**1. Introduction**

Nowadays, with all the advantages that technology offers us, many businesses are virtualized like for example the books that before we used to read them physically but now, we can even read them online. Considering the previous factors, in this report I will analyse a book dataset which has different details of users, books, rating, among others.

Recommendation systems are used and applied in different businesses to attract the attention of the customers or users and offer them items that could be of their interest. Also, Market Basket Analysis is developed to offer them deals or understand the customer behaviour; that is why in the first part of this report, I will be developing the answers of Machine learning implementing different techniques and explaining how they work and why were they applied. Besides, Data Visualization techniques will be applied crating an interactive dashboard for seniors (+65 years old), in which with simple visualizations, I will summarize the important characteristics.

**2. Data**

The data was taken from Kaggle in the following link: <https://www.kaggle.com/datasets/arashnic/book-recommendation-dataset/data> (Kaggle, 2023)

**2.1 Characterization of the dataset**

The dataset was compound of 3 excel files and I decided to work with 3 data frames with different dimensions since in every Recommended system and visualizations I selected the necessary features for their analysis and not losing the information they have inside.

* df\_book has 271360 rows and 8 columns.
* df\_rating has 11499780 rows and 8 columns.
* df\_users has 278858 rows and 3 columns.

**2.2 Data Dictionary**

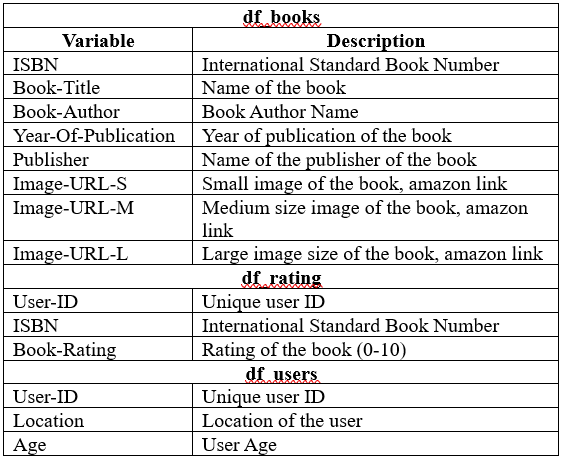


Figure 1: Data Dictionary of the dataset

As we can see in Figure 1, the datasets share common variables to adjust the necessary information for the different questions.

**2.3 Data Preparation**

After applying Data Preparation and Data Cleaning techniques, I got the next decisions and information:

* Drop 3 columns in df\_book where they had no information about year of publication.
* The 3 data frames don’t have duplicates
* As we can see in Figure 2, Null values present in “Book-Author” and “Publisher” were replaced by “Unknown” since they represented less than 1 percent in df\_books and “Age” column was dropped since it had around 39% of missing values which can cause variance to the results.

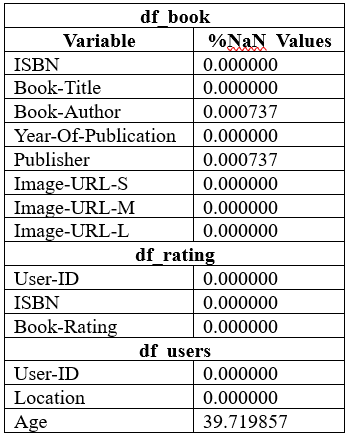


Figure 2: Percentage of Null values in the data frames

**3. Machine Learning**

**3.1 Question 1**

**3.1.1 Discuss and explain the purpose of a recommendation system for online retail business in machine learning.**

A recommendation system in online retail has the purpose to suggest customers or users services or products that they would be interested to get, buy or read according to previous data like previous sells, streaming services, demographics among others characteristics analysed in which using Machine Learning Recommendation Systems we can predicts what are the interests of the customers and give us the option to offer it to them, the final purpose is to give more value to the company making the user be more interested on us and for the user the benefit is the time since we are offering products or services he/she is interested.

**3.1.2 Briefly compare Content and Collaborative filtering using any dataset of your choice.**

**Content Based Recommended System**

According to Kulkarni, A.B. (Kulkarni et al., 2022) Content-based filtering is used in recommending products or items very similar to those being clicked or liked. User recommendations are based on the description of an item and a profile of the user’s interest. Content-based recommender systems are widely used in e-commerce platforms. It is one of the basic algorithms in the recommendation engine. Content-based filtering can be triggered for any event; for example, on click, on purchase, or add to cart.

To start working with this system I merged df\_book with df\_rating creating columns of Average\_Rating and rating count, but as we can see in Figure 3, many people read books and don’t rate them showing as zero. So, I didn’t consider the ratings as zero for this analysis.

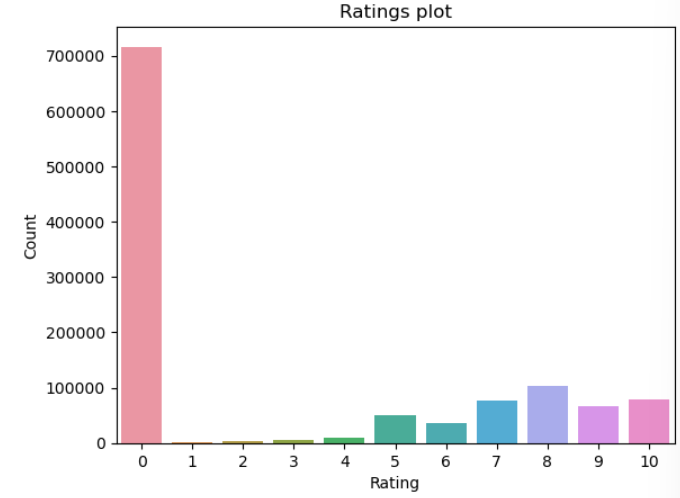


Figure 3: Ratings plot according to the number of users

To start I calculated the weighted rating with the next formula in Figure 4:

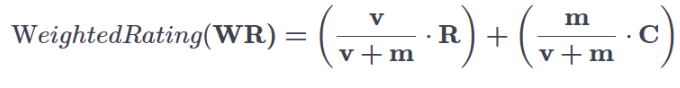


Figure 4: Weighted Rating Formula

Where:

v: number of people that rated the books (Rating\_Count)

m: minimum of rated books required to be listed in the chart

R: Average Rating of the books

C: mean rated count of the books across the dataframe

In my results I got that:

* C = 7.53; considering that the rating values have as the highest rate 10.
* m = 24; representing the number of books in the 99th percentile which means that people didn’t rate a large number of books.
* Filtering with those values the new data frame has 1586 rows and 10 columns.

The results are the next ones shown in Figure 5.

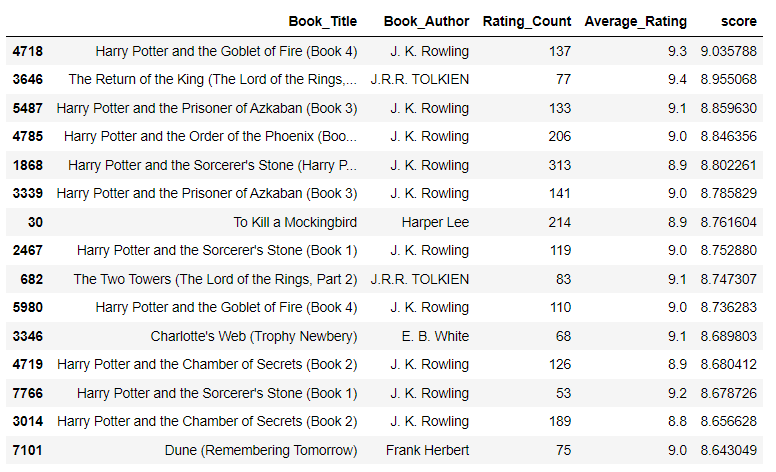


Figure 5: 15 best scored books applying weighted mean rating.

The score calculated with weighted rating function is the one in which we can trust because it considers how many users voted in the movie, the minimum requires, the average rating and the mean across the whole data which gives us a result in which we can trust. In the rating count we can see that many people rated those books which confirms as well that we can trust in m value calculated before.

In addition, with df\_book dataset I analysed Based Recommender focusing “Book-Author”, “Year of Publication” and “Publisher” in which I cleaned the data converting words to lower case, dealing with extra spaces and creating a soup of the mentioned features to vectorize them creating a matrix with a shape of 271357 rows and 112629 columns and applying cosine similarity function I got the recommendations for the books as we can see in Figure 6.

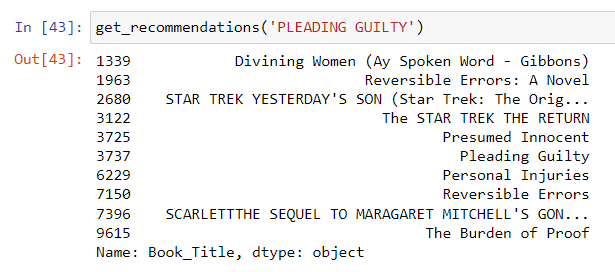


Figure 6: 10 Most similar movies recommendation applying cosine similarity function

Sciencedirect in his website (Sciencedirect.com, 2019) states that Cosine similarity measures the similarity between two vectors of an inner product space. It is measured by the cosine of the angle between two vectors and determines whether two vectors are pointing in roughly the same direction. It is often used to measure document similarity in text analysis.

**Collaborative Filtering System**

Google developers (Google Developers, n.d.) mentions that 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. Furthermore, the embeddings can be learned automatically, without relying on hand-engineering of features.

As I am analysing books, this method will recommend books according to other users that read the same and similar books. I have a big data frame in df\_rating, and that could cause error in our next functions. So, I will filter and just use the movies that were rated for 50 users or more and ratings marked as zeros will not be considered since they represent users that read the book but didn't give a rating and for this example I used user-user Collaborative filtering.

According to Maklin, C. (Maklin, 2022) User-based — User-based collaborative filtering makes recommendations based on the user’s preferences that are similar to other users. For example, if a user gives a similar rating to movies as the user in question. We could assume that they have similar interests. Thus, if the other user has seen and liked a movie that the user hasn’t seen, we would recommend it.

The variables were defined as “X” containing all the df\_users dataset with a data frame names as Colab\_F and “y” represents the User\_ID. Before splitting I balanced the classes in the dataset since when we count the “y values” we need to have at least 2 of each split to make correctly the stratification, and after splitting the data in 20% test size; I got the next table of results as shown in Figure 7.

|  |  |
| --- | --- |
| **Methods applied** | **Root Mean Squared Error** |
| Score baseline | 3.3110361664645906 |
| Mean Rating Filtering | 1.73515138499277 |
| Weighted mean with cosine similarity | 2.8679984594623487 |

Figure 7: Results with Colaborative Filtering Item-Item

According to the mean score the best model is the Mean Rating filtering model, and in second place comes the weighted mean model; both of them are better than the base line score gotten. Although Mean Rating Filtering shows a better result, I trust and decide to use in this case the weighted Mean Score since due to the use of cosine similarity we can compare between users according to the preferences of other readers and captures the reader behaviour. As the weighted model is lower than the baseline Root Squared Error metric, this is the best for me.

**3.1.3 Train and test machine learning models for the user-user or item-item collaborative filtering. Justify your recommendations for the considered scenario by providing a conceptual insight.**

Item-item collaborative filtering is one kind of recommendation method which looks for similar items based on the items users have already liked or positively interacted with. (qutbuddin, 2020)

I performed item-item filtering using the same collaborative filtering used in the previous item-item, but including “Book\_Title” from “df\_book”.

To justify my recommendations, I transformed a binary matrix for rating for each rating more than 3 return a value of one; otherwise, zero, and applied k-nearest neighbors (knn) with cosine similarity metric. For testing it I tried with the books liked by user “276747” shown in Figure 8.

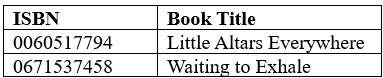


Figure 8: Books liked by user “276747”

And the model recommended the next books according to his preferences in Figure 9:

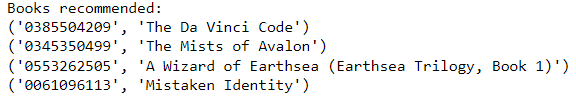


Figure 9: Books recommended for user “276747”.

Practically in the matrix is filtering if the user liked the items or not and comparing those items with cosine similarity the model suggests the books that the user could be interested in reading them.

**3.2 Question 2**

**3.2.1 Perform Market Basket Analysis on the chosen dataset by using Apriori and FP growth algorithms. Can you express major divergence between these models? Compare and contrast the machine learning results obtained based on both algorithms.**

I used the same data frame used in item - item Collaborative filtering system, but I added the country in which the user is located from df\_users because that will improve the model understanding cultural differences and applying different metrics.

According to Ph.D, S.C.S. (Ph.D,2020) Support: This is the easiest metric to calculate, as it’s simply the proportion of all your transactions that contain an association rule.

Confidence brings a bit more specificity to your judgment of this association rule. In this case, it’s the proportion of all the transactions that contain all the items in the itemset over the proportion of transactions containing just one of them. (Ph.D,2020)

The lift metric lets us know whether our assumption of “no relationship” between the items — that they are independent — is reality or not. (Ph.D,2020)

The same steps were applied for both algorithms models Apriori and FP growth just changing the code of evaluation; the data was already cleaned and first I evaluated first for Canada country with a min\_support = 0.001 that shows the items that occur at least 0.1% in our data. Then I applied the rules focusing in lift metric and a min\_treshold of zero to display all the values in this case displaying antecedents and consequents as we can see in Figure 10

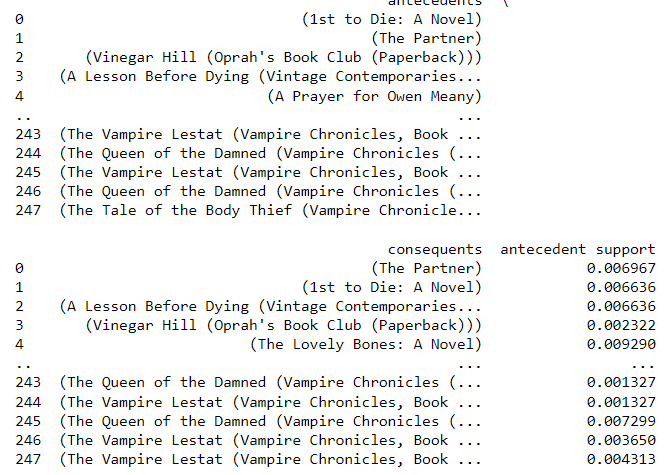


Figure 10: First results of Apriori

Then, I applied the rules lift>=5 and confidence >=0.8 to know exactly the results of the next books that the reader can be really interested to get with and 80% of confidence with a good likelihood to acquire the consequent book from the antecedent and we can see them in Figure 11 and 12.

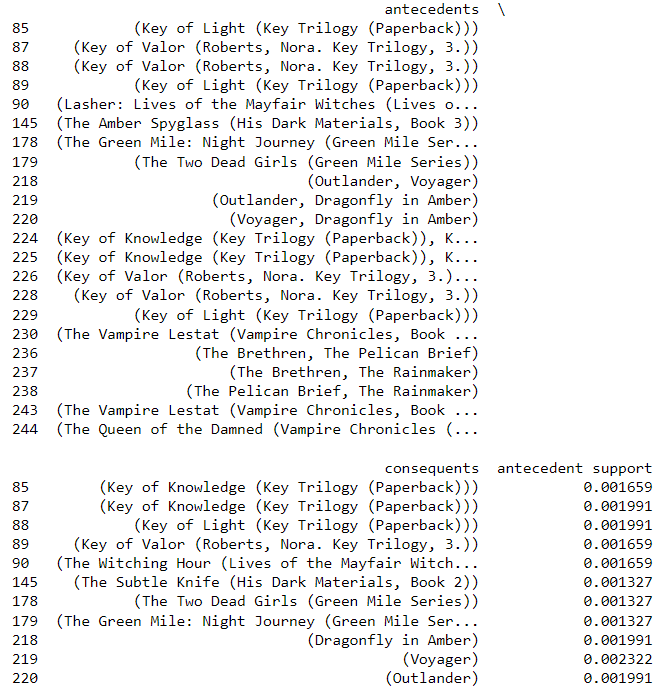


Figure 11: Part 1 Apriori results with rules applied

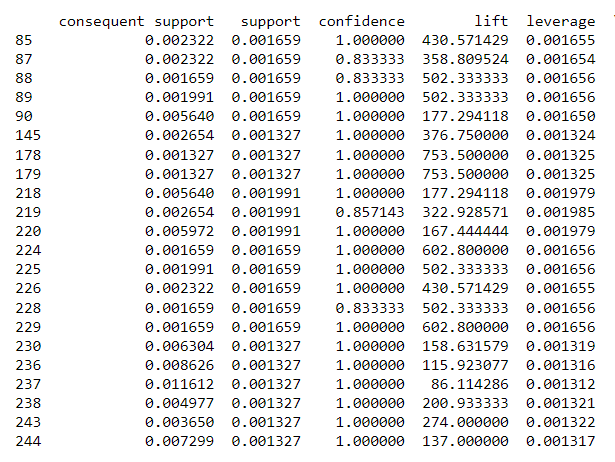


Figure 12: Part 2 Apriori results with rules applied

For FP Growth algorithm first I encoded the data and with a lift >=5 and confidence 0.8, I got the results in figure 13 and 14.

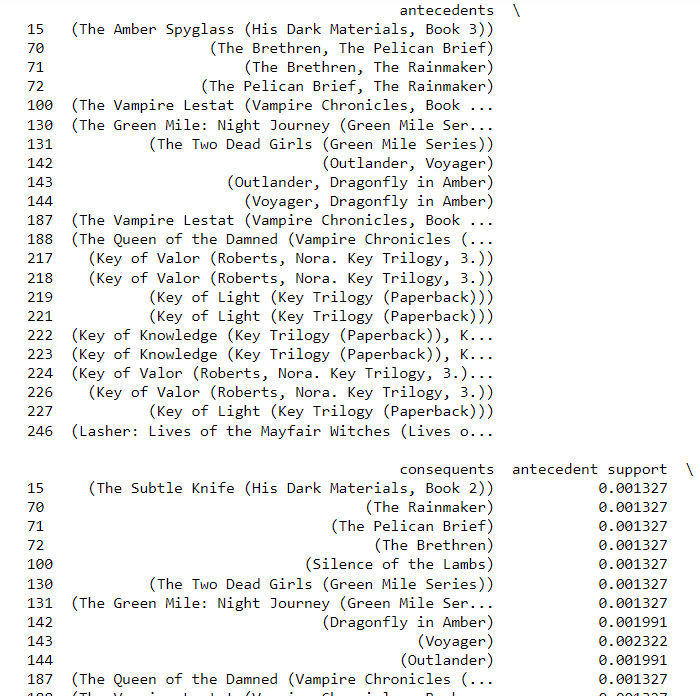


Figure 13: Part 1 FP growth results with rules applied

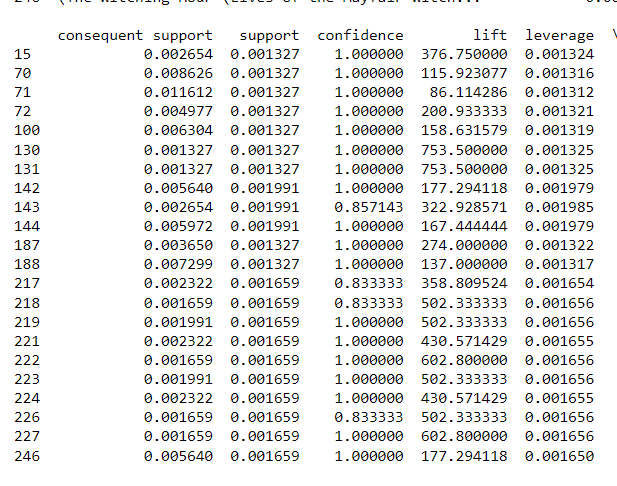


Figure 14: Part 2 FP growth results with rules applied

After applying the same codes for United States represented as “usa” in the data I compared the time of both algorithms in seconds shown in the next table:

|  |  |
| --- | --- |
| Apriori execution time for “usa” | 73.63069939613342 seconds |
| FP growth execution time for usa | 2.9268031120300293 seconds |

Figure 12: Part 2 Apriori results with rules applied

After exploring carefully both methods give really good results showing a good confidence and lift, the big difference is the time to implement both methos as we could see in figure 2 FP growth algorithm has a great advantage in time efficiency for these calculations because Apriori uses more memory for the calculations, but as we could see FP growth is encoded handling with the data if it is very sparsed and it is really good for large datasets as is this one.

**Data Visualization**

**3) Create an interactive Dashboard aimed at older adults (65+) with specific features to summarise the most important aspects of the data and identify through your visualisation why this dataset is suitable for Machine Learning models in an online retail business. Explain how your dashboard is designed with this demographic in mind.**

www.kaggle.com. (2023). Book Recommendation Dataset. [online] Available at: <https://www.kaggle.com/datasets/arashnic/book-recommendation-dataset/data>

Kulkarni, A.B., Adarsha Shivananda, Kulkarni, A. and V Adithya Krishnan (2022). Content-Based Recommender Systems. Apress eBooks, pp.63–87. doi:https://doi.org/10.1007/978-1-4842-8954-9\_3.

Sciencedirect.com. (2019). Cosine Similarity - an overview | ScienceDirect Topics. [online] Available at: <https://www.sciencedirect.com/topics/computer-science/cosine-similarity>.

Google Developers. (n.d.). Collaborative Filtering | Recommendation Systems. [online] Available at: <https://developers.google.com/machine-learning/recommendation/collaborative/basics>.

qutbuddin, muffaddal (2020). Comprehensive Guide on Item Based Recommendation Systems. [online] Medium. Available at: <https://towardsdatascience.com/comprehensive-guide-on-item-based-recommendation-systems-d67e40e2b75d#:~:text=Item%2Ditem%20collaborative%20filtering%20is>.

Maklin, C. (2022). Memory Based Collaborative Filtering — User Based. [online] Medium. Available at: <https://medium.com/@corymaklin/memory-based-collaborative-filtering-user-based-42b2679c6fb5#:~:text=User%2Dbased%20%E2%80%94%20User%2Dbased>.

Ph.D, S.C.S. (2020). Market Basket Analysis 101: Key Concepts. [online] Medium. Available at: https://towardsdatascience.com/market-basket-analysis-101-key-concepts-1ddc6876cd00.