CONTENT

1.0 Problem Background

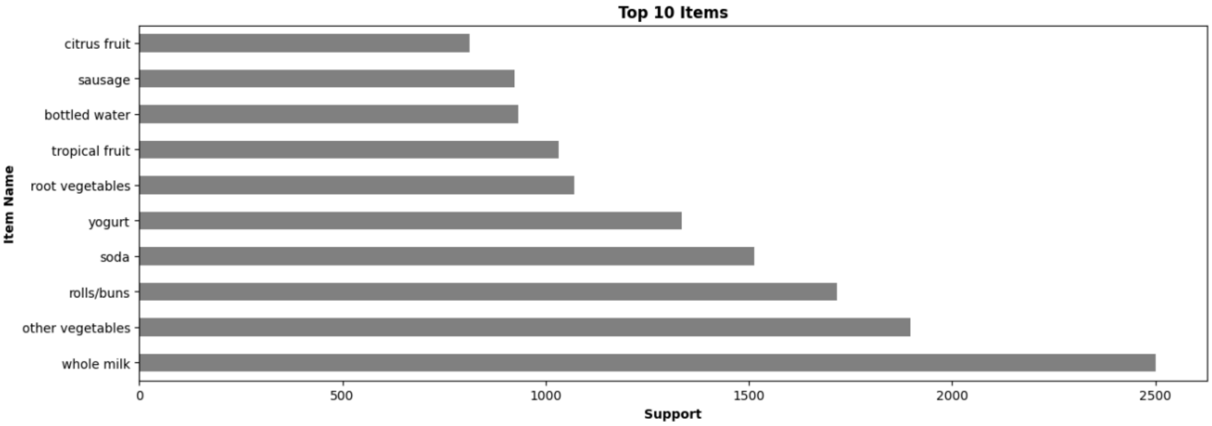
Market Basket Analysis is a robust technique employed by retailers to delve into customer purchasing patterns and unravel associations among frequently co-purchased items in a retail environment. Utilizing widely adopted algorithms such as Apriori and FP-Growth, this analysis seeks to discover recurring sets of items and construct association rules based on transactional data. In this context, the Apriori algorithm employs a level-wise approach, generating candidate item sets and trimming non-frequent item sets based on a minimum support threshold[1]. On the other hand, the FP-Growth algorithm constructs a frequent pattern tree (FP-tree) from transaction data, iteratively mining the tree to unveil frequent item sets efficiently. The dataset under consideration comprises 38,765 rows of purchase orders from grocery stores. This dataset presents an opportunity for thorough analysis and the generation of association rules using Market Basket Analysis, particularly through algorithms like the Apriori Algorithm[7]. Market Basket Analysis, facilitated by the Apriori and FP-Growth algorithms, empowers retailers to gain insights into customer behavior. This includes identifying items frequently purchased together, recommending related products, optimizing product arrangement, and refining marketing and sales strategies. Apriori and FP-Growth algorithms are not merely data analysis tools; they are gateways to strategic insights[8].

By applying these algorithms, retailers can make informed decisions, enhance the shopping experience, and achieve sustainable success in the dynamic and competitive retail landscape. The significance extends beyond transactional understanding, enabling strategic utilization of information to improve customer satisfaction and drive business growth.

2.0 Data Understanding & Integration

The dataset, comprising 38,765 rows of purchase orders collected from Kaggle, pertains to individual shopping transactions at grocery stores. The substantial number of rows provides an opportunity for in-depth analysis, enabling a comprehensive understanding of customer behavior patterns[3]. The order details can be analyzed using association rules generated by Market Basket Analysis algorithms such as Apriori.

The large number of rows opens avenues for deep analysis, allowing for a detailed understanding of customer behaviors, including preferences and quantities purchased. These data can be systematically presented in a table containing diverse variables such as purchased items, member numbers, and purchase times. This broad diversity facilitates a comprehensive analysis of order information, unlocking valuable insights. Leveraging this extensive dataset enhances our comprehension of purchasing habits and details, contributing to the improvement of marketing strategies and the overall shopping experience for customers.

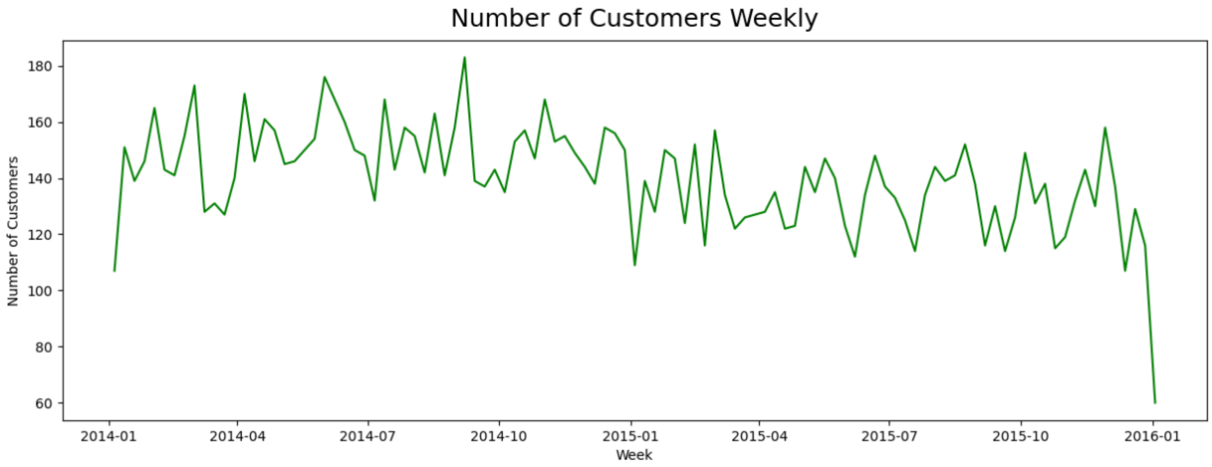


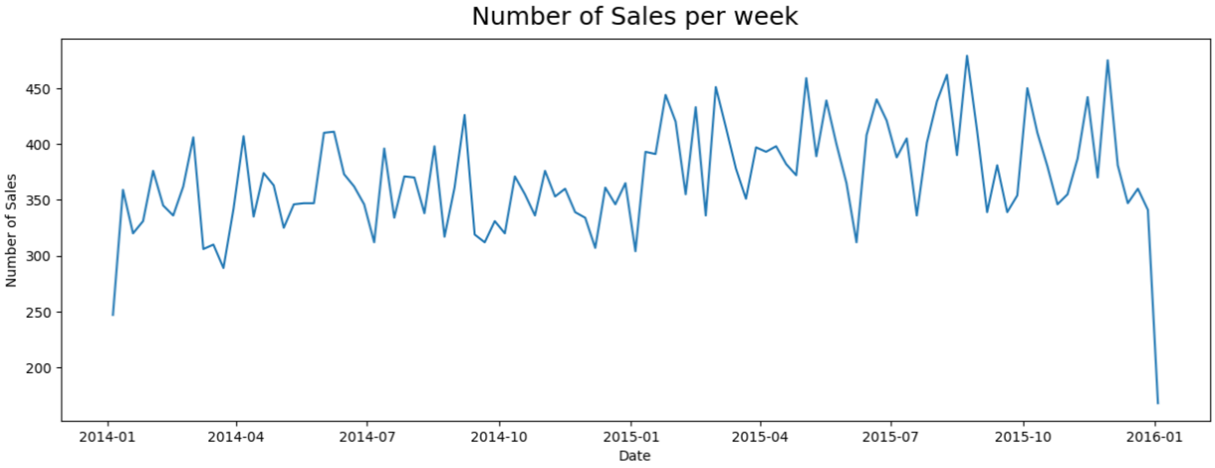
The Fig. shows the top 10 items purchased by customers at a grocery store. The top item is whole milk, followed by wheat milk, yogurt, other vegetables, tomatoes, carrots, onions, citrus fruit, sausage, and bottled water.

Here are some of the insights that can be gleaned from this data:

Milk is the most popular item, with both whole milk and wheat milk appearing in the top 10 items. This suggests that there is a demand for both dairy and non-dairy milk options.

Yogurt is also a popular item, which could be due to several factors, such as its health benefits, versatility, or affordability. Vegetables are well-represented on the list, with other vegetables, tomatoes, carrots, and onions all making the top 10. This suggests that customers are making an effort to eat healthily. Fruit is also present on the list, with citrus fruit and tropical fruit making the top 10. This suggests that customers are looking for a variety of fruits to eat. Sausage is the only meat item on the list, which is somewhat surprising. This could be because the data is from a grocery store, and not a butcher shop. This could be because the data is from a grocery store and not a convenience store. Overall, the data suggests that customers at this grocery store are looking for healthy, affordable, and convenient items. It is important to note that this is just a small sample of data, and it may not be representative of all grocery stores or all customers. However, it does provide some interesting insights into the shopping habits of grocery store customers.





The graphs show the number of customers and sales per week at a store between January 2014 and the end of 2015. Here are some of the key trends I observed:

Seasonality: There is a clear seasonal pattern to the data, with the number of customers peaking in the summer months (June-August) and dipping in the winter months (December-February). This could be due to several factors, such as the weather, holidays, and back-to-school season.

Growth: The overall trend is one of growth, with the number of customers increasing steadily over the two years. This could be due to several factors, such as population growth, increased marketing efforts, or an improving economy.

Fluctuations: There are some short-term fluctuations in the data, such as the dip in customers in July 2014 and the spike in customers in December 2014. These fluctuations could be due to several factors, such as promotions, special events, or changes in the weather.

Overall, the graph suggests that the store is doing a good job of attracting customers and that the number of customers is growing over time. However, there is also some seasonality to the data, so it is important to keep this in mind when planning marketing and staffing levels.

Here are some additional questions that you could ask about the data:

What is the average number of customers per week?

What is the busiest day of the week?

What is the least busy day of the week?

How has the number of customers changed over time?

What are the busiest and least busy times of the year?

By answering these questions, we can gain a better understanding of our customer base and make informed decisions about how to market our store and staff our business.

3.0 Pre-processing options

Grocery basket analysis is a process that analyzes the purchasing habits of customers by finding associations between different items in the customer's shopping cart. This association is needed to find out what items the customer may buy at the same time. This analysis is very helpful for business owners in improving their marketing strategy. Market basket analysis can be analyzed using the association rule. The purpose of market basket analysis is to find out which products may be purchased simultaneously.

3.1 Association Rules

Association rule mining is a data mining technique that involves the discovery of frequent patterns known as associations among sets of items or objects in transaction databases, relational databases, and other information repositories. The primary goal is to identify relationships or associations between different items based on their co-occurrence in the data[3]. This technique is commonly used in market basket analysis, where the focus is on finding connections between products that are frequently purchased together. Association rule mining employs measures such as support, confidence, and lift to quantify and evaluate the strength and significance of these discovered associations[6].

- **Apriori Algorithm**

The data mining types of association rules are included in the apriori algorithm. Frequent pattern mining is one of the stages of association analysis that has drawn the interest of numerous academics to create effective algorithms. It is crucial to determine whether an association can be determined by two benchmarks, namely: support and confidence[1]. The strength of the relationship between the items in the association rule is known as confidence (value of certainty), whereas support (value of support) is the percentage of this combination of items in the database[7]. The first step of the a priori method is the analysis of high-frequency patterns, which is essentially locating item combinations in the database that satisfy the minimal requirements of the support value. An item's support value is determined by the following formula:

$$Support = \frac{number\ of\ transactions}{\text{total transactions}}$$

The item set frequency shows the item set that has an appearance frequency of more than the specified minimum value. The next step is the formation of association rules, that is, after all high-frequency patterns are found, then you can search for association rules that meet the minimum confidence requirements, by calculating the value A => B. The formula is as follows:

$$\text{Confidence}(\text{A} => \text{B}) = \frac{Support(A \cup B)}{\text{Support}(A)}$$

- **FP-Growth**

FP-Growth (Frequent Pattern Growth) algorithm is a data mining algorithm used for discovering frequent item sets and generating association rules from transactional databases. It constructs an FP-tree (Frequent Pattern tree) to efficiently represent and analyze patterns in the data, offering advantages in terms of computational efficiency, especially for large datasets. The algorithm employs a divide-and-conquer strategy, recursively building conditional FP-trees and mining frequent item sets directly from the FP-tree structure, eliminating the need for multiple database scans. FP-Growth is known for its effectiveness in scenarios with substantial amounts of transactional data.

3.2 Analysis step

The steps in the analysis are as follows:

1. Recap sales data.
2. Descriptive analysis.
3. Determine the value of minimum support and confidence.
4. The formation of associative rules with the Apriori algorithm and FP-Growth is as

Follows:

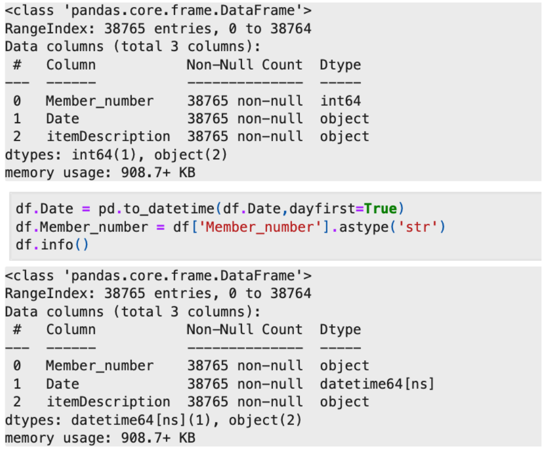
a) Select a frequency greater than or equal to the minimum limit that has been determined.
b) Make combinations of 2 dataset items.
c) Make combinations of 3 dataset items.
d) Apply association rules.
e) Analyze and interpret the meaningful rules generated.

4.0 Experiment and Analysis Using Apriori Algorithm and FP-Growth Algorithms

Data processing

Before starting this experiment, data processing is required first. The dataset has already been introduced in the previous paragraph, so we won't repeat it here. Here, we convert `Data` into a datetime type for better expression. It is shown in Fig.1. Data Transformation

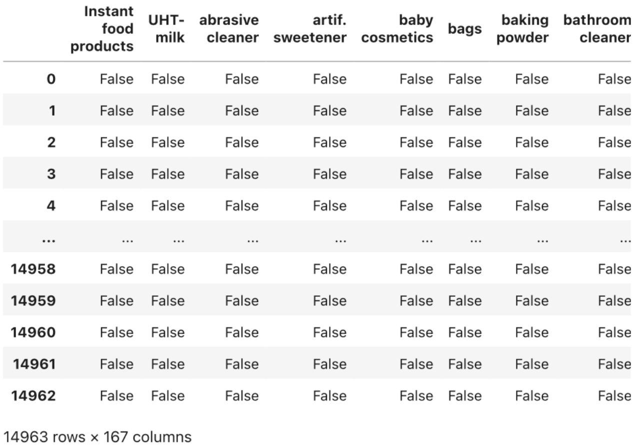
The process `itemDescription`, turns each subitem into an element in the list, in this format can be well applied to this one hot-like function, there we use `TransactionEncoder () `. After that, we get like only have False and True matrix (Fig.2.Converted data).



```
<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
```

```
df.Date = pd.to_datetime(df.Date,dayfirst=True)
df.Member_number = df['Member_number'].astype('str')
df.info()
```

```
<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  object        
 1   Date             38765 non-null  datetime64[ns]
 2   itemDescription  38765 non-null  object        
dtypes: datetime64[ns](1), object(2)
memory usage: 908.7+ KB
```



	Instant food products	UHT-milk	abrasive cleaner	artif. sweetener	baby cosmetics	bags	baking powder	bathroom cleaner
0	False	False	False	False	False	False	False	False
1	False	False	False	False	False	False	False	False
2	False	False	False	False	False	False	False	False
3	False	False	False	False	False	False	False	False
4	False	False	False	False	False	False	False	False
...	...	...	...	...	...	...	...	...
14958	False	False	False	False	False	False	False	False
14959	False	False	False	False	False	False	False	False
14960	False	False	False	False	False	False	False	False
14961	False	False	False	False	False	False	False	False
14962	False	False	False	False	False	False	False	False

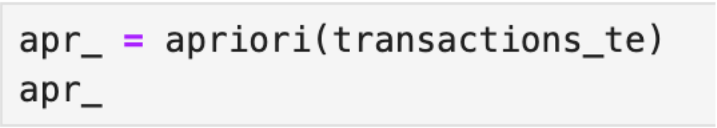
14963 rows × 167 columns

Fig.1. Data Transformation

Fig.2. Converted data

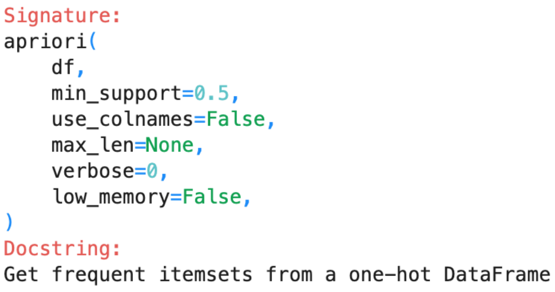
Apriori Algorithm

The field processed above is called `transactions_te`, put this into `apriori () ` function without setting the parameters, all default parameters.



```
apr_ = apriori(transactions_te)
apr_
```

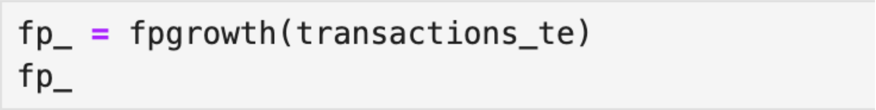
With this code, no results are displayed, cause the default parameter min_support in this apriori function is 0.5, and in the `apr_` has no support value more than 0.5.



```
Signature:
apriori(
    df,
    min_support=0.5,
    use_colnames=False,
    max_len=None,
    verbose=0,
    low_memory=False,
)
Docstring:
Get frequent itemsets from a one-hot DataFrame
```

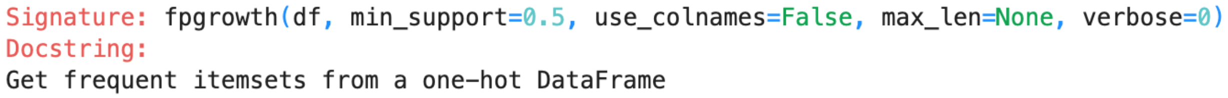
FP-Growth Algorithm

We put the `transactions_te` into `fpgrowth () ` the same as using apriori algorithm, without any parameters.



```
fp_ = fpgrowth(transactions_te)
fp_
```

Running this code still has no result output, cause the min_support is 0.5, in the `fp_` has no support value more than 0.5.



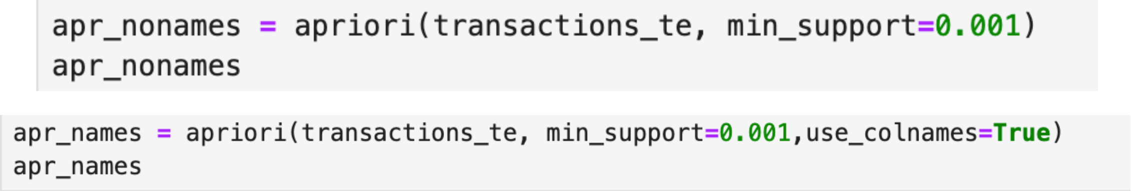
```
Signature: fpgrowth(df, min_support=0.5, use_colnames=False, max_len=None, verbose=0)
Docstring:
Get frequent itemsets from a one-hot DataFrame
```

4.1 Setting the parameters for the algorithm and running the algorithm.

In this section, the experiment involves using the Apriori Algorithm and FP-Growth algorithm to perform frequent pattern mining on a dataset. The algorithm is applied with different parameters, such as minimum support values, and the results are presented in the form of itemsets and their corresponding support values.

Apriori Algorithm

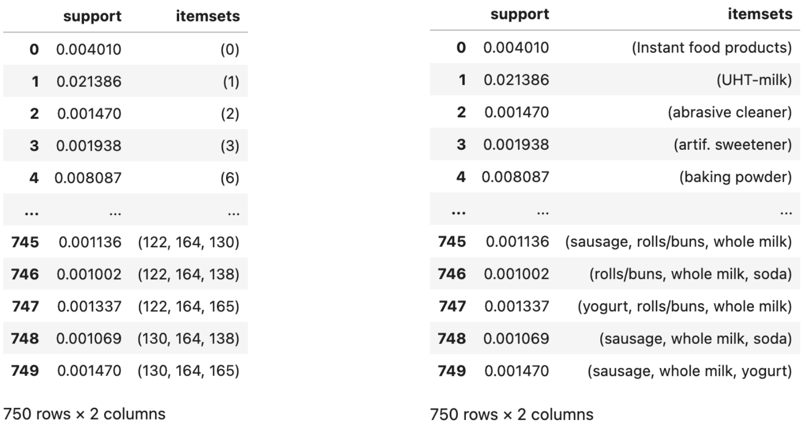
we are going to start setting some parameters in the function, e.g., set the min_support value to 0.001, use_colnames = True.



```
apr_nonames = apriori(transactions_te, min_support=0.001)
apr_nonames
```

```
apr_names = apriori(transactions_te, min_support=0.001,use_colnames=True)
apr_names
```

Fig.3.Apriori Codes



	support	itemsets
0	0.004010	(0)
1	0.021386	(1)
2	0.001470	(2)
3	0.001938	(3)
4	0.008087	(6)
...	...	...
745	0.001136	(122, 164, 130)
746	0.001002	(122, 164, 138)
747	0.001337	(122, 164, 165)
748	0.001069	(130, 164, 138)
749	0.001470	(130, 164, 165)

750 rows × 2 columns

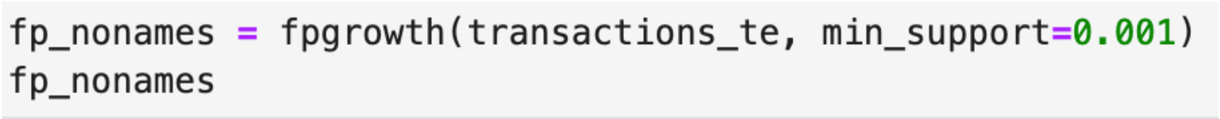
	support	itemsets
0	0.004010	(Instant food products)
1	0.021386	(UHT-milk)
2	0.001470	(abrasive cleaner)
3	0.001938	(artif. sweetener)
4	0.008087	(baking powder)
...	...	...
745	0.001136	(sausage, rolls/buns, whole milk)
746	0.001002	(rolls/buns, whole milk, soda)
747	0.001337	(yogurt, rolls/buns, whole milk)
748	0.001069	(sausage, whole milk, soda)
749	0.001470	(sausage, whole milk, yogurt)

750 rows × 2 columns

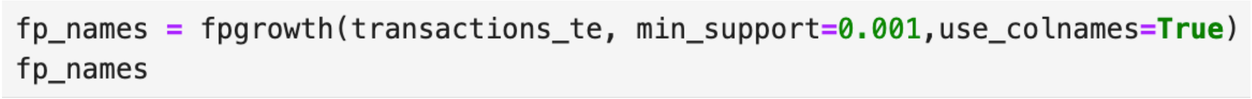
Fig.4.Apriori Results

FP-Growth Algorithm

The same as Apriori algorithm, we set the min_support value to 0.001.

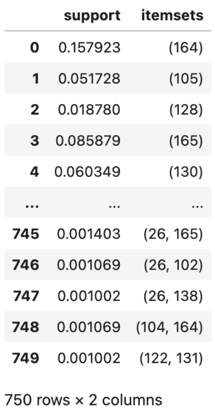


```
fp_nonames = fpgrowth(transactions_te, min_support=0.001)
fp_nonames
```



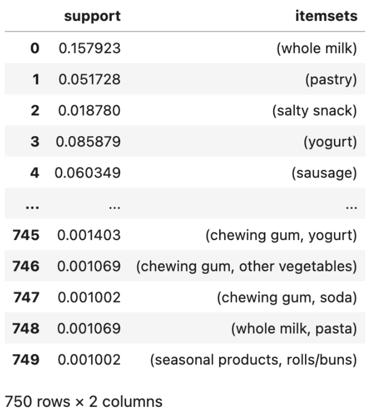
```
fp_names = fpgrowth(transactions_te, min_support=0.001,use_colnames=True)
fp_names
```

Fig.5.FP-Growth Codes



	support	itemsets
0	0.157923	(164)
1	0.051728	(105)
2	0.018780	(128)
3	0.085879	(165)
4	0.060349	(130)
...	...	...
745	0.001403	(26, 165)
746	0.001069	(26, 102)
747	0.001002	(26, 138)
748	0.001069	(104, 164)
749	0.001002	(122, 131)

750 rows × 2 columns



	support	itemsets
0	0.157923	(whole milk)
1	0.051728	(pastry)
2	0.018780	(salty snack)
3	0.085879	(yogurt)
4	0.060349	(sausage)
...	...	...
745	0.001403	(chewing gum, yogurt)
746	0.001069	(chewing gum, other vegetables)
747	0.001002	(chewing gum, soda)
748	0.001069	(whole milk, pasta)
749	0.001002	(seasonal products, rolls/buns)

750 rows × 2 columns

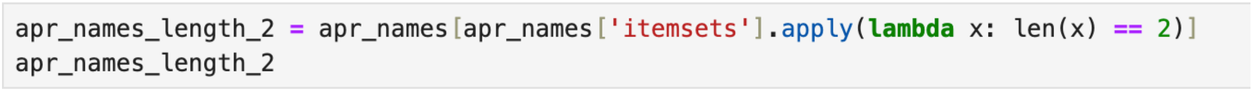
Fig.6.FP-Growth Results

Running these codes, we can get the result (Fig.4 and Fig.6). The result is a table with two columns: `itemsets` and `support`. The `itemsets` column contains the combinations of items, and the `support` column contains the support values for each itemset[4]. The support value represents the frequency of occurrence of each itemset in the dataset. Each row in the table represents a different itemset and its corresponding support value.

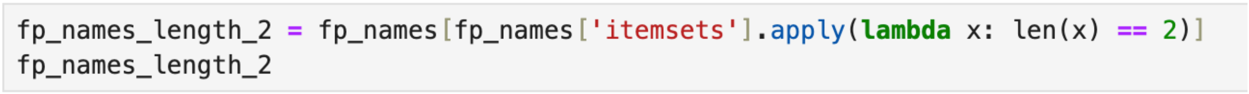
4.2 Inspect frequent items.

In this section, we will display and analyze the Analysis steps b) and c), using the Apriori algorithm and FP-Growth algorithm.

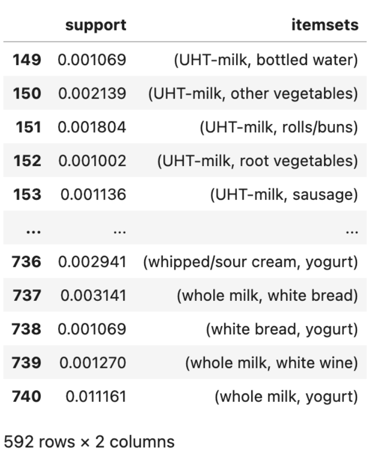
Make combinations of 2 dataset items



```
apr_names_length_2 = apr_names[apr_names['itemsets'].apply(lambda x: len(x) == 2)]
apr_names_length_2
```

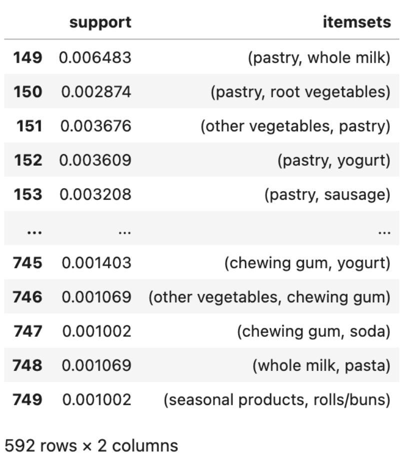


```
fp_names_length_2 = fp_names[fp_names['itemsets'].apply(lambda x: len(x) == 2)]
fp_names_length_2
```



	support	itemsets
149	0.001069	(UHT-milk, bottled water)
150	0.002139	(UHT-milk, other vegetables)
151	0.001804	(UHT-milk, rolls/buns)
152	0.001002	(UHT-milk, root vegetables)
153	0.001136	(UHT-milk, sausage)
...	...	...
736	0.002941	(whipped/sour cream, yogurt)
737	0.003141	(whole milk, white bread)
738	0.001069	(white bread, yogurt)
739	0.001270	(whole milk, white wine)
740	0.011161	(whole milk, yogurt)

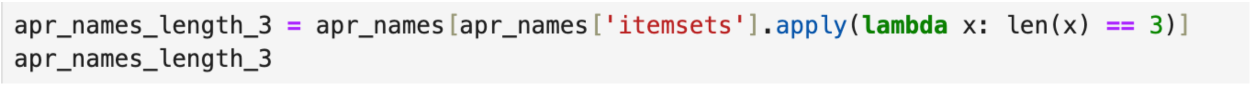
592 rows × 2 columns



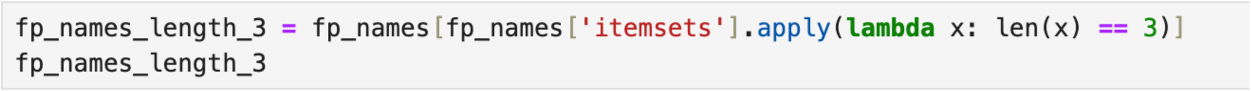
	support	itemsets
149	0.006483	(pastry, whole milk)
150	0.002874	(pastry, root vegetables)
151	0.003676	(other vegetables, pastry)
152	0.003609	(pastry, yogurt)
153	0.003208	(pastry, sausage)
...	...	...
745	0.001403	(chewing gum, yogurt)
746	0.001069	(other vegetables, chewing gum)
747	0.001002	(chewing gum, soda)
748	0.001069	(whole milk, pasta)
749	0.001002	(seasonal products, rolls/buns)

592 rows × 2 columns

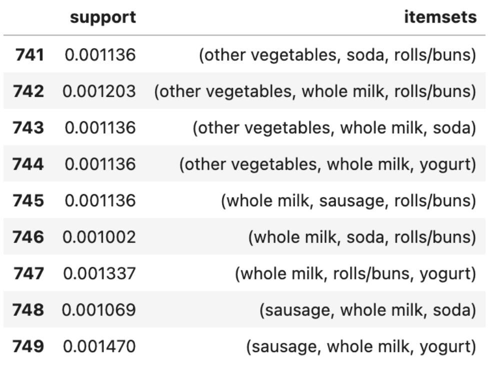
Make combinations of 3 dataset items



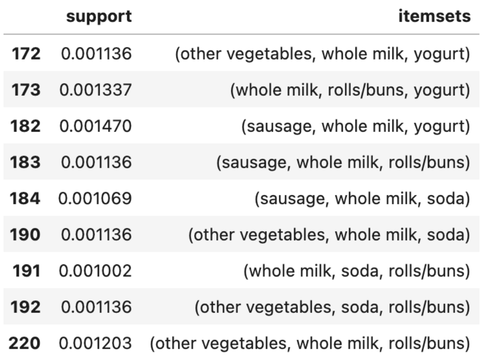
```
apr_names_length_3 = apr_names[apr_names['itemsets'].apply(lambda x: len(x) == 3)]
apr_names_length_3
```



```
fp_names_length_3 = fp_names[fp_names['itemsets'].apply(lambda x: len(x) == 3)]
fp_names_length_3
```



	support	itemsets
741	0.001136	(other vegetables, soda, rolls/buns)
742	0.001203	(other vegetables, whole milk, rolls/buns)
743	0.001136	(other vegetables, whole milk, soda)
744	0.001136	(other vegetables, whole milk, yogurt)
745	0.001136	(whole milk, sausage, rolls/buns)
746	0.001002	(whole milk, soda, rolls/buns)
747	0.001337	(whole milk, rolls/buns, yogurt)
748	0.001069	(sausage, whole milk, soda)
749	0.001470	(sausage, whole milk, yogurt)



	support	itemsets
172	0.001136	(other vegetables, whole milk, yogurt)
173	0.001337	(whole milk, rolls/buns, yogurt)
182	0.001470	(sausage, whole milk, yogurt)
183	0.001136	(sausage, whole milk, rolls/buns)
184	0.001069	(sausage, whole milk, soda)
190	0.001136	(other vegetables, whole milk, soda)
191	0.001002	(whole milk, soda, rolls/buns)
192	0.001136	(other vegetables, soda, rolls/buns)
220	0.001203	(other vegetables, whole milk, rolls/buns)

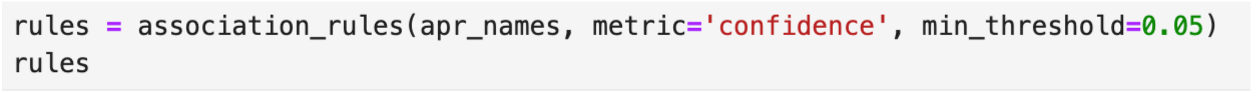
From the results, we can conclude that the experiment has successfully calculated the frequency of items in the dataset, these items have support values indicating their frequency of occurrence. From the itemsets' length of 3: (sausage, whole milk, rolls/buns) in `itemsets` has the highest supported value, it indicates that they are frequently purchased together.

4.3 Generate association rules and analyze the results using confidence, support, and lift.

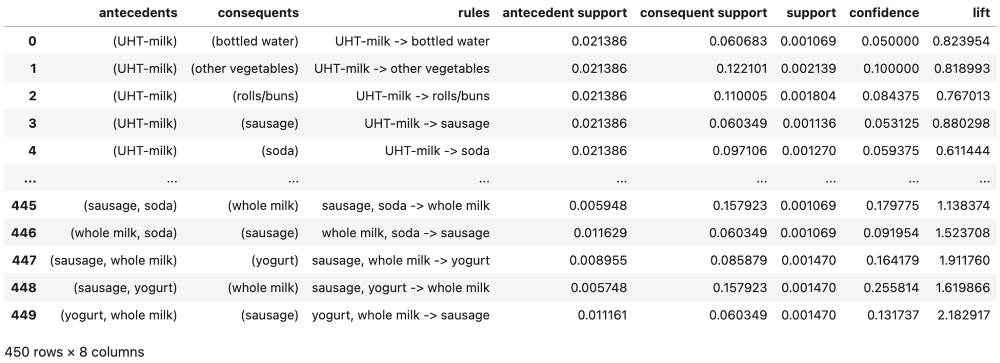
The purpose of lift is to determine how much more often the antecedent and consequent occur together than would be expected if their occurrences were independent of each other.

- If the lift is greater than 1, it suggests that the presence of the items on the LHS (A) has increased the probability that the items on the right-hand side will occur on this transaction.
- If the lift is below 1, it suggests that the presence of the items on LHS(B) make the probability that the items on the RHS (B) will be part of the transaction is lower.
- If the lift is 1, it indicates that the items on the left and right are independent.

$$Lift(X \rightarrow Y) = \frac{Support(X \cup Y)}{Support(Y) * Support(X)}$$

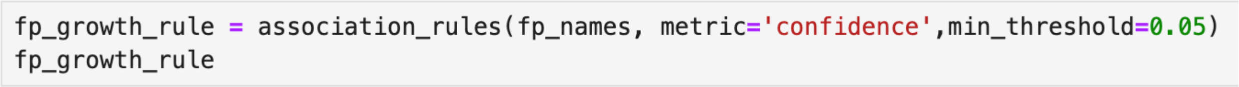


```
rules = association_rules(apr_names, metric='confidence', min_threshold=0.05)
rules
```

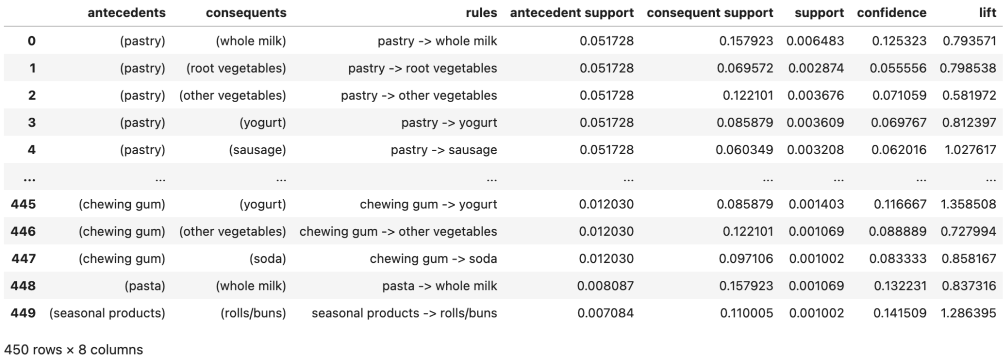


	antecedents	consequents	rules	antecedent support	consequent support	support	confidence	lift
0	(UHT-milk)	(bottled water)	UHT-milk -> bottled water	0.021386	0.060683	0.001069	0.050000	0.823954
1	(UHT-milk)	(other vegetables)	UHT-milk -> other vegetables	0.021386	0.122101	0.002139	0.100000	0.818993
2	(UHT-milk)	(rolls/buns)	UHT-milk -> rolls/buns	0.021386	0.110005	0.001804	0.084375	0.767013
3	(UHT-milk)	(sausage)	UHT-milk -> sausage	0.021386	0.060349	0.001136	0.053125	0.880298
4	(UHT-milk)	(soda)	UHT-milk -> soda	0.021386	0.097106	0.001270	0.059375	0.611444
...	...	...	...	...	...	...	...	...
445	(sausage, soda)	(whole milk)	sausage, soda -> whole milk	0.005948	0.157923	0.001069	0.179775	1.138374
446	(whole milk, soda)	(sausage)	whole milk, soda -> sausage	0.011629	0.060349	0.001069	0.091954	1.523708
447	(sausage, whole milk)	(yogurt)	sausage, whole milk -> yogurt	0.008955	0.085879	0.001470	0.164179	1.911760
448	(sausage, yogurt)	(whole milk)	sausage, yogurt -> whole milk	0.005748	0.157923	0.001470	0.255814	1.619866
449	(yogurt, whole milk)	(sausage)	yogurt, whole milk -> sausage	0.011161	0.060349	0.001470	0.131737	2.182917

450 rows × 8 columns



```
fp_growth_rule = association_rules(fp_names, metric='confidence',min_threshold=0.05)
fp_growth_rule
```



	antecedents	consequents	rules	antecedent support	consequent support	support	confidence	lift
0	(pastry)	(whole milk)	pastry -> whole milk	0.051728	0.157923	0.006483	0.125323	0.793571
1	(pastry)	(root vegetables)	pastry -> root vegetables	0.051728	0.069572	0.002874	0.055556	0.798538
2	(pastry)	(other vegetables)	pastry -> other vegetables	0.051728	0.122101	0.003676	0.071059	0.581972
3	(pastry)	(yogurt)	pastry -> yogurt	0.051728	0.085879	0.003609	0.069767	0.812397
4	(pastry)	(sausage)	pastry -> sausage	0.051728	0.060349	0.003208	0.062016	1.027617
...	...	...	...	...	...	...	...	...
445	(chewing gum)	(yogurt)	chewing gum -> yogurt	0.012030	0.085879	0.001403	0.116667	1.358508
446	(chewing gum)	(other vegetables)	chewing gum -> other vegetables	0.012030	0.122101	0.001069	0.088889	0.727994
447	(chewing gum)	(soda)	chewing gum -> soda	0.012030	0.097106	0.001002	0.083333	0.858167
448	(pasta)	(whole milk)	pasta -> whole milk	0.008087	0.157923	0.001069	0.132231	0.837316
449	(seasonal products)	(rolls/buns)	seasonal products -> rolls/buns	0.007084	0.110005	0.001002	0.141509	1.286395

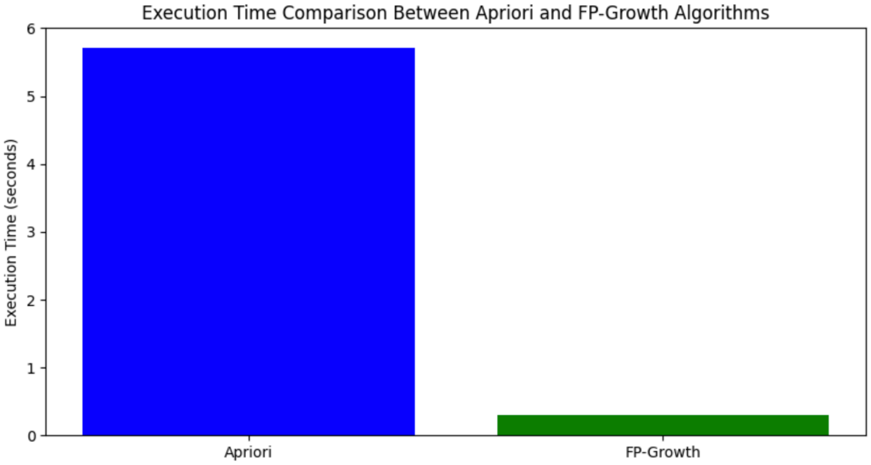
450 rows × 8 columns

In a shut, we are looking for rules with a lift > 1 and preferably with a higher level of support. E.g., seasonal products-> rolls/buns lift is 1.286395, this suggests a positive association between "seasonal products" and "rolls/buns", indicating that they are more likely to be purchased together than if their occurrences were independent.

5.0 Conclusion

In this paper, we talk about the description of the dataset, data processing, what is Apriori algorithm and FP-Growth algorithm, how to apply these algorithms in the dataset, and how well do they perform. But we know that the Apriori algorithm and FP-Growth algorithm end up with the same results, the process is different.

Here is a graph comparing the speed of running in the same dataset. From the graph, we can see that FP-Growth is more efficient than Apriori, FP-Growth is an efficient mining method of frequent patterns in large datasets.



In a shut, the analysis using Apriori and FP-Growth algorithms on the grocery store dataset reveals valuable insights into customer purchasing patterns, provides a very useful and powerful decision-making method for sellers.

References

[1] Kurnia, Y., Isharianto, Y., Giap, Y. C., Hermawan, A., & Riki, R. (2019). Study of application of data mining market basket analysis for knowing sales pattern (association of items) at the O! Fish restaurant using apriori algorithm. Journal of Physics: Conference Series, 1175, 012047. https://doi.org/10.1088/1742-6596/1175/1/012047

[2] Kaur, M., & Kang, S. (2016). Market Basket Analysis: Identify the changing trends of market data using association rule mining. Procedia Computer Science, 85, 78–85. https://doi.org/10.1016/j.procs.2016.05.180

[3] Efrat, A., Gernowo, R., & Farikhin. (2020). Consumer purchase patterns based on market basket analysis using apriori algorithms. Journal of Physics, 1524(1), 012109. https://doi.org/10.1088/1742-6596/1524/1/012109

[4] Nasreen, S., Azam, M. A., Shehzad, K., Naeem, U., & Ghazanfar, M. A. (2014). Frequent Pattern Mining Algorithms for finding associated frequent patterns for data streams: a survey. Procedia Computer Science, 37, 109–116. https://doi.org/10.1016/j.procs.2014.08.019

[5] S. S. Khedkar and S. Kumari, "Market Basket Analysis using A-Priori Algorithm and FP-Tree Algorithm," 2021 International Conference on Artificial Intelligence and Machine Vision (AIMV), Gandhinagar, India, 2021, pp. 1-6, doi: 10.1109/AIMV53313.2021.9670981.

[6] Shariff, S.s & Bakri, Zurriyati & Hamzah, Pa'ezah. (2016). Association Rules for Purchase Dependency of Grocery Items. Social and Management Research Journal. 13. 61. 10.24191/smrj.v13i2.5271.

[7] Petimar, J., Moran, A. J., Grummon, A. H., Anderson, E., Lurie, P., John, S., Rimm, E. B., & Thorndike, A. N. (2023). In-Store Marketing and Supermarket Purchases: associations overall and by transaction SNAP status. American Journal of Preventive Medicine, 65(4), 587–595. https://doi.org/10.1016/j.amepre.2023.02.029

[8] Shelke, R. R. (2017). Data mining for supermarket sale analysis using association rule. International Journal of Trend in Scientific Research and Development, Volume-1(Issue-4). https://doi.org/10.31142/ijtsrd94