Association rule/pattern mining for recommender system

Kusal Bista
a1881044

The University of Adelaide
4433_COMP_SCI_7306 Mining Big Data
Lecturer: Dr. Alfred Krzywicki

Table of Contents

1. Executive Summary

The current scenario of the retail market is very competitive; for a business to stand out it requires a prompt understanding of customer behaviour and addressing their preferences dynamically. The traditional marketing approach often lacks customer personalization and fails to provide effective engagement of customers. This report challenges the traditional approach and uses advanced mining data to analyze the basket offering the tailored recommendation of the items according to the customer's preferences.

Basket analysis and recommendation are the key strategies used by major retailers to unveil connections between the products. Using big data mining we can easily identify the connection of items that frequently occur together during the transactions. Essentially, it empowers retailers to distinguish the relationships between the products purchased by customers which facilitates informed decision making. This project uses collaborative filtering that filters out the items that a user might like on the transaction history of other similar users. It traverses through the large group of people and finds a smaller set of users with the similar taste. It is a personalized recommending technique which filters information such as interaction data from other similar users. (TechTarget Contributor)

This project is scalable as it can incorporate larger data for increasing customers. It uses collaborative filtering techniques for efficient performance even when the dataset increases. We use a rating matrix where we use frequency information to find the similarity to facilitate the determination of purchasing patterns. Lastly, we implement Alternating Least Square (ALS) algorithm to solve problems like popularity bias, cold-start problem and scalability issue. (Liao, 2018)

In addition, the project discusses the model performance resulting in an RMSE of 0.14 for user based collaborative filtering in big dataset and 0.15 in small dataset. Here, RMSE (Root Mean Square Error) helps measure our accuracy by calculating the difference between the prediction of the model from the actual outcome. In our case, RMSE for both test and train set is low indicating that our model is making accurate predictions.

2. Introduction

Most of the businesses are now digital and collect large amounts of data every day; however, simply storing this data does not provide valuable insights for the business. Recommendation system implements collaborative filtering to discover underlying patterns in customer behaviour and helps identifying relationships between purchased items. With the knowledge of the pattern of a user, websites can understand what items are commonly bought together, this helps in enhancing business strategies in optimizing the item placement and building customized marketing strategies to enhance user engagement. This project aims to implement a recommendation system on the historical datasets of a customer to reveal hidden relationships between items and provide practical insights for business. The major objective is to improve user preferences and understanding cross-selling opportunities. Ultimately, leveraging the insights gained from analysis can give businesses a competitive edge and drive for sustainable growth in today's dynamic market. (Chaudhary, 2022)

Mining data by association rule is popular and has been extensively researched in various industries like retail, e-commerce, hospitality, healthcare, banking, telecommunication, entertainment etc. Besides utilizing algorithms like Apriori, FP-Growth, and ECLAT, researchers have discovered significant relationships between entities in datasets, particularly in retail and e-commerce sectors. (Hermina, Gopalakrishnan, & Balajishankar, 2022) Poel et al. conducted a market basket analysis at a "Do It Yourself" DIY retailer to identify complementary product pairs and recommend promotional strategies. Chen et al. addressed the challenge of relaxing shopping patterns in chain stores by integrating inventory and timing data into association rule development. Yun et al. introduced an Apriori-like algorithm for multi-store environments that surpassed traditional methods in simulating diverse store sizes and product assortments. Kuo et al. employed association rules to detect disease associations in health insurance data using an ant colony system and clustering techniques. Erdem and Özdağoğlu analyzed emergency department data to enhance department reorganization by identifying associations with patient profiles and service parameters. (Sagin & Ayvaz, 2018)From market analysis to healthcare, association rule mining has been beneficial for extracting insights from large datasets, aiding decision-making and strategy development processes. (Chaudhary, 2022)

In this project we implement a recommendation system using collaborative filtering techniques to improve user experience and boosting business turnover. It uses explicit and implicit rating system for labelling transaction data and finding strong rules to determine the association between the items. This allows retailers to analyze customer behaviour from which personalized product recommendations can be generated by mining the transaction history. As an implicit rating, matrix factorization is implemented using ALS algorithm. As it involves predicting user ratings, it is a regression task. Hence, we use RMSE for evaluation where we achieved RMSE of 1.2403 for train data and 1.0389 for test data.

3. Methodology

3.1. Dataset

Initially, for a small dataset we have 50000 entries of sales transaction records which includes BillNo, Itemname, Quantity, Date, Price, CustomerID, and Cost associated with the transaction.
Following is the detail information about each field:

- BillNo: This is a unique integer value for each transaction as each transaction has a unique receipt.
- Itemname: This field contains a detailed name of the item being purchased in each transaction. In a small dataset we have 2641 unique items whereas big data has 3680 unique items.
- Quantity: It refers to the number of units of the item being purchased in a single transaction. On average, each transaction involves buying around 4 items. The transaction range is from 1 to 10.
- Date: It is a timestamp of each transaction record.
- Price: It represents the cost of a single unit of the item.
- CustomerID: It is a unique identifier assigned for each unique customer. It tracks the transaction made by the user.
- Cost: It refers to the total amount paid for one item and is calculated by multiplying the quantity purchased by the price per unit. It represents the monetary value of each transaction.

Similarly, big dataset is also loaded for the testing purpose. It consists of 200000 data with the same field name.

3.2 Collaborative filtering

It is a technique used to recommend items to users by analysing the response of similar users. It works by analyzing the trend of the large group of identical users and generates personalized lists of recommendations. Firstly, it predicts the association rule and then provides the recommendation. There are two types of approach for collaborative filtering.

a. User based collaborative filtering: We calculate the similarity of different users by implementing similarity measures and use it to predict the ratings. To predict the rating of an unrated item I for a user U, collaborative filtering selects N similar users from a list based on their rating vectors. These users have previously rated item I. The rating for item I for user U is then computed by averaging the ratings given by these N users. (Eve, 2023)

The cosine similarity between two vectors$A$ and Bcan be calculated using the following formula:

$$\text{cosine\_similarity}(A, B) = \frac{A \cdot B}{\|A\|\|B\|}$$

Where:
- The dot product of vectors $A$and B is denoted by $A \cdot B$.
- Euclidean norms of vectors A and B is denoted by $||A||$ and $||B||$.

The dot product of two vectors A and B is computed as:

$$A \cdot B = \sum_{i=1}^{n} A_i \times B_i$$

Where A and B are the components of vectors Bat index i, respectively, and n is the dimensionality of the vectors. (Eve, 2023)

b. Item based collaborative filtering: We assess the similarity of various items by implementing similarity measures and utilizing them to predict the rating. To predict the rating of an item I for a user U, who has not rated it yet, we first identify a set of similar items based on rating vectors provided by other users. From this similarity list, we select N items that user U has rated and use their ratings to calculate the predicted rating for item I. (Eve, 2023)We use cosine similarity between two vectors, representing the items and is given by:

$$\text{Cosine Similarity} = \frac{\sum_{i=1}^{n} A_i \times B_i}{\sqrt{\sum_{i=1}^{n} (A_i)^2} \times \sqrt{\sum_{i=1}^{n} (B_i)^2}}$$

Where,
- A_i and $B_i$are the components of vectors A and B respectively.
- n is the dimensionality of the vectors (i.e., the features or ratings). (Eve, 2023)
- The dot product of the vectors A and B is $\sum_{i=1}^{n} A_i \times B_i$.
- $\sqrt{\sum_{i=1}^{n} (A_i)^2}$ and $\sqrt{\sum_{i=1}^{n} (B_i)^2}$ are the Euclidean norms of vectors A and B respectively.

As a shortcoming of these collaborative filtering, we are following issues:

a. Popularity bias: The system tends to suggest product based on number of interactions they've had, without considering any personalization.

b. Cold-start problem: These kinds of problem occur when newly added items have no interactions within the system.

c. Scalability issue: It refers to the issue when the system lacks the capability to efficiently handle the increasing dataset as more users or items are added to the database. (Liao, 2018)

3.3 Matrix Factorization

It is a factorization technique of a matrix that decomposes a given matrix into the product of multiple of matrices. In the context of recommendation systems, factorization algorithms decompose the user-item interaction matrix into the product of two lower-dimensional matrices representing latent factors for users and items. Implementing this does the following task:

a. When the rating matrix is factorized into users and items representation, the model can predict better for personalized item.

b. This facilitates less-utilized items to have rich latent representations as much as popular items have. Doing so can drastically enhance model to recommend less-known items as well. (Liao, 2018)

The predicted rating $\hat{r}_{ui}$ that user *u* will give to item *i* can be computed using matrix factorization as the sum of the element-wise product of the user and item latent factor vectors:

$$\hat{r}_{ui} = \sum_{k=1}^{K} u_{uk} \cdot v_{ik}$$

Here, K is the number of latent factors, u_{uk} is the k-th element of the user latent factor vector $u_{u\cdot}$, and V_{ik} is the k-th element of the item latent factor vector $v_{i\cdot}$. In the above formula, the hyperparameter of latent factors can be tuned. Now, the latent factor represents the user preferences. (Liao, 2018)

3.4 Alternating Least Square (ALS)

It is the matrix factorization algorithm that operates autonomously in a parallel function. It optimizes collaborative filtering in context of matrix by utilizing a large matrix into two smaller matrices to approximate the original matrix. Instead of explicitly rating, we use implicit rating that represent a "confidence" and an interaction with the item. If any item has higher number of ratings by a user, then the item has more weight in rating matrix. After implicitly rating, ALS iteratively optimizes the approximation between the original matrix and its factorized approximation. It can switch between optimizing one matrix while holding another matrix as fixed and vice versa. This alternating optimization continues until the convergence, where the changes is negligible. It tries to minimize the square difference between actual ratings and the predicted rating obtained from the current approximation of the matrices. In case of overfitting during the optimization ALS can incorporate regularization techniques helping to generalize the learned latent factors and improve model ability to make accurate prediction. Due to the fact, that it incorporates parallelism, it is very suitable for large-scale recommendation system. (Liao, 2018)

3.5 Evaluation of a system:

Root Mean Square Error (RMSE) is one of the popular metrics and evaluates by penalizing more when the rating prediction is way off and penalizing less when the prediction is reasonably close. It is given by:

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2}$$

Where,

- n is the number of observations or data points.
- y_i is the actual value of the target variable for the ith observation.
- $\hat{y}_i$ is the predicted value of the target variable for the ith observation.

Instead of summing the absolute values between predicted and actual rating, we sum the rating prediction errors. (Sen, 2020)

4. Exploratory Analysis

Following EDA was performed after loading the small dataset with 40000 data:

a) **Checking for null values:** Firstly, null values are checked as it is very essential to ensure data integrity. The dataset does not have any missing values.
b) **Checking duplicate values:** The duplicate values are detected and deleted. The first value is kept, and other duplicate values are dropped. Our data had 797 duplicate data in the training set and 150 duplicate entries in test data.
c) **Top 10 Popular Sold Items:** This insight provides the most popular item among customers. Here, “white hanging heart t-light holder” is the most popular item and 372 items have been sold just once. The dataset has altogether had 2641 unique items in the basket.
d) **Top 10 Member Numbers:** This insight provides the top customer IDs based on the frequency of transactions. Customer ID 14606.0 has 767 unique transactions. The dataset has 1660 unique customers.
e) **Checking negative values:** This ensures that the dataset has no negative values.
f) **Monthly Income graph:** This insight provides the overview of the business’s revenue over time. Monthly income in December 2010 was much higher than January, February and March of 2011. The sales drastically dropped from 149459 to 56291 in the months February to March 2011.

From the above EDA on the provided dataset, we can visualize the popular items sold and top customers by transaction frequency. Besides, with the help of a unique count we can decide which collaborative filtering is suitable for this project. We have 2641 unique items and 1660 unique customers, hence item-based collaborative filtering will be appropriate. Using item-based collaborative filtering we can exploit similarities between items and provide robust personalized recommendations based on the customer transaction history. Same EDA process is carried out for the big data as well.

5. Implementation and Testing

5.1 Explict Rating

a) **Data loading and preprocessing:** The data is loaded into the system and splitted to 80-20 training and testing split. After loading the data into a dataframe preprocessing and EDA including checking null data, removing duplicate values are executed to ensure the integrity and readiness of the data.

b) **Rating matrix creation:** To facilitate collaborative filtering, we make a rating matrix representing the interactions between users and items. This acts as a foundation to determine the similarities between items or users. In our case, we implement both item based and user based collaborative filtering. As this matrix represents user and items in row and column respectively, each cell corresponds to the user's rating for an item. If the user has some interaction with the item, then the corresponding cell contains the rating value else zero. Doing so serves as an input matrix for computing item-item or user-user similarities. 'Train_rating_piv_df' and 'Test_rating_piv_df' are rating matrices for training and testing data respectively.

c) **Calculating similarities:** This project has implemented cosine similarity for the calculation of similarity between items or users. It calculates the similarity between two non-zero vectors in an inner product space. Basically, it measures the cosine of the angle between two vectors where 1 indicates perfect similarity, 0 means no similarity and -1 represents perfect dissimilarity. (Eve, 2023) The function 'cosine_similarity' is used where each row represents an item vector. These vectors are computed pairwise to generate the similarities between items. After the computation, the item-item similarity score is assigned where higher similarity means more similar and lower similarity indicates less similar. 'cosine_similarity 'function is used to compute the similarity.

d) **Predicting function:** We have a 'get_prediction' function responsible for estimating the rating that a user would give to a specific item based on the rating of similar items and the user's past transaction history. Next, the function would compute a weighted sum of the rating provided by the user for all the similar items by multiplying each similarity score by the corresponding rating and computing the sum of those products. With the intention of aggregating the ratings of similar items we calculate the weights sum and divide it by the sum of similarity scores to normalize. Finally, the returned normalized weighted sum is the predicted rating that the user would give to the specified item based on the item-item similarity or user-user similarity.

e) **Evaluation:** For evaluation we use RMSE score which aggregates the errors across all the pairs in the training and testing pair. For each user-item pair, the system predicts the user's rating for the item based on their past interactions and similarity with other items. After getting a prediction from the prediction function, we find the differences between predicted rating and actual rating. The RMSE for training and testing dataset is 1.2403 and 1.0389 respectively for item-item based collaborative filtering whereas 1.245 and 1.04 for user-user based collaborative filtering hence, item based collaborative filtering has better performance than user based collaborative filtering. (Sen, 2020) The predictions of the item based collaborative filtering are closer to the actual rating provided by users whereas, the predictions were more deviated in user based collaborative filtering as compared to item based.

5.2 Implicit Rating

a) **Data preparation:** Initially, we count the unique items and customers to calculate the interaction between item and customer. A threshold of '6' is set to filter out the customers who has purchased items fewer than that.

b) **Sparse matrix creation:** We construct a sparse matrix representation using 'sparse_customer_item' of customer-item interactions. This matrix represents the quantity of each item purchased by each customer.

c) **Model Training:** Then model is trained using 'implicit' library. The model will learn latent factors for both customer and items based on the interaction in the sparse matrix.

d) **Finding similar Items:** After training the model, it finds the similar items for each item in the data using 'similar_items' function. For each item, we retrieve the top N similar items along with the corresponding scores.

e) **Generating recommendation:** We generate the recommendation resolving the customer index. The score and item liked information is stored.

f) **Train-Test split for evaluation:** Finally, the dataset is splitted in training and testing, and performance is measured in RMSE. (Li, 2019) (Liao, 2018)

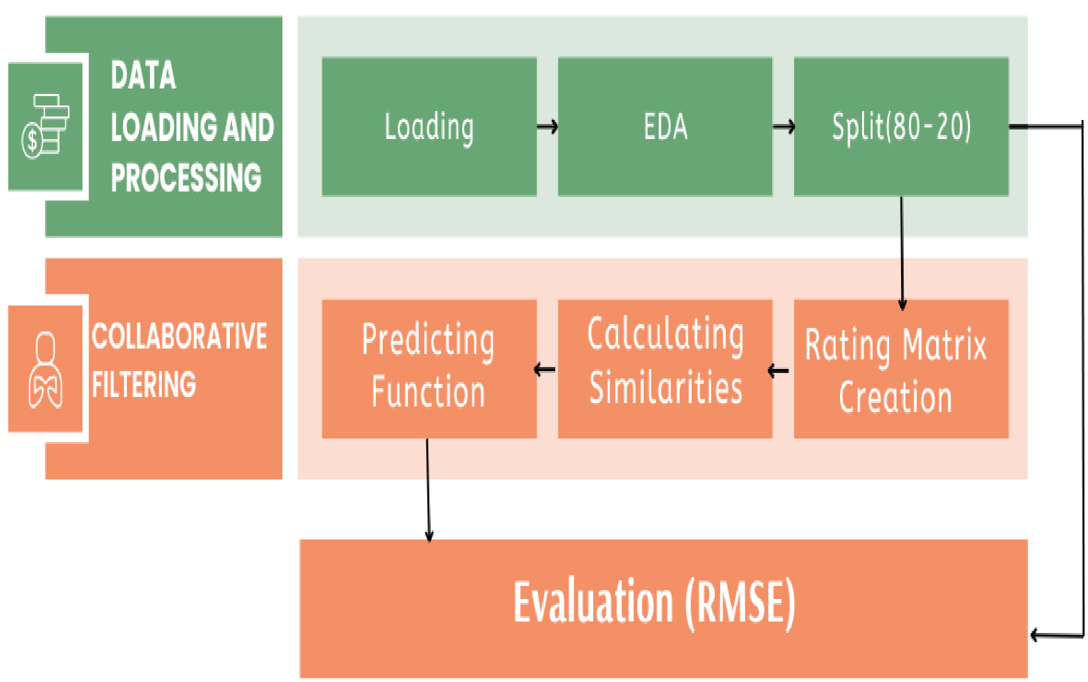

Fig: Recommendation system implementation

6. Discussion of Results

6.1. Five examples of recommendations:

Item Name	Recommended Items
10 Colour Spaceboy Pen	Curious Images Gift Wrap Set, Birdcage Decoration Tealight Holder, Pack of 12 Pink Polkadot Tissues, Hearts Wrapping Tape, Pink Love Heart Shape Cup
Travel Card Wallet Union Jack	Travel Card Wallet Keep Calm, Travel Card Wallet Suki, Travel Card Wallet Skulls, Travel Card Wallet Pantry, Black Mini Tape Measure,
Set of 6 Spice Tins Pantry Design	Set of 3 Cake Tins Pantry Design, Set of 6 Herb Tins Sketchbook, Pantry Rolling Pin, Pack of 12 Traditional Crayons, Traditional Modelling Clay, Pac, Jam Making Set with Jars, Set 2 Pantry Design Tea Towels
Jumbo Bag Red Retrospot	Jumbo Bag Baroque Black White, Jumbo Bag Pink Vintage Paisley, Jumbo Bag Pink Polkadot, Jumbo Bag Strawberry, Jumbo Storage Bag Skulls, Red Retrospot Charlotte Bag, Jumbo Bag Spaceboy Design
Pink Regency Teacup and Saucer	Green Regency Teacup and Saucer, Roses Regency Teacup and Saucer, Regency Cakestand 3 Tier, Button Box, Deluxe Sewing Kit, Feltcraft Doll Molly, Cake Plate Lovebird Pink, Alarm Clock Bakelike Red, Red Retrospot Small Milk Jug, Make Your Own Flowerpower Card Kit

Table 1: Example of item based collaborative filtering in small dataset.

a) **Example1:** As “10 colour spaceboy pen” is a likely a space-themed pen with 10 different colour. The recommended items include various accessories like gift wrap, tealight holder, wrapping tape and a cup, all sharing a similar aesthetic appeal as a spaceboy pen.

b) **Example 2:** The product named “Travel Card Union Jack” has several recommendations related to travels like travel card wallet keep calm, travel card wallet suki, travel card wallet skulls, travel card wallet pantry.

c) **Example 3:** As customer bought spice tines pantry design, the customer was recommended with similar kitchen accessories.

d) **Example 4:** For the product name “Jumbo Bag Red Restrosport” similar bags with the same product ‘Retrospot’ is recommended. Basically, all retro-inspired aesthetic bags are recommended.

e) **Example 5:** This item named “Pink Regency Teacup and Saucer” recommended item includes similar teacup sets with different designs, along with home decorations.

6.2 Metrics with discussion

Approach	**Dataset**	**Value**
Item based	Small dataset	Train data RMSE = 1.240
		Test data RMSE = 1.038
	Big dataset	Train data RMSE = 1.688
		Test data RMSE = 1.291
User based	Small dataset	Train data RMSE = 1.245
		Test data RMSE = 1.0417
	Big dataset	Train data RMSE = 1.7219
		Test data RMSE = 1.348
User based using ALS	Small dataset	Test RMSE = 0.154
User based using ALS	Big dataset	Test RMSE = 0.1402

Table 2: Metrics with discussion

For experiment, we test both items based, and user based collaborative filtering in small and large dataset. Also, we test user based collaborative filtering using Alternating Least Square algorithm to over come the issue of sparsity, cold start problems. Both item-based and user-based approaches performed better on smaller dataset however suffered from higher RMSE values on the bigger dataset, indicating overfitting. To facilitate the scalability of the system we utilize user-based collaborative filtering using ALS. The ALS model delivered lower RMSE values on both small and big datasets indicating the adjustment to the sparsity and scalability towards larger datasets.

7. Conclusion and Recommendations

Recommender systems is a powerful technology for customer satisfaction and efficient business. This project attempts to build a collaborative filter using item-based explicit rating approach and user-based implicit rating approach. It utilizes the rating matrices representing the user-item interaction to serve a foundation for computing similarities between users and items. After explicit rating, cosine similarity is employed to calculate the similarities between items, where higher score indicates the greater similarity and vice versa. The final prediction was made by aggregating the rating of similar items by computing a weighted sum, normalized by the sum of similarity scores. Whereas ALS algorithm is used for matrix factorization to address the issue generated by the explicit rating.

For evaluation, RMSE is used to quantify the accuracy of the predictions. We achieved RMSE score of 1.2403 on the training dataset and 1.0389 on the testing dataset for item based collaborative filtering and RMSE of 0.14 for user-based collaborative filtering suing ALS. Hence, user based performed better than item based.

Collaborative filtering is a very flexible technique to adapt the changes in customer behavior over time making the system dynamic. Besides, collaborative filtering works for any kind of items, we don't need to define a set of features that define the item. On the other hand, collaborative filtering has a cold start problem because it needs enough targets or users in the system to find a match. In addition, the sparseness of the user/product matrix makes it difficult to find users with similar preferences, increasing the "first rate problem" where unrated products are difficult to recommend. Popularity bias further complicates matters by favoring mainstream products, potentially creating filter bubbles and contributing to problems such as radicalization.

To improve these issues, matrix factorization techniques like Alternating Lease Squares (ALS) has been implemented to decompose the user-item interaction matrix into lower-dimensional matrices. This not only helped to capture latent factors but also helped reducing the dimensionality of the data to mitigate the sparsity issue.

Besides, hybrid approaches can be used to combine collaborative filtering with content-based or knowledge-based methods to enhance recommendations system. Another technique could be by handling implicit cues like purchase history or browsing behavior where explicit feedback is not enough. We can use regularization techniques such as L1 or L2 to prevent overfitting in collaborative filtering models. To solve cold start problem in recommendation systems we can address through using popularity-based recommendation where we can suggest popular or trending items. Implementing algorithms such as Thompson Sampling or Upper Confidence Bound (UCB) can help balance exploration when recommending items to users with limited historical data.

8. Reflection

After completing this task of building a recommendation system using collaborative filtering, I acknowledge the comprehensive understanding of data mining algorithms and their practical applications in real-world applications, especially in the business industry. One of the significant take aways is that a developer should be able to understand the business objectives and work around to enhance customer experience. The project utilized collaborative filtering techniques as the choice of this algorithm is well suited for analyzing transaction history and generating personalized recommendations based on customer behavior. I got valuable insights from EDA process which helped me to handle missing values, removing duplicates, and exploring patterns in the data and implement it in the algorithm accordingly. I learned the techniques to evaluate the models such as MAE, RMSE, F1, Recall. Furthermore, I understood different mining techniques like Apriori algorithm, association mining techniques. I found out that even though Apriori algorithm provides valuable insights into the relationship between items purchased together by customer however it cannot provide personalized recommendations like collaborative filtering. Understanding the shortcomings of collaborative filtering led me to explore techniques like matrix factorization.

9. References

Ajitsaria, A. (n.d.). *Build a Recommendation Engine With Collaborative Filtering*. Retrieved from RealPython: https://realpython.com/build-recommendation-engine-collaborative-filtering/

Chaudhary, S. (2022, January). *Collaborative Filtering in Recommender System: An Overview.* Retrieved from Turing: https://medium.com/@evelyn.eve.9512/collaborative-filtering-in-recommender-system-an-overview-38dfa8462b61

Eve, E. (2023, November 5). *Collaborative Filtering in Recommender System: An Overview.* Retrieved from Medium: https://medium.com/@evelyn.eve.9512/collaborative-filtering-in-recommender-system-an-overview-38dfa8462b61

Hermina, C., Gopalakrishnan, B., & Balajishankar, A. (2022, 11 18). *MARKET BASKET ANALYSIS FOR A SUPERMARKET.* Retrieved from Reasearchgate: https://www.researchgate.net/publication/365489098_MARKET_BASKET_ANALYSIS_FOR_A_SUPERMARKET

Kadlaskar, A. (2024, February 26). *Market Basket Analysis: A Comprehensive Guide for Businesses.* Retrieved from Analytics Vidhya: https://www.analyticsvidhya.com/blog/2021/10/a-comprehensive-guide-on-market-basket-analysis/

Li, S. (2019, April 20). *towardsdatascience.* Retrieved from Building a Collaborative Filtering Recommender System with ClickStream Data: https://towardsdatascience.com/building-a-collaborative-filtering-recommender-system-with-clickstream-data-dffc86c8c65

Liao, K. (2018, November 17). *Towards Data Science.* Retrieved from Prototyping a Recommender System Step by Step Part 2: Alternating Least Square (ALS) Matrix Factorization in Collaborative Filtering: https://towardsdatascience.com/prototyping-a-recommender-system-step-by-step-part-2-alternating-least-square-als-matrix-4a76c58714a1

Natsuki Sano, N. M. (2015). *Recommendation System for Grocery Store Considering Data Sparsity.* Retrieved from ScienceDirect: 10.1016/j.procs.2015.08.216

Palvel, S. (2023, September 13). *Recommender Systems using Matrix Factorization.* Retrieved from medium: https://subashpalvel.medium.com/recommender-systems-using-matrix-factorization-14ac3ac43e7a

Sagin, A., & Ayvaz, B. (2018, 05 10). *Determination of Association Rules with Market Basket Analysis: Application in the Retail Sector.* Retrieved from researchgate: 10.21533/scjournal.v7i1.149

Saluja, C. (2018, March 6). *Collaborative Filtering based Recommendation Systems exemplified.* Retrieved from towardsdatascience: https://towardsdatascience.com/collaborative-filtering-based-recommendation-systems-exemplified-ecbffe1c20b1

Sen, S. (2020, July 21). *Evaluating Recommender Systems.* Retrieved from Medium: https://medium.com/the-owl/evaluating-recommender-systems-749570354976

TechTarget Contributor. (n.d.). *market basket analysis.* Retrieved from techtarget: https://www.techtarget.com/searchcustomerexperience/definition/market-basket-analysis