SJIF Impact Factor 2022: 8.197| ISI I.F. Value:1.241| Journal DOI: 10.36713/epra2016 ISSN: 2455-7838(Online)
EPRA International Journal of Research and Development (IJRD)
Volume: 7 | Issue: 5 | May 2022 - Peer Reviewed Journal

MINING FREQUENT PATTERNS AND ASSOCIATION RULES USING SIMILARITIES

V.Sreeja[1] , B.Raghuram[2], A.Vishal[3], S.Sreesurya[4], L.Pavan Kalyan[5], M. Mahanth Preeth Reddy[6]

[1]Department of CSE,KITS Warangal, Telangana, India
[2]Department of CSE,KITS Warangal, Telangana,India
[3]Department of CSE,KITS Warangal,Telangana, India
[4]Department of CSE,KITS Warangal, Telangana, India
[5]Department of CSE, KITS Warangal, Telangana, India
Department of CSE, KITS Warangal, Telanagan, India

Article DOI: https://doi.org/10.36713/epra10267
DOI No: 10.36713/epra10267

ABSTRACT

Because of the intensive usage of database systems, the university library has gathered a mass-circulation historical data. We employed data mining algorithms to reveal useful information in the investigation of circulation data. The book recommender system described in this work provides clients with book recommendations based on an apriori algorithm and association rules applied to transactional data. A web application is introduced and used to provide readers with recommendation information. Bookstores, information retrieval systems, and network reference databases can all benefit from the recommendation model.

KEYWORDS: *Recommendation system, Apriori algorithm, association rule.*

1. INTRODUCTION

Books are a significant source of information for anyone, particularly students at big colleges. Libraries are the primary source of knowledge, which is in high demand. However, as the quantity of books circulates and increases, so does the number of borrowers and information about the books, making book management increasingly difficult. Borrowers can study more successfully and efficiently manage their books with the help of an excellent book management system for information recommendation. The Book Recommendation System is software that helps customers find comparable books based on their previous purchases or interests. This system examines a big book database and generates a list of books that fit the buyer's requirements. Most ecommerce businesses now use recommendation algorithms to entice customers to spend more by suggesting things that they are likely to like. It primarily employs two types of information filtering: content-based and collaborative filtering.

Here we proposed an effective book recommender system for stakeholder needs. The Apriori algorithm, a commonly utilised Association rule method, is used in this system. The algorithm is evaluated using a synthesised dataset of 2000 records.

2. LITERATURE SURVEY

Maalej and Thurimella [1] presented a few general ideas for using recommender systems, investigated the potentials for developing recommender systems, and evaluated the semantic similarity of the obtained items.

In India, Smita R.Sankhe works as a writer. Kavita Kelkar et al. [2] investigated how to improve the efficiency of association rule creation by reducing the execution time of traditional Apriori and an updated Apriori method.

Patel Tushar S, Panchal Mayur, and colleagues [3] investigated a few essential elements as well as the inner workings of several mining methods. Apriori, DHP, Eclat, FP-growth, and H-mine algorithms were evaluated for their strengths and limitations. When the complexity of the need selection transcends a user's ability to scan and choose, Ninaus et al. postulated in [4] that recommender systems aid in the discovery of related requirements. The two most common types of recommendation systems tested were collaborative and content-based filtering.

3. ANALYSIS OF SYSTEM

3.1 PROPOSED SYSTEM

We use collaborative filtering and association rule mining to recommend books to customers in the proposed system. We recommend goods to one person based on comparable users' interests via collaborative filtering. We create book store web application and Customers' purchase histories were used to create a transaction book dataset. On the transactioned dataset obtained, the Apriori Algorithm is used. Filter the data that is less commonly purchased by consumers with the minimum of support and confidence using association rule mining. Rank and recommend the products to the customers.

3.2 ADVANTAGES OF PROPOSED SYSTEM

This system offer products to users based on their likely interests, and it is vital to alleviate the search and selection pressures caused by the expanding volume of item data. It also aids profit-making for retailers.

3.3.1 SYSTEM ARCHITECTURE

We used a Kaggle book dataset with 6000 records for our study. The missing and null values are then removed from the dataset, which is subsequently analysed and cleaned. We receive a dataset of 2000 records after conducting data pre-processing, which we employ for our research. This dataset's books have been uploaded to an ecommerce web app called Online Book Store. Customers order books from a variety of categories, and the details of the books ordered are saved in the database.

Now we'll take a look at the transactioned dataset, which includes order-ids and book-ids for books purchased in that order. The Apriori algorithm is used on a transactioned dataset to obtain support and confidence for each item using association rules.

3.3.1 ALGORITHM

In our project, Apriori algorithm is applied on the transactioned dataset by using minimum support and confidence to get frequent itemset. Support and confidence can be calculated by applying association rules.

3.3.1.1 ASSOCIATION RULE MINING

It is based on the IF-THEN rule. It's all about building rules.. It's a two-part process:

1. To find frequent itemsets, algorithms such as Apriori and FP-growth are used.
2. Using a large number of often occurring items to create strong association rules. Association rules with a high level of support and confidence are considered strong.

According to the association rule AB, if someone buys A, they must also buy B.

Support, confidence, and lift are all measures of association.

Support: The amount of times an itemset exists in all transactions is represented by this metric. Support is usually expressed as a percentage of total transactions in which the itemset appears.

Support = freq(A, B)/N,

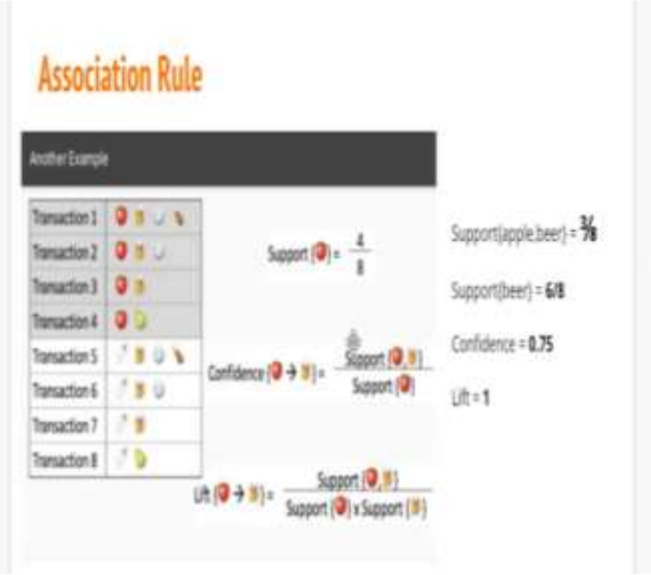
here N is the number of transactions done.

Confidence: This metric displays how often item B is purchased with item A. It refers to the dependent likelihood of the subsequent occurring given the antecedent which means is a data item, in technical words. A consequence is a thing discovered in combination with the antecedent.

The degree of confidence in an association rule with a frequently occurring outcome is always high.

Freq(A, B)/ freg=confidence (A)

Lift: Lift decides the likelihood that thing B will be bought if thing A is bought, considering the popularity of thing B.



The lift of apple - > beer is one in Table, showing that there is no connection between the two. Assuming thing X is bought, a lift esteem bigger than one demonstrates that thing Y is probably going to be bought, while a worth short of what one shows that thing Y is less likely to be bought.

3.3.1.2 APRIORI ALGORITHM

This was created by R. Agrawal. This method is used to identify a set of frequently used products. It's an iterative strategy for locating frequently occuring itemsets. Frequent itemsets are itemsets that appear frequently in a dataset. It's a prune-and-join approach. The basic purpose of the apriori algorithm is to construct a rule of association between distinct items. The relationship between two or more entities is defined by the association rule. On a large database with a significant number of transactions, the Apriori method is often utilised.

It's based on the Apriori property, which claims that something already exists before it's made. Every non-empty subsets of the itemset which is frequent should likewise be frequent. If {1,2,3} is a frequently occurring itemset, then

{1},{2},{3},{1,2},{1,3},{2,3} must also be frequently occurring.

Consider the following dataset

TID	Itemset
10	I1, I2, I5
20	I1, I5
30	I1, I2, I3, I5
40	I2, I5
50	I5, I3
60	I1, I5, I4
70	I2, I3, I5

Table 1: Sample Dataset

Let min support is 2
Step-1:
Now generate the candidate item set
Here we get table C1

Itemset	Sup
I1	4
I2	4
I3	3
I4	1
I5	7

Fig 2: Candidate Set

Then compare the candidate support count to the minimum support level, and remove the itemset that have less support than the minimum.

Itemsets	Sup count
I1	4
I2	4
I3	3
I5	7

Fig 3: L1

Step-2
To receive a set of candidate itemset, join the itemset. When l1 and l2 are considered itemset in l(k-1), the itemset generated by connecting them.

Itemset	Sup count
I1, I2	2
I1, I3	1
I1, I4	1
I1, I5	3
I2, I3	2
I2, I4	0
I2, I5	4
I3, I4	0
I3, I5	3
I4, I5	1

Fig 4:

Check the candidate support count against the minsup count, then delete the itemsets with fewer support.

Itemset	Sup count
I1, I2	2
I1, I5	3
I2, I3	2
I2, I5	4
I3, I5	3

Fig 5: Itemsets Greater Than Min Support

Continue the same process until there are no frequent itemset found and finally we get,

Itemset	Sup count
I1, I2, I5	2
I2, I3, I5	2

Fig 6: Final Itemset

Confidence is calculated as,
So here, by taking an example of any frequent itemset, we will show the rule generation.
Consider itemset {I2, I3, I5}
Its non-empty itemsets are
{I2, I3}, {I2, I5}, {I3, I5}, {I2}, {I3}, {I5}.

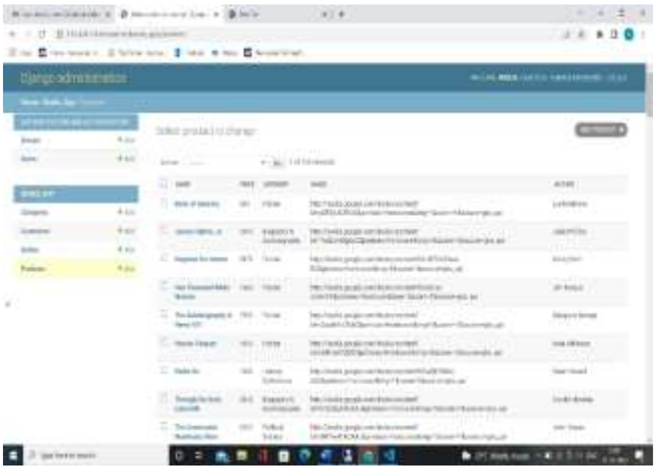
the association rules are,
{I2, I3}→I5, confidence=2/2=100 percent
{I1, I5}→ I2, and confidence=2/2=100%
{I5, I2}→I3, 2/2=100 percent confidence
I3 →(I2, I5); confidence=2/6=33%
I2→(I3, I5); confidence=2/7=29%
I5→ (I3, I2), confidence = 2/2=100%
Only the first, second, third, and last rules are generated if the minimum confidence is 70%.

Advantages of Apriori algorithm:

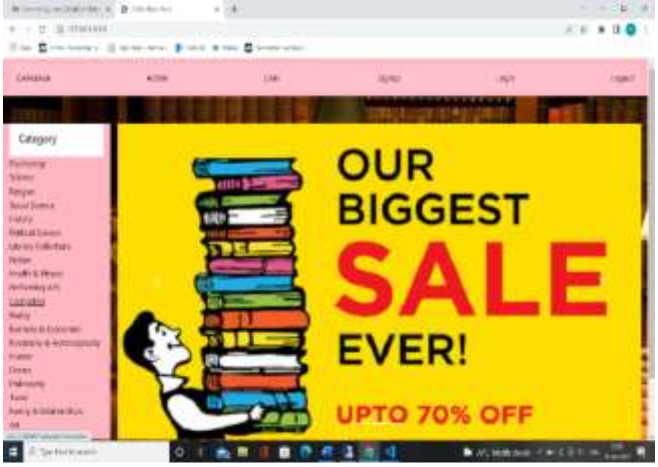
It's easy to put into action. Make the most of the large itemset option.
Disadvantages of Apriori algorithm:

The apriori algorithm is a time-consuming method of locating support because it must run over the full database. It gets computationally more expensive when a large number of candidate rules is required.

Initially, an online book store is established as an ecommerce web application using the Django framework. It has 20 different categories of works from diverse authors.

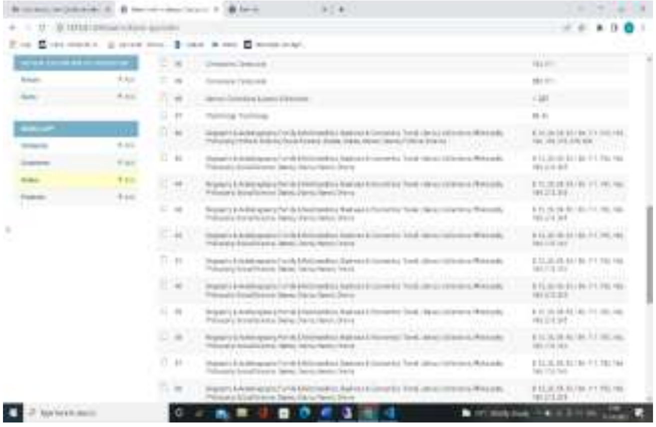


Books

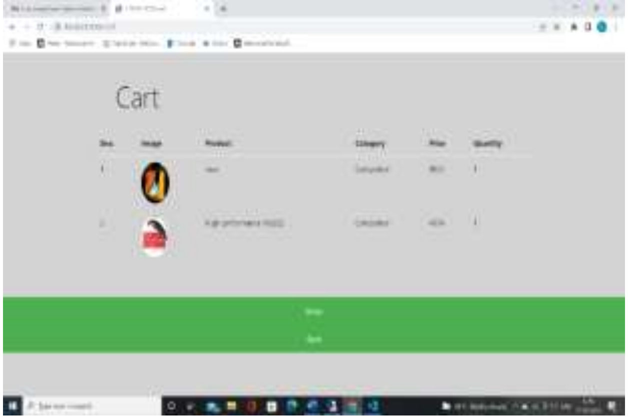


Home page

Customers can now order certain book categories depending on their interests. Ordered details, such as order id, category, and book ids of books purchased, are maintained as transactional data.

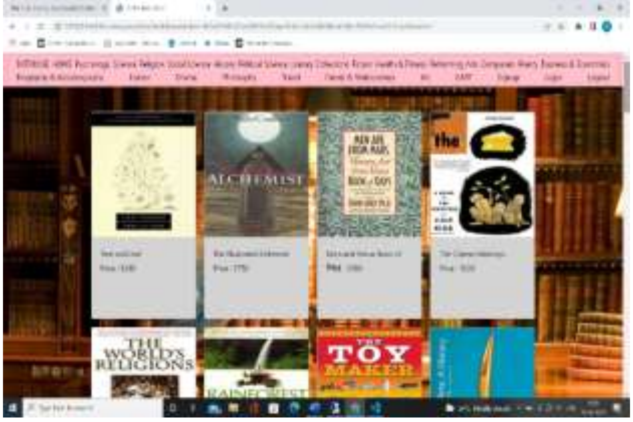


On the transaction dataset, the Apriori algorithm is used. A minimum level of support and confidence is established. Support is determined for all things, and items with support less than the minimal support are eliminated. This produces frequent items. Then, over things, apply association rules to gain confidence. In terms of consequences, this yields more likely purchased items.



Ordered books

The buyer is then shown books from the commonly purchased book category as recommended books.

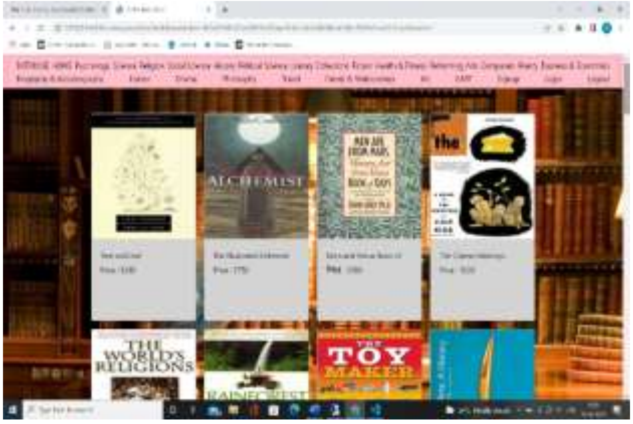


4. EXPERIMENT

The descriptions and outcomes of experiments are presented in this section. When a consumer buys a book, the details of the order are saved in the backend (order id, book ids of books bought in that order, categories of books ordered).

On the transactioned dataset obtained, the Apriori technique is used to obtain frequently purchased books and recommend the related books.

SAMPLE OUTPUT OF THE PROJECT



Recommended books

5. CONCLUSION

In this Project, we recommend products to customers and conclude that Apriori algorithm works better for large datasets and gives frequent items more efficiently.

6. FUTURE ENHANCEMENT

In order to progress the conversation in the field of recommendation systems about future generations of techniques, we also want to design an algorithm that overcomes the limits of previously developed recommender systems.

REFERENCES

1. *W. Maalej and A. K. Thurimella, "Towards a research agenda for recommendation systems", in Managing the Requirements knowledge (MARK), 2009 International Workshop on, pp. 32-39, IEEE 2009.*
2. *Smita R. Sankhe, Kavita Kelkar, Optimization of Execution Time using Association Rule Mining Algorithms International Journal of Computer Applications (0975 – 8887) Volume 59– No.11, December 2012 18.*
3. *Patel Tushar S.1, Panchal Mayur2, Ladumor Dhara2, Kapadiya Jahnvi2, Desai Piyusha2, Prajapati Ashish3 and Prajapati Reecha4, Association An Analytical Study of Different Frequent Itemset Mining Algorithms, Research Journal of Computer and Information Technology Sciences Vol. 1(1), 6-9, February (2013) Res. J. Computer & IT Sci. International Science Congress.*
4. *G. Ninaus, A. Felfernig, M. Stettinger, S. Reiterer, G. Leitner, L. Weninger, and W. Schanil, "Intellireq: Intelligent techniques for software requirements engineering.," in ECAI, pp. 1161–1166, 2014.*
5. *R. Burke, "Hybrid recommender systems: Survey and experiments," User modeling and user-adapted interaction, vol. 12, no. 4, pp. 331– 370, 2002.*