International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

Enhancing Retail Strategies through Apriori, ECLAT& FP Growth Algorithms in Market Basket Analysis

Harshali Patil
Department of Computer Engineering
Thakur College of Engineering & Technology
Mumbai, India
harshali.patil@thakureducation.org

Tarunima Mukherjee
Department of Computer Engineering
Thakur College of Engineering & Technology
Mumbai, India
tarunima.mukherjee@tcetmumbai.in

Kanak Pandit
Department of Computer Engineering
Thakur College of Engineering & Technology
Mumbai, India
1032200176@tcetmumbai.in

Hiteshu Jani
Department of Computer Engineering
Thakur College of Engineering & Technology
Mumbai, India
1032200188@tcetmumbai.in

Pawan Kumar Jha
Department of Computer Engineering
Thakur College of Engineering & Technology
Mumbai, India
1032200229@tcetmumbai.in

Vinit Agarwal
Department of Computer Engineering
Thakur College of Engineering & Technology
Mumbai, India
1032200290@tcetmumbai.in

Abstract—-"Market basket analysis" is a method employed in data mining to discover items that are commonly bought together by customers in a retail store. It is a crucial tool for retailers to understand consumer purchasing behavior and to improve their sales and marketing strategies. In this research paper, we present a comprehensive study on market basket analysis using three popular algorithms: Apriori, ECLAT, and FPGrowth. The paper begins with a brief synopsis of market basket analysis and the techniques adopted for itemset mining. We then introduce the dataset used in this study, which consists of real-life transaction data collected from a retail store. Next, we perform a thorough evaluation of the Apriori, ECLAT, and FPGrowth algorithms in terms of their computational time and the quality of the association rules generated. The results show that the FPGrowth algorithm is the fastest of the three algorithms, while the Apriori algorithm generates the most comprehensive and high-quality association rules.In addition, we also present a comparison of the performance of these algorithms that involve different assessment criteria like support, confidence, and lift. Our study highlights the importance of selecting the appropriate algorithm for market basket analysis depending on the specific requirements and constraints of the task. The paper concludes with an analysis on the limitations and future directions of research in this area. Overall, our study provides insights into the strengths and weaknesses of the Apriori, ECLAT, and FPGrowth algorithms and functions as a valuable resource for professionals and researchers in the field of market basket analysis.

Keywords- Apriori, ECLAT, FPGrowth, Market Basket Analysis, Support, Confidence, Lift, Leverage

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

I. INTRODUCTION

In recent times, “data mining” has garnered significant interest both within the information field and in society at large. This raising interest is primarily attributed to the abundant availability of vast datasets and the pressing requirement to transform such data into valuable insights and knowledge. Data mining, as a concept, involves the retrieval and extraction of knowledge from extensive data sources. [1] Numerous data mining techniques are instrumental in the processing and examination of such data to derive valuable insights. Association rule mining, in particular, assumes a pivotal role in various applications, including market basket analysis and business-related tasks. [2] Market basket analysis is a method used by businesses to understand customer buying habits. It involves examining transaction data to discover which products are commonly bought together, often referred to as "item sets" or "association rules." This information holds significance in decision-making processes, such as determining product placement and devising strategies for sales promotions. It serves as a critical tool for businesses aiming to improve customer satisfaction and increase their sales revenue. It offers a comprehensive understanding of the relationships between products, enabling businesses to make choices about inventory, pricing, and marketing strategies. With the expansion of real-world datasets, the importance of swift and scalable approaches for mining frequent itemsets has never been more pertinent. [3]

It aims to compute and analyze which products are commonly purchased together by customers, as if they are tossing all their selections into a shopping cart during grocery shopping. Understanding these group purchases can greatly benefit retailers and other companies.

Today's retail industry is highly customer-focused, with retailers actively seeking new ways to understand their customers better. Market basket analysis has become an attractive tool for many retail companies. It involves identifying customer behavior, purchase patterns, and the relationship between products and the content provided by retailers, either in physical stores or online. With the assistance of this tool, retailers can not only pinpoint their target markets but also enhance their sales by offering a superior customer experience. It allows retailers to quickly analyze the size, quantity, and value of customers' shopping baskets to discern purchasing patterns.

Such an analysis holds significance for several reasons:

a. Customer insights: It helps to understand the customer purchasing behavior, preferences, and inclinations, which can guide business decisions and drive sales. [4]
b. Increased sales: By identifying frequently co-purchased products, companies can design targeted promotions and packages, resulting in heightened sales and customer satisfaction.
c. Improved inventory management: By recognizing popular combinations of items, businesses can optimize their inventory management and prevent overstocking or understocking.
d. Enhanced product placement: Market basket analysis can help businesses determine the ideal placement for products in stores or online, leading to improved sales and customer experience.
e. Cost savings: Market basket analysis assists businesses in reducing waste by avoiding overstocking low-demand products and focusing on stocking items popular with customers, also taking into account green consumerism.[5]

II. LITERATURE SURVEY

There is a wide array of data mining techniques and algorithms designed for the discovery of meaningful patterns and rules. Various methods include:

a. Classification: In this process, the characteristics of a newly introduced item are examined, and is allocated to a predefined category. For instance, categorizing credit applicants into high, medium and low risk.
b. Association: The primary objective here aims to establish relationships between items existing within the market. Tools like the Apriori algorithm and the Weka toolkit are commonly used for this purpose.
c. Prediction: This functionality involves predicting unidentified or absent attribute values based on available information. For instance, forecasting the sales value for the next seven days using existing data.
d. Clustering: Data Mining, in this case, organizes data into meaningful subcategories or clusters. These clusters ensure that elements within the group share similarities while being as distinct as possible from points in other groups. This method operates as an unsupervised classification.
e. Analysis: In this, Data Mining is conducted to recognize and elucidate anomalies. As an illustration, an outlier could refer to a business-transaction that occurs exceptionally. [6]

It involves examining a set of items to discover associations that can be leveraged in various ways. Some applications include:

- Product placement. Recognizing products frequently bought together and strategically positioning these items, whether in a catalog or on a website, to prompt customers to purchase them as a bundle. [7]
- Physical shelf arrangement. Another purpose for physically placing products in a store is to segregate items that are typically bought together, enticing customers to explore the store in search of what they need and possibly increasing the likelihood of extra purchases
- Customer retention. Maintaining customer loyalty involves employing strategies when customers reach out to terminate their relationship with a business. In such instances, a company representative may utilize these methods to identify suitable incentives aimed at retaining the customer's business.

III. METHODOLOGY

A. Hardware and Software Requirements

1. Google Colab – It serves an open-source, Jupyter based

environment. It involves running code in “Python” and “Python-based third-party tools” and machine learning frameworks [8].

2. GPU- Graphics Processing Units (GPUs) are highly parallel devices suitable for executing parallel tasks efficiently. These accelerators are widespread, readily available, and offer a high GFlops per Dollar ratio. [9] Google provides the use of free GPU for the notebooks. Choose a GPU, and your notebook will utilize the available cloud-based GPU for processing without any cost.

B. Description of Libraries used

1. Numpy
NumPy is a library used for working with arrays. It also includes functions for performing operations in the fields of linear algebra, Fourier transforms, and matrix manipulations. [10]

2. Pandas
Pandas is a widely utilized open-source Python package primarily employed for activities associated with data science, analyzing data, and machine learning. [11]

3. Itertools
Itertools is a Python module designed for iterating over data structures that can be traversed using a for-loop.

4. Mlxtend
Mlxtend, short for "machine learning extensions," is a Python library that offers a set of valuable tools for various data science tasks in your daily work. [12]

5. Apriori
Apriori is a technique that involves identifying frequently individual items happening in a database and then expanding them to form bigger sets of items, continuing this process provided that those sets of items occur with sufficient frequency in the database. [13]

6. Matplotlib
Matplotlib is a flexible library employed for creating static, animated, and interactive visualizations within the Python programming language.

7. Seaborn
It is a Python library primarily employed for crafting statistics. It is built on the foundation of Matplotlib and tightly integrated with pandas data structures in the Python programming language. [14]

8. Plotly
It is an open-source tool which offers support for more than 40 distinct chart types, encompassing a broad spectrum of applications including statistics, finance, geography, science, and three-dimensional visualizations

C. Relevant Algorithms

1. Apriori:
The Apriori algorithm serves as the foundational algorithm in basket analysis. It entails a series of steps for identifying the most frequently occurring itemsets within a given database. The Apriori algorithm follows a simple principle: if the support value of an item set surpasses a specific threshold, it qualifies as a frequent item set. The algorithm utilizes a "bottom-up" approach, systematically expanding common subsets by adding one item at a time. These sets of potential items are subsequently evaluated against the dataset. The algorithm concludes when there are no more successful expansions can be identified. A limitation of the Apriori algorithm is that, with larger datasets, it demands increased memory resources, resulting in a performance burden and prolonged execution times, thereby diminishing the algorithm's efficiency. [15]

2. ECLAT:
It is structured differently from the Apriori algorithm. ECLAT employs a Depth-First Search (DFS) approach for discovering frequent item sets, while Apriori uses a Breadth-First Search (BFS) approach. ECLAT's vertical pattern-oriented design makes it a faster algorithm compared to Apriori.

In ECLAT, Transaction ID sets, also known as sets, are used to calculate the “support” of a dataset. To circumvent the creation of subsets when the minimum support is reached, the support value for an item is computed using this method. Each item-transaction set is compared to every other pair in the function to generate new candidate item sets. If a candidate item set is found to be common, the list of shared partners is expanded. Importantly, if a frequency set includes a pair of items, the frequent item set also encompasses its subsets.

In the ECLAT algorithm, the process involves obtaining the tidlist for each database object, leading to an exhaustive search of the database. The tidlist represents the list of transactions containing a particular item, denoted as "a." Moreover, a new transaction list is created by intersecting the transaction lists of both item "a" and item "b" to identify transactions in which both items are involved. This approach effectively aids in discovering frequent item sets in a dataset. ECLAT is employed within the MapReduce framework and is implemented using the Java programming language. [16]

3. FPGrowth:
FP-growth stands as an enhanced iteration of the Apriori Algorithm, a popular tool for the extraction of recurring patterns, also known as “Association Rule Mining”. It is employed as a process of analysis for discovering frequent patterns or associations within datasets. This algorithm initiates by condensing the

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

---

input database to construct an instance of a frequent pattern tree. The condensed database is subsequently partitioned into distinct conditional databases, with each database representing a unique frequent pattern. [17]

FP-growth accomplishes this by representing frequent items through a structure known as a "frequent pattern tree" or FP-tree. This tree-like pattern tree dispenses with the need for candidate generation when identifying frequent patterns. The FP-growth algorithm. using the concept of *"recursive elimination scheme"*, encodes the database in this tree pattern, which is referred to as the FP-tree.[18]

It is constructed using the initial item sets from the database, forming a structured representation of the dataset. In the FP-tree, each node signifies an item within the item set, thereby preserving the associations or relationships between these items. The database is segregated using one frequent item, leading to the creation of pattern fragments. The item sets within these fragmented patterns are then examined, effectively reducing the search for frequent item sets. This approach streamlines the process of identifying frequent patterns in the dataset.

D. Implementation

Several models were trained to evaluate their performance, and a comparison was conducted among all of them.

1. Data Description

   To begin, various libraries were loaded, and the data was imported. The read.transactions() function from the arules package was utilized to create a transactions object.

The dataset comprises 15,010 observations and includes:

a) Date and time which are both categorical variables,
b) Transaction: A "quantitative variable" that enables the distinction of transactions. Rows sharing the same value in this field are part of the same transaction, which is why there are fewer transactions than observations.
c) Item: A categorical variable that stores information about the products.[19]

In the first 5 rows of the dataset, there are three distinct columns:

i. Member_number: A unique numerical identifier for each customer.
ii. Date: Represents the transaction date.
iii. ItemDescription: Indicates the specific product purchased on that date.

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 38765 entries, 0 to 38764
Data columns (total 3 columns):
 #   Column           Non-Null Count  Dtype
---  ------           --------------  -----
 0   Member_number    38765 non-null  int64
 1   Date             38765 non-null  object
 2   itemDescription  38765 non-null  object
dtypes: int64(1), object(2)
memory usage: 908.7+ KB
```

Figure 1: Data Information

2. Data Preparation

The verification of null values is a crucial step in data preprocessing to ensure that the dataset is prepared properly for training machine learning models.

```
# Checking for the missing values
nan_values = data.isna().sum()
nan_values
```

```
memberID    0
Date        0
itemName    0
dtype: int64
```

Figure 2: Computation of Missing Data

The analysis plan is designed to address the following questions:

A. What are the items that are sold most frequently?
B. What are the consequent items associated with the chosen items?
C. How confident are the consequent items that follow the chosen items?
D. Which items are the most essential and should always be stocked in the store?
E. What does the network of item relationships look like?

According to the presented results, it seems that there are no absent values in any of the columns within the dataset.

```
# Converting Date column into correct datatype which is datetim
e
data.Date = pd.to_datetime(data.Date)
data.memberID = data['memberID'].astype('str')
data.info()  # They are in correct datatype now
```

```
<class 'pandas.core.frame.DataFrame'>
RangeIndex: 38765 entries, 0 to 38764
Data columns (total 3 columns):
 #   Column    Non-Null Count  Dtype
---  ------    --------------  -----
 0   memberID  38765 non-null  object
 1   Date      38765 non-null  datetime64[ns]
 2   itemName  38765 non-null  object
dtypes: datetime64[ns](1), object(2)
memory usage: 908.7+ KB
```

Figure 3: Conversion into datetime

International Journal on Recent and Innovation Trends in Computing and Communication
ISSN: 2321-8169 Volume: 11 Issue: 9
***Article Received:** 25 July 2023 **Revised:** 12 September 2023 **Accepted:** 30 September 2023*

3. *Exploratory Data Analysis*

To begin, various libraries were loaded, and the data was imported. The read.transactions() function from the arules package was utilized to create a transactions object.

Figure 4: Total Number of Items Sold by Date

As depicted in the figure, the highest sales occurred on the 21st of November, 2014, and just before the 9th of May, 2014. To determine the count of distinct items with the average number of items sold, the subsequent code was executed, and the output was documented.

```
total_items = len(d)
total_days = len(np.unique(d.index.date))
total_months = len(np.unique(d.index.month))
average_items = total_items / total_days
unique_items = d.itemDescription.unique().size

print("There are {} ye unique items sold ".format(unique_items))
print("Total {} items sold in {} days throughout {} months".format(total_items, total_days, total_months))
print("With an average of {} items sold daily".format(average_items))

There are 167 ye unique items sold
Total 38765 items sold in 728 days throughout 12 months
With an average of 53.24862637362637 items sold daily
```

Figure 5: Unique, Average and Total Items sold

We also found out the overall number of items sold each month in that year:

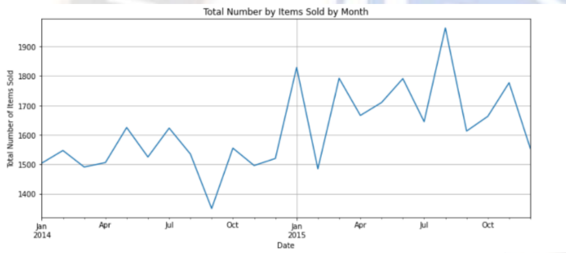


Figure 6: Overall Number of Items sold by Date

It is noticed that most number of sales were made during the month of August followed by Jan 2015.

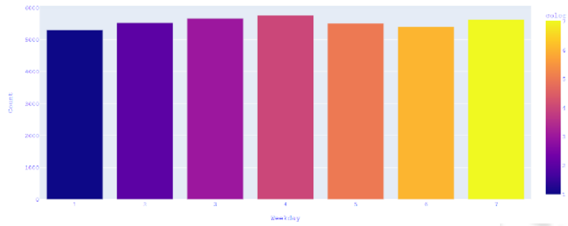


Figure 7: Count of Items sold by weekday

Also, count of weekly sales is made by calculating the number of sales on every day.

Also, for curiosity about the item frequencies, we got the following output:

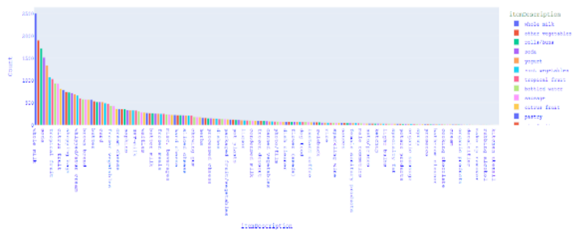
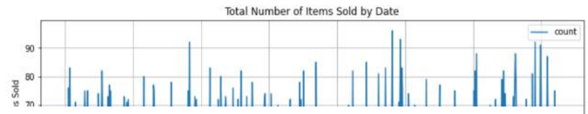


Figure 8: Count of Items

4. *Choice* of *Model and Model Training*

i. Apriori

It is an algorithm utilized for mining sets of frequent items and discovering association rules within relational databases. Its



	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
668	(sausage, yogurt)	(whole milk)	0.005748	0.157923	0.001470	0.255814	1.619866	0.000563	1.131541
978	(sausage, rolls/buns)	(whole milk)	0.005347	0.157923	0.001136	0.212500	1.345594	0.000292	1.069304
1079	(soda, sausage)	(whole milk)	0.005948	0.157923	0.001069	0.179775	1.138374	0.000130	1.026642
526	(semi-finished bread)	(whole milk)	0.009490	0.157923	0.001671	0.176056	1.114825	0.000172	1.022008
828	(rolls/buns, yogurt)	(whole milk)	0.007819	0.157923	0.001337	0.170940	1.082428	0.000102	1.015701

operation involves identifying distinct items in the database that occur frequently and then progressively extending these items into larger item sets, provided these larger sets also appear frequently in the database.

Figure 9: Association Table

The Apriori algorithm takes into account three crucial factors when establishing association rules among elements or items:

Support: Denoted as I, it is calculated as the number of transactions with that item divided by the overall number of transactions.

Confidence: It gauges the ratio of transactions that involve item I1 and also include item I2. More precisely, the confidence between two items, I1 and I2, within a transaction is determined by the total number of transactions containing both I1 and I2 divided by the total number of transactions containing I1. [20]

Lift: It is defined as the ratio between "confidence" and "support".

These factors are essential for evaluating and generating association rules that reveal relationships and trends among different items in a dataset.

ii. FP Growth

It offers an effective method for discovering frequent item sets without the need for candidate generation, which enhances performance. It employs a divide-and-conquer strategy, and its core relies on a specialized data structure called the frequent-pattern tree (FP-tree) to capture item set associations.

The algorithm operates as outlined:

Initially, it condenses the input database to construct an "FP-tree", which represents frequent items.

After this compression process, the condensed database is split into a series of conditional databases, each linked to a particular frequent pattern.

Subsequently, each of these conditional databases is mined separately. Each transaction is processed, and it is mapped onto a path within the FP-tree. This process continues until all transactions have been processed. The compactness of the tree is maintained through overlapping paths for different transactions with common subsets.

The FP-tree is constructed using the initial item sets from the database, and its primary purpose is to mine the most frequent patterns. In this tree structure, each node represents an item

ISSN: 2321-8169 Volume: 11 Issue: 9
Article Received: 25 July 2023 Revised: 12 September 2023 Accepted: 30 September 2023

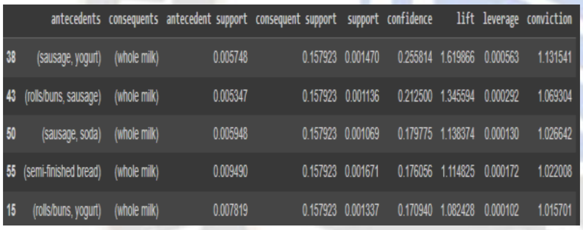
within the item set. This efficient approach contributes to improved performance in finding frequent item sets.

iii. ECLAT Algorithm

It is a prominent technique for Association Rule mining. It represents a more effective and scalable substitute to the Apriori algorithm. As the execution of the Apriori algorithm operates in a horizontal manner, resembling the Breadth-First Search of a graph, the ECLAT algorithm follows an upright approach, similar to the graph's Depth-First Search. This vertical nature of the ECLAT algorithm expedites the process and more efficient compared to the Apriori algorithm. Here's how the ECLAT algorithm works:-

The fundamental concept is to utilize intersections of sets of transaction IDs (tidsets) to calculate the support value of a candidate item set while preventing the creation of subgroups that are not present within the prefix tree. In the initial function call, all individual items are utilized alongside their corresponding tidsets. Subsequently, the function is invoked recursively, and with each recursive call, individual item-tidset pairs are examined and amalgamated with additional item-tidset pairs. This process continues until no further candidate item-tidset pairs can be merged. This efficient approach to item set discovery contributes to the speed and effectiveness of the ECLAT algorithm.

IV. Result and Analysis



	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
38	(sausage, yogurt)	(whole milk)	0.005748	0.157923	0.001470	0.255814	1.619866	0.000563	1.131541
43	(rolls/buns, sausage)	(whole milk)	0.005347	0.157923	0.001136	0.212500	1.345594	0.000292	1.069304
50	(sausage, soda)	(whole milk)	0.005948	0.157923	0.001069	0.179775	1.138374	0.000130	1.026642
55	(semi-finished bread)	(whole milk)	0.009490	0.157923	0.001671	0.176056	1.114825	0.000172	1.022008
15	(rolls/buns, yogurt)	(whole milk)	0.007819	0.157923	0.001337	0.170940	1.082428	0.000102	1.015701

Figure 10: Association Rules Analysis

Antecedent and Consequent: In association rule terminology, the "IF" component is referred to as the antecedent, and the "THEN" component is known as the consequent. These two components, the antecedent and the consequent, are disjoint, meaning they have no items in common. As you can see in the figure 10, different combinations can be witnessed in both the columns.

Confidence: It is the conditional probability of the consequent occurring given the occurrence of the antecedent. Users have the flexibility to set a minimum confidence threshold for rules.

Support: It indicates the popularity of an item across all transactions. The support of an item is computed by dividing the number of transactions involving that item by the total number of transactions.

Lift: It serves as a metric to assess the performance of a targeting model, specifically an association rule, in predicting a particular outcome in comparison to a random choice.

Leverage: Retailers leverage this method to identify combinations of goods or menu items that frequently appear together in transactions. This enables them to uncover associations in customer purchasing patterns, facilitating the creation of new products and pricing models to generate additional revenue streams.

Conviction: It can be understood as the ratio of the anticipated frequency of X occurring without Y to the observed frequency of such incorrect predictions.

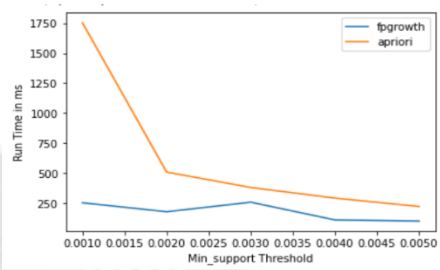


Figure 11: Minimum threshold support of FPgrowth and Apriori

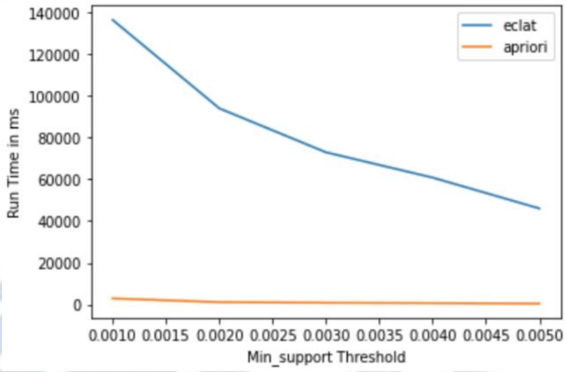


Figure 12: Minimum threshold support of ECLAT and Apriori

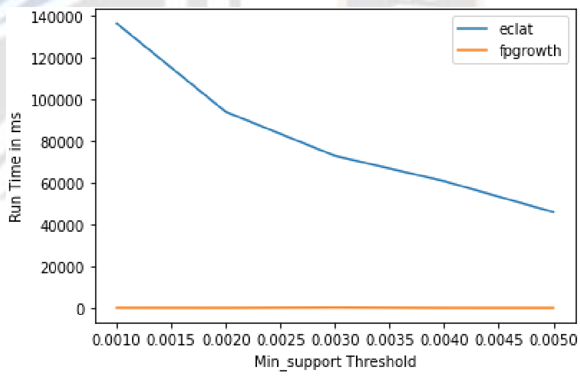


Figure 13: Minimum threshold support of FPgrowth and ECLAT

According to the above charts, FPgrowth is the fastest for huge datasets, while ECLAT is incredibly slow.

Table 1: Comparison between Apriori, ECLAT and FPGrowth

Aspect	**Apriori**	**ECLAT**	**FPGrowth**
Algorithm type	Array-based	Tree-based	Tree-based
Techniques	Join and Prune	Tidset Intersection	Pattern Growth

Search Approach	Breadth-First Search	Depth-First Search	Depth-First Search
Pattern Generation Method	Level-wise (1 item, 2 items, 3 items, etc.)	Vertical (Based on Tidset Intersection)	Existing Patterns in the Database
Runtime Characteristics	Exponential Increase	Linear Increase (Depending on Transactions)	Linear Increase (Depending on Transactions)
Efficiency	Candidate generation is slow	Efficient for large datasets	Efficient for large datasets

V. CONCLUSION

In our exploration of association rules and interestingness measures for frequent item patterns, we gained an overview of the ECLAT algorithm, as well as the implementation of the Apriori and FP-Growth algorithms using Python, which were presented in a straightforward manner.

Considering the variety of algorithms available for performing ECLAT, it's worth delving into these alternatives, as they can prove useful in cases where Apriori or FP-Growth algorithms are not meeting performance requirements or delivering the desired results.

The Apriori algorithm, with its ability to scan the database only once and significantly reduce the size of item sets, offers efficient performance in data mining. Consequently, data mining aids both consumers and industries during the decision-making process.

For those interested in Python implementations, the Mlxtend library is a valuable resource as it provides implementations of Apriori, FP-Growth, and ECLAT algorithms for such applications. [21]

In future research, it would be fascinating to gain a deeper enhancing comprehension of association rules by assessing alterations in lift and confidence values, possibly through the calculation of standard deviation. This approach could enable us to witness the evolution of association rules over time.

Moreover, future work could explore the discovery of association rules using time series clustering methods, offering new dimensions to the analysis.

Overall, the comparison between Apriori and FP-Growth algorithms, as well as the potential for further research in this domain, opens up exciting opportunities for enhancing decision-making processes and recommendation systems.

REFERENCES

[1] R. Sumithra and S. Paul, "Using distributed apriori association rule and classical apriori mining algorithms for grid based knowledge discovery", 2010 Second International conference on Computing, Communication and Networking Technologies, Karur, India, 2010, pp. 1-5, doi: 10.1109/ICCCNT.2010.5591577.

[2] A. Aggarwal and D. Toshniwal, "Frequent pattern mining on time and location aware air quality data", in IEEE Access, vol. 7, pp. 98921-98933, 2019, doi: 10.1109/ACCESS.2019.2930004.

[3] K. -W. Chon, E. Yi and M. -S. Kim, SGMiner, "A fast and scalable GPU-based frequent pattern miner on SSDs," in IEEE Access, vol. 10, pp. 62502-62519, 2022, doi: 10.1109/ACCESS.2022.3179592.

[4] Pillai, Anup & Jolhe, Dhananjay. (2021). "Market basket analysis: case study of a supermarket", 10.1007/978-981-15-3639-7_87.

[5] Prakash Rao, Sreejith Balasubramanian, Nitin Vihari, Shazi Jabeen, Vinaya Shukla & Janya Chanchaichujit | Albert W. K. Tan (Reviewing editor) (2021) "The e-commerce supply chain and environmental sustainability: An empirical investigation on the online retail sector", Cogent Business & Management, 8:1, DOI: 10.1080 /23311975 .2021.1938377

[6] Manpreet Kaur, Shivani Kang, Market Basket Analysis: "Identify the changing trends of market data using association rule mining", Procedia Computer Science, Volume 85, 2016, Pages 78-85, ISSN 1877-0509, https://doi.org/10.1016/j.procs.2016.05.180.

[7] Kouki, Pigi & Fountalis, Ilias & Vasiloglou, Nikolaos & Yan, Nian & Ahsan, Unaiza & Jadda, Khalifeh & Qu, Huiming. (2019). "Product collection recommendation in online retail", 486-490. 10.1145/3298689.3347003.

[8] A. Das, M. Wasif Ansari and R. Basak, "Covid-19 face mask detection using tensorflow, keras and openCV", 2020 IEEE 17th India Council International Conference (INDICON), New Delhi, India, 2020, pp. 1-5, doi: 10.1109/INDICON49873.2020.9342585.

[9] T. Carneiro, R. V. Medeiros Da NóBrega, T. Nepomuceno, G. -B. Bian, V. H. C. De Albuquerque and P. P. R. Filho, "Performance analysis of Google colaboratory as a tool for accelerating deep learning applications," in IEEE Access, vol. 6, pp. 61677-61685, 2018, doi: 10.1109/ACCESS.2018.2874767.

[10] Harris, C.R., Millman, K.J., van der Walt, S.J. et al. "Array programming with NumPy". Nature 585, 357–362 (2020). https://doi.org/10.1038/s41586-020-2649-2

[11] Mckinney, Wes. (2011). "Pandas: a foundational python library for data analysis and statistics". Python High Performance Science Computer.

[12] Raschka, Sebastian. (2018). MLxtend: "Providing machine learning and data science utilities and extensions to Python's scientific computing stack", Journal of Open Source Software. 3. 638. 10.21105/joss.00638.

[13] Xie, Haoyu. (2021) "Research and case analysis of Apriori algorithm based on mining request item-sets". Open Journal of Social Sciences. 09. 458-468. 10.4236/jss.2021.94034.

[14] Waskom, Michael. (2021). seaborn: "Statistical data visualization. journal of open source 2`Software". 6. 3021. 10.21105/joss.03021.

[15] Gulzar K, Ayoob Memon M, Mohsin SM, Aslam S, Akber SMA, Nadeem MA. "An efficient healthcare data mining approach using Apriori algorithm: A case study of eye disorders in young adults. Information" 2023; 14(4):203. https://doi.org/10.3390/info14040203.

[16] Bansal, Urvashi. (2014). "ECLAT Algorithm for Frequent Item sets Generation", International Journal of Computer System.

[17] Sidhu, Shivam & Meena, Upendra & Nawani, Aditya & Gupta, Himanshu & Thakur, Dr. Narina. (2014). "FP growth algorithm implementation", International Journal of Computer Applications. 93. 6-10. 10.5120/16233-5613.

[18] Borgelt, Christian. (2010). "An implementation of the FP-growth algorithm" Proceedings of the 1st International Workshop on

Open Source Data Mining: Frequent Pattern Mining Implementations. 10.1145/1133905.1133907.

[19] Panjaitan, Suprianto & Sulindawaty, & Amin, Muhammad & Lindawati, Sri & Watrianthos, Ronal & Sihotang, Hengki & Sinaga, Bosker. (2019). "Implementation of Apriori algorithm for analysis of consumer purchase patterns". Journal of Physics: Conference Series. 1255. 012057. 10.1088/1742-6596/1255/1/012057.

[20] Bagui, Sikha & Just, Jiri & Bagui, Subhash. (2009). "Deriving strong association mining rules using a dependency criterion, the lift measure". International Journal of Data Analysis Techniques and Strategies. 1. 297-312. 10.1504/IJDATS.2009.024297.

[21] Raschka, Sebastian. (2018). MLxtend: "Providing machine learning and data science utilities and extensions to Python's scientific computing stack". Journal of Open-Source Software. 3. 638. 10.21105/joss.00638.