Instacart Market Basket Analysis

ISYE 7406 - Data Mining and Statistical Learning

April 14 2024

—

Stephen (Hyun Gil) Kim	Gatech ID: hkim3183	Email: hkim3183@gatech.edu
Nildip Chaudhuri	Gatech ID: nchaudhuri6	Email:nchaudhuri6@gatech.edu
Son Tran	Gatech ID: stran64	Email: sontran@gatech.edu
Mehul Dodia	Gatech ID: mdodia7	Email: mehulcd@gatech.edu

Abstract

The shift toward online grocery shopping, accelerated by pandemic-induced lockdowns, has highlighted the need for retailers to adapt swiftly to changing consumer preferences and expectations. The increased reliance on platforms like Instacart for grocery delivery services has underscored the importance of using predictive analytics to anticipate and fulfill customer needs effectively.

In response to this demand, our project delves into the realm of data analytics with the Instacart Market Basket Analysis. Through this endeavor, we aim to deliver actionable insights that can potentially help retailers (both physical and online) offer more personalized shopping experiences and streamline operations, thereby increasing customer satisfaction and business efficiency.

Introduction

Background

In the digital commerce era, online marketplaces like Amazon, eBay, Alibaba, and JD.com, and many others have emerged as dominant players, collectively responsible for over one-third of global online shopping transactions. According to a survey conducted by Statista 2023, 15 percent of digital shoppers allocated their spending to e-grocers and supermarkets, reflecting a notable preference for online grocery shopping. As a result, understanding customer purchasing behavior emerges as a pivotal challenge, with transaction analysis serving as a key technique to glean actionable insights.

Problem Statement

Transaction analysis holds profound implications for businesses across various sectors, especially in the retail domain, where comprehension of customer purchasing patterns is instrumental in driving sales. Despite the availability of robust datasets like Instacart's, many retailers struggle to efficiently utilize such data to gain actionable insights. Traditional analysis methods can be time-consuming and may not effectively handle the volume and complexity of the data. Advanced algorithms such as A-Priori, FP-Growth, and ECLAT offer solutions to these challenges, yet their comparative effectiveness in real-world scenarios is not well-documented, especially in the context of grocery shopping data.

Objectives

The primary objective of this project is to employ common association rules embedded within the historical transactions recorded in the Instacart dataset. Leveraging the A-Priori, FP-Growth, and ECLAT algorithms - known for their efficacy in frequent itemset mining - we aim to extract actionable insights from the dataset. Each algorithm boasts unique strengths and strategies, offering versatile approaches to association rule mining tailored to diverse datasets and mining scenarios. Unlike probabilistic ensemble methods, which necessitate extensive feature engineering and may yield suboptimal results due to inherent biases, our chosen algorithms offer a more direct and interpretable approach to uncovering important association rules between products. Moreover, given the sparsity of the dataset, we endeavor to explore association rules not only between individual products but also between departments and aisles. By leveraging department and aisle associations, retailers (both online and physical) can potentially enhance marketing strategies, optimize product placements, and improve inventory management.

In the subsequent sections of this report, we detail our exploratory data analysis and the methodology for association rule mining (including the comparison of these algorithms in terms of efficiency, scalability, and practical applicability), present our findings, and offer insights garnered from the application of these methodologies.

Data Sources

For our data source, we utilized the Instacart dataset available on Kaggle, which was originally provided by Instacart for a previous competition. This extensive dataset comprises anonymized records of over 3 million grocery orders, originating from a diverse pool of more than 200,000 real Instacart users.

Each user's data within the dataset encompasses a subset of their orders, ranging from 4 to 100 orders, offering a comprehensive view of their purchasing behavior over time. Crucially, each order entry within the dataset meticulously documents the sequence of products purchased, providing invaluable insights into evolving buying habits and preferences. In addition to the product sequences, the dataset contains rich metadata, including details such as the week and hour of the day each order was placed, as well as the time elapsed between orders. This wealth of information enables us to analyze temporal trends, discern recurring patterns, and ultimately develop robust predictive models.

The dataset is structured into several files, each serving a specific purpose in our analysis:

- Aisles.csv: Contains aisle_id and aisle as primary columns, providing categorization of products by aisle.
- Departments.csv: Includes department_id and department as primary columns, offering further categorization of products by department.
- Order_products_*.csv: Multiple files capturing the relationship between orders and products, including details such as order ID, product ID, and add-to-cart order.
- Orders.csv: Provides comprehensive information about each order, including user ID, order ID, order number, order day of week, and order hour of day.
- Products.csv: Contains product information, including product ID, product name, aisle ID, and department ID.

Our use case is atypical to that of the problems we have faced during the semester. The reason is because there is no clear loss function and evaluation metric. Our implementation is mainly for quick recommendations and consumer purchase analysis such as what products are purchased together, which departments and aisle within the department should be placed together, etc. We have created a synthetic “evaluation function” in the conclusion to see the correlation/overlap with the Instacart frequent itemsets in their websites and our findings. This is not a perfect evaluation function but is a great litmus test to see how well our algorithms have performed based on our hyperparameter tuning.

The extensive nature of this dataset, coupled with its diverse array of features and metadata, serves as a robust foundation for our predictive analysis.

You can access the dataset through the following link. Or you can get the data locally via the kaggle API for python.

Methodology

Association Rule Mining

We conduct association rule mining using the A-Priori, FP-Growth, and ECLAT algorithms. These algorithms offer distinct methodologies for frequent itemset mining, allowing us to explore different strategies for uncovering association rules within the dataset.

1. A-priori Algorithm

The Apriori algorithm is a classical algorithm for frequent itemset mining and association rule learning over transactional databases.It works based on the principle of generating candidate itemsets of increasing size and then scanning the database to determine the support of each candidate.

The worst-case time complexity of the Apriori algorithm is exponential, specifically O(2^N), where N is the number of items in the database. However, optimizations such as pruning techniques can improve performance in practice.

Strengths:

- Ease of implementation
- Incremental updates just require a new run of the algorithm

Weaknesses

- Does suffer when trying to scale to extremely large datasets (billions of instances)
- Requires high memory usage unless you are using a map-reduce functionality

2. FP-Growth Algorithm (Frequent Pattern Growth)

FP-Growth is an efficient algorithm for mining frequent itemsets without candidate generation. It constructs a compact data structure called the FP-tree, which represents the input dataset and exploits the downward closure property of frequent itemsets.

Building the FP-tree has a time complexity of O(N * log(M)), where N is the number of transactions and M is the average length of transactions. After constructing the FP-tree, the time complexity for mining frequent itemsets is O(N * d), where N is the number of transactions and d is the average depth of the FP-tree.

Strengths:

- More efficient than the Apriori algorithm. Eliminates the need for generating candidate itemsets, which reduces the computational overhead.
- Efficiently handles large datasets by constructing a compact FP-tree structure.
- Suitable for datasets with high dimensionality and sparse data.

Weaknesses:

- Requires more memory
- Harder to implement that Apriori (code implementation)

3. ECLAT Algorithm (Equivalence Class Clustering and Bottom-Up Lattice Traversal):

ECLAT is an algorithm for mining frequent itemsets by exploiting vertical data format representation. It constructs an intersection lattice based on the frequency of itemsets and uses a depth-first traversal to find frequent itemsets.

The time complexity of the ECLAT algorithm is similar to that of Apriori and heavily depends on the size of the transaction database and the number of frequent itemsets. It involves constructing an intersection lattice, which has a time complexity of O(N^2), where N is the number of frequent items. Traversing the lattice to find frequent itemsets has a time complexity of O(3^N), where N is the number of frequent items.

Strengths:

- Efficiently handles vertical data format, which is suitable for datasets with a large number of items and sparse transactions.

    - This is our use case
- Does not generate candidate itemsets, leading to reduced computational overhead compared to Apriori.
- Can be more memory-efficient compared to Apriori for certain datasets.

Weaknesses:

- May require more memory than FP-Growth for constructing the intersection lattice.
- Performance can degrade when dealing with datasets with low average transaction width.

We had to implement this algorithm from scratch as there are no popular modules. The latter might be the reason why it took significantly longer to execute as we did not optimize in the backend.

4. Comparison

The Apriori algorithm is a traditional algorithm that is easy to understand but may suffer from scalability issues. On the other hand, the FP-Growth algorithm is a more efficient alternative to Apriori that eliminates the need for candidate generation, making it suitable for large datasets. ECLAT is efficient for datasets with vertical data representation, offering advantages over Apriori in memory usage but may not be as scalable as FP-Growth for certain datasets.

ETL Pipeline

Our ETL pipeline was fairly straight forward. We used the Kaggle API to get the flat files. Next we decided to use Neo4J to store the data efficiently. The schema (bipartite graph structure) can be seen in the appendix. We wanted to emulate how a production environment would store the data in the real world. The latter is because a graph structure is more efficient when dealing with a lot of joins in queries.

Our Novel Approach

In traditional market basket analysis (MBA), the focus lies on understanding the associations between individual products purchased together. However, in the context of physical retail stores, there exists a rich hierarchy of product organization, typically categorized into departments and further subdivided into aisles. Recognizing the potential value in exploring associations at these hierarchical levels, we embarked on a novel approach to enhance MBA by incorporating Isle IDs and Department IDs into the analysis.

Rather than solely examining associations between individual products, we extended the scope of MBA to encompass associations between Isle IDs and Department IDs. By doing so, we sought to capture higher-level patterns of consumer behavior within retail stores. This approach not only adds depth to the analysis but also addresses the challenge of dealing with sparse datasets containing numerous orders and products. To accomplish this, we utilized association rule mining techniques, similar to those employed in traditional MBA. However, instead of focusing solely on product-level associations, we generated association rules between Isle IDs and Department IDs. These rules reveal insights into which aisles are frequently visited within specific departments, providing valuable information for optimizing store layout, product placement, and marketing strategies.

Advantages of this Modified Approach

Reduced Sparsity Effect: By analyzing associations at the Isle and Department level, we mitigated the impact of sparsity inherent in datasets with extensive product variations and diverse customer preferences.

Applicability to Physical Retail Stores: This approach extends the applicability of market basket analysis to physical retail stores. Understanding the relationships between Isle IDs and Department IDs can inform decisions related to store layout, aisle organization, and product assortment, ultimately enhancing the shopping experience for customers.

Complementary Insights: By combining product-level associations with Aisle and Department-level associations, this approach provides a comprehensive understanding of consumer behavior within retail environments. This holistic view enables retailers to make informed decisions that cater to the diverse needs and preferences of their customer base.

Analysis and Results

Exploratory Data Analysis (EDA)

Before diving into association rule mining results, we conduct exploratory data analysis (EDA) to gain insights into the structure and characteristics of the dataset. This will involve visualizing the distribution of variables, identifying patterns, and detecting outliers or anomalies that may affect the subsequent analysis.

EDA will help us understand the underlying trends and relationships within the data, guiding our approach to association rule mining and informing decisions related to data preprocessing and feature engineering.

We have identified the following insights through exploratory data analysis:

- Customers place significantly more orders on Sunday and Monday (Figure 1)
- Majority of orders happen about 1 week or 1 month since the previous order (Figure 2)
- Most orders have less than 10 grocery items, with 5 being the most popular (Figure 3)
- Two kinds of Bananas, strawberries, spinach, and avocado are the top 5 most popular products (Figure 4)
- Produce, dairy eggs, snacks, beverages, and frozen are the top 5 departments (Figure 5)
- Customers usually place their order between 10 AM and 4 PM across the week (Figure 6)

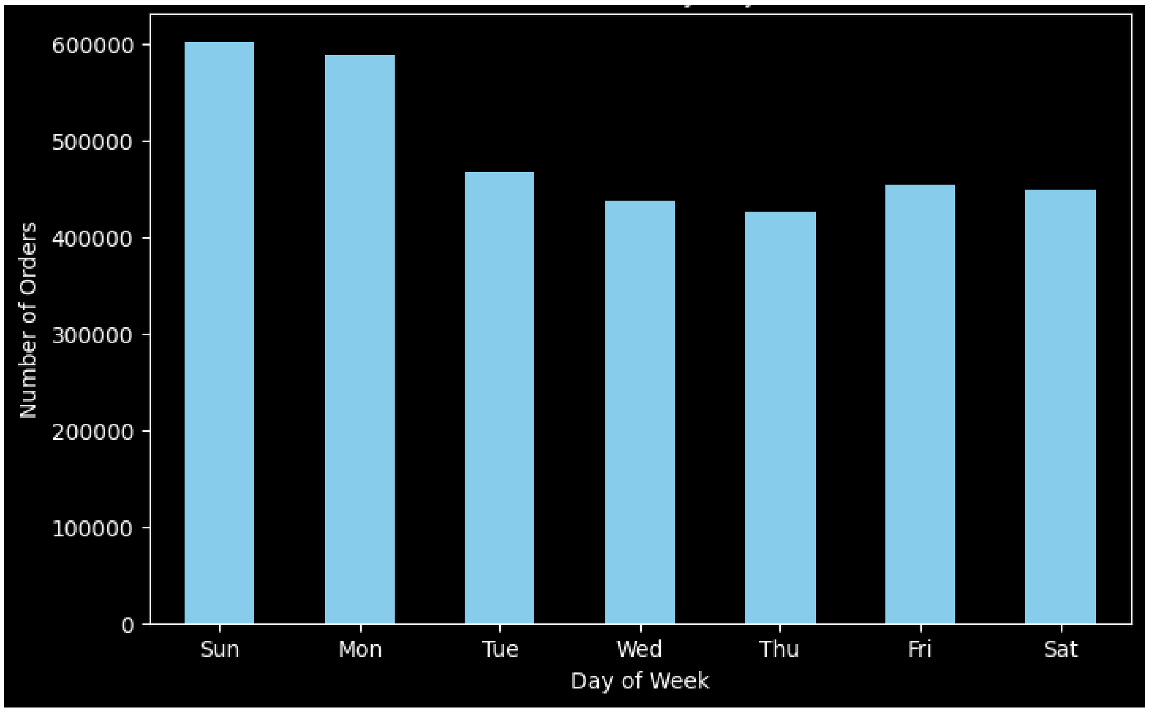


Figure 1. Orders Distribution by Day of Week

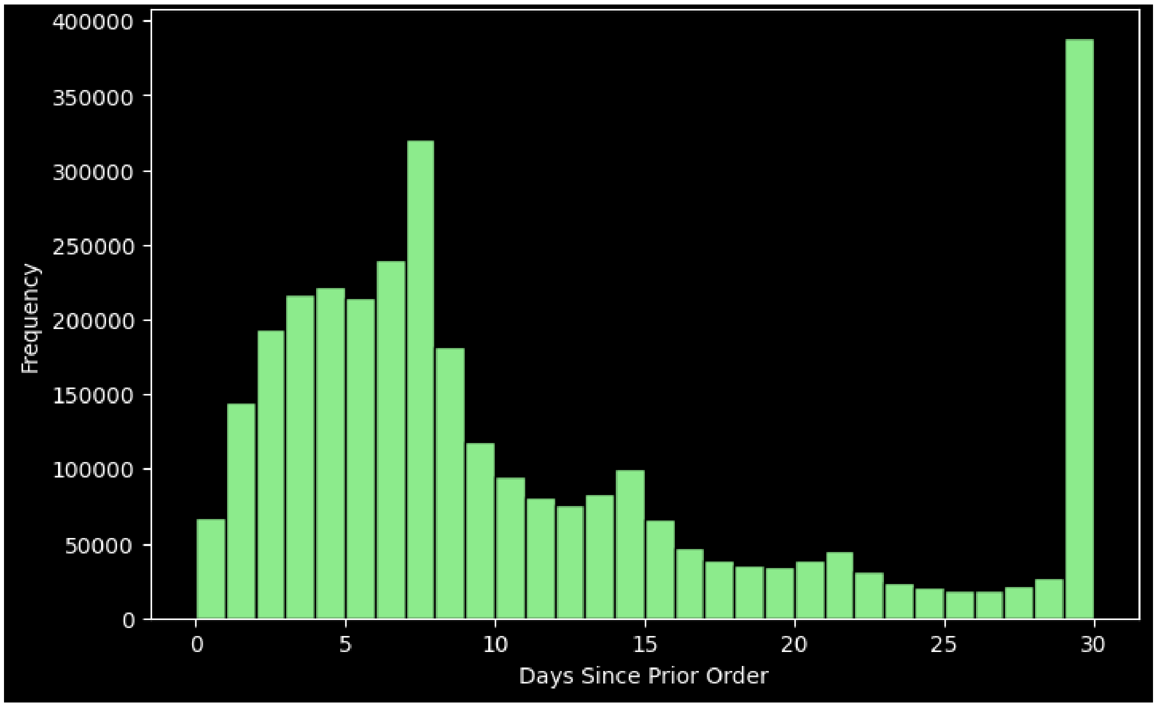


Figure 2. Distribution of Days since Prior order

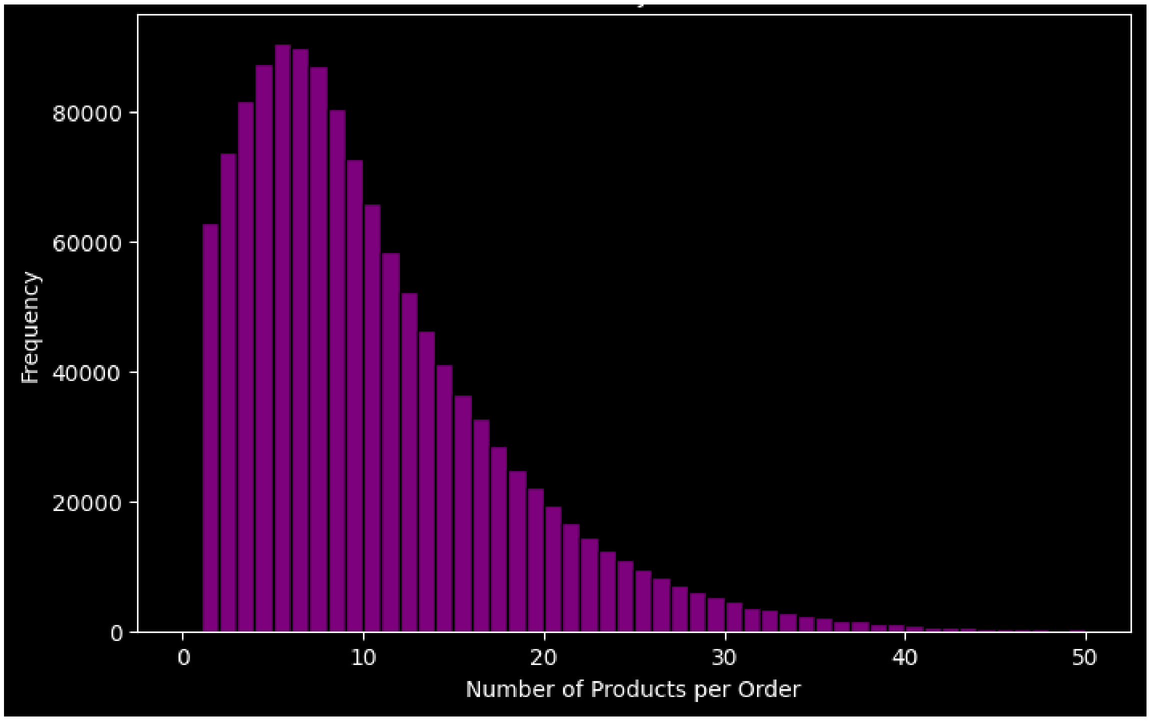


Figure 3. Order quantity distribution

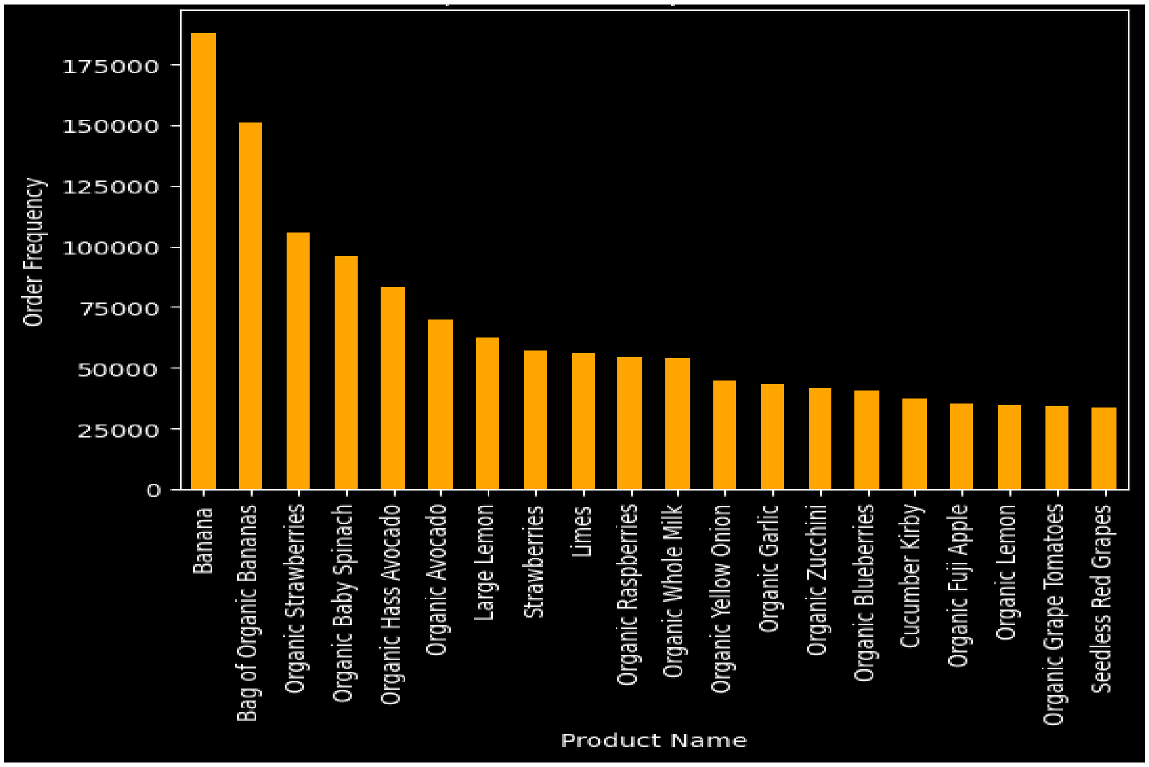


Figure 4. Top 20 most frequent products

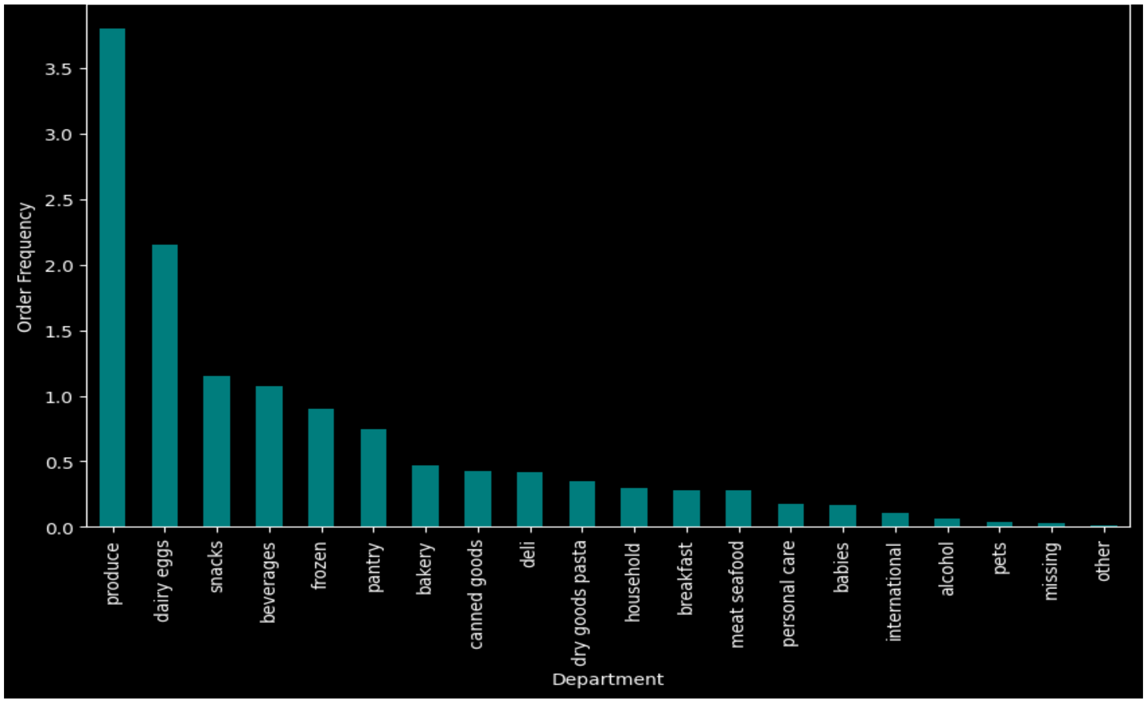


Figure 5. Top 20 departments by order frequency

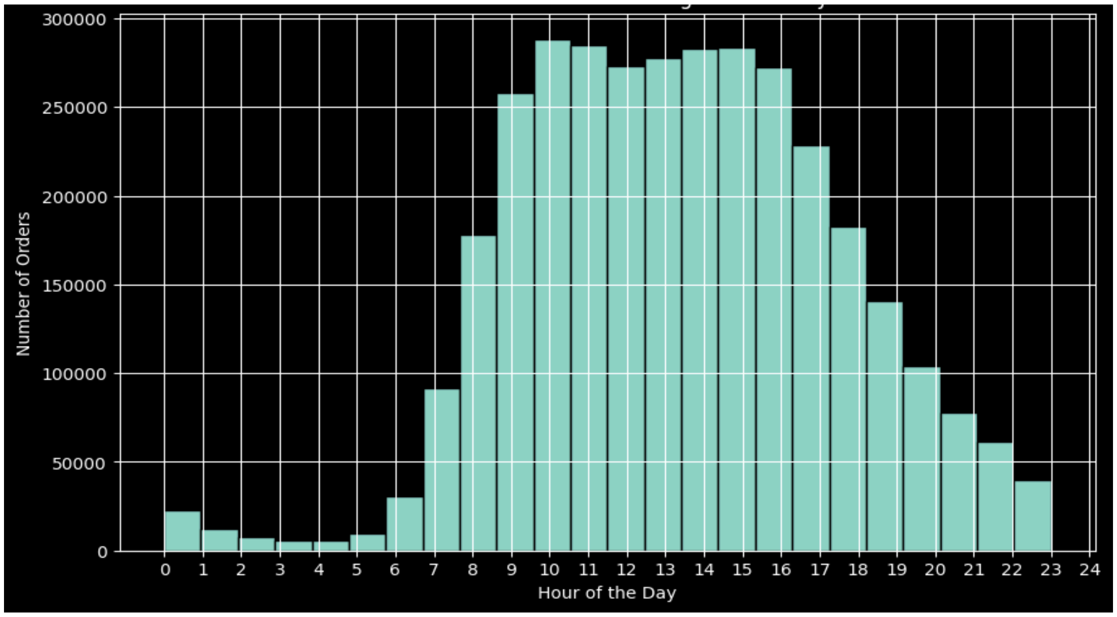


Figure 6. Distribution of order throughout the day

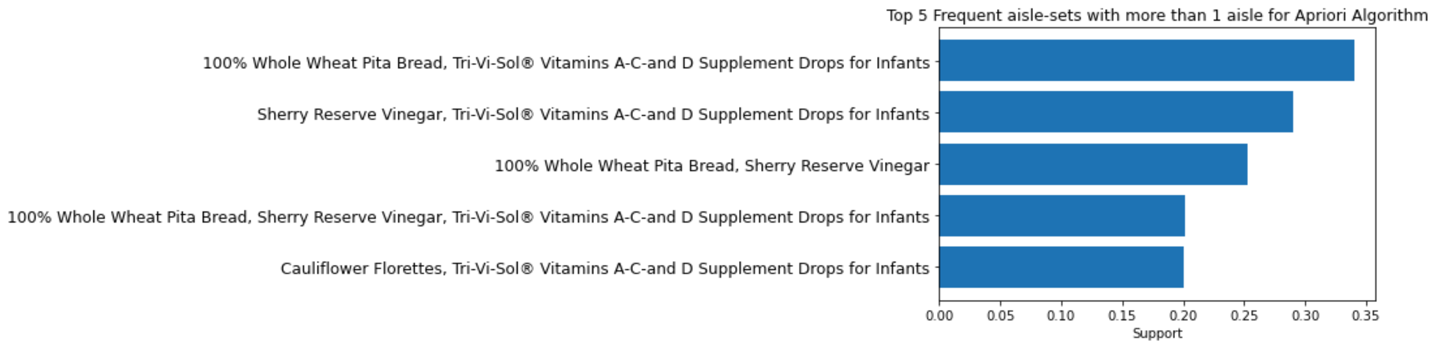
Results

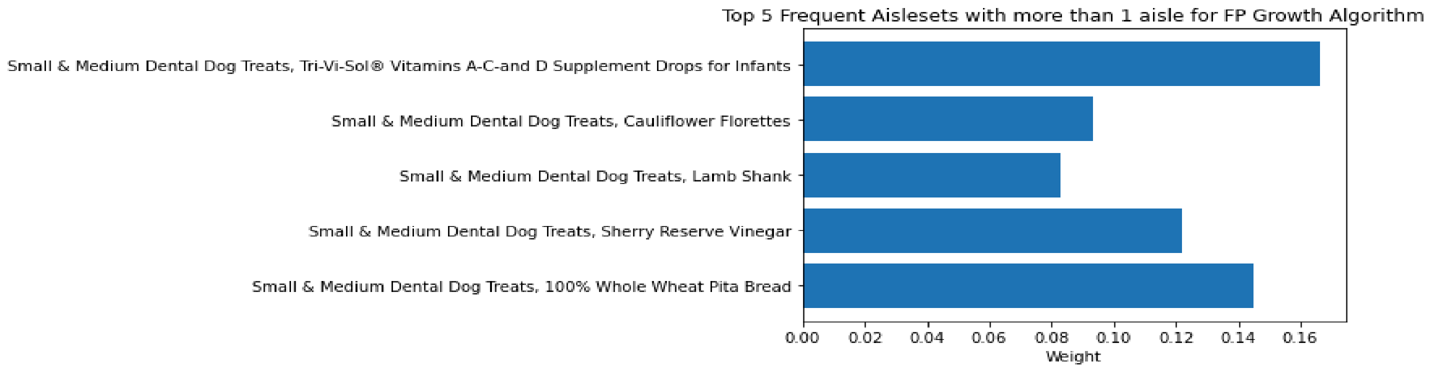
Product/Itemset mining

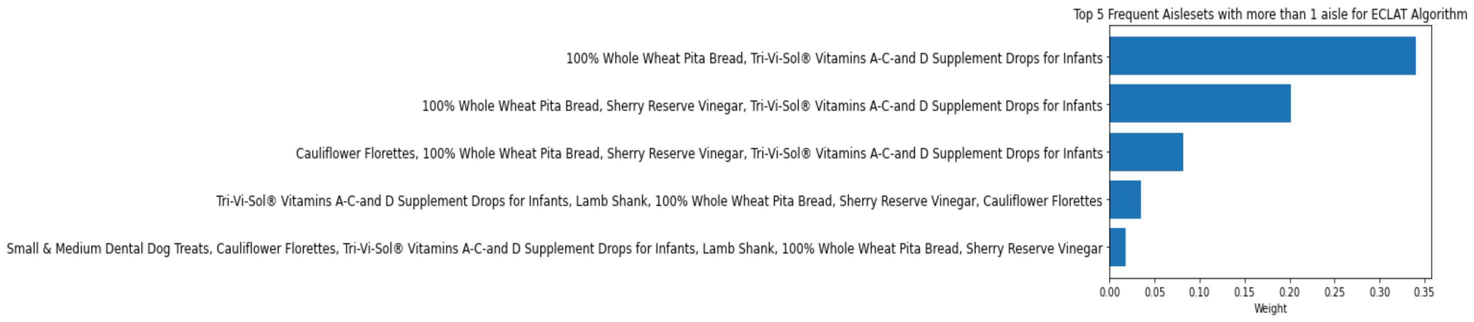
- Banana, Avocados and Strawberries seem to be the most frequently purchased products
- It also seems like Spinach goes well in combination with the above three
- Apriori and FP growth gave identical itemsets for top 10 frequent itemsets
- Eclat found more interesting itemsets like Banana-Onion, Banana-Zucchini, Spinach-Lemon

Aisle/Department mining

- A-priori and ECLAT both gave a higher weight to Pita bread and Vitamin supplements, suggesting it’s better if their aisles are nearby each other
- Vinegar and Bread was another interesting combination with a higher association between their aisles
- It also seems like Dog treats are great combined with other items suggesting its better to keep them scattered in multiple little aisles


Top 5 Frequent aisle-sets with more than 1 aisle for Apriori Algorithm
100% Whole Wheat Pita Bread, Tri-Vi-Sol® Vitamins A-C-and D Supplement Drops for Infants
Sherry Reserve Vinegar, Tri-Vi-Sol® Vitamins A-C-and D Supplement Drops for Infants
100% Whole Wheat Pita Bread, Sherry Reserve Vinegar
100% Whole Wheat Pita Bread, Sherry Reserve Vinegar, Tri-Vi-Sol® Vitamins A-C-and D Supplement Drops for Infants
Cauliflower Florettes, Tri-Vi-Sol® Vitamins A-C-and D Supplement Drops for Infants
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
Support


Top 5 Frequent Aislesets with more than 1 aisle for FP Growth Algorithm
Small & Medium Dental Dog Treats, Tri-Vi-Sol® Vitamins A-C-and D Supplement Drops for Infants
Small & Medium Dental Dog Treats, Cauliflower Florettes
Small & Medium Dental Dog Treats, Lamb Shank
Small & Medium Dental Dog Treats, Sherry Reserve Vinegar
Small & Medium Dental Dog Treats, 100% Whole Wheat Pita Bread
0.00
0.02
0.04
0.06
0.08
0.10
0.12
0.14
0.16
Weight


Top 5 Frequent Aislesets with more than 1 aisle for ECLAT Algorithm
100% Whole Wheat Pita Bread, Tri-Vi-Sol® Vitamins A-C-and D Supplement Drops for Infants
100% Whole Wheat Pita Bread, Sherry Reserve Vinegar, Tri-Vi-Sol® Vitamins A-C-and D Supplement Drops for Infants
Cauliflower Florettes, 100% Whole Wheat Pita Bread, Sherry Reserve Vinegar, Tri-Vi-Sol® Vitamins A-C-and D Supplement Drops for Infants
Tri-Vi-Sol® Vitamins A-C-and D Supplement Drops for Infants, Lamb Shank, 100% Whole Wheat Pita Bread, Sherry Reserve Vinegar, Cauliflower Florettes
Small & Medium Dental Dog Treats, Cauliflower Florettes, Tri-Vi-Sol® Vitamins A-C-and D Supplement Drops for Infants, Lamb Shank, 100% Whole Wheat Pita Bread, Sherry Reserve Vinegar
0.00
0.05
0.10
0.15
0.20
0.25
0.30
0.35
Weight

Conclusions

Synthetic Performance Evaluation

Given that we did not have labeled data and the nature of our problem was not supervised machine learning, we weren't able to leverage traditional performance metrics such as F1 score. Therefore, we determined that comparing our recommendations to Instacart's own publication: 10 for 10: Instacart's Food Trend Hall of Fame￼ would help us assess the efficacy of our findings. We see dog treats, milk, and produce such as avocados and cauliflower come up in their rankings, which is a great signal that our recommendations can help boost profitability by our suggested pairings.

Lessons we have learned

First, we learned that we had limited scalability of the chosen algorithms due to processing requirements, particularly for A-Priori. A-Priori algorithm has an exponential time complexity therefore wasn't the best suitable for our use case.

Secondly, we learned that our analysis could be better suited for traditional, brick-and-mortar retailers not just for grocery delivery. Given that our final output is association mining,, our research is optimized for brick-and-mortar retailers in which customers visit the physical store. However, our analysis is still valuable since distribution centers can optimize product placing for faster delivery.

Finally, we learned that there are multiple opportunities for Instacart to leverage data science. While we don't have the data to analyze, Instacart's internal team could leverage internal datasets to see a more holistic data science effort, processes, one which includes delivery route optimization, product pairings, increasing clickthrough rates, amongst other projects.

Alternative Methods: Graph Neural Network Link Prediction

We had originally wanted to implement a graph neural network model (link prediction). The structure would be similar to that of GraphSage and PinSage, but our use case is a bit different as our graph is a Bipartite Graph (order nodes and product nodes). The reason why we abandoned this early on is because we would need to create our own map-reduce logic like they did in the PinSage paper. This is because a Graph Model is usually very memory/compute intensive. Our initial preparations projected a use of an excess of 250 GBs without optimization. As a result, to implement this model we would need cloud DB vendors Neo4J (aura DB) and cloud compute platforms which would get too expensive.

Appendix

Research

- https://www.kaggle.com/competitions/instacart-market-basket-analysis/overview
- https://ieeexplore.ieee.org/abstract/document/6781357
- https://www2.deloitte.com/content/dam/Deloitte/us/Documents/consumer-business/2022-retail-industry-outlook.pdf
- https://papers.ssrn.com/sol3/papers.cfm?abstract_id=4461121
- https://link.springer.com/chapter/10.1007/978-3-031-40395-8_16

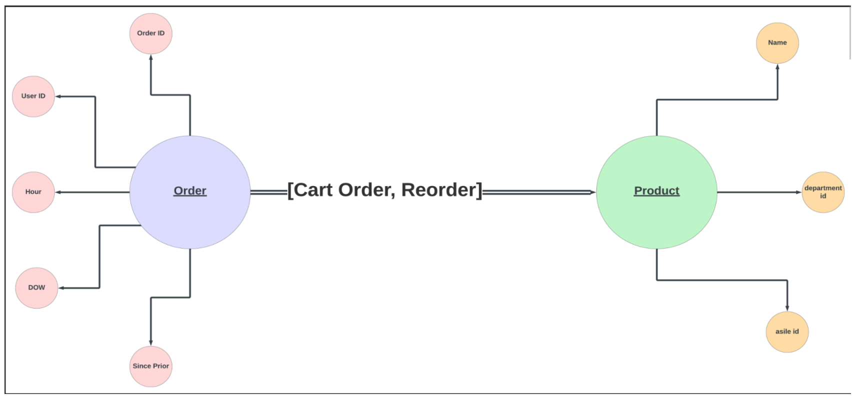
Data source

- https://www.kaggle.com/competitions/instacart-market-basket-analysis/data

Code Repository

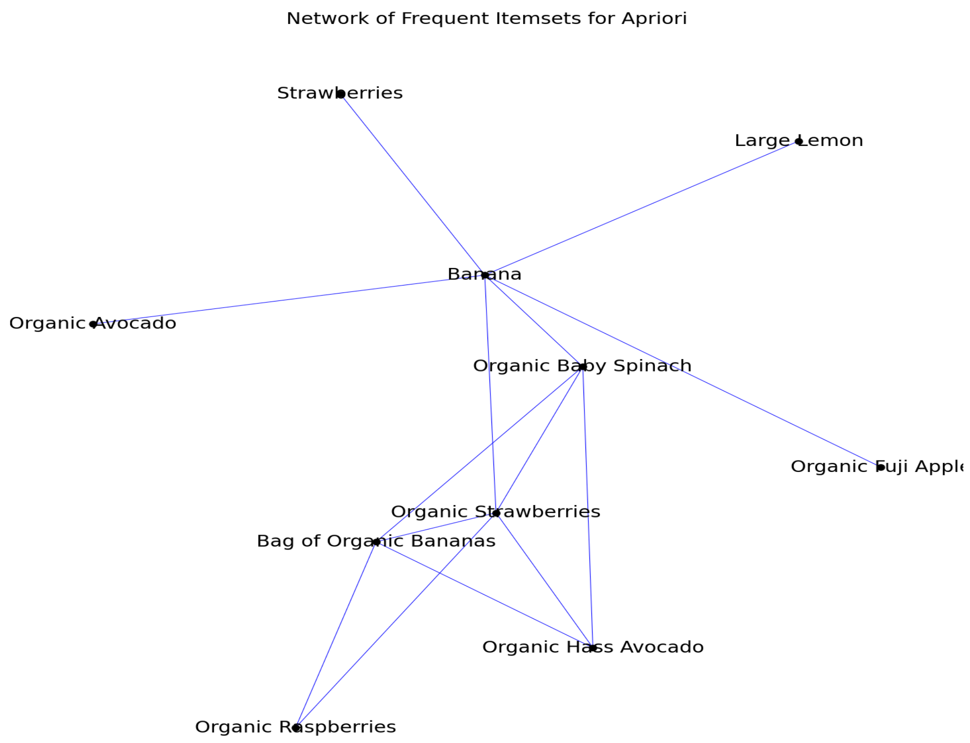
- https://github.gatech.edu/nchaudhuri6/ISYE_7406_GROUP_PROJECT

ETL Pipeline

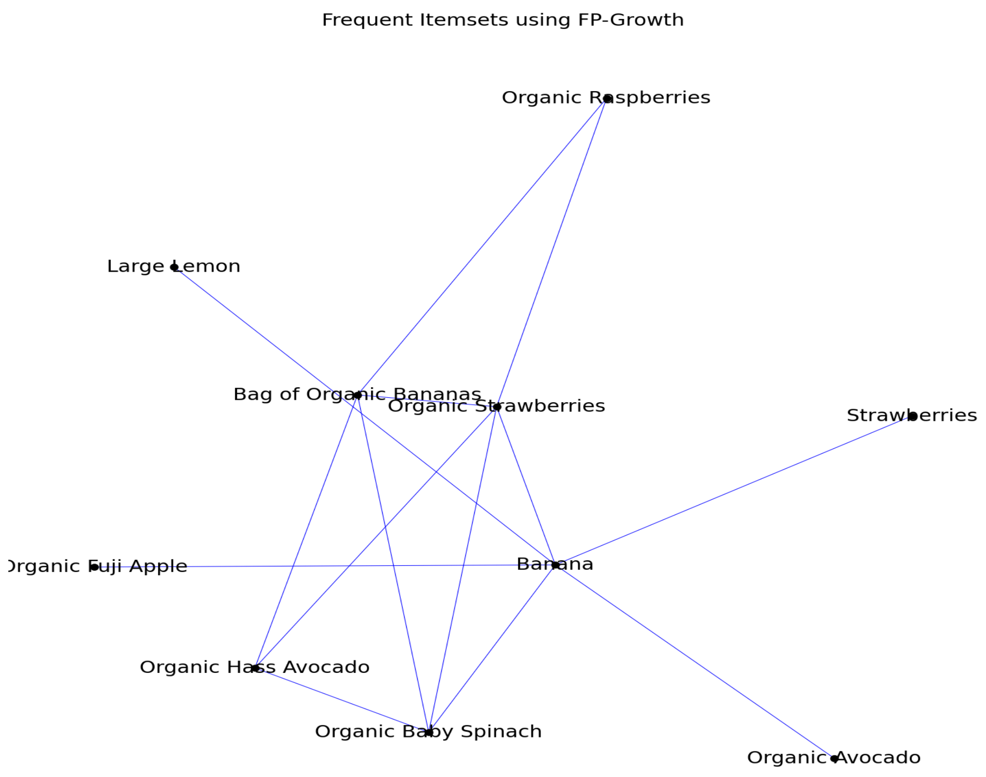


Network Diagrams

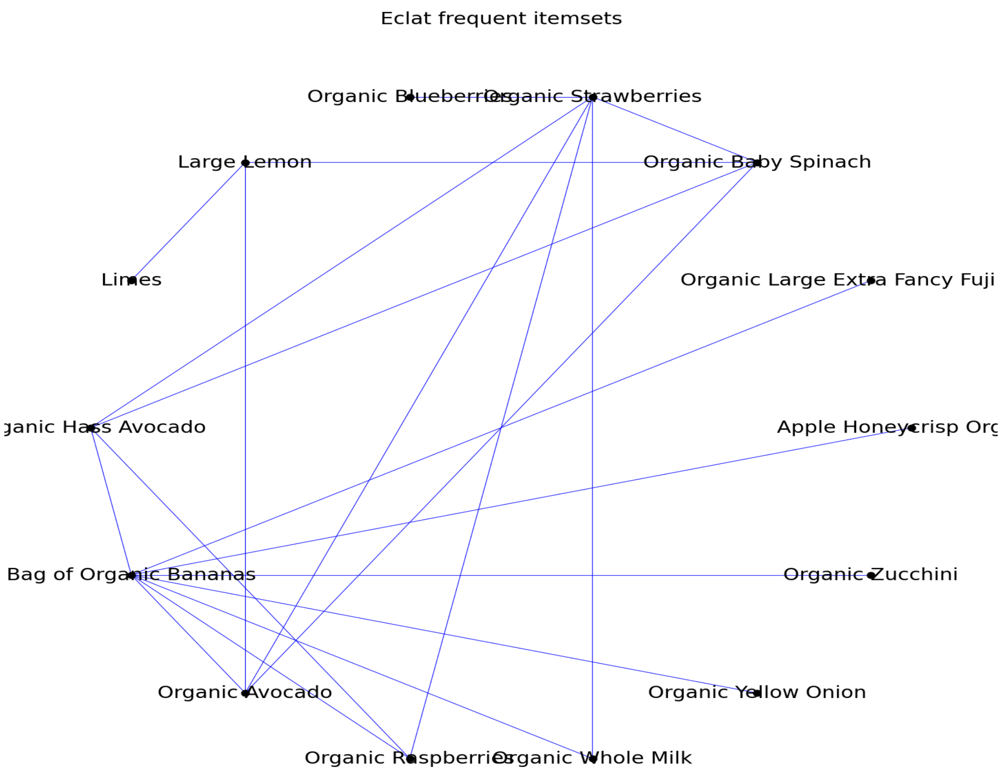
Apriori :



FP-Growth :

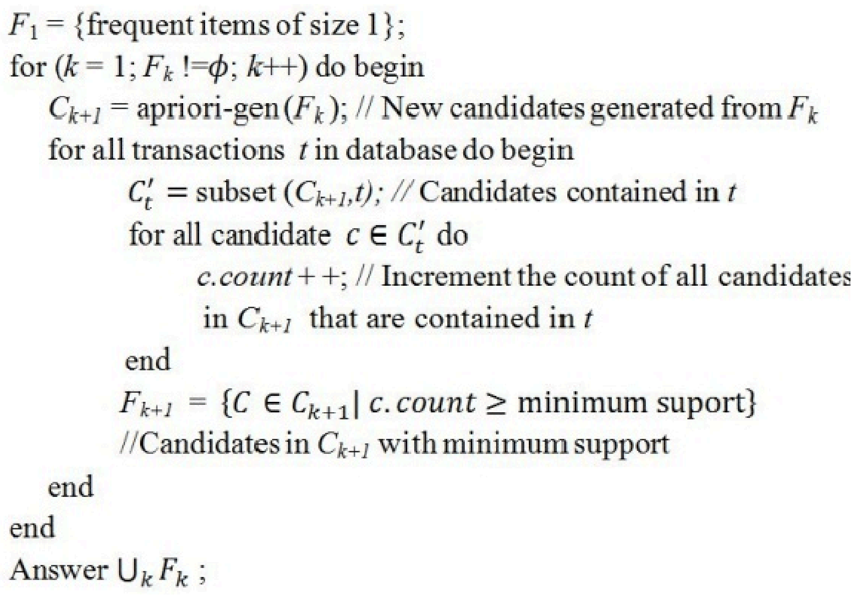

Frequent Itemsets using FP-Growth
Organic Raspberries
Large Lemon
Bag of Organic Bananas
Organic Strawberries
Strawberries
Organic Fuji Apple
Banana
Organic Hass Avocado
Organic Baby Spinach
Organic Avocado

Eclat :


Eclat frequent itemsets
Organic Blueberries
Organic Strawberries
Large Lemon
Organic Baby Spinach
Limes
Organic Large Extra Fancy Fuji
ganic Hass Avocado
Apple Honeycrisp Org
Bag of Organic Bananas
Organic Zucchini
Organic Avocado
Organic Yellow Onion
Organic Raspberries
Organic Whole Milk

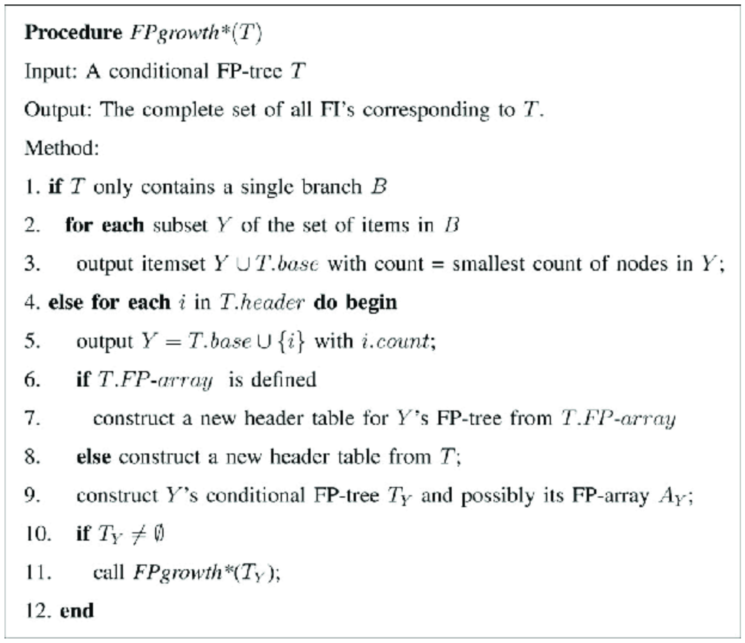
Details of the algorithms:

A-priori Algorithm



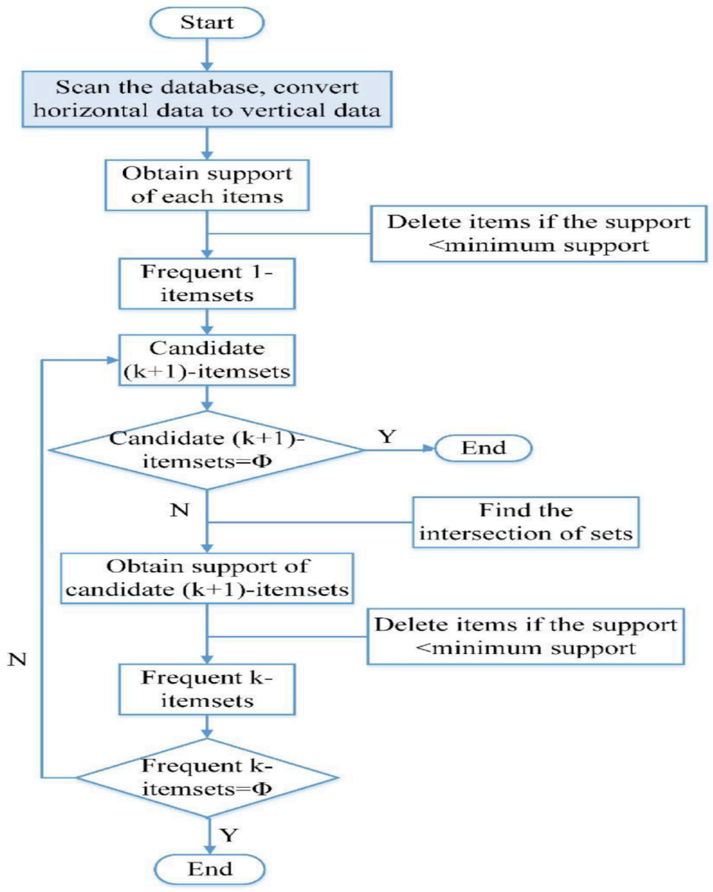
```
$F_1$ = {frequent items of size 1};
for ($k$ = 1; $F_k$ !=$\phi$; $k$++) do begin
    $C_{k+1}$ = apriori-gen($F_k$); // New candidates generated from $F_k$
    for all transactions  $t$ in database do begin
            $C'_t$ = subset ($C_{k+1}$,$t$); // Candidates contained in $t$
            for all candidate  $c \in C'_t$ do
                    $c.count++$; // Increment the count of all candidates
                    in $C_{k+1}$  that are contained in $t$
            end
            $F_{k+1}$ = $\{C \in C_{k+1} | \; c.count \geq$ minimum suport$\}$
            //Candidates in $C_{k+1}$ with minimum support
    end
end
Answer $\cup_k F_k$ ;
```

FP-Growth Algorithm (Frequent Pattern Growth)



```
**Procedure** *FPgrowth**($T$)
Input: A conditional FP-tree $T$
Output: The complete set of all FI's corresponding to $T$.
Method:
1. **if** $T$ only contains a single branch $B$
2.   **for each** subset $Y$ of the set of items in $B$
3.     output itemset $Y \cup T.base$ with count = smallest count of nodes in $Y$;
4. **else for each** $i$ in $T.header$ **do begin**
5.     output $Y = T.base \cup \{i\}$ with $i.count$;
6.     **if** $T.FP\text{-}array$ is defined
7.       construct a new header table for $Y$'s FP-tree from $T.FP\text{-}array$
8.     **else** construct a new header table from $T$;
9.     construct $Y$'s conditional FP-tree $T_Y$ and possibly its FP-array $A_Y$;
10.    **if** $T_Y \neq \emptyset$
11.      call *FPgrowth**($T_Y$);
12. **end**
```

ECLAT Algorithm (Equivalence Class Clustering and Bottom-Up Lattice Traversal):


Start
Scan the database, convert horizontal data to vertical data
Obtain support of each items
Delete items if the support <minimum support
Frequent 1-itemsets
Candidate (k+1)-itemsets
Candidate (k+1)-itemsets=Φ
Y
End
N
Find the intersection of sets
Obtain support of candidate (k+1)-itemsets
Delete items if the support <minimum support
N
Frequent k-itemsets
Frequent k-itemsets=Φ
Y
End