IJRAR.ORG E-ISSN: 2348-1269, P-ISSN: 2349-5138

INTERNATIONAL JOURNAL OF RESEARCH AND ANALYTICAL REVIEWS (IJRAR) | IJRAR.ORG

An International Open Access, Peer-reviewed, Refereed Journal

FREQUENT ITEMSET MINING FOR MARKET BASKET ANALYSIS

[1]Pranshu shukla,[2]Yeshiv Sahu,[3] Avinash Kashyap,[4]Shubham verma,[5]Lalita Panika

[1]B.Tech Student, [2]B.Tech Student, [3]B.Tech Student, [4]B.Tech Student,

[5]Assistant Professor

[1]Computer Science Engineering

[1]Bhilai Institute of Technology Raipur,Chhattisgarh India

Abstract: Frequent item mining is a crucial task in data mining, which involves identifying sets of items that frequently appear together in a given dataset. By identifying frequent itemsets, or sets of items that frequently appear together in a given dataset, frequent item mining enables the analysis of market baskets, web clickstreams, and other data sources that exhibit transactional behavior. These frequent itemsets can then be used to derive valuable insights and knowledge from the data, such as identifying association rules, recommending products, and predicting customer behavior. In this paper, we provide an overview of frequent item mining, including its definition, importance, and various algorithms. We focus on two popular algorithms for frequent item mining, Apriori and FP-Growth, and compare their advantages and disadvantages.

Keywords:- Frequent item mining, Apriori algorithm, Market basket analysis, Support, Confidence, Lift .

1.INTRODUCTION :

In todays data-driven world, organizations and businesses are generating massive amounts of data. While this data can provide valuable insights and help make informed decisions, its sheer size and complexity make it challenging to extract useful information through manual analysis. To address this issue, data mining techniques like frequent item mining are becoming increasingly important. By using algorithms to identify patterns and relationships within the data, these techniques enable organizations to extract meaningful insights and make data-driven decisions. Therefore, frequent item mining and other data mining techniques are becoming essential tools for businesses to stay competitive in today's data-driven world.

1.1 Introduction to Itemset :

An itemset is a group of two or more items, where an item can be any element in a collection X. If an itemset has k-items, it is called a k-item set. An itemset S is a subset of itemset X if all the items in S are also in X. A frequent item is an item that appears regularly.

1.2 Introduction to the Frequent Item Set:

A frequent item is a subset of an item that occurs with a certain regularity in a dataset. A frequent itemset combines elements that occur repeatedly together. A set of items is considered frequent if it meets a minimum threshold of support and confidence. Support displays transactions in which multiple items were purchased in a single transaction. Confidence displays transactions in which the items are purchased sequentially. We consider only those transactions that meet the minimum threshold support and confidence requirements for the frequent item mining method. For example, a set of mobile supplies such as "backcovers,", "screenguards,", and "chargers" is a frequent item if enough people purchase it.

1.3 Frequent Item Mining:

Frequent item mining is a powerful data mining technique that involves identifying sets of items that frequently appear together in a given dataset. This technique has been widely applied in various domains such as market basket analysis, web usage analysis, healthcare, social networks, and many more. The identification of frequent itemsets allows analysts to discover hidden patterns and relationships in the data that might be difficult to discern through other means. These insights can be used to make informed decisions, enhance customer experience, improve business performance, and advance scientific research.

1.4 Why Frequent Itemset Mining? :

A frequent method called itemset mining or pattern mining is used to extract relationships from patterns in client shopping behaviour in physical and online establishments in order to find association rules. It aims to identify groups of items that are frequently purchased together. The frequency of purchases of related products is shown by association rules. After the guidelines are set, these collections of comparable products can be used to arrange the products on display shelves in brick-and-mortar stores or online shops to their best advantage, or they can even suggest which products should be tightly grouped.

1.5 Fundamentals of Itemset Mining:

Basic explanations of item-set mining methods are provided below:

Support: This is a measurement of the popularity of a product in a transaction database. Support is calculated by dividing the total number of transactions in a database by the number of transactions of a specific item.

Confidence: Confidence suggests that it's possible that the customers purchased two goods at once. To gain confidence, divide the total number of transactions in a database by the number of transactions with both items.

Lift: Lift is the measure of the strength between the two items in an itemset. It is the ratio of the support of the itemset to the product of the support of the individual items in the itemset. Lift values greater than 1 indicates positive association while values less than 1 indicates a negative association between the items.

1.6 Market Basket Analysis :

Market basket analysis is a data mining approach used by retailers to boost sales by better understanding customers and their buying habits. In simple terms, a market basket analysis in data mining is to analyse the combination of products that have been bought together.The market basket analysis mainly works with the association rule "if" and "then.

- IF denotes an antecedent, which is a component or part of the data.
- THEN denotes the consequent, something that is discovered along with the antecedent.

1.7 Apriori Algorithm:

The first algorithm to be suggested for frequent item set mining was the apriori algorithm. Later, it was enhanced by R Agarwal and R Srikant, and the result was known as Apriori. The "join" and "prune" steps of this algorithm are used to reduce the search space. Finding the most common item sets is done iteratively. The algorithm is known as Apriori since it makes use of frequent item qualities that are known in advance.

2. LITERATURE REVIEW:

This section consists of the literature review.

“YAFIM is a Spark-based version of the Apriori algorithm.” Spark was created specifically for in-memory parallel computing in order to support iterative & interactive data mining. YAFIM achieved an 18x speedup on average for all apriori-based algorithms that were implemented over MapReduce. YAFIM creates the candidate itemsets in a separate step by using a Cartesian product between (k-1 frequent itemset) & (k-1 frequent itemset), then iterates through the dataset (transaction by transaction) to determine the support of each candidate. [1]

“Deep Frequent Itemset Mining with Dynamic Memory Networks” by Wei Zhang, Jun Zhou, and Xiaolong Jin. This paper proposes a new approach to frequent itemset mining that utilizes dynamic memory networks (DMNs) to capture the relationships between items in a dataset. The DMNs are used to generate a distributed representation of the items, which is then used to identify frequent itemsets. The proposed method achieves state-of-the-art performance on several benchmark datasets.[2]

"A Parallelized Approach for Mining Frequent Itemsets from Big Data" by Aman Kumar and Ritu Garg. This paper presents a parallelized approach for frequent itemset mining that is designed to handle large-scale datasets. The approach uses the MapReduce framework to distribute the mining process across multiple nodes, which improves scalability and performance. The proposed approach is evaluated on several datasets and compared with existing methods.[3]

"Frequent Itemset Mining Using Particle Swarm Optimization with Local Search" by Ahmed M. Zeki, Sabah M. Hassan, and Yasir Ali Omar. This paper presents a hybrid approach to frequent item mining that combines particle swarm optimization (PSO) with local search. The PSO algorithm is used to explore the search space and find promising candidate itemsets, while local search is used to refine the solutions and improve the overall performance.[4]

"Efficient Parallel Mining of Maximal Frequent Itemsets on Hadoop" by Junchen Hu, Hui Liu, and Jiajia Sun. This paper presents a parallelized algorithm for mining maximal frequent itemsets using the Hadoop platform. The approach improves performance by using multiple MapReduce jobs to efficiently filter out infrequent itemsets and reduce the size of the search space.[5]

“Online Frequent Pattern Mining Using Multi-Scale Convolutional Neural Networks” by Xiaodong Xu, Wenpeng Feng, and Xiaolin Zhang. This paper proposes an online approach to frequent pattern mining that uses multi-scale convolutional neural networks (CNNs) to process streaming data. The CNN-based approach achieves high accuracy and can adapt to changes in the data distribution over time. The proposed method is evaluated on several real-world datasets and compared with existing methods.[6]

"Efficient and Scalable Frequent Itemset Mining on GPUs" by Yunqi Zhang, Jun Wang, and Wei Xu. This paper proposes a GPU-accelerated approach for frequent itemset mining that achieves high performance and scalability. The approach uses a parallelized implementation of the Apriori algorithm and takes advantage of the high memory bandwidth and computational power of GPUs. The proposed method is evaluated on several large-scale datasets and compared with existing methods.[7]

"Leveraging the Strength of Online Gradient Descent for Frequent Itemset Mining" by Zhendong Chu, Jia Chen, and Wei Wang. This paper proposes an online gradient descent-based approach for frequent itemset mining that can handle streaming data. The approach uses a novel update rule to dynamically adjust the learning rate and momentum of the algorithm, which improves the accuracy and efficiency of the mining process. The proposed method is evaluated on several real-world datasets and compared with existing methods.[9]

3. *METHODOLOGY:*

The frequent item set mining for market basket analysis system is built by the techniques namely apriori algorithm , association rule mining ,fp growth algorithm. In the suggested system, users interact with a web portal, providing data that is then recorded in a raw log file after going through a number of preprocessing, data cleaning, and other procedures to separate the useful information from the raw data and put it into a structured format. After that, the clean data is used to find pattern find patterns and recommend the top n products to each user. The system architecture is depicted below in:

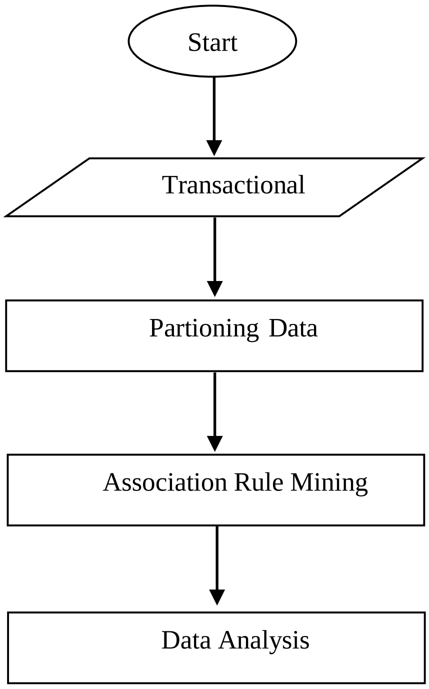


Fig. 4.1 Methodology

Proposed work as follow:

a) Data acquisition - Extraction of relevant information, data transformation, and loading of the data into the target program are all steps in the data acquisition process.
b) Data Cleaning- data that is collected during acquisition phase may have null value, missing data etc so those inaccurate record must be corrected to perform data mining techniques.
c) Data mining- After data cleaning the data mining techniques are applied on it.
d) The Apriori algorithm working can be explained on the basis of the below steps:

Apriori algorithm was the first algorithm for finding the frequent item sets and association rule mining. The Apriori algorithm is divided in two major steps: join and prune.

The new candidate seta is generated in the join step. Depending on the support count, the candidate set can be defined as frequent or infrequent. For generating higher level candidate itemsets (Ci) previous level frequent itemsets Li-l are joined. In the pruning step, the infrequent candidate item sets are filtered out. This step ensures that every subset of a frequent itemset is also frequent. Hence, if the candidate item set contains more infrequent item sets, will be removed from the process of frequent itemset and association mining. This process is called pruning.Apriori Algorithm Input D. , a database of transactions Min_sup, the minimum threshold support Output L.K. Maximal frequent item sets in D Ck Set of Candidate k-itemsets

Method:

- L1 = Frequent items oflength l.
- For (k = 1; Lk! = ; k ++) do.
- Ck + 1 = candidates generated from Lk.
- For each transaction t in database D do.
- Increment the count of all candidates in Ck + 1 that are contained in t.
- Lk + 1 = candidates in Ck + 1 with minimum support.
- End do.
- Return the set Lk as the set of all possible frequent item sets The main notation for association rule mining that is used in Apriori algorithm is the following:

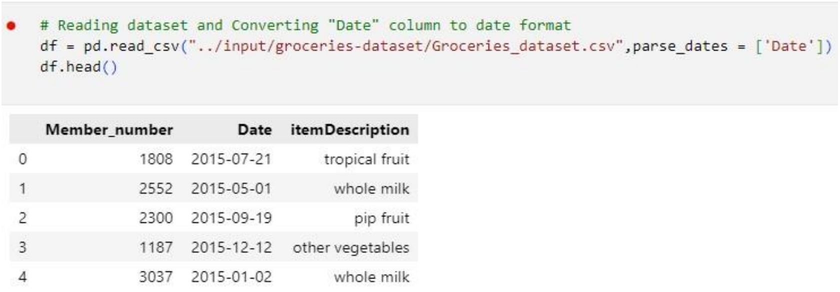
- A k -itemset is a set ofk items.
- The set Ck is a set of candidte k-itemsets that are potentially frequent.
- The set Lk is a subset of Ck and is the set of k-itemsets that are frequent.

4. RESULT AND DISCUSSION:

For this project work Python is used to develop a recommendation analysis, and the IDE used for executing the code is anaconda prompt. To perform a Market Basket Analysis implementation with the Apriori Algorithm, we have used the Groceries dataset which has been downloaded from Kaggle. The dataset contains following properties:

- Member _number: It indicates the ID of particular user.
- Date: It provides the purchasing date of the user.
- ItemDescription: Basically describes the type of product of the grocery.

Step-1: The first step is to read the dataset from its path.



```
# Reading dataset and Converting "Date" column to date format
df = pd.read_csv("../input/groceries-dataset/Groceries_dataset.csv",parse_dates = ['Date'])
df.head()
```

	Member_number	Date	itemDescription
0	1808	2015-07-21	tropical fruit
1	2552	2015-05-01	whole milk
2	2300	2015-09-19	pip fruit
3	1187	2015-12-12	other vegetables
4	3037	2015-01-02	whole milk

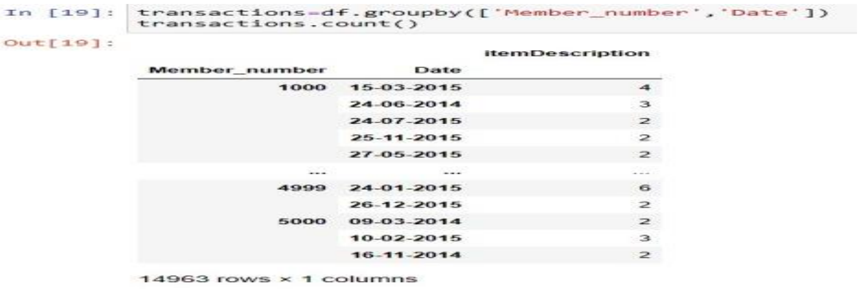
Fig. 5.1 Reading Groceries Dataset (from kaggle)

Step-2: Checking of any null values in the dataset.

No null values were found in taken dataset.

Step-3: Calculate the number of transaction.

Here the number of transaction done by each customer is calculated. Example- a customer_number 1000 bought 3 items in 2014-06-21.



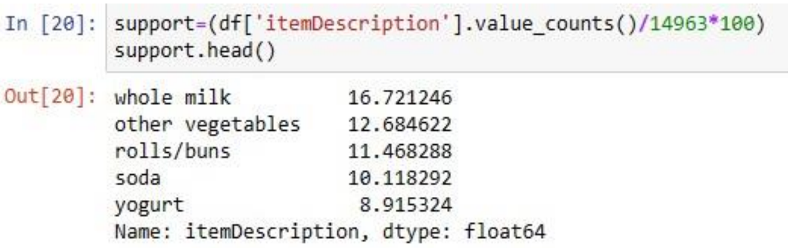
```
In [19]: transactions=df.groupby(['Member_number','Date'])
         transactions.count()
Out[19]:
```

		itemDescription
Member_number	Date	
1000	15-03-2015	4
	24-06-2014	3
	24-07-2015	2
	25-11-2015	2
	27-05-2015	2
...	...	...
4999	24-01-2015	6
	26-12-2015	2
5000	09-03-2014	2
	10-02-2015	3
	16-11-2014	2

14963 rows × 1 columns

Fig. 5.3 Number of transactions

Step-4: Calculate support count of each item.



```
In [20]: support=(df['itemDescription'].value_counts()/14963*100)
         support.head()

Out[20]: whole milk          16.721246
         other vegetables    12.684622
         rolls/buns          11.468288
         soda                10.118292
         yogurt               8.915324
         Name: itemDescription, dtype: float64
```

Fig. 5.4 Support count of each item

Step-5: Represent top 10 products .

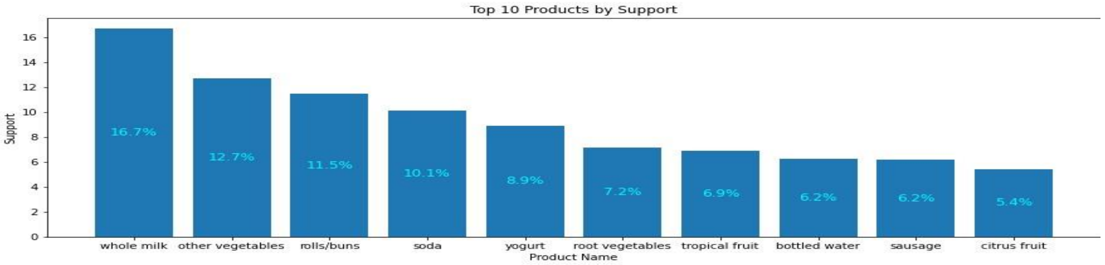
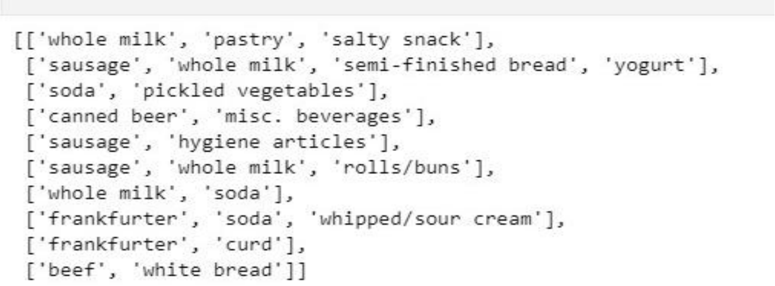


Fig. 5.5 Top 10 product in our dataset.

Step-6: Convert the number of transactions into a list.

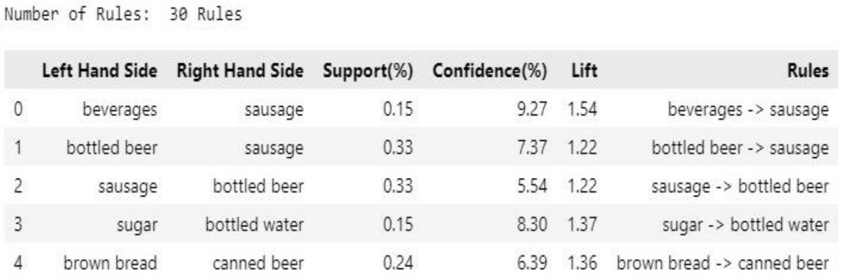


```
[['whole milk', 'pastry', 'salty snack'],
 ['sausage', 'whole milk', 'semi-finished bread', 'yogurt'],
 ['soda', 'pickled vegetables'],
 ['canned beer', 'misc. beverages'],
 ['sausage', 'hygiene articles'],
 ['sausage', 'whole milk', 'rolls/buns'],
 ['whole milk', 'soda'],
 ['frankfurter', 'soda', 'whipped/sour cream'],
 ['frankfurter', 'curd'],
 ['beef', 'white bread']]
```

.

Fig. 5.6 Transactions in form of list

Step-7: Creating a new dataframe of rules and calculating number of rules.



```
Number of Rules:  30 Rules
```

	Left Hand Side	Right Hand Side	Support(%)	Confidence(%)	Lift	Rules
0	beverages	sausage	0.15	9.27	1.54	beverages -> sausage
1	bottled beer	sausage	0.33	7.37	1.22	bottled beer -> sausage
2	sausage	bottled beer	0.33	5.54	1.22	sausage -> bottled beer
3	sugar	bottled water	0.15	8.30	1.37	sugar -> bottled water
4	brown bread	canned beer	0.24	6.39	1.36	brown bread -> canned beer

.

Fig. 5.7 Dataframe of rules

Step-8: Visualizing the association rules.

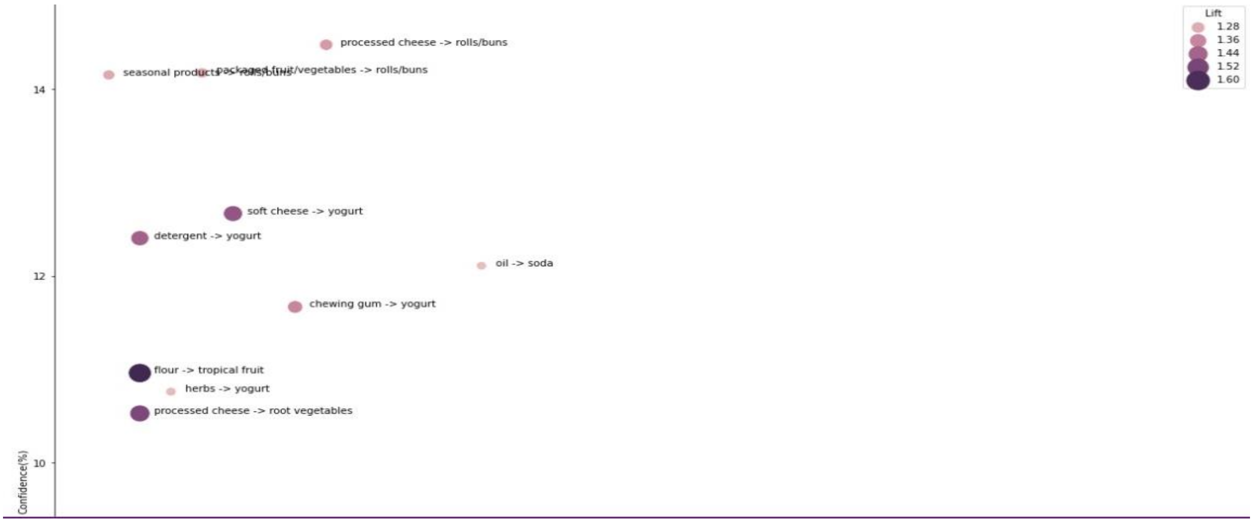
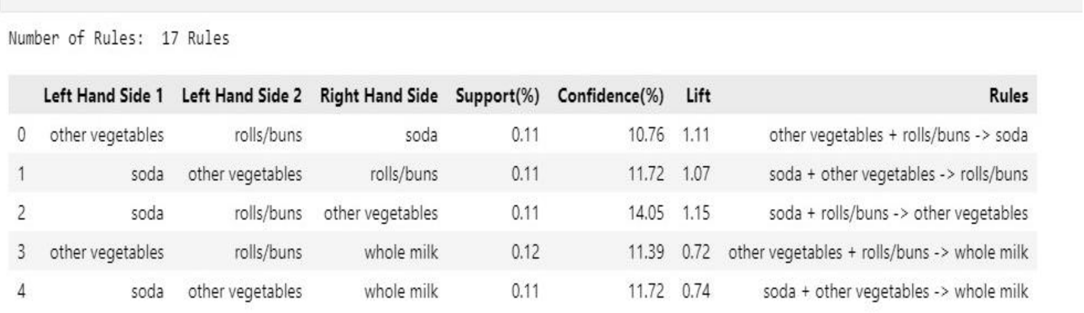


Fig. 5.8 2D scatter plot of association rule

Step-9: Creating a new dataframe of new rules and calculating number of rules.



```
Number of Rules:  17 Rules
```

	Left Hand Side 1	Left Hand Side 2	Right Hand Side	Support(%)	Confidence(%)	Lift	Rules
0	other vegetables	rolls/buns	soda	0.11	10.76	1.11	other vegetables + rolls/buns -> soda
1	soda	other vegetables	rolls/buns	0.11	11.72	1.07	soda + other vegetables -> rolls/buns
2	soda	rolls/buns	other vegetables	0.11	14.05	1.15	soda + rolls/buns -> other vegetables
3	other vegetables	rolls/buns	whole milk	0.12	11.39	0.72	other vegetables + rolls/buns -> whole milk
4	soda	other vegetables	whole milk	0.11	11.72	0.74	soda + other vegetables -> whole milk

Fig. 5.9 Number of rules.

Step-10: Plotting support, confidence and lift of new rules.

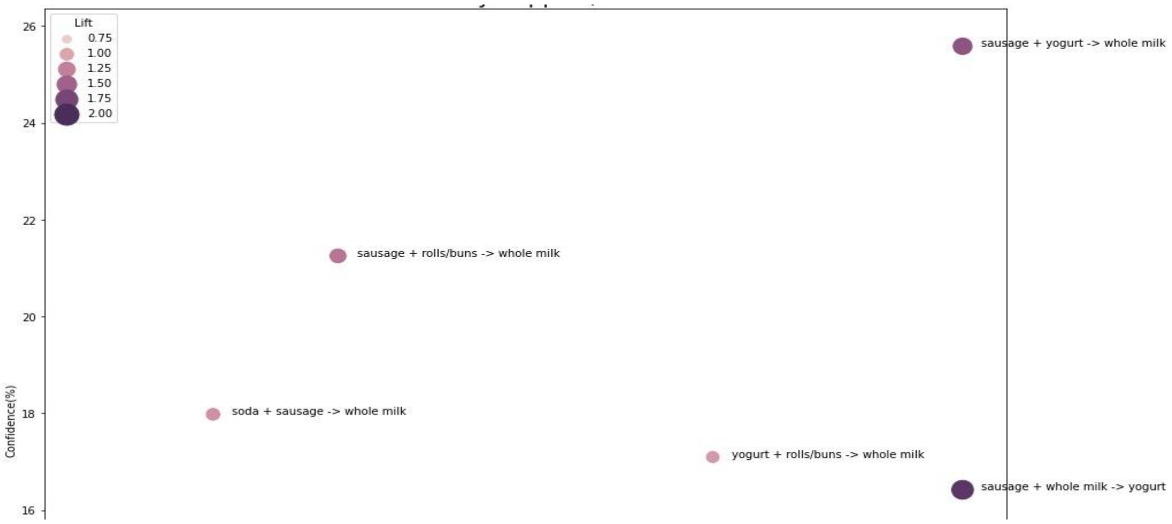


Fig. 5.10 Plotting support, confidence and lift of new rules

5. CONCLUSION:

Frequent item mining is a powerful data mining technique used to identify the most common items or itemsets in a dataset. It can help uncover important patterns and relationships in large datasets and has numerous applications in various industries and research fields, including market basket analysis, recommendation systems, and web mining.One of the most popular algorithms for frequent item mining is the Apriori algorithm, which works by generating candidate itemsets of increasing size and counting the support of each candidate itemset. The support of an itemset is the proportion of transactions in the dataset that contain the itemset. Candidate itemsets with support below the threshold are discarded, while frequent itemsets are retained and used to generate new candidate itemsets.Association rules are another important concept in frequent item mining, which represent the relationships between two or more items that frequently occur together in a dataset. The strength of an association rule is measured by its support and confidence, which are the proportions of transactions that contain both A and B, and the proportion of transactions that contain A that also contain B, respectively.

6. REFERENCES:

[1] Qiu, H., Gu, R., Yuan, C., & Huang, Y. (2014). Yafim: “A parallel frequent itemset mining algorithm with Spark.” *2014 IEEE International Parallel & Distributed Processing Symposium Workshops*.

[2] Wei Zhang, Jun Zhou, and Xiaolong Jin. (2021). “Deep Frequent Itemset Mining with Dynamic Memory Networks.” IEEE Transactions on Knowledge and Data Engineering.

[3] Aman Kumar and Ritu Garg. (2021). "A Parallelized Approach for Mining Frequent Itemsets from Big Data." IEEE Access.

[4] Ahmed M. Zeki, Sabah M. Hassan, and Yasir Ali Omar.(2021). "Frequent Itemset Mining Using Particle Swarm Optimization with Local Search." International Journal of Swarm Intelligence and Evolutionary Computation.

[5] Junchen Hu, Hui Liu, and Jiajia Sun.(Jan. 2021). "Efficient Parallel Mining of Maximal Frequent Itemsets on Hadoop." Journal of Grid Computing.

[6] Xiaodong Xu, Wenpeng Feng, and Xiaolin Zhang.(Jan. 2023) "Online Frequent Pattern Mining Using Multi-Scale Convolutional Neural Networks." IEEE Transactions on Knowledge and Data Engineering.

[7] Zhang, Y., Wang, J., & Xu, W. (2021). “Efficient and Scalable Frequent Itemset Mining on GPUs.” IEEE Transactions on Knowledge and Data Engineering, 33(5), 1965-1979.

[8] Zhang, Y., Lu, J., & Xiao, X. (2022). “Fast and Accurate Frequent Itemset Mining with Adaptive Bloom Filters.” IEEE Transactions on Knowledge and Data Engineering, 34(1), 216-229.

[9] Chu, Z., Chen, J., & Wang, W. (2022). “Leveraging the Strength of Online Gradient Descent for Frequent Itemset Mining.” IEEE Transactions on Knowledge and Data Engineering, 34(2), 433-447.