data-oriented Python packages. The Python package known as Pandas is recognized for its exceptional utility in facilitating data importation and analysis. During the course of data analysis, it is frequently necessary for the user to access distinct values in a given column. This objective can be conveniently achieved by employing the `unique()` function provided by the Pandas library. The methodology employed for determining the percentage of ratings as per authors involves the use of the `unique()` function for gathering distinct authors..

```
author_list= list(data1['authors'].unique())
```

Also, we used `ratings_1` and `ratings_2` columns from `book.csv` file.

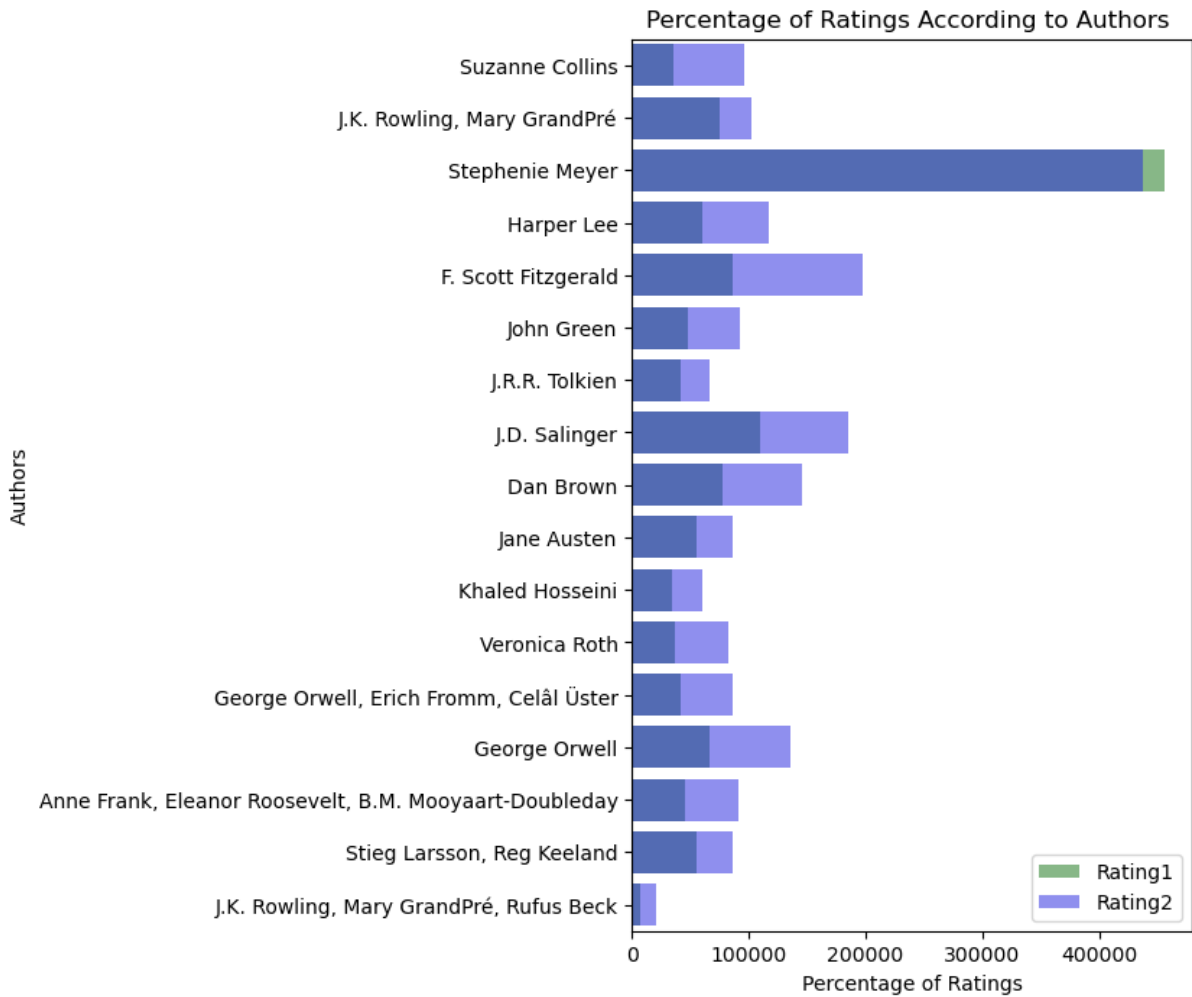


Figure 23 Visualize Percentage of Ratings

4.5.3 Rating comparisons

Here visualize rating and Average Ratings of 2004/2005/2006 for comparing data. Here we used **scatter()** function.

Scatterplots are commonly utilized in data visualization to investigate the correlation between two variables, whereby each data point is represented by a single dot on the graph. The scatter plot is drawn through the implementation of the `scatter()` method, which is an integral component of the matplotlib library. Scatter plots are frequently employed in the representation of the relationship among variables and the degree to which alterations in one variable impact the other.



Figure 24 Average Ratings

In 2003 Dan Brown had the most ratings it was more than 800k but same time average rating was below 3.6. Nicholas Sparks rating was 166.129k with 4.53 average rating.

In 2004 David Sedaris rating was 456.191 k with 3.57 average rating.

In 2005 J.k Rowling and Mary GrandPra bot ratings were 436.802k and the average rating of 3.57.

Average rating of First 10 Books

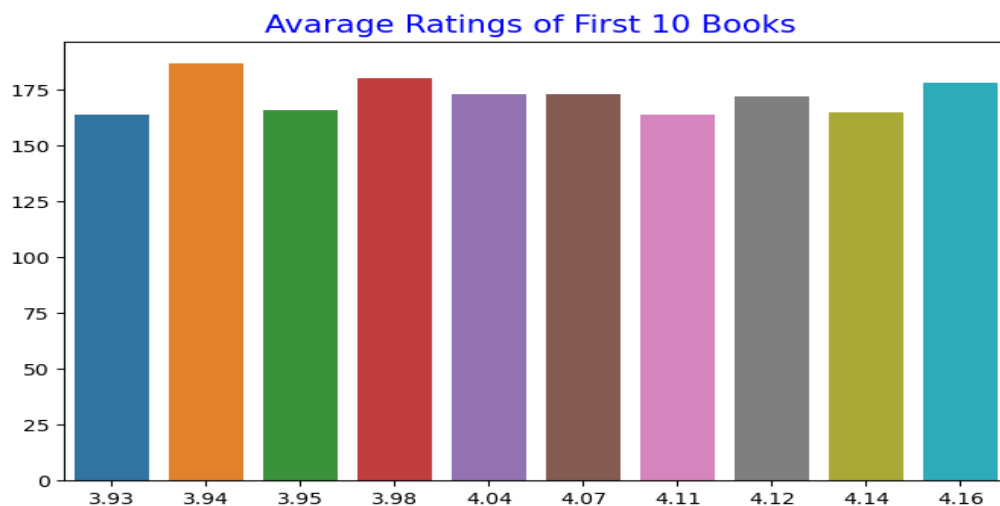


Figure 25 Average rating of First 10 Books

Here is average rating of first 10 Books. Visualize this graph with barplot() function. Where
`X=books.average_rating.value_counts().index[:10]`
`y=books.average_rating.value_counts().values[:10]`

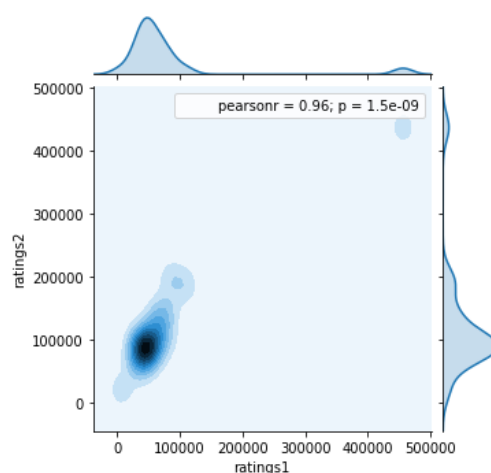
4.5.4 Correlation research

The correlation coefficient is a statistical technique utilized to ascertain the magnitude and direction of the association between two numerical variables that display a linear relationship. The Pearson correlation coefficient is classified as a descriptive statistic given its capacity to provide a concise summary of the key features of a dataset. This passage delineates the magnitude and orientation of the linear association between two quantitative variables.

The Pearson's correlation coefficient formula is:

$$r = [n(\Sigma xy) - \Sigma x \Sigma y] / \sqrt{[n(\Sigma x^2) - (\Sigma x)^2][n(\Sigma y^2) - (\Sigma y)^2]}$$

In this formula, x is the independent variable, y is the dependent variable, n is the sample size, and Σ represents a summation of all values.



- The correlation coefficient is a statistical technique utilized to ascertain the magnitude and direction of the association between two numerical variables that display a linear relationship. The Pearson correlation coefficient is classified as a descriptive statistic given its capacity to provide a concise summary of the key features of a dataset. This passage delineates the magnitude and orientation of the linear association between two quantitative variables.

Pearson correlation coefficient (r)

$r < 0.2$ very weak relationship or no correlation

$0.2 < r < 0.4$ poor correlation between

$0.4 < r < 0.6$ moderate correlation between

$0.6 < r < 0.8$ high correlation between

$r > 0.8$ is interpreted to be very high correlation.

-If the correlation coefficient is negative, there is an inverse proportion between the two variables, that means when the value of one variable increases, the other decreases. If the correlation coefficient is positive, when the value of one variable increases, the other increases as well.

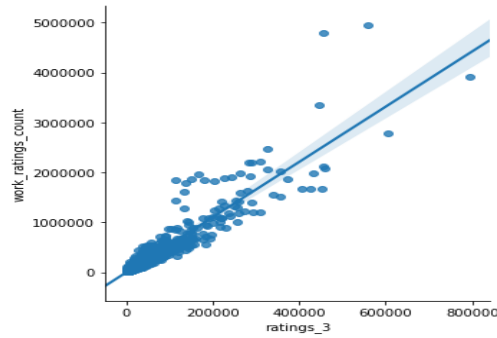


Figure 26 lmplot

Lm Plot shows the results of a linear regression within each dataset can be used in Machine learning (for instance when solving a regression problem).

4.5.5 Data Visualization

Distribution of ratings

It has been observed that individuals commonly assign favorable ratings to literature. The majority of ratings fall within the range of 3-5, whereas a negligible proportion of ratings fall within the range of 1-2.

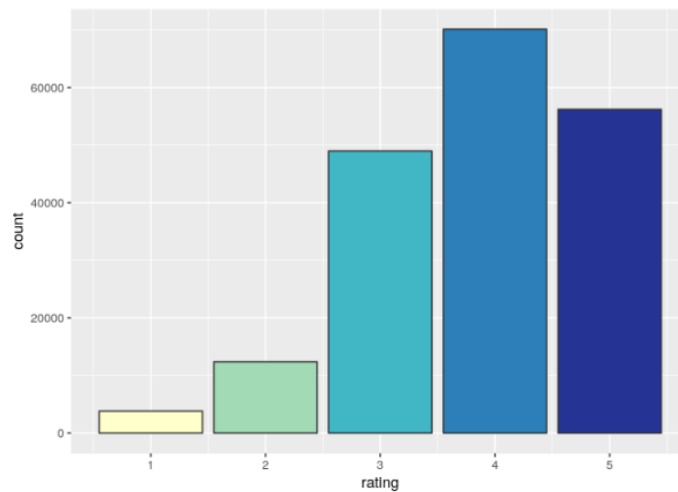


Figure 27 Distribution of ratings

Number of ratings per user

Upon filtering our ratings, it has been observed that all users possess a minimum of three ratings. Nevertheless, it is discernible that some users exhibit a multitude of evaluations. The present observation is intriguing as it offers the opportunity for subsequent investigation into potential discrepancies in book ratings between frequent and infrequent raters.

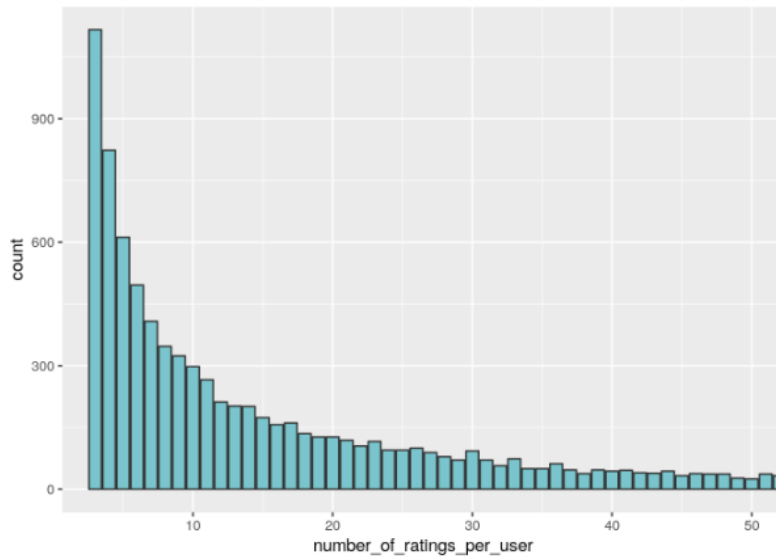


Figure 28 Number of ratings per user

Distribution of mean user rating

Individuals display distinctive tendencies in assessing literature. Within the realm of book review literature, a plethora of diverging perspectives can be observed. Certain critics appear to readily bestow a five-star rating upon literary works of only modest quality, whereas others reserve such a high evaluation for texts that flawlessly satisfy their uncompromising criteria. The illustrated tendencies are readily discernible in the visual depiction presented below. The emergence of a projection on the right-hand side of the graph can be attributed to users who have assigned a mean rating of 5, indicating their inclination towards recommending the books under review. This phenomenon may potentially serve as an indication of their genuine regard for all literary works, or alternatively, it may signify their inclination to selectively evaluate and assign high ratings solely to books that have left a favorable impact on their reading experience. The data suggests a noticeable dearth of individuals who demonstrate a persistent tendency to assign a rating of 1 to every book, among the subject population under scrutiny. These trends are considered indispensable for the efficacy of collaborative filtering in the foreseeable future and are frequently tackled through the application of a user's personalized ratings by subtracting the mean rating.

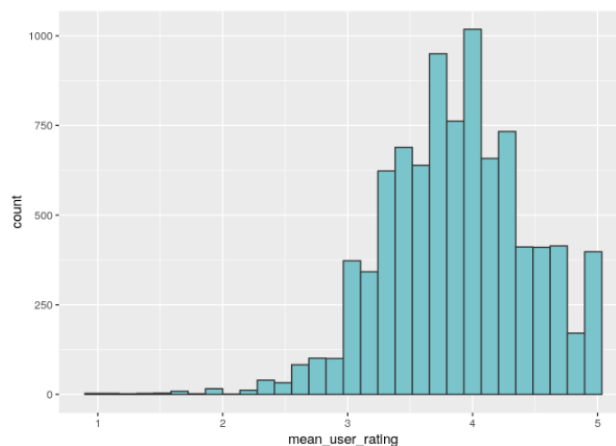


Figure 29 Distribution of mean user ratings

Number of ratings per book

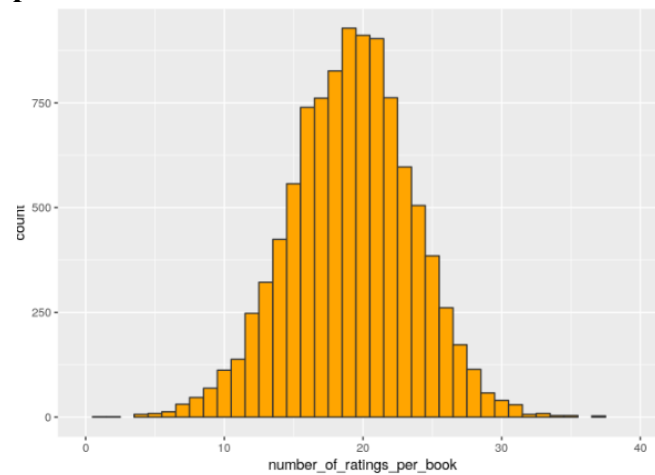


Figure 30 Number of ratings per book

We can see that in the subsetting dataset most books have around 18-20 ratings.

Top Rated Books and Their ratings

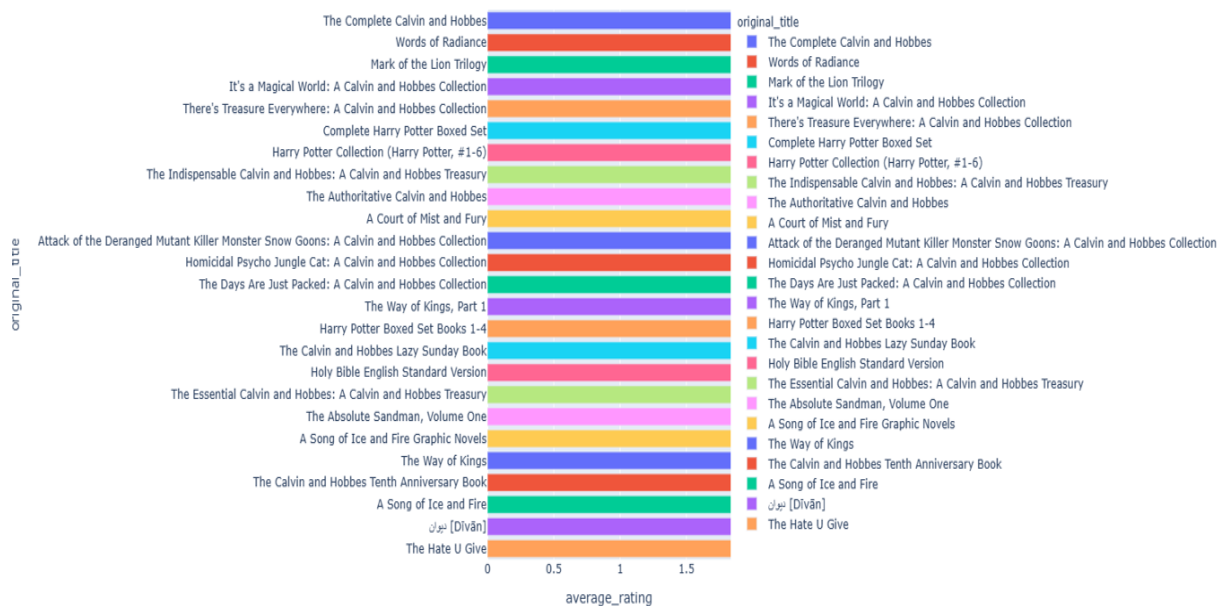


Figure 31 Top Rated Books

The complete Calvin and Hobbes is the top-rated book with 4.82 average rating.

All top-rated book average ratings from 4.62 to 4.82.

Top Popular books

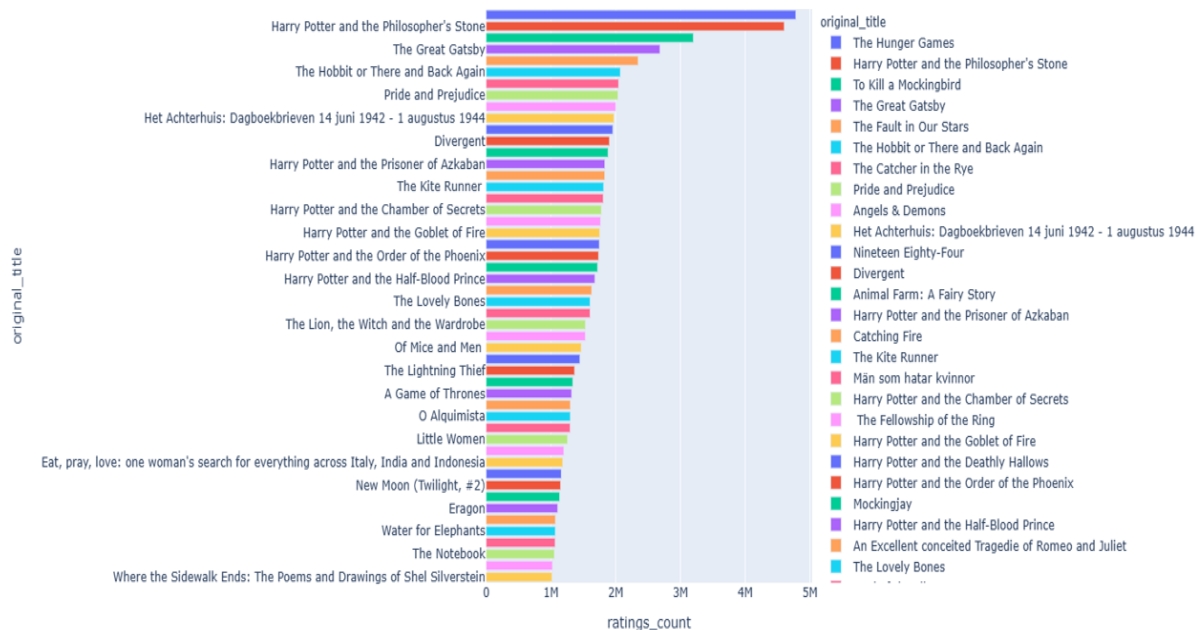


Figure 32 Popular books

Most popular book is The Hunger Games with 4.780653M rating count. Harry Potter and the Philosopher is the 2nd popular book with 4.602479M rating count.

4.6 Related Technology

4.6.1 Python

Python is the foundation dialect for AI. In any case, the ventures do vary from a conventional program venture, in this way, it is vital to jump more profound into the subject. The core of building an AI career is by learning Python – a programming dialect that's cherished by all since it is both steady and adaptable. It is presently broadly utilized for machine learning applications and why not, it has become one of the leading choices over businesses. Huge bundle of libraries/frameworks.

It is regularly a dubious assignment to select what best fits whereas running an ML or an AI calculation algorithm. It is vital to have the correct set of libraries, a well-structured environment for developers to come up with the most excellent coding arrangement. To ease their improvement timings, most engineers depend on Python libraries and systems. In a program library, there are already pre-written codes that the designers see up to illuminate programming challenges. Typically, where Python's pre-existing broad set of libraries play a major part in giving them with the set of libraries and systems to select from.

To name a few are: –

- SciPy – advanced computing
- Keras – machine learning and deep learning models

- Scikit-learn – data modeling
- NumPy – data cleaning and data manipulation
- Seaborn – data visualization
- Caffe – image processing
- Pandas – general usage for analysis of data
- PyTorch – training deep learning models
- OpenCV – image processing

It is noteworthy that the number of online repositories housing specialized Python software packages exceeds 140,000. An example of the practical implementation of Python programming is through the installation of libraries such as SciPy, NumPy, and Matplotlib, which can be seamlessly integrated into a Python-based application. The integration of packages in machine learning facilitates the identification of patterns from vast datasets by engineers specializing in artificial intelligence. The popularity of Python is sufficiently acknowledged such that it is utilized by Google to conduct web page crawling activities. Pixar employs it as a means of creating films within the domain of animation. Unexpectedly, the music streaming platform, Spotify, makes use of the programming language Python for its song recommendation system.

4.6.2 Matrix Factorization

In 1992, the concept of a recommendation system was first introduced. Through the endeavors of researchers, recommender systems have currently come to encompass a diverse range of algorithms, including Collaborative Filtering, Frequent Item Set, Cluster Analysis, Regression Analysis, Graphical Model, Restricted Boltzmann Machine, Tensor Decomposition, and Deep Learning.

Nonetheless, scholars are still grappling with the issue of insufficiency of data. When considering the data as a matrix, it can be determined that the matrix possesses a large dimensionality. However, it is important to note that only a limited number of entries within the matrix have non-zero values. The constraint of large matrix dimensions necessitates the implementation of storage-efficient strategies, as well as optimizing computational efficiency to ensure low time complexity. Typically, the size of the data matrix reaches a maximum of 105 in dimension. In this particular scenario, a time complexity of $O(n^2)$ is deemed unsatisfactory. Hence, the utilization of dimensionality reduction is imperative for resolving the aforementioned issue. Matrix factorization represents a methodology for generating hidden attributes by multiplying disparate classes of entities. Collaborative filtering is an approach that involves the utilization of matrix factorization technique aimed at establishing connections between items and users' entities. Utilizing users' ratings on shop items, a prediction model is proposed with the goal of forecasting user-rated preferences. This model enables personalized recommendations for the users based on the predictive outcomes.

Assume we have the customers' ranking table of 5 users and 5 movies, and the ratings are integers ranging from 1 to 5, the matrix is provided by the table below.

	Movie1	Movie2	Movie3	Movie4	Movie5
U1		5	4	2	1
U2	1			5	3
U3	1	4	4	1	
U4			2		2
U5	3	1	1		

Figure 33 Users' ratings table on movie

Due to incomplete participation by users, the matrix containing ratings for movies exhibits a high number of missing values, thereby resulting in a sparse matrix. Consequently, the unknown or missing values that have not been provided by the users will be imputed with a value of 0 to ensure that quantitative operations involving multiplication are appropriately implemented. As an illustration, it has been observed that a pair of individuals may assign elevated appraisal scores to a particular film if the leading roles are portrayed by their preferred actor and actress, or if the film falls within the action genre, among other factors. Based on the information presented in the table, it can be observed that users 1 and 3 have assigned elevated ratings to movies 2 and 3. Consequently, through the utilization of matrix factorization, it becomes feasible to uncover these underlying characteristics that facilitate the estimation of user ratings, reflecting the congruity between their preferences and interactions.

In a given scenario, it was observed that User 4 refrained from assigning a rating to the movie 4. We seek to ascertain whether user 4 expresses an interest in watching movie 4. The proposed approach involves identifying and analyzing the ratings provided by individuals who share analogous preferences to user 4 with respect to the movie under consideration. The goal is to generate predictions regarding the likelihood of user 4 enjoying the movie in question.

4.6.3 Euclidean Distance

The Euclidean distance is a metric defined over the Euclidean space (the physical space that surrounds us, plus or minus some dimensions). In a few words, the Euclidean distance measures the shortest path between two points in a smooth n-dimensional space.

We can define the Euclidean distance only in flat spaces: on curved surfaces, strange things happen, and straight lines are not necessarily the shortest.

The Euclidean distance is defined through the Cartesian coordinates of the points under analysis. We can think of it as the translation vector between two points. In our Euclidean distance calculator, we teach us how to calculate:

- The Euclidean distance between two or three points in spaces from one to four dimensions;
- The Euclidean distance between a point and a line in a 2D space; and
- The Euclidean distance between two parallel lines in a 2D space.

To find the Euclidean distance between two points, we need to know the coordinates of these points.

Take a generic point p . We can write its coordinates as:

$$p = (p_1, p_2, p_3, \dots)$$

The number of components depends on the dimensionality of the space.

To calculate the distance between the point p and the point q , we apply a generalized form of the Pythagorean theorem: Where n is the dimensionality of the space.

$$d(p, q) = \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \dots}$$

$$= \sqrt{\sum_{i=1}^n (q_i - p_i)^2}$$

Chapter 5 Model Building

Collaborative filtering is a standard method for product recommendations. To get the general idea consider this example:

Imagine we want to read a new book, but we don't know which one might be worth reading. We have a certain friend, with whom we have talked about some books and we typically have had quite a similar opinion on those books. It would then be a good idea to ask this friend whether he read and liked some books that we don't know yet. These would be good candidates for our next book.

What I described above is exactly the main idea of the so-called user-based collaborative filtering. It works as follows:

1. We first identify other users similar to the current user in terms of their ratings on the same set of books. For example, if we liked all the "Lord of the rings" books, we identify users which also liked those books.
2. If we found those similar users, we take their average rating of books the current user has not yet read ... So, how did those "Lord of the rings" lovers rate other books? Maybe they rated "The Hobbit" very high.
3. And recommend those books with the highest average rating to him.

Accordingly, "The Hobbit" has a high average rating and might be recommended to us.

These three steps can easily be translated into an algorithm.

However, before we can do that, we have to restructure our data. For collaborative filtering data are usually structured that each row corresponds to a user and each column corresponds to a book. This could for example look like this, for 3 users and 5 books. Note that not every user rated every book. For example, user 1 only rated book 3, while user 2 rated book 1 and book 2.

```
##          book_id
## user_id 1  2  3  4  5
##          1 NA NA  4 NA NA
##          2  2  1 NA NA NA
##          3 NA NA  3 NA  3
```


Collaborative Recommendation Model step by step

Book_id column has available in ratings and book file. Merge book and rating based on book id we get new data-frame.

Sort out each books average rating also number of ratings

Merge average rating and number of ratings based on title columns

Now filter those books has number of ratings is more than 200 and ascending based on average rating. Same time drop duplicates title. Then we get top popular book based on number of rating and average rating.

User based Filtering We sort out those users who has rating count is more than 100 books and ascending based on average rating.

Book based filtering At first, we filter based on book title and count number of rating of each book. We need only those books which has number of ratings more than 50.

Pivot table in our project for pivot table we use book title as index, user id as columns and rating as values.

User id=who rated more than 100 books after drooped duplicate values.

Ratings= Number of ratings is more than 50.

Shape=909 rows × 1100 columns

Each book can represent with help of 1100. That's means each book is a vector in 1100-dimensional space. Now we find Euclidean Distance among all books. With Euclidean distance we find the similarity.

For Euclidean Distance we use cosine_similarity() function. We get data shape(909x909) basically we calculate 909 books Euclidean Distance with 909 books.

Euclidean distance is a distance measure that can be used to calculate the distance between two points in a multidimensional space. It is calculated as the square root of the sum of the squared differences between corresponding elements of two vectors. Euclidean distance can be used to calculate similarity between two vectors by taking the inverse of the distance. The closer the vectors are, the higher their similarity score will be.

Now we make a function that will return us five books as suggestion. At first need index fetch and enumerate so we can see now all similarity with index. And ready for recommend.

Chapter 6 Experimental Results and Analysis

Here we can see all result and output from recommendation system.

We merge ratings and books based on book id:

```
1 ratings.merge(books, on='book_id')
```

✓ 0.8s

	book_id	user_id	rating	goodreads_book_id	best_book_id	work_id	books_count	isbn	isbn13	authors
0	1	314	5	2767052	2767052	2792775	272	439023483	9.780439e+12	Suzanne Collins
1	1	439	3	2767052	2767052	2792775	272	439023483	9.780439e+12	Suzanne Collins
2	1	588	5	2767052	2767052	2792775	272	439023483	9.780439e+12	Suzanne Collins
3	1	1169	4	2767052	2767052	2792775	272	439023483	9.780439e+12	Suzanne Collins
4	1	1185	4	2767052	2767052	2792775	272	439023483	9.780439e+12	Suzanne Collins
...
981751	10000	48386	5	8914	8914	11817	31	375700455	9.780376e+12	John Keegan
981752	10000	49007	4	8914	8914	11817	31	375700455	9.780376e+12	John Keegan
981753	10000	49383	5	8914	8914	11817	31	375700455	9.780376e+12	John Keegan
981754	10000	50124	5	8914	8914	11817	31	375700455	9.780376e+12	John Keegan
981755	10000	51328	1	8914	8914	11817	31	375700455	9.780376e+12	John Keegan

981756 rows × 25 columns

Figure 34 Ratings and books merge

Top 50 books with the greatest number of rating and average rating:

	title	authors	num_ratings	avg_ratings
0	The Complete Calvin and Hobbes	Bill Watterson	100	4.82
1	Harry Potter Boxed Set, Books 1-5 (Harry Potte...	J.K. Rowling, Mary GrandPré	100	4.77
2	It's a Magical World: A Calvin and Hobbes Coll...	Bill Watterson	100	4.75
3	There's Treasure Everywhere: A Calvin and Hobb...	Bill Watterson	100	4.74
4	Harry Potter Boxset (Harry Potter, #1-7)	J.K. Rowling	100	4.74
5	The Authoritative Calvin and Hobbes: A Calvin ...	Bill Watterson	100	4.73
6	Harry Potter Collection (Harry Potter, #1-6)	J.K. Rowling	100	4.73
7	The Revenge of the Baby-Sat	Bill Watterson	100	4.71
8	The Way of Kings, Part 1 (The Stormlight Archi...	Brandon Sanderson	100	4.67
9	The Harry Potter Collection 1-4 (Harry Potter,...	J.K. Rowling, Mary GrandPré	100	4.66
10	The Calvin and Hobbes Lazy Sunday Book	Bill Watterson	100	4.66
11	The Absolute Sandman, Volume One	Neil Gaiman, Mike Dringenberg, Chris Bachalo, ...	100	4.65
12	The Essential Calvin and Hobbes: A Calvin and ...	Bill Watterson	100	4.65
13	The Way of Kings (The Stormlight Archive, #1)	Brandon Sanderson	100	4.64
14	A Song of Ice and Fire (A Song of Ice and Fire...	George R.R. Martin	100	4.63
15	Jesus the Christ	James E. Talmage	100	4.63
16	A Song of Ice and Fire (A Song of Ice and Fire...	George R.R. Martin	100	4.63
17	The Sandman: King of Dreams	Alisa Kwitney, Neil Gaiman	100	4.61
18	Calvin and Hobbes	Bill Watterson, G.B. Trudeau	100	4.61
19	Harry Potter and the Chamber of Secrets: Sheet...	John Williams	100	4.61
20	Vampire Academy Collection (Vampire Academy, #...	Richelle Mead	100	4.61
21	Queen of Shadows (Throne of Glass, #4)	Sarah J. Maas	100	4.60
22	A Voice in the Wind (Mark of the Lion, #1)	Francine Rivers, Richard Ferrone	100	4.60
23	An Echo in the Darkness (Mark of the Lion, #2)	Francine Rivers	100	4.60
24	BookRags Summary: A Storm of Swords	BookRags	100	4.59
25	J.R.R. Tolkien 4-Book Boxed Set: The Hobbit an...	J.R.R. Tolkien	100	4.59
26	The Kindly Ones (The Sandman #9)	Neil Gaiman, Marc Hempel, Richard Case, D'Isra...	100	4.59
27	Percy Jackson and the Olympians (Percy Jackson...	Rick Riordan	100	4.58
28	The Annotated Sherlock Holmes: The Four Novels...	Arthur Conan Doyle, William S. Baring-Gould	100	4.58
29	Collected Fictions	Jorge Luis Borges, Andrew Hurley	100	4.58
30	Empire of Storms (Throne of Glass, #5)	Sarah J. Maas	100	4.58

Figure 35 Top 30 Books

Pivot table:

	user_id	35	173	178	274	314	368	439	588	589	725	...	52929	52956	52965	52994	53145	53173	53245	53292	53293	53366
	title																					
	Salem's Lot	0	0	0	0	0	0	0.0	0	0	0.0	...	4	0	0	0	0	0	3.5	0	0	0
	11/22/63	0	0	0	0	0	3	0.0	0	0	5.0	...	0	0	0	0	0	0	0.0	0	0	0
	1984	0	0	0	0	0	0	0.0	0	0	0.0	...	0	0	0	0	0	0	0.0	0	0	0
	1st to Die (Women's Murder Club, #1)	0	0	0	0	0	0	0.0	0	0	0.0	...	0	0	0	0	0	0	0.0	0	0	0
	2001: A Space Odyssey (Space Odyssey, #1)	0	0	0	0	0	0	0.0	0	0	0.0	...	5	0	0	0	0	0	0.0	0	5	0

	World Without End (The Kingsbridge Series, #2)	0	0	0	0	0	0	0.0	0	0	0.0	...	0	0	0	0	0	0	0.0	0	0	0
	Wuthering Heights	0	0	0	0	0	0	3.0	0	0	0.0	...	0	0	0	0	0	0	0.0	0	0	0
	Xenocide (Ender's Saga, #3)	0	0	0	0	0	0	0.0	0	4	0.0	...	0	3	0	0	0	0	0.0	0	0	0
	Year of Wonders	4	0	0	0	0	0	0.0	0	0	0.0	...	0	0	0	0	0	0	0.0	0	0	0
	Zen and the Art of Motorcycle Maintenance: An Inquiry Into Values	0	0	0	0	0	0	0.0	0	0	0.0	...	0	0	0	0	0	0	0.0	0	5	0

Figure 36 Pivot table

6.1 Test

At first, we test our model with similarity score:

Test 1 data = 2

```
[ (2, 1.0000000000000002),
  (70, 0.7039171804814047),
  (441, 0.6346678183909346),
  (209, 0.6306504551539561),
  (318, 0.6162158506256372),
  (269, 0.5902385227283753),
  (621, 0.5851413664845332),
  (567, 0.5741682154236565),
  (608, 0.5637800432574369),
  (850, 0.5603539880374004),
  (634, 0.5590274446696226),
  (349, 0.5542926227139883),
  (369, 0.5529006848068173),
  (598, 0.5518139586730748),
  (407, 0.5167640596249053),
```

Figure 37 Test where index 2

Test 2 data=209

```
[ (209, 0.9999999999999999),
  (70, 0.6865075070044274),
  (2, 0.6306504551539561),
  (621, 0.5746329253067539),
  (608, 0.5608633088058983),
  (349, 0.5491923528503386),
  (269, 0.541663001414477),
  (567, 0.5299040744613589),
  (98, 0.5283494448835366),
  (441, 0.5266917442404498),
```

Figure 38 Test where index 209

Observations: In figure 37 where we search with index 2 got similar score from Index 70,441,209. Then In figure 38 we search similarity for index 209 and we can same

similarly score from index 441,70,2 like figure 37.

Now we search with book title:

Test data: The Hotel New Hampshire, 1984, State of Wonder, A Brief History of Time

```
1 recommend('The Hotel New Hampshire')
✓ 0.0s
[['The Cider House Rules'],
 ['The World According to Garp'],
 ['A Prayer for Owen Meany'],
 ['The Bonfire of the Vanities'],
 ['The Shipping News']]
```

Test 1

```
1 recommend('1984')
✓ 0.0s
[['Animal Farm'],
 ['Pride and Prejudice'],
 ['Fahrenheit 451'],
 ['Jane Eyre'],
 ["Harry Potter and the Sorcerer's Stone (Harry Potter, #1)"]]
```

Test 2

```
1 recommend('State of Wonder')
✓ 0.0s
[["Major Pettigrew's Last Stand"],
 ['The Light Between Oceans'],
 ['Bel Canto'],
 ['Beautiful Ruins'],
 ['Olive Kitteridge']]
```

Test 3

```
1 recommend('A Brief History of Time')
✓ 0.0s
[['A Short History of Nearly Everything'],
 ['Guns, Germs, and Steel: The Fates of Human Societies'],
 ['Outliers: The Story of Success'],
 ['Do Androids Dream of Electric Sheep?'],
 ['A Study in Scarlet']]
```

Test 4

6.2 Evaluate the Collaborative recommend-er model

After the completion of the training process with the provided set of data, the model can be implemented to compute the error that arises from the predictions produced on the test data, such as root-mean-square error (RMSE). Moreover, alternative methods exist for the appraisal of the model's performance.

Actual ratings given by the users:

	user_id	35	173	178	274	314	368	439	588	589	725	...	52929	52956	52965	52994	53145	53173	53245	53292	53293	53366
	title																					
	'Salem's Lot	0	0	0	0	0	0	0.0	0	0	0.0	...	4	0	0	0	0	0	3.5	0	0	0
	11/22/63	0	0	0	0	0	3	0.0	0	0	5.0	...	0	0	0	0	0	0	0.0	0	0	0
	1984	0	0	0	0	0	0	0.0	0	0	0.0	...	0	0	0	0	0	0	0.0	0	0	0
	1st to Die (Women's Murder Club, #1)	0	0	0	0	0	0	0.0	0	0	0.0	...	0	0	0	0	0	0	0.0	0	0	0
	2001: A Space Odyssey (Space Odyssey, #1)	0	0	0	0	0	0	0.0	0	0	0.0	...	5	0	0	0	0	0	0.0	0	5	0

Figure 39 User Rating

Average ACTUAL rating for each item:

```
user_id
35      0.235424
173     0.258526
178     0.140814
274     0.265127
314     0.394939
dtype: float64
```

Figure 40 Item Rating

6.3 Bugs encountered

The data-set I initially worked with contained nearly 10000 records. While loading this data into data frame it uses to throw me memory exceeded error. To fix this issue I changed the code

```
from books = pd.read_csv(books,sep='\t') # memory error
to books = pd.read_csv(to_reads,sep='\t',low_memory=False)
```

I fixed the error and also, I reduced some records because as less data takes less time for analysis.

6.4 Limitations

The effectiveness of Collaborative Filtering as a recommendation system has been demonstrated, indicating its ability to maintain robustness despite operating at a level of granularity that is potentially suboptimal. Notwithstanding, particular contextual factors can potentially impose certain constraints.

A product that is introduced into the market without any pre-existing customer reviews or ratings is considered to exhibit a "cold start." In the absence of supplementary product information, it would be imprudent to proffer any recommendations pertaining to its procurement. The Collaborative Filtering methodology has been identified as inadequate in terms of its transparency and capacity for explanation in relation to the quantity of data that it analyzes.

To tackle such scenarios, it is recommended to employ a Hybrid recommender system. This system integrates both content-based filtering and collaborative-based filtering algorithms to produce recommendations with a heightened likelihood of effectiveness.

The utilization of Collaborative Filtering has demonstrated its efficacy in constructing sustainable recommendation systems, despite not requiring an exceedingly high level of granularity as could be potentially realized. However, in certain particular circumstances, specific limitations may be imposed.

An item that has not received any customer feedback or ratings at the outset of its introduction to the market is said to experience a "cold start" phenomenon. In the absence of supplementary product information, it would be imprudent to proffer any suggestions with respect to its acquisition. The Collaborative Filtering method is identified to possess a deficit in terms of transparency and explain ability regarding the extent of information that it handles.

To tackle the aforementioned circumstances, it is proposita to adopt a Hybrid recommender system that fuses both content-based filtering and collaborative-based filtering algorithms. Such an approach is anticipated to yield recommendations with increased probability of effectiveness.

The conceptual frameworks or theoretical constructs utilized in a given research or study, commonly referred to as the models, play a crucial role in enabling researchers to investigate and analyze complex phenomena.

Chapter 7 Conclusion

The recommender system that has been developed is anticipated to offer significant assistance to customers and e-commerce enterprises. Specifically, it will provide personalized product recommendations based on prior product usage experiences. An enhancement may be made to the recommendation engine by incorporating Deep Learning Techniques, such as inclusion of Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), and additional layers for the purpose of training the model to achieve higher levels of accuracy. In addition to the utilization of Deep Hybrid Models in recommendation systems, a variety of neural architecture components may be assimilated to construct more robust and intricate models with enhanced predictive capability. Collaborative filtering is a personalized recommendation system that derives its recommendations from the user's past behavior, without relying on any supplementary information. In the field of collaborative filtering, the recommendation of items such as books is facilitated by analyzing the similarities between the user profiles of various individuals. Based on this analysis, the system is able to identify users whose profiles are most similar to ours and subsequently suggests items that align with their respective preferences. The present approach is subject to the challenge known as the "cold-start problem." Specifically, when a novel item is introduced to the system and no prior user feedback is available, instances where said item may be preferred, notwithstanding the absence of data indicating such preferences, will be omitted from the list of suggested items. In the present system, domain knowledge is rendered unnecessary as the embeddings are acquired automatically. The proposed model has the potential to aid users in the exploration and identification of previously unknown areas of personal interest. When operating alone, the machine learning (ML) system may lack knowledge regarding a user's level of interest in a particular item. Nevertheless, the model could still make a recommendation for this item based on the fact that comparable users demonstrate an interest in it. The training of a matrix factorization model can, to a certain degree, be facilitated solely by the feedback matrix in the system. Specifically, the system does not require contextual features. In practical applications, this method serves as one of several potential candidate generators. The present study has revealed that the recommendation system implemented has demonstrated the capability to suggest items that are not commonly favored. As a result, the proposed system can be deemed advantageous to both consumers and e-commerce enterprises alike, as it can provide accurate product recommendations founded on past product interactions. The recommendation engine can be further enhanced through the integration of sophisticated Deep Learning Techniques, such as the incorporation of Recurrent Neural Networks (RNNs), Convolutional Neural Networks (CNNs), and additional layers designed to enhance the accuracy of the training process. Furthermore, the incorporation of multiple neural components in Deep Hybrid Models Based Recommendation can effectively formulate models that are both highly potent and expressive. Collaborative Filtering represents a personalized recommendation system that relies on the past behavior of the user, independent of any supplementary information. The collaborative-filtering approach recommends items, such as books, by assessing the level of similarity between a user's profile and those of other

users. This method identifies users with profiles that are most akin to ours, and subsequently suggests items for which they have displayed a preference. The proposed methodology is challenged by a commonly known issue referred to as the cold-start problem. This phenomenon arises when a new book is introduced to the system, and there exists no pre-existing data related to user preferences for it, consequently leading to its exclusion from the user's recommended reading list, even if it is a highly desirable suggestion. In this particular system, domain expertise is not deemed essential as the embedding process is acquired autonomously. The model has the capacity to facilitate users in identifying novel areas of interest. When an ML system operates in isolation, it may lack awareness of the specific user's interest in a certain item. Nonetheless, the model could still suggest the item based on the fact that comparable users have demonstrated interest in it. To a certain degree, the training of a matrix factorization model requires solely the utilization of a feedback matrix within the system. Specifically, the aforementioned system is not reliant on contextual characteristics. In practical application, this can serve as one of several potential sources of candidates. It has been observed that recommendations can be generated for items that lack popularity.

References

- [1] Francesco Ricci and Lior Rokach and Bracha Shapira, Introduction to Recommender Systems Handbook, Recommender Systems Handbook, Springer, 2011, pp. 1–35.
- [2] John S. Breese, David Heckerman, and Carl Kadie, Empirical Analysis of Predictive Algorithms for Collaborative Filtering, 1998 Archived 19 October 2013 at the Wayback Machine.
- [3] John S. Breese, David Heckerman, and Carl Kadie, Empirical Analysis of Predictive Algorithms for Collaborative Filtering, 1998 Archived 19 October 2013 at the Wayback Machine.
- [4] Collaborative Filtering: Lifeblood of The Social Web Archived 22 April 2012 at the Wayback Machine.
- [5] Ghazanfar, Mustansar Ali; Prügel-Bennett, Adam; Szedmak, Sandor (2012). "Kernel-Mapping Recommender system algorithms". *Information Sciences*. 208: 81–104. CiteSeerX 10.1.1.701.7729. doi:10.1016/j.ins.2012.04.012.

- [6] Koren, Yehuda; Bell, Robert; Volinsky, Chris (August 2009). "Matrix Factorization Techniques for Recommender Systems". *Computer*. 42 (8): 30–37. CiteSeerX 10.1.1.147.8295. doi:10.1109/MC.2009.263. S2CID 58370896.
- [7] Bi, Xuan; Qu, Annie; Wang, Junhui; Shen, Xiaotong (2017). "A group-specific recommender system". *Journal of the American Statistical Association*. 112 (519): 1344–1353. doi:10.1080/01621459.2016.1219261. S2CID 125187672.
- [8] Cao, Jian; Hu, Hengkui; Luo, Tianyan; Wang, Jia; Huang, May; Wang, Karl; Wu, Zhonghai; Zhang, Xing (2015). Distributed Design and Implementation of SVD++ Algorithm for E-commerce Personalized Recommender System. *Communications in Computer and Information Science*. Vol. 572. Springer Singapore. pp. 30–44. doi:10.1007/978-981-10-0421-6_4. ISBN 978-981-10-0420-9.
- [9] Fang, Yi; Si, Luo (27 October 2011). "Matrix co-factorization for recommendation with rich side information and implicit feedback". *Proceedings of the 2nd International Workshop on Information Heterogeneity and Fusion in Recommender Systems - Het Rec '11*. ACM. pp. 65–69. doi:10.1145/2039320.2039330. ISBN 9781450310277. S2CID 13850687.
- [10] Dacrema; Ferrari (2021). "A Troubling Analysis of Reproducibility and Progress in Recommender Systems Research". *ACM Transactions on Information Systems*. 39 (2): 39.2. arXiv:1911.07698. doi:10.1145/3434185. S2CID 208138060.
- [11] Smith, Karl (2013), *Precalculus: A Functional Approach to Graphing and Problem Solving*, Jones & Bartlett Publishers, p. 8, ISBN 978-0-7637-5177-7
- [12] Andreescu, Titu; Andrica, Dorin (2014), "3.1.1 The Distance Between Two Points", *Complex Numbers from A to ... Z* (2nd ed.), Birkhäuser, pp. 57–58, ISBN 978-0-8176-8415-0.
- [13] Bell, Robert J. T. (1914), "49. The shortest distance between two lines", *An Elementary Treatise on Coordinate Geometry of Three Dimensions* (2nd ed.), Macmillan, pp. 57–61.
- [15] Ciarlet, Philippe G. (2013), *Linear and Nonlinear Functional Analysis with Applications*, Society for Industrial and Applied Mathematics, p. 173, ISBN 978-1-61197-258-0.
- [16] Ricci, Francesco; Rokach, Lior; Shapira, Bracha (2022). "Recommender Systems: Techniques, Applications, and Challenges". In Ricci, Francesco; Rokach, Lior; Shapira, Bracha (eds.). *Recommender Systems Handbook* (3 ed.). New York: Springer. pp. 1–35. doi:10.1007/978-1-0716-2197-4_1. ISBN 978-1-0716-2196-7.
- [17] H. Chen, L. Gou, X. Zhang, C. Giles Collabseer: a search engine for collaboration discovery, in *ACM/IEEE Joint Conference on Digital Libraries (JCDL)* 2011.

Appendix

Here I insert all important coding part that used on our project.

```
1 x=ratings_wit_name.groupby('user_id').count()['rating']>100
2 readers_users=x[x].index

1 filtered_rating=ratings_wit_name[ratings_wit_name['user_id'].isin(readers_users)]

1 y=filtered_rating.groupby('title').count()['rating']>=50
2 famous_books=y[y].index

1 final_ratings=filtered_rating[filtered_rating['title'].isin(famous_books)]

1 pt=final_ratings.pivot_table(index='title',columns='user_id',values='rating',fill_value=0)

1 pt.fillna(0,inplace=True)
```

Figure 41 Coding for prepare data

1 pt

	user_id	35	173	178	274	314	368	439	588	589	725	...	52929	52956	52965	52994	53145	53173	53245	53292	53293	53366	Python
	title																						
	'Salem's Lot	0	0	0	0	0	0	0.0	0	0	0.0	...	4	0	0	0	0	0	3.5	0	0	0	
	11/22/63	0	0	0	0	0	3	0.0	0	0	5.0	...	0	0	0	0	0	0	0.0	0	0	0	
	1984	0	0	0	0	0	0	0.0	0	0	0.0	...	0	0	0	0	0	0	0.0	0	0	0	
	1st to Die (Women's Murder Club, #1)	0	0	0	0	0	0	0.0	0	0	0.0	...	0	0	0	0	0	0	0.0	0	0	0	
	2001: A Space Odyssey (Space Odyssey, #1)	0	0	0	0	0	0	0.0	0	0	0.0	...	5	0	0	0	0	0	0.0	0	5	0	
	
	World Without End (The Kingsbridge Series, #2)	0	0	0	0	0	0	0.0	0	0	0.0	...	0	0	0	0	0	0	0.0	0	0	0	
	Wuthering Heights	0	0	0	0	0	0	3.0	0	0	0.0	...	0	0	0	0	0	0	0.0	0	0	0	
	Xenocide (Ender's Saga, #3)	0	0	0	0	0	0	0.0	0	4	0.0	...	0	3	0	0	0	0	0.0	0	0	0	
	Year of Wonders	4	0	0	0	0	0	0.0	0	0	0.0	...	0	0	0	0	0	0	0.0	0	0	0	
	Zen and the Art of Motorcycle Maintenance: An Inquiry Into Values	0	0	0	0	0	0	0.0	0	0	0.0	...	0	0	0	0	0	0	0.0	0	5	0	

909 rows × 1100 columns

Figure 42 Pivot table for recommendation

```

1 from sklearn.metrics.pairwise import cosine_similarity

1 similarity_scores=cosine_similarity(pt)

1 sorted(list(enumerate(similarity_scores[209])),key=lambda x:x[1],reverse=True)

```

Output exceeds the [size limit](#). Open the full output data [in a text editor](#)

```

[(209, 0.9999999999999999),
 (70, 0.6865075070044274),
 (2, 0.6306504551539561),
 (621, 0.5746329253067539),
 (608, 0.5608633088058983),
 (349, 0.5491923528503386),
 (269, 0.541663001414477),
 (567, 0.5299040744613589),
 (98, 0.5283494448835366),
 (441, 0.5266917442404498),
 (407, 0.5249000324663803),
 (318, 0.5244135793764957),

```

Figure 43 Similarity Score

```

1 def recommend(book_name):
2     index = np.where(pt.index==book_name)[0][0]
3     similar_items = sorted(list(enumerate(similarity_scores[index])),key=lambda x:x[1],reverse=True)[1:6]
4     data = []
5     for i in similar_items:
6         item = []
7         temp_df = books[books['title'] == pt.index[i][0]]
8         item.extend(list(temp_df.drop_duplicates('title')['title'].values))
9         #item.extend(list(temp_df.drop_duplicates('title')['authors'].values))
10        #item.extend(list(temp_df.drop_duplicates('title')['image_url'].values))
11        data.append(item)
12
13    return data

```

```

1 recommend('1984')

```

```

[['Animal Farm'],
 ['Pride and Prejudice'],
 ['Fahrenheit 451'],
 ['Jane Eyre'],
 ["Harry Potter and the Sorcerer's Stone (Harry Potter, #1)"]]

```

Figure 44 Recommendation Function

Display in website:

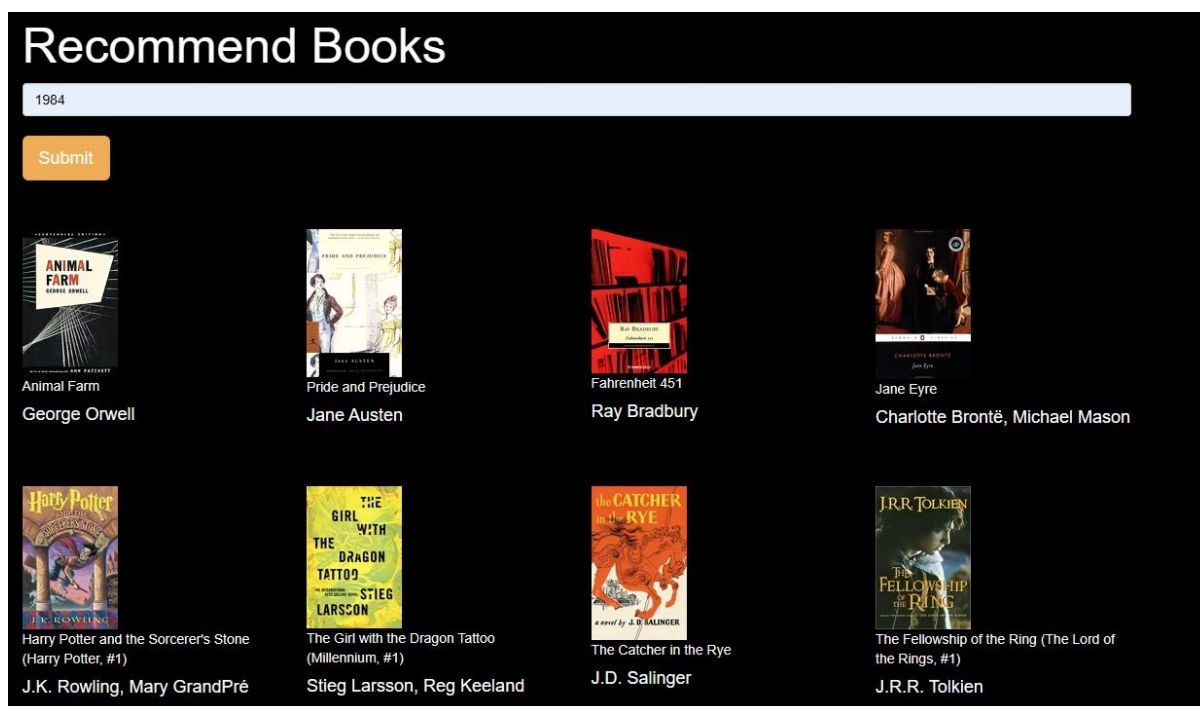


Figure 47 recommends Books

Acknowledgement

I would want to use this chance to offer my sincere gratitude and appreciation to everyone who helped me finish this thesis.

First and foremost, I want to express my gratitude to Yu Lixing, my supervisor, for her leadership, persistence, and knowledge. Their invaluable counsel, insightful criticism, and unfailing support were crucial in determining the focus and caliber of this study. My sincere gratitude goes out to them for serving as my mentors and for imparting their wisdom to me along the way.

I also want to express my profound gratitude to my parents for their everlasting support, love, and belief in me. Their unwavering encouragement, compassion, and sacrifices have been the impetus behind my successes. I will always be grateful that they were in my life and that they instilled resilience and tenacity in me.

In addition, I want to thank my pals Yeasin Arafat and Rashik Jahangir for their never-ending support, inspiration, and companionship. Their relationship has served as a constant source of encouragement and joy, and their presence has added flavor and enjoyment to this trip. I am appreciative of their assistance in both happy and difficult times.