WQD7001 PRINCIPLES OF DATA SCIENCE 1/2022/2023

BOOK RECOMMENDATION SYSTEM USING CONTENT-BASED FILTERING ALGORITHM

GROUP ASSIGNMENT 1

PDS G3(RL) – Group 1		
No.	**MATRIC ID**	**NAME**
1	22056764	Devayani A/P Balkrishnan
2	S2195613	Li Xin Qi
3	S2155659	Lim Yu Xuan
4	S2192763	Navaneeta A/P P Shanmugam

1. Project Background 3

2. Problem Statement 4

3. Research Questions 4

4. Problem Objectives 4

5. Project Scope 5

6. Literature Study / Information Gathering Analysis 5

7. Description of Methodology 8

7. 1 Obtain 8

7.2 Scrub 11

7.2.1 Data reduction 11

7.2.2 Data cleaning 11

7.2.2.1 Remove Unwanted Data 11

7.2.2.2 Handle Missing Values and Noisy Data 11

7.3 Explore 12

8. Impact of the Project on the society 14

9. References 15

10. Appendix 17

10.1 OSEMN Framework 17

10.2 Data attributes and data cleaning techniques 18

10.3 EDA 25

10.3.1 Descriptive Statistics 25

10.3.2 Book 27

10.3.3 Publication 29

10.3.4 Price 30

1. Project Background

Systems that make recommendations to users based on a variety of parameters are known as recommender systems. These systems forecast the product that customers will be most interested in and likely to buy. This project focuses on building a recommender system for books and is named “What’s Next, Shakespeare?”. The need for this product comes from the current buying trend that has evolved during the modern era where consumers would rather buy items online than going out due to its convenience. This buying pattern has further been pushed into becoming the new norm in the market environment by the COVID-19 pandemic. In addition, books were the chosen media because unlike books, mediums such as music and movies have established household names when it comes to recommenders (i.e., Spotify and Netflix).

There are multiple approaches that can be used to make recommender systems, depending on the scope of the project and the data that is accessible. Collaborative filtering uses similarities between users and items simultaneously to provide recommendations. This allows for serendipitous recommendations; that is, collaborative filtering models can recommend an item to user A based on the interests of a similar user B. Association rule mining finds interesting associations and relationships among large sets of data items. This rule shows how frequently an itemset occurs in a transaction. On the other hand, content-based filtering uses item features to recommend other items similar to what the user likes, based on their previous actions or explicit feedback. This model is advantageous as it doesn't need any data about other users, since the recommendations are specific to this user. This makes it easier to scale to a large number of users. The model can capture the specific interests of a user, and can recommend niche items that very few other users are interested in.

This project aims to build a book recommendation system through content-based filtering algorithm. This system serves the purpose of narrowing down the options available to users by predicting and making suggestions on books that they may be interested in. It can more effectively provide relevant and useful recommendations to customers along with their change in interest over time. This recommendation system would also benefit book sellers by boosting their book sales with the increase in public exposure, gaining more profits in return.

The data used to model the book recommender is obtained by conducting web scraping on Google Books. Google Books API was employed to perform full-text searches and retrieve information such as publisher, author, viewability, eBook availability, etc. The primary resources used for the data pre-processing, EDA, and modelling are Jupyter Notebook (used to deploy Python) and GitHub (used for code sharing). As for the reproducibility of the research, a recommender system's transparency may be hampered by a number of design and implementation decisions, making it nearly impossible to reproduce the stated settings and insights. These decisions span from the use of various model parameters to the way particular system components are implement (Bellogín & Said, 2021).

2. Problem Statement

Users can find books, news, movies, online courses, and research articles with the use of a personalized recommendation system. One of the issues with the current recommendation system is the inaccuracy in determining user's book preference due to vulnerability of data quality. An accurate recommendation system relies heavily on sufficient data to be able to churn out high quality data for analysis and prediction. Besides that, with the exponentially increase volume of books found online, overflowing information has pushed towards users and it becomes extremely challenging for them in finding relevant books. This has also led to problems with overwhelming options, causing trouble for users in making purchasing decisions. Moreover, some of the current recommendation systems are not built according to user preference but focused on the sellers' interests, which is to increase their book sales, resulting in redundancy and irrelevant recommendation to users. By building a book recommendation system with high quality data, it can help users by recommending the right choice according to their preference. Therefore, building a book recommendation system with high quality data benefits the user.

3. Research Questions

1. Which media does not have commonly used recommendation systems?
2. Which factors can be deemed relevant to narrow down recommendations to users?
3. Which method of obtaining data will give the widest variety of books to improve data quality?
4. Which approach will be best suited based on the type of data collected?
5. How can a balance be struck between sellers' and consumers' interest by the platform?

4. Problem Objectives

In this project, our aims are to:

a) Explore the best avenue of data collection which will result in a wide variety of books
b) Determine the key features of relevancy that will yield personalized recommendations to users
c) Evaluate different algorithms and techniques to build the content-based filtering book recommendation system

5. Project Scope

In this project, we have identified the problems of online users having difficulty in identifying relevant books with the vast e-books available online. Therefore, our goal of this proposed study is to build a recommendation system through content-based filtering algorithm in order to ease online users by predicting and suggesting the books that they may be interested in. From the bookseller's perspective, our system would be able to recommend their books to the right users, resulting in profit-gaining. We have chosen Google Books, one of the e-book platforms that is available for web scrapping. For this project, over 112,000 rows of data were obtained via web scrapping using API. The scope of this study has limited English language books as default language.

6. Literature Study / Information Gathering Analysis

Most common methods proposed earlier by researchers when building a book recommendation system are collaborative filtering method, content-based filtering and association rule mining. The techniques evolved into deep learning models in recent years.

Khan et al. (2017) proposed online recommendation system using collaborative filtering. The researchers applied user-based and item-based approaches in collaborative filtering and found out some limitations, for example. data sparsity and data scalability which will lead to data inaccuracy. Data sparsity occurs when items have no ratings and data. These limitations are open for future research to improve data accuracy.

In the studies done by Tian et al. (2019), a personalized recommendation system was built for library of Inner Mongolia University of Technology for better allocation of library resources. With the combination of collaborative filtering and content-based filtering methods, they can overcome the data sparsity and cold start problem. The performance was evaluated by comparing content-based and collaborative filtering separately with the combination of both methods using the precision evaluation criteria. K-means clustering was used to reduce data sparsity. It is concluded that hybrid algorithm improved the efficiency and quality of the recommendation system.

Tewari & Priyanka (2014) uses collaborative and association rule mining algorithm in their book recommendation system that were designed for college students. The mathematical formula is derived from book price, publisher’s name and predicted book rating to get book recommendation score. Top five books with the highest scores will be recommended. The research is concluded that the recommendation system helps students in finding books of their price range and preferred publisher.

Tewari et al. (2014) proposed a book recommendation system to readers that suits their interests with the combination features of content-based filtering, collaborative filtering and association rule mining. They mainly focused on the quality of the books by using collaborative filtering that includes user’s opinion as content-based filtering itself is unable to differentiate between good and bad articles when both are using same terminology. Association rule finds the correlation between items in large dataset using market basket analysis. Content based filtering is used to find out similar books based on user's purchasing history; collaborative filtering to get the rating order of recommendation; and association rule on transaction database to anticipate book recommended. It was concluded that with these parameters, the model can predict buyer’s interest and recommend books accordingly. Similarly, Mathew et al. (2016) proposed the same methods with the additional inclusion of keyword-based filtering for providing efficient recommendation to users.

The paper “Book Recommendation System using Opinion Mining Technique” written by Sohail et al. (2013) presents a recommendation technique base that proposes top ranked books on the discipline of computer science. The study applies user review to classify features like occurrence, helpfulness, material, availability, irrelevancy and price. The weights are given to each feature, and it is assigned according to the feature importance and user requirements. Books with the maximum normalized scores are sorted in a descending order of scores and the top ten ranked books of a specific topic is attained. It is concluded that the proposed technique helps customers to seek the best books available in the market and can be improved by automating the recommendation system.

Hou (2022) used deep learning model to build a personalized book recommendation algorithm for university library. The personalized book recommendation algorithm has been built according to the characteristics and laws of user savings in university library. To improve the deep autoencoder (DAE), the long short-term memory network (LSTM) was first used. This has been done to make sure that the model can extract the temporal feature of the data. Then, to obtain the book recommendation result of the current user, the SoftMax function was used. This method recommends future book borrowings according to the user’s borrowing history. The actual library lending data was used for this project. It was found that the current method has higher accuracy compared with some other existing recommendation systems.

On the article titled “Book Recommendation for Library Automation Use in School Libraries by Multi Features of Support Vector Machine” by Puritat et al., (2021), they have presented the book recommendation algorithm using support vector machine (SVM). Their aim was to increase the satisfaction of user and reduce the task of librarians by developing the open-source library automation implementation with a recommendation system. They used both the qualitative and quantitative approach for this research. Sixty-four students of Banpasao

Chiang Mai school were selected for qualitative approach. The web monitoring and precision measures were used for quantitative approach. There was a positive response from the system. They have concluded that this system is preferred for use at small libraries and for library automation in the Thai language.

Sarma et al., (2021) used the clustering algorithm to increase the predicting capability of a recommendation system. This study offered a mechanism for internet users to score books using clustering and then find novels that are like those books to recommend new books. To measure the distance the K-means cosine distance function was used and to find the similarity the cosine similarity was used. The model's performance was measured using the Sensitivity, Specificity and F1 score. It was found that the model can remove boring books from the recommendation list. Mostly all the datasets stayed close to the diagonal ideal classifier line in this study.

Ijaz (2020) used machine learning for book recommendation system. The main purpose of this work is to construct a book recommendation using collaborative item. Two machine learning algorithms which are the K-NN, and matrix factorization were used by the author. The data was obtained from the official Kaggle website from BX Books. The root mean square error (RMSE) was used to calculate the accuracy of the prediction. It has been observed that the proposed device's RMSE cost meets the same fees as the current approach, but with fewer commenters. The RMSE value of the proposed system is 0.3.

A subject based book recommendation platform which is built using the convolutional neural network was built by Wadikar et al., (2020). Two approaches were used which are recommendation using text processing and similarity measures and recommendation using image classification. For the recommendation using text processing and similarity measures, the data were web scraped from websites like Amazon and Flipkart whereas for recommendation using image classification, 987 book cover images were web scraped from Amazon. Different parameters such as ratings, book name, book cover image, price and many more were used in this work. It was concluded that the implementation was successful and has able to find the similar books from both Amazon and Flipkart using cosine similarity.

7. Description of Methodology

Note: All figures and tables are shown in appendix.

There are several commonly used data science methodologies to describe the end-to-end flow of a data science project, for instance CRISP-DM, OSEMN, TDSP etc. It is essential in forming a concrete research problem, gathering feedbacks from different teams after model deployment as well as extending the project beyond data science to business field. In this project, the OSEMN framework is applied, and each step of the data science project lifecycle is discussed thoroughly. Figure 1 shows the OSEMN framework, and the details of each stage in the framework (Lau, 2019). The first 3 stages of the OSEMN framework, which include Obtain, Scrub and Explore, is probed in this project phase, and the last 2 stages (Modelling and Interpret) will be further discussed in the next phase.

Python is used as the main programming language to construct the book recommendation system, while other programming language and tools for building a web application like Flask, Django, HTML and CSS may be considered in the next phase of the project. Until the model stage, Jupyter Notebook will be utilized to deploy Python, and Visual Studio Code will be used for the last stage, which is the interpretation of the OSEMN framework.

7. 1 Obtain

Qualitative and quantitative data, including book title, genre, author, ratings, price, etc., are needed to build a book recommendation system. There are plenty of available data sources on book data, and different methods can be used to extract such data, for instance downloading organized data from Kaggle which typically stored in Comma Separated Values (CSV) or Tab Separated Value (TSV), scrape data from online bookstore like Amazon Books, eBay Books, Google Play Books and Kinokuniya using web scraping tools or crawl data from bookstore Web Application Programming Interface (Web API).

Initially, multiple online stores were scraped for book data prior to deciding on the best option. Some bookstores (such as Kinokuniya and Amazon) did not allow for scraping, while others did not yield enough information (Book Depository and BookXcess). Thus, in this project, the book data is obtained from Google Books, a platform maintained by Google, which enables users to conduct web searches across millions of books and magazines published in numerous languages, including rare, out-of-print, and unobtainable volumes outside of the library system. It is ideal to use Google Books, which employs the Google search engine, as the data source for developing the book recommendation system. Google is the most used and reliable search engine due to its dependability and relevant search results. Apart from that, it contains countless of public domain books which are free to access, and preview option is allowed in cases where the book is out of copyright or is given permission by the publisher. These features not only provide the opportunity for book lovers to access books for free and at affordable price, but also encourage individuals to read by easily browsing books or magazines in their area of interest and millions of them are at their pick. Google Books is chosen as the project data source because of its wide selection of books and extensive set of features, and it also unquestionably satisfies the project's goals, which is to recommend books as of reader interest and encourage the reading habit.

There are numerous ways to fetch data from Google Books, including web scraping from Google Books and data crawling from Google Books API (also known as Books API). The process of web scraping from Google Books entails gathering a sizable number of book URLs, sending GET requests to these URLs, using an element locator to spot the necessary book data in the HTML source, gaining the data from the Google Books website, and finally saving the data in a structured format. In contrast, getting data using the Books API is as easy as sending a GET request and thus can result to the acquisition of more detailed book information in an organised way. In a nutshell, the Books API method is recommended over web scraping for getting the book data. The Books API enables finding and browsing books on Google Books as well as retrieving comprehensive information about a book, including title, author, description, price, availability etc. The volume collection and bookshelf collection are the two types of collections that the Books API data model supports. To clarify, a volume is information about a book or magazine that is stored in Google Books, whereas a bookshelf is where a user's volumes are stored. According to the Books API developers guide, only books that meet the requirements of the search queries can be accessed in a volume collection, and bookshelves must always be referenced in the context of a single user's library. Hence, only the volume collection data model is utilized to gather the book information to build the content-based book recommendation system.

According to the Books API documentation, no authentication is needed to perform a search via the GET request, hence in this project, no API key or OAuth 2.0 token is required to extract the book information. More detail on the Books API's functioning is presented below.

1. A HTTP GET request with the following format is delivered to the globally unique Uniform Resource Identifier (URI) to call the Books API and request resource of the book information in the volume collection:
   https://www.googleapis.com/books/v1/volumes?q=search+term
   a. q: to input specific keywords to search for volumes
   b. term: to refine the search results by applying filter in certain fields

2. If the GET request is valid, the Books API will make a call to the Google Books server.
3. The server will then perform the request and send a response to the Books API with the requested information
4. The Books API will finally return all information about the resource requested and translated into human-readable JavaScript Object Notation (JSON) format.

The detailed steps for using the Books API to acquire data are explained below.

1. A list of book genre is predefined before sending any request to the Books API. For instance, action, adventure, romance, art etc. is included in the book genre list.
2. An empty dictionary is created for storing the book data later.
3. With the book genre list, an URI with following format is defined:
   https://www.googleapis.com/books/v1/volumes?q='+genre+'+book&printType=books&langRestrict=en&startIndex='+str(idx)+'&maxResults=40'
   Note: text in red indicates the variable defined, the value will change by iterating for loop (discussed more in next step)
   a. *q='+genre+'+book*
   inputted the genre (genre in the books genre list) and book keyword to search for volumes in the specific genre
   b. *printType=books*
   returned only books
   c. *langRestrict=en*
   restricted the volumes returned to English language
   d. *startIndex='+str(idx)+'&maxResults=40*
   paginated the book information by setting the startIndex and maxResults returned, the maxResults for each request is 40
4. A nested for loop is used to iterate over the genre in the genre list and increment the start index in the URI. As a result, the genre and startIndex elements in the URI will change, allowing for the sending of GET requests to that particular URI and the retrieval of book information for each genre from each page of Google Books.
5. The information returned is stored in a dictionary created in step 2, with genre as the key, and book information (list data type) as value.
6. The book information for each genre is combined by merging the book data for each genre into a master list, then transform the master list into a dataframe.

The book dataset obtained consists of 60 attributes and 112,833 records, each attribute is described thoroughly in the table 1.

7.2 Scrub

After obtaining the data, the next stage is Scrub, where the data cleaning process is involved to ensure that the data is accurate and reliable for analysis purpose, therefore smart business decisions and successful real-life application. The data in real life is often rogue data, which is inaccurate, corrupted, improperly formatted, duplicate, or incomplete, thus data cleaning is fit in place to detect these data, then correct or remove them.

7.2.1 Data reduction

The goal of data reduction is to lessen the number of records and attributes in the dataset, thus results in a smaller representation of the data whilst maintaining the outcomes. To achieve this, the attributes which are appropriate, relevant, and required for analysis and modelling are selected. Out of the 60 attributes in the dataset, 33 attributes are chosen, while the remaining 27 attributes are removed from the dataset. The full list of chosen attributes is shown in the table 1 in appendix (9.2 Data Attributes Summary).

7.2.2 Data cleaning

7.2.2.1 Remove Unwanted Data

The irrelevant and duplicate data are redundant; hence deletion of these data is needed not only to ensure the efficiency of training model, but more importantly increase the accuracy of the model. In this project, the id attribute, which is the unique identifier of a volume in Google Books, is used to check if there are any duplicate books in the dataset. As a result, 88,513 books are unique, hence the remaining 24,320 duplicate books are removed, with the first record of the book is remained. The records with empty "volumeInfo.title" are deleted, and only English books are selected, resulting 88,321 books left to be cleaned.

7.2.2.2 Handle Missing Values and Noisy Data

Most of the machine learning model will failed due to missing values, therefore it is essential to handle missing values using the suitable method to prevent bias in the model. Noisy data indicates the data which cannot be understood and interpreted correctly by the machine, such as unknown encoding, inconsistent data, inappropriate formats, unstructured text etc., and these data will seriously impact the outcome of data analysis.

The techniques used to handle the dirty data in each attribute of the dataset is described thoroughly in the table 2. Upon checking, missing values are found in 16 attributes in the dataset, and different imputation method is applied to deal with different types of missing values, such as Missing completely at random (MCAR), Missing at random (MAR) and Missing not at random (MNAR).

7.3 Explore

Exploratory data analysis (EDA) is performed in this stage to analyse the dataset by summarizing their characteristics using visualization tools. In summary, the cleaned dataset consists of 88,321 records and 36 attributes in total, where 6 attributes are numerical, and the remaining are all categorical data, as shown in the table 3. There are no duplicates and missing values in the dataset as it is either removed, imputed, or replaced with other values in the scrubbing stage.

Table 4 displayed the descriptive statistics of the numerical attributes, including the count, mean, standard deviation, minimum, maximum and quartiles, which can then be used to determine the distribution of these numerical attributes. Based on the table, it is observed that the record counts are same and no negative values for all numerical attributes. Apart from that, it is noticeable that the minimum, and the 25, 50 and 75 quartiles values in 4 attributes are all 0, showing that at least 75% of the values in these attributes are 0. This is sensible, as most of the books are not for sale and have no ratings. Also, the difference between the 75 percentiles and the maximum values is huge for all attributes except the “volumeInfo.publishedYear”, which is an indicator that these attributes have outliers. Hence, box plots for each numerical attribute are drawn as shown in Figure 7 further assured that the outliers exist in these attributes. The standard deviation of the “volumeInfo.pageCount” attribute is the highest among all, however it may because of the values in the attribute are higher than other attributes. Thus, the coefficient of variation (CV) is calculated to determine the extent of variability in relation to the mean of the attribute, where higher CV indicates greater dispersion. The results showed that the CV of “volumeInfo.pageCount” attribute is still lower than the 4 attributes with 0 in minimum and quartiles.

A subset of categorical variables in the book dataset is chosen to perform EDA, including book categories, publishers, authors, viewability, accessibility, saleability, and e-book availability. Figure 16 shows the top 15 most common books in the dataset. According to the bar chart, fiction is the most common category with 12,346 books, followed by Juvenile fiction with a massive difference of 7,701 books as compared to fiction. In contrast, there are not much difference in the number of books for the remaining categories. It is also noted that juvenile fiction is also a type of fiction, albeit for a younger audience, thus can conclude that the fiction books hold at least 19% of the entire dataset. According to the top 15 publishers bar chart (Figure 20), it is observed that Routledge published the most, with 2,324 books, followed by Hachette UK and Simon and Schuster with 1,844 and 1,700 books published, respectively. Based on publishers with most ratings count (Figure 21), Saga DLX Ed Hc and National Geographic Books outperformed the other publishers with a high rating of 4.5 and 3.93 ratings respectively. The top 15 author with most published books did not show meaningful information as most of the authors in the dataset are publishers.

Based on the histogram of books publication year (Figure 19), it is clearly seen that most books published after the year 2000, and the trend reached at the peak around 2010s. As shown in Figure 2, the page count of the books is varied in this dataset, still there are a significant number of books fall between 300 and 400 pages as seen in the histogram of the books page count (Figure 15). Almost 80% of the books offered in Google Books are not for sale, whereas majority of the saleable books are sold with less or equal to RM100 as observed from the bar chart of price range of books for sale (Figure 24). Hence, a histogram is drawn to understand the distribution of the books with price ranged between RM1 and RM100 (Figure 25), and as of results, most books are sold at price ranged between RM20 and RM40, which are affordable to most consumers. Looking into the ratings aspect of the books (Figure 17), it is noticed that 'Saga' has the highest rating count among all the books, with the rating count at least 2.5x more than the other books. There is no significant difference in ratings count if compared between the rest of the books. Out of the top 15 books, it is observed that 'Unbroken' and 'Saga' have the highest ratings, i.e., 4.5, followed by 'The Immortal Life of Henrietta Lacks', 'Heaven is for Real', 'Where the Wild Things are', 'The Giving Tree' and 'Crazy Love/Forgotten God 2 in 1 Custom Project' which scored 4.0 (Figure 18). Since the number of ratings is taken into consideration, hence no books in the top 15 rated 5.0.

The viewability of a book is important for customers to know if they can either have full access or segment view to the books, and therefore help them to decide if it is worth to purchase. Based on the pie chart (Figure 11), roughly 38% of the books are allowed to preview, and 2% of them are available as full public domain, which can be legally used or referenced without permission. Thus, all books under public domain have full viewing access available online. Looking deeper into the non-public domain books which allowed preview (Figure 10), majority of the books, around 99% of them only provide partial preview. The book accessibility describes the various formats designed to cater with different needs of books lover, for instance e-Pub and PDF access increase consumer’s ability to obtain the books by giving them more avenues to read it. Nearly 45% of the books provide at least one alternative option, either PDF or ePub for consumers to access its digital copy while the eBook availability is only 24%, which indicates that not all books with PDF and ePub versions are considered as eBooks (Figure 12).

The key features of the dataset will be determined and discussed further in the next phase of the project, then applied in the recommendation system.

8. Impact of the Project on the society

A significant problem of information overload that prevents timely access to things of interest on the Internet has been caused by the exponential development in the volume of digital information available and the number of Internet users. This issue has been largely resolved by information retrieval systems like Google, DevilFinder, and Altavista, but prioritisation and personalization (where a system matches accessible content to a user's interests and preferences) of information were lacking (Isinkaye et al., 2015). As a result, recommender systems are more in demand than ever. By selecting important information fragments from a large volume of dynamically created material based on the user's choices, interests, or observed behaviour about the item, recommender systems are information filtering systems that address the issue of information overload. Based on the user's profile and a set of criteria, recommender systems can determine whether a specific user will favour an item or not.

Both service providers and customers can benefit from recommender systems. They lower the transaction costs associated with locating and choosing products in an online buying setting. It has been shown that recommendation systems enhance the effectiveness and process of decision-making. Because they are efficient ways to sell more things, recommender systems increase revenues in an e-commerce environment. Recommender systems in libraries assist users by enabling them to go beyond catalogue searches. Therefore, it cannot be overstated how important it is to deploy effective and accurate recommendation processes within a system that will offer users reliable and relevant recommendations.

9. References

Bellogín, A., & Said, A. (2021). Improving accountability in recommender systems research through reproducibility. *User Modeling and User-Adapted Interaction*, *31*(5), 941–977. https://doi.org/10.1007/s11257-021-09302-x

Google. (n.d.-a). *About Google Books – Google Books*. Www.google.com. https://www.google.com/intl/en/googlebooks/about/index.html

Google. (n.d.-b). Class: Google::Apis::BooksV1::Volume. Retrieved December 8, 2022, from https://googleapis.dev/ruby/google-api-client/v0.40.0/Google/Apis/BooksV1/Volume

Google. (n.d.-c). *Collaborative Filtering | Recommendation Systems*. Google Developers. https://developers.google.com/machine-learning/recommendation/collaborative/basics

Google. (n.d.-d). *Content-based Filtering | Recommendation Systems*. Google Developers. https://developers.google.com/machine-learning/recommendation/content-based/basics

Google. (n.d.-e). *Volume | Google Books APIs*. Google Developers. https://developers.google.com/books/docs/v1/reference/volumes

Google. (2016, March). *Getting Started | Google Books APIs | Google Developers*. Google Developers. https://developers.google.com/books/docs/v1/getting_started

Hou, D. (2022). Personalized Book Recommendation Algorithm for University Library Based on Deep Learning Models. *Journal of Sensors*, *2022*, 1–6. https://doi.org/10.1155/2022/3087623

Ijaz, F. (2020). Book Recommendation System using Machine learning. *Easychair.org*. https://easychair.org/publications/preprint/Ks4N

Isinkaye, F. O., Folajimi, Y. O., & Ojokoh, B. A. (2015). Recommendation systems: Principles, methods and evaluation. *Egyptian Informatics Journal*, *16*(3), 261–273. https://doi.org/10.1016/j.eij.2015.06.005

Khan, B. M., Mansha, A., Khan, F. H., & Bashir, S. (2017). Collaborative filtering based online recommendation systems: A survey. *2017 International Conference on Information and Communication Technologies (ICICT)*. https://doi.org/10.1109/icict.2017.8320176

Lau, D. C. H. (2019, January 10). *5 Steps of a Data Science Project Lifecycle*. Medium. https://towardsdatascience.com/5-steps-of-a-data-science-project-lifecycle-26c50372b492

Mathew, P., Kuriakose, B., & Hegde, V. (2016). Book Recommendation System through content based and collaborative filtering method. *2016 International Conference on Data Mining and Advanced Computing (SAPIENCE)*. https://doi.org/10.1109/sapience.2016.7684166

Puritat, K., Julrode, P., Ariya, P., Sangamuang, S., & Intawong, K. (2021). Book Recommendation for Library Automation Use in School Libraries by Multi Features of Support Vector Machine. *International Journal of Advanced Computer Science and Applications*, *12*(4). https://doi.org/10.14569/ijacsa.2021.0120426

Sarma, D., Mittra, T., & Shahadat, M. (2021). Personalized Book Recommendation System using Machine Learning Algorithm. *International Journal of Advanced Computer Science and Applications*, *12*(1). https://doi.org/10.14569/ijacsa.2021.0120126

Sohail, S. S., Siddiqui, J., & Ali, R. (2013, August 1). *Book recommendation system using opinion mining technique*. IEEE Xplore. https://doi.org/10.1109/ICACCI.2013.6637421

Tewari, A. S., Kumar, A., & Barman, A. G. (2014, February 1). *Book recommendation system based on combine features of content based filtering, collaborative filtering and association rule mining*. IEEE Xplore. https://doi.org/10.1109/IAdCC.2014.6779375

Tewari, A. S., & Priyanka, K. (2014). Book recommendation system based on collaborative filtering and association rule mining for college students. *2014 International Conference on Contemporary Computing and Informatics (IC3I)*. https://doi.org/10.1109/ic3i.2014.7019651

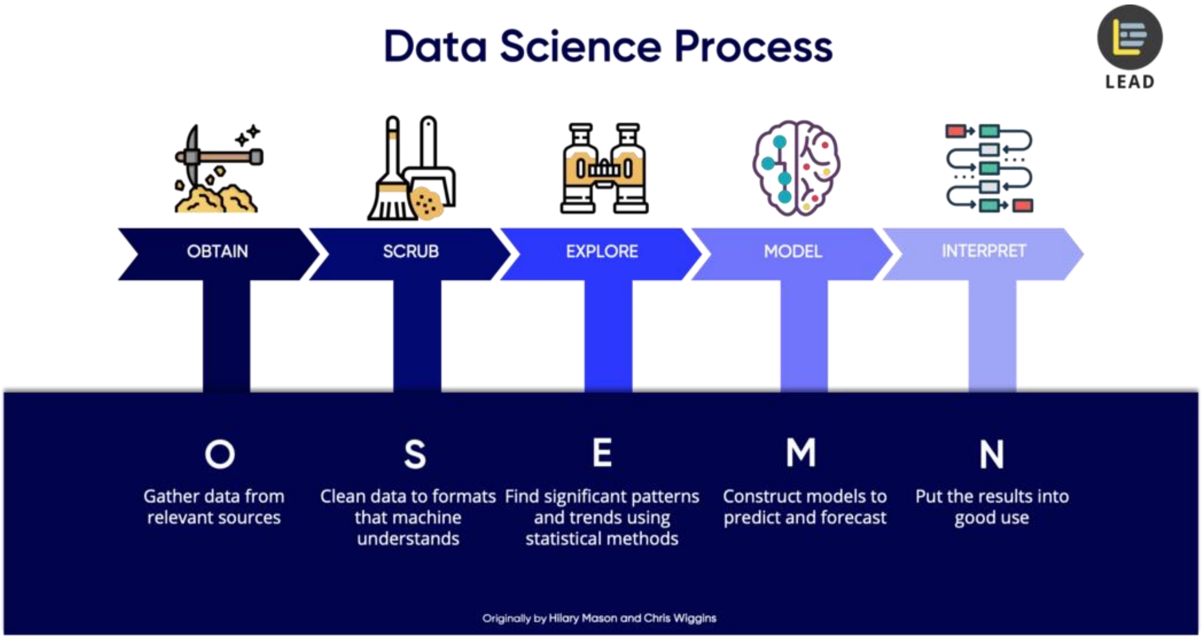
Tian, Y., Zheng, B., Wang, Y., Zhang, Y., & Wu, Q. (2019). College Library Personalized Recommendation System Based on Hybrid Recommendation Algorithm. *Procedia CIRP*, *83*, 490–494. https://doi.org/10.1016/j.procir.2019.04.126

Wadikar, D., Kumari, N., Bhat, R., & Shirodkar, V. (2020). Book recommendation platform using deep learning. *International Journal of Engineering and Technology*, *07*, 6764–6770.

10. Appendix

10.1 OSEMN Framework

Figure 1 OSEMN framework and description of each stage in OSEMN


Data Science Process
LEAD
OBTAIN
SCRUB
EXPLORE
MODEL
INTERPRET
O
Gather data from relevant sources
S
Clean data to formats that machine understands
E
Find significant patterns and trends using statistical methods
M
Construct models to predict and forecast
N
Put the results into good use
Originally by Hilary Mason and Chris Wiggins

10.2 Data attributes and data cleaning techniques

Table 1 Book Dataset Attributes and its description (before cleaned)

Attributes	Type	Description	Remain / Drop
kind	string	Resource type for a volume.	Drop
id	string	Unique identifier for a volume.	Remain
etag	string	Opaque identifier for a specific version of a volume resource.	Remain
selfLink	string	URL to this resource.	Remain
volumeInfo.title	string	Volume title.	Remain
volumeInfo.subtitle	string	Volume subtitle.	Remain
volumeInfo.authors	list	The names of the authors and/or editors for this volume.	Remain
volumeInfo.publisher	string	Publisher of this volume.	Remain
volumeInfo.publishedDate	string	Date of publication.	Remain
volumeInfo.description	string	A synopsis of the volume. The text of the description is formatted in HTML and includes simple formatting elements, such as b, i, and br tags.	Remain
volumeInfo.industryIdentifiers	list	Industry standard identifiers for this volume.	Remain
volumeInfo.readingModes.text	boolean	Whether the text reading mode is available for this volume.	Remain
volumeInfo.readingModes.image	boolean	Whether the image reading mode is available for this volume.	Remain
volumeInfo.pageCount	integer	Total number of pages.	Remain
volumeInfo.printType	string	Type of publication of this volume. Possible values are BOOK or MAGAZINE.	Drop
volumeInfo.categories	list	A list of subject categories, such as "Fiction", "Suspense", etc.	Remain
volumeInfo.maturityRating	string	A maturity rating of the volume to show which age group is suitable to watch the specific volume.	Remain
volumeInfo.allowAnonLogging	boolean	Whether anonymous logging should be allowed.	Drop
volumeInfo.contentVersion	string	An identifier for the version of the volume content (text & images).	Drop

volumeInfo.panelizationSummary.containsEpubBubbles	boolean	Whether or not contains epub bubbles	Drop
volumeInfo.panelizationSummary.containsImageBubbles	boolean	Whether or not contains images bubbles	Drop
volumeInfo.imageLinks.smallThumbnail	string	Image link for small thumbnail size (width of ~80 pixels).	Drop
volumeInfo.imageLinks.thumbnail	string	Image link for thumbnail size (width of ~128 pixels).	Drop
volumeInfo.language	string	Best language for this volume (based on content). It is the two-letter ISO 639-1 code such as 'fr', 'en', etc.	Remain
volumeInfo.previewLink	string	URL to preview this volume on the Google Books site.	Drop
volumeInfo.infoLink	string	URL to view information about this volume on the Google Books site.	Drop
volumeInfo.canonicalVolumeLink	string	Canonical URL for a volume.	Drop
saleInfo.country	string	The two-letter ISO_3166-1 country code for which this sale information is valid.	Remain
saleInfo.saleability	string	Whether or not this book is available for sale or offered for free in the Google eBookstore for the country listed above. Possible values are FOR_SALE, FREE, NOT_FOR_SALE, or FOR_PREORDER.	Remain
saleInfo.isEbook	boolean	Whether or not this volume is an eBook (can be added to the My eBooks shelf).	Remain
saleInfo.listPrice.amount	double	Amount in the currency listed below.	Remain
saleInfo.listPrice.currencyCode	string	An ISO 4217, three-letter currency code.	Remain
saleInfo.retailPrice.amount	double	Amount in the currency listed below.	Remain
saleInfo.retailPrice.currencyCode	string	An ISO 4217, three-letter currency code.	Remain
saleInfo.buyLink	string	URL to purchase this volume on the Google Books site.	Drop
saleInfo.offers	list	Available offers	Drop

accessInfo.country	string	The two-letter ISO_3166-1 country code for which this access information is valid.	Drop
accessInfo.viewability	string	The read access of a volume. Possible values are PARTIAL, ALL_PAGES, NO_PAGES or UNKNOWN. This value depends on the country listed above. A value of PARTIAL means that the publisher has allowed some portion of the volume to be viewed publicly, without purchase. This can apply to eBooks as well as non-eBooks. Public domain books will always have a value of ALL_PAGES.	Remain
accessInfo.embeddable	boolean	Whether this volume can be embedded in a viewport using the Embedded Viewer API.	Remain
accessInfo.publicDomain	boolean	Whether or not this book is public domain in the country listed above.	Remain
accessInfo.textToSpeechPermission	string	Whether text-to-speech is permitted for this volume. Values can be ALLOWED, ALLOWED_FOR_ACCESSIBILITY, or NOT_ALLOWED.	Remain
accessInfo.epub.isAvailable	boolean	Is a flowing text epub available either as public domain or for purchase.	Remain
accessInfo.epub.acsTokenLink	string	URL to retrieve ACS token for epub download.	Drop
accessInfo.pdf.isAvailable	boolean	Is a scanned image pdf available either as public domain or for purchase.	Remain
accessInfo.pdf.acsTokenLink	string	URL to retrieve ACS token for pdf download.	Drop
accessInfo.webReaderLink	string	URL to read this volume on the Google Books site. Link will not allow users to read non-viewable volumes.	Drop
accessInfo.accessViewStatus	string	Combines the access and viewability of this volume into a single status field for this user. Values can be FULL_PURCHASED, FULL_PUBLIC_DOMAIN, SAMPLE or NONE.	Remain
accessInfo.quoteSharingAllowed	boolean	Whether or not sharing quote of the specific volume is allowed	Drop
searchInfo.textSnippet	string	A text snippet containing the search query.	Remain
volumeInfo.averageRating	double	The mean review rating for this volume. (min = 1.0, max = 5.0)	Remain

volumeInfo.ratingsCount	integer	The number of review ratings for this volume.	Remain
accessInfo.epub.downloadLink	string	URL to download epub.	Drop
accessInfo.pdf.downloadLink	string	URL to download pdf.	Drop
volumeInfo.panelizationSummary.imageBubbleVersion	string	The version of the image bubble in the top-level summary of the panelization info in this volume.	Drop
volumeInfo.comicsContent	boolean	Whether the volume has comics content.	Drop
volumeInfo.seriesInfo.kind	string	The resource type of the volume series	Drop
volumeInfo.seriesInfo.bookDisplayNumber	integer	The number of displayed pages of the volume series.	Drop
volumeInfo.seriesInfo.volumeSeries	list	The series information of the volume series.	Drop
volumeInfo.panelizationSummary.epubBubbleVersion	string	The version of the epub bubble in the top-level summary of the panelization info in this volume.	Drop
volumeInfo.seriesInfo.shortSeriesBookTitle	string	The short book title in the context of the series.	Drop

Table 2 Attributes with dirty values and description of methods to handle them

Attributes	Missing?	Issues	Techniques
volumeInfo.title	Yes - MCAR	- HTML character entity - Special characters	- Records with missing values are removed. - HTML entities are decoded, and special characters are replaced with empty string.
volumeInfo.subtitle	Yes - MCAR	- Incorrect data formats (Dates and Integer) - HTML character entity - Special characters	- Values in integer and date are replaced with NA. - Records with missing values are replaced with "Missing". - HTML entities are decoded, and special characters are replaced with empty string.
volumeInfo.authors	Yes - MCAR	- Values stored in list - HTML character entity - Special characters	- Record with missing values are replaced with "Missing". - Lists are converted to string. - HTML entities are decoded, and special characters are replaced with empty string.
volumeInfo.publisher	Yes - MCAR	- Incorrect data formats (Integer) * - HTML character entity - Special characters	- Values in integer are replaced with NA. - Records with missing values are replaced with "Missing". - HTML entities are decoded, and special characters are replaced with empty string.
volumeInfo.publishedDate	Yes - MCAR	- Incorrect data - Different data formats (Most contain year)	- Year is extracted from the date. - Year not between 1500 and 2022 are replaced with NA. - Records with missing values are imputed using attribute median.
volumeInfo.description	Yes - MCAR	- HTML character entity - Special characters	- Records with missing values are replaced with "Missing". - HTML entities are decoded, and special characters are replaced with empty string.

volumeInfo. industryIdentifiers	Yes - MCAR	- Values stored in a list of dictionaries	- List of dictionaries is flattened and pivoted into attributes based on types of industry identifier. - Records with missing values are replaced with "Missing".
volumeInfo.pageCount	Yes - MCAR	- Incorrect data (Contain 0)	- 0s are replaced with NA. - Records with missing values are imputed using attribute mean.
volumeInfo.categories	Yes - MCAR	- Values stored in list	- Records with missing values are imputed using forward fill. - Lists are converted to string.
saleInfo.listPrice. amount	Yes - MNAR (Missing values for non-saleable books)	- Incorrect data (Contain 0 for saleable books)	- 0s are replaced with attribute median. - Records with missing values are replaced with 0.
saleInfo.retailPrice. amount			
saleInfo.listPrice. currencyCode		-	- Records with missing values are replaced with "MYR".
saleInfo.retailPrice. currencyCode			
searchInfo.textSnippet	Yes - MCAR	- HTML character entity - Special characters	- Records with missing values are replaced with "Missing". - HTML entities are decoded, and special characters are replaced with empty string.
volumeInfo.averageRating	Yes - MCAR	-	- Records with missing values are replaced with 0
volumeInfo.ratingsCount	Yes - MCAR	-	- Records with missing values are replaced with 0

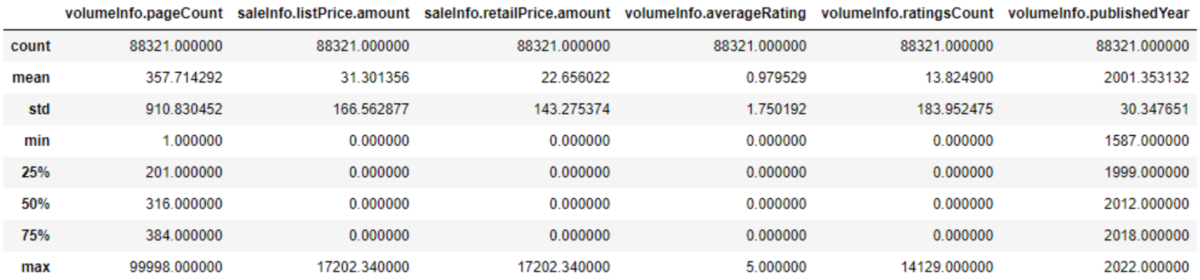
Table 3 Book Dataset Attributes (after cleaned)

Attributes	Data Type		Notes
id	Categorical	String	-
etag	Categorical	String	-
selfLink	Categorical	String	-
volumeInfo.title	Categorical	String	-
volumeInfo.subtitle	Categorical	String	-
volumeInfo.authors	Categorical	String	-
volumeInfo.publisher	Categorical	String	-
volumeInfo.description	Categorical	String	-
volumeInfo.readingModes.text	Categorical	Boolean	-
volumeInfo.readingModes.image	Categorical	Boolean	-
volumeInfo.pageCount	Numerical	Integer	-
volumeInfo.categories	Categorical	String	-
volumeInfo.maturityRating	Categorical	String	-
volumeInfo.language	Categorical	String	-
saleInfo.country	Categorical	String	-
saleInfo.saleability	Categorical	String	-
saleInfo.isEbook	Categorical	Boolean	-
saleInfo.listPrice.amount	Numerical	Float	-
saleInfo.listPrice.currencyCode	Categorical	String	-
saleInfo.retailPrice.amount	Numerical	Float	-
saleInfo.retailPrice.currencyCode	Categorical	String	-
accessInfo.viewability	Categorical	String	-
accessInfo.embeddable	Categorical	Boolean	-
accessInfo.publicDomain	Categorical	Boolean	-
accessInfo.textToSpeechPermission	Categorical	String	-
accessInfo.epub.isAvailable	Categorical	Boolean	-
accessInfo.pdf.isAvailable	Categorical	Boolean	-
accessInfo.accessViewStatus	Categorical	String	-
searchInfo.textSnippet	Categorical	String	-
volumeInfo.averageRating	Numerical	Float	-
volumeInfo.ratingsCount	Numerical	Float	-
volumeInfo.publishedYear	Numerical	Integer	Derived from volumeInfo.publishedDate
volumeInfo.industryIdentifiers.ISBN_10	Categorical	String	Derived from volumeInfo.industryIdentifiers
volumeInfo.industryIdentifiers.ISBN_13	Categorical	String	
volumeInfo.industryIdentifiers.ISSN	Categorical	String	
volumeInfo.industryIdentifiers.OTHER	Categorical	String	

10.3 EDA

10.3.1 Descriptive Statistics

Table 4 Descriptive Statistics for numerical attributes



	volumeInfo.pageCount	saleInfo.listPrice.amount	saleInfo.retailPrice.amount	volumeInfo.averageRating	volumeInfo.ratingsCount	volumeInfo.publishedYear
count	88321.000000	88321.000000	88321.000000	88321.000000	88321.000000	88321.000000
mean	357.714292	31.301356	22.656022	0.979529	13.824900	2001.353132
std	910.830452	166.562877	143.275374	1.750192	183.952475	30.347651
min	1.000000	0.000000	0.000000	0.000000	0.000000	1587.000000
25%	201.000000	0.000000	0.000000	0.000000	0.000000	1999.000000
50%	316.000000	0.000000	0.000000	0.000000	0.000000	2012.000000
75%	384.000000	0.000000	0.000000	0.000000	0.000000	2018.000000
max	99998.000000	17202.340000	17202.340000	5.000000	14129.000000	2022.000000

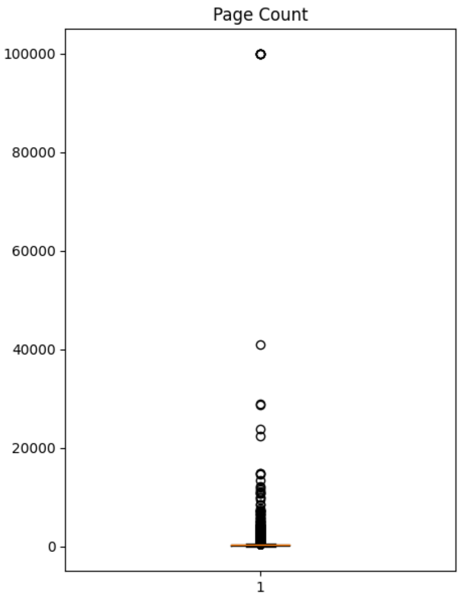


Figure 2 Boxplot for page count

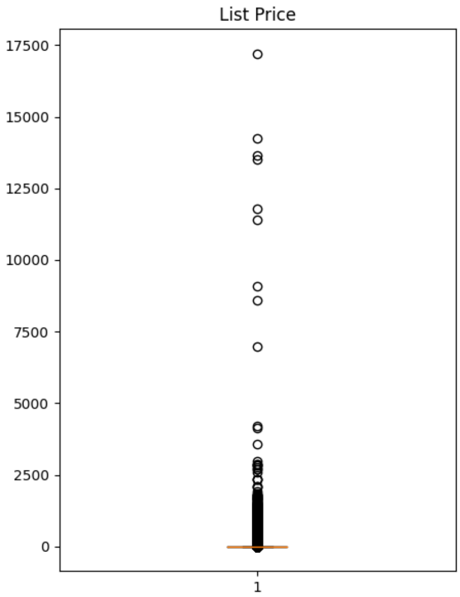


Figure 3 Boxplot for list price

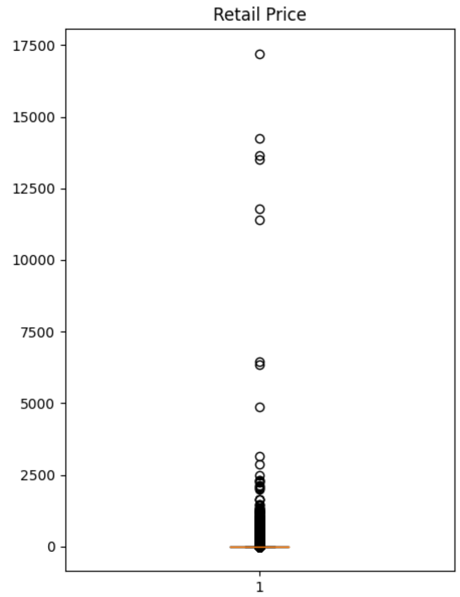


Figure 4 Boxplot for retail price

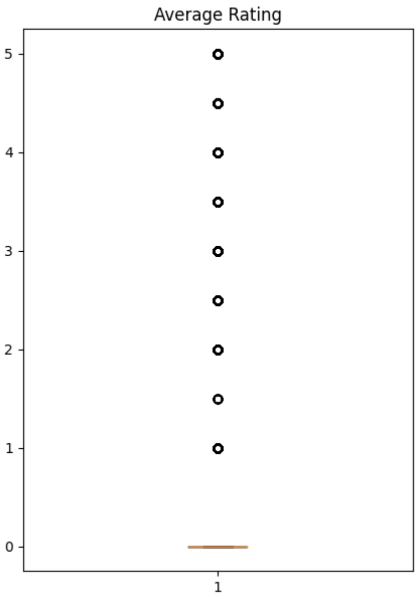


Figure 5 Boxplot for average rating

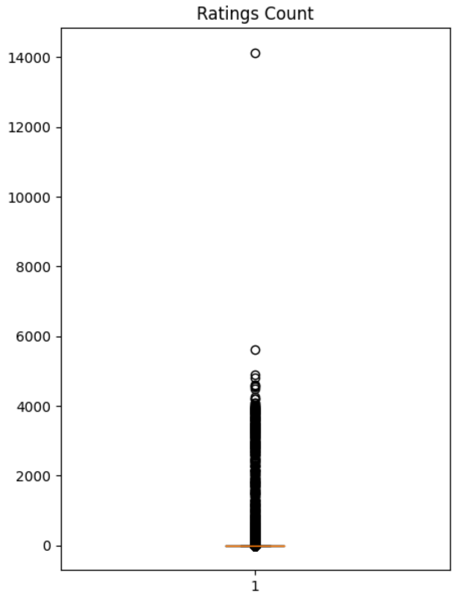


Figure 6 Boxplot for ratings count

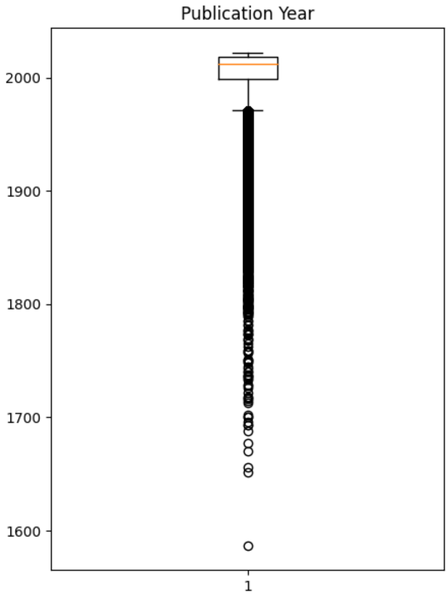


Figure 7 Boxplot for publication year

10.3.2 Book

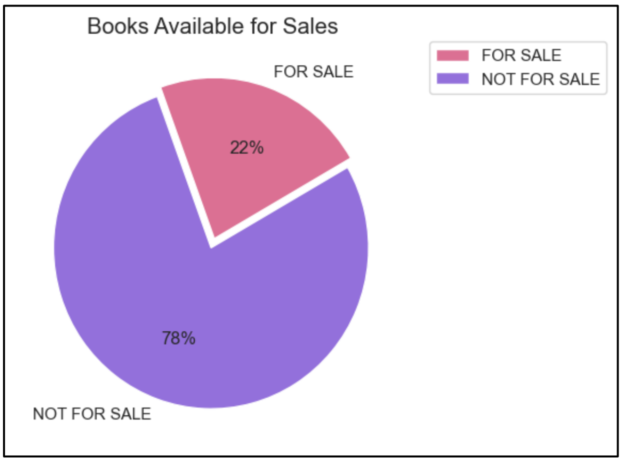


Figure 8 Pie chart of Books Available

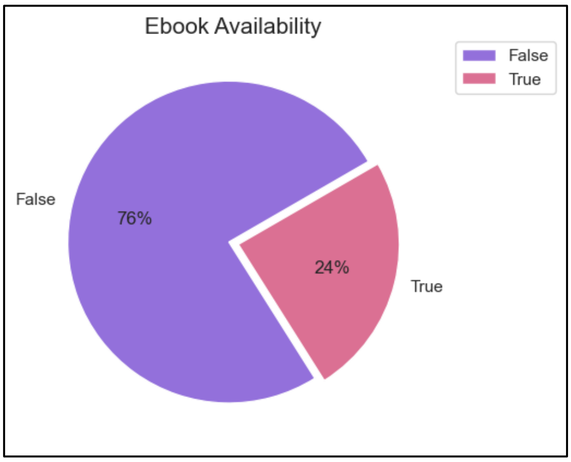


Figure 9 Pie Chart for E-book Availability

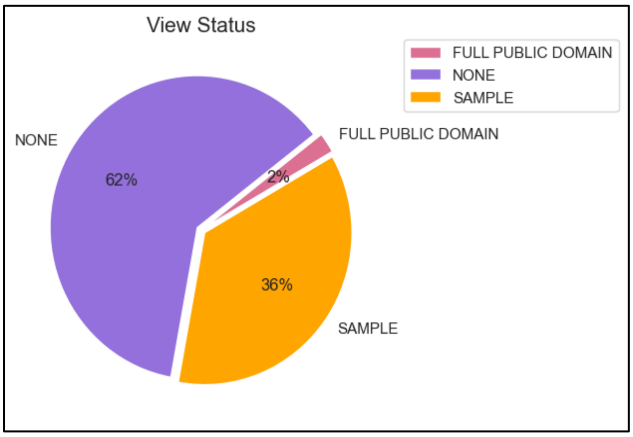


Figure 10 Pie Chart for Books View Status

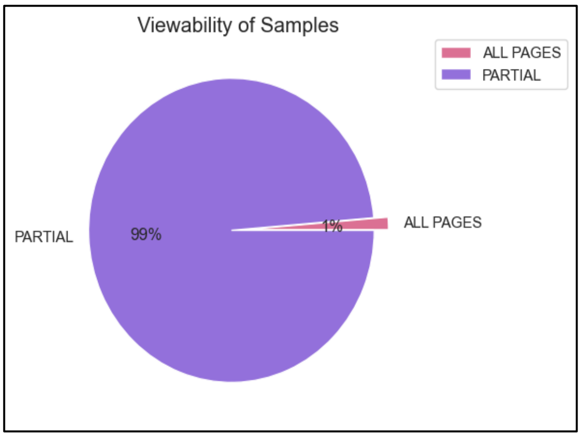


Figure 11 Pie Chart for Viewability of Samples

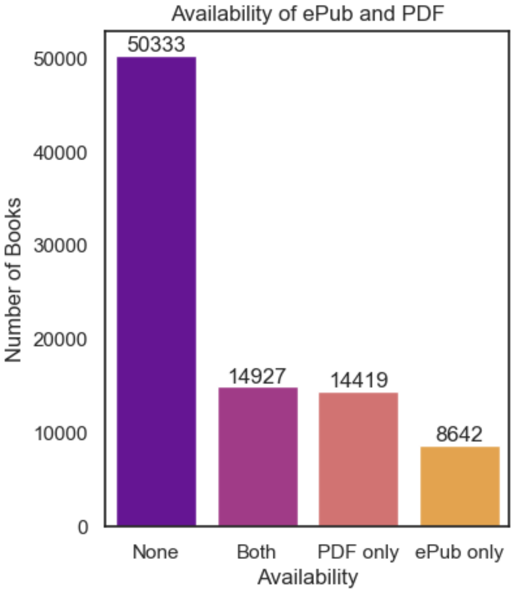


Figure 12 Bar Chart for Availability of e-Pub and PDF

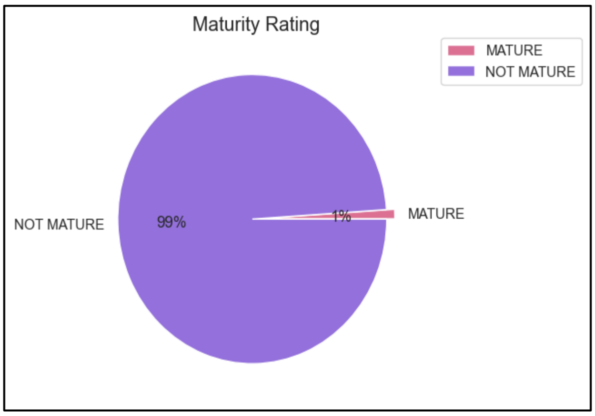


Figure 13 Pie Chart for Maturity Rating

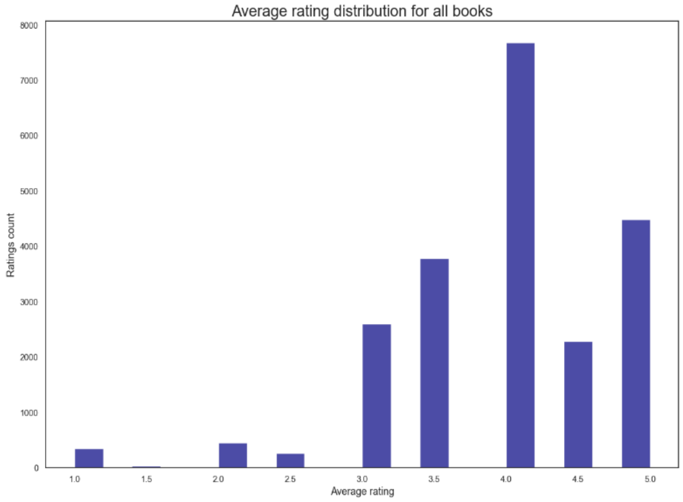


Figure 14 Average Rating of all books

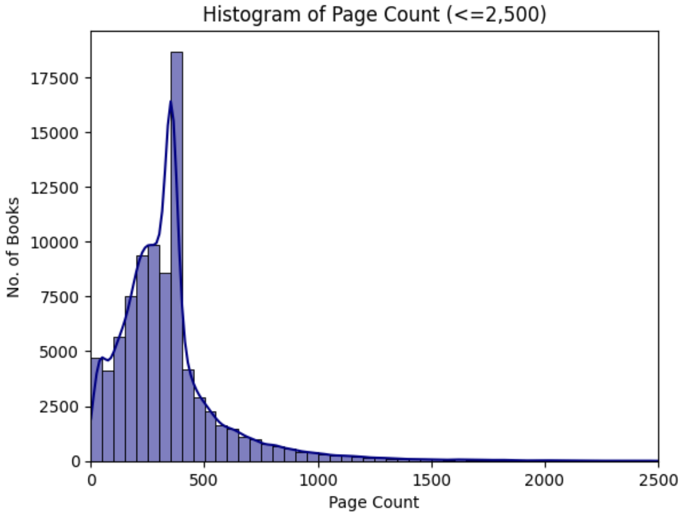


Figure 15 Histogram of Page Count (≤ 2,500)

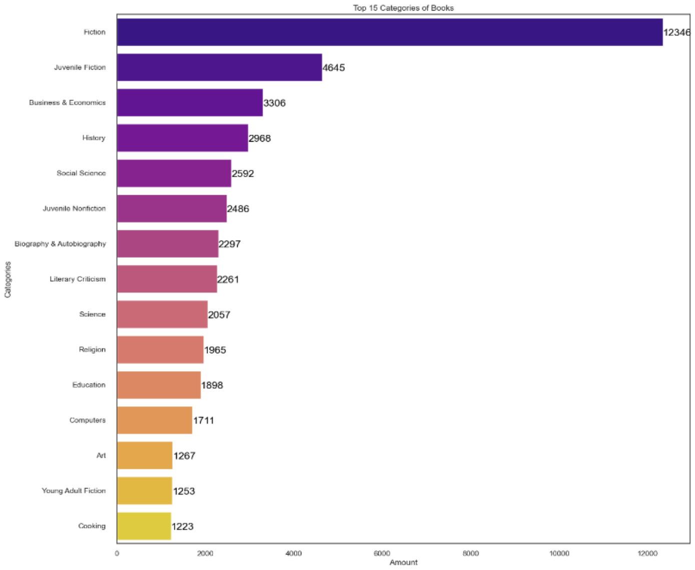


Figure 16 Top 15 Books Categories

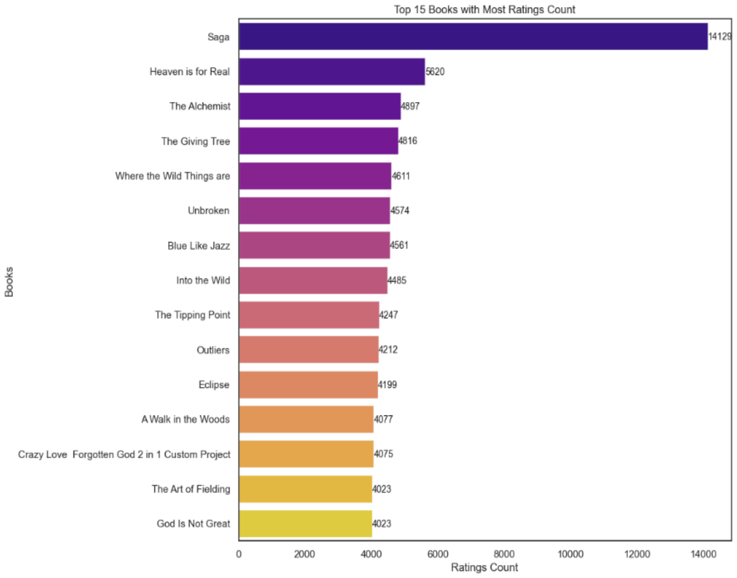


Figure 17 Top 15 Books with Highest Rating Count

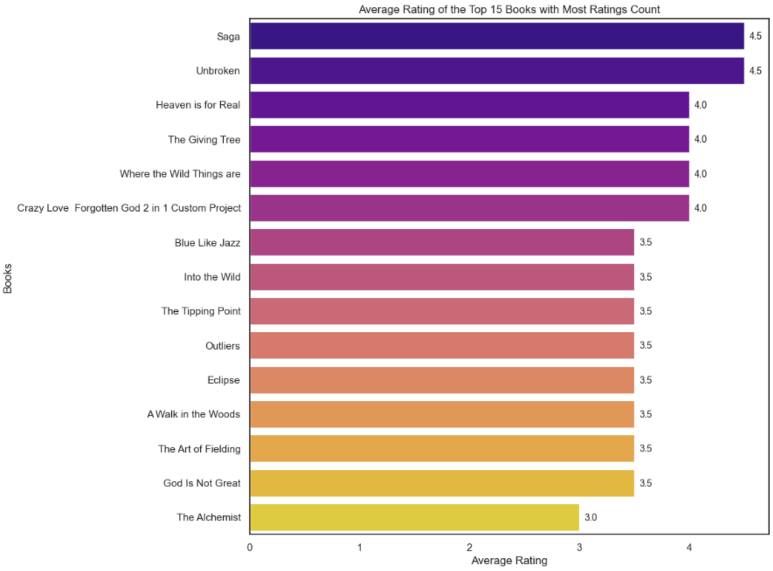


Figure 18 Average Rating of the top 15 Books with Highest Rating Count

10.3.3 Publication

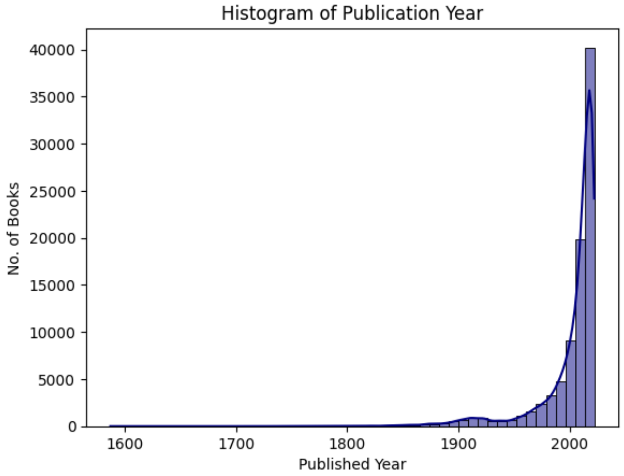


Figure 19 Histogram of Publication Year

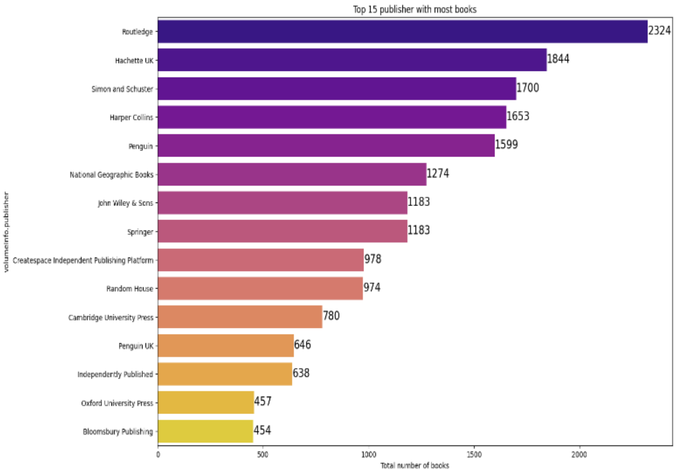


Figure 20 Top 15 Publishers with Most Books Published

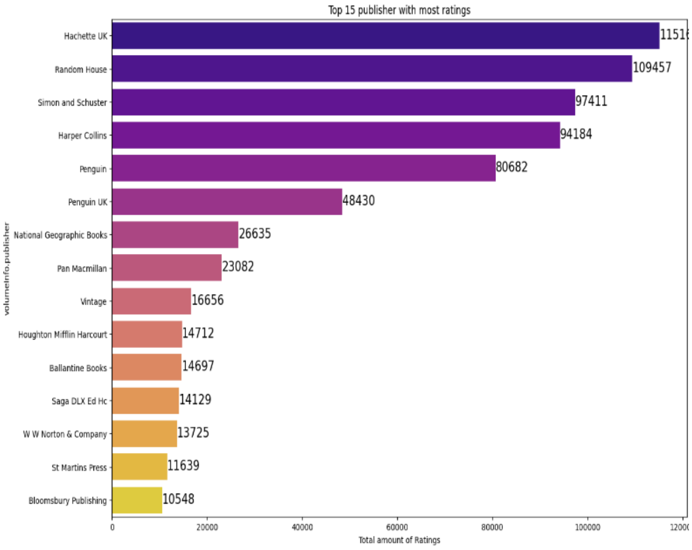


Figure 21 Top 15 Publishers with Most Ratings

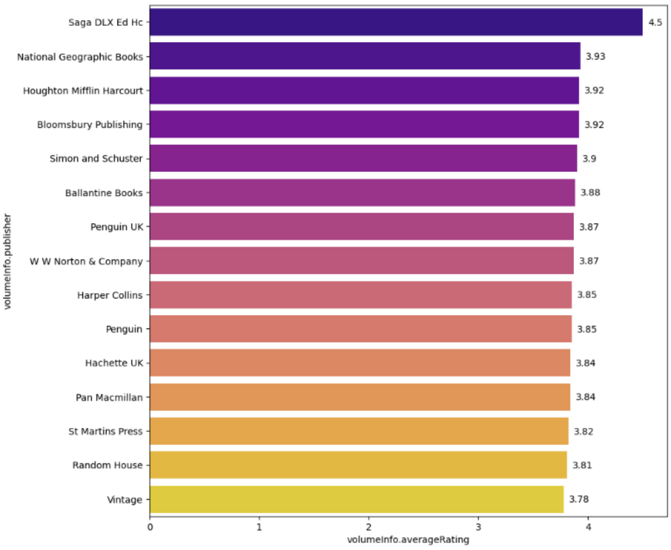


Figure 22 Average Rating of Top 15 Publishers with Most Ratings Count

10.3.4 Price

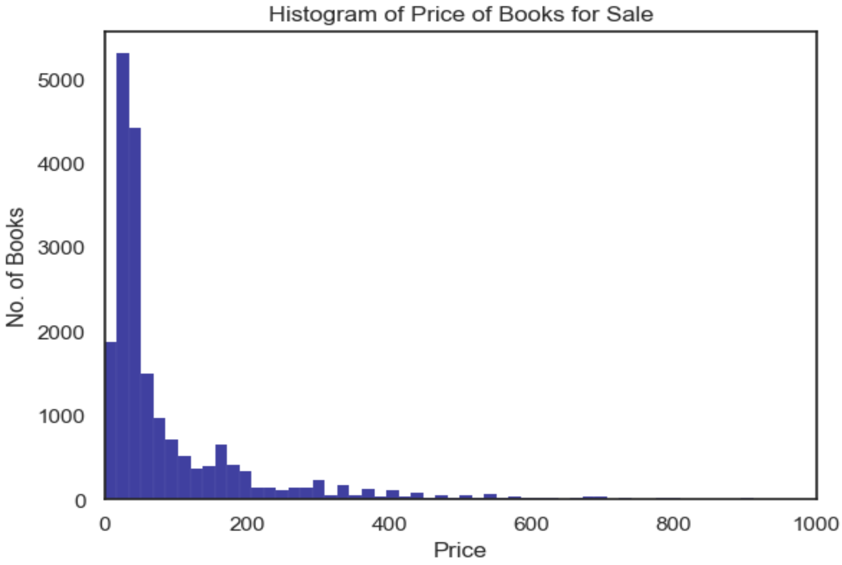
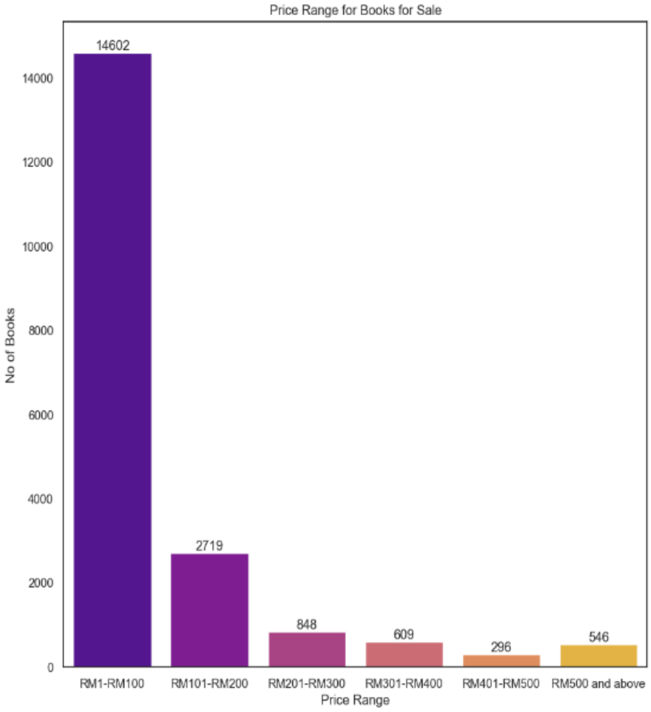


Figure 23 Histogram of Price of Books for Sale



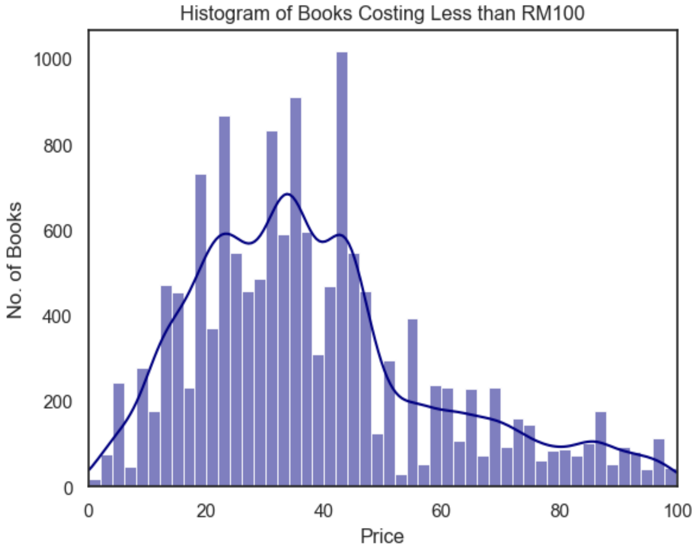


Figure 24 Bar chart of Price Range for Books for Sale

Figure 25 Histogram of Books Costing Less than RM100