Market Basket Analysis (MBA) in Retail

Ziafat Shehzad

Abstract

A data mining technique called Market Basket Analysis (MBA) is used to find buying patterns in any retail environment. MBA essentially searches for the product combinations that appear most frequently in transactions. MBA use the association rule to forecast the possibility that two products will be bought together. Association mining is used to identify patterns and identify regulations. By looking at how often other things happen in a transaction, this mining method can find rules that predict how often a particular item will happen again. These connections can be used to make more profit through cross-selling, recommendations, promotions, or even the way products are placed on a menu or in a store. The Apriori and FP Growth algorithms are well-known ways to process data sets that include information about transactions, like in market basket analysis. In this study, the layout and planning of the availability of products are determined using the market basket analysis with FP-Growth and the apriori algorithm. Both are used to find associations between items in a dataset, and then we can design association rules. One of the most important things is that any algorithm for finding frequently used items should use less time and memory. These algorithms differ from one another in some fundamental ways. The experimental findings demonstrate that the FP-Growth algorithm can assess customer purchase patterns more quickly and effectively than the apriori algorithm. So, it is concluded that the FP Growth algorithm takes less computation time to find an association between the items than the Apriori algorithm.

Contents

List of Figures

1 Introduction to the project

1.1 Background

From the general stores on every corner in the early 1900s to the rise of e-commerce, which has completely changed the retail business, retail has changed a lot over the years. This process of change has opened up new opportunities for businesses and consumers. Today's consumers have various options, regardless of the sector of commerce. When a customer needed to make a purchase in the past, he was limited to selecting a product from the store's catalog. But in this age of technology and globalization, the number of options has grown by a huge amount. Customers can now pick from a vast selection of goods. Geographic and seasonal restrictions are no longer a problem. Products that were once thought of as luxury items are now considered everyday objects. All of this is possible because businesses nowadays have countless opportunities. However, this abundance of options also brought us many brand-new rivals. Companies have had to come up with new ways to market themselves in order to get new customers and keep the ones they already have.

So, the technique which is used to identify associations between items is called Market Basket Analysis(MBA). Market basket analysis includes a wide range of analytic tools intended to identify relationships and linkages between particular objects, as well as customer behaviors and relationships between products. If a buyer purchases a particular group of goods, they are more (or less) likely to purchase a different group of goods. For instance, it is commonly known that, most of the time, when a consumer buys beer, they also buy chips. The client was interested in the behaviors that led to the purchases. In order to develop new tactics that enhanced the advantages of the business and the experience of the customers, the client was interested in analyzing which things were purchased together.

Many algorithms have been proposed to identify the association between items. In simple terms, finding an association between objects is called "mining association," and the mining association rule is a crucial measurement. For example, Apriori and FP Growth are two algorithms for market basket analysis that are used to find connections between items and find groups of items that are often bought together. A lot of studies have been done to find the associations between items. One author, for example, used the Apriori algorithm to figure out the relationship between two things by setting the minimum support at 0.05 and the minimum confidence at 0.5. (Arora and Arora, 2022)

1.2 Problem statement

This change process has given rise to new opportunities for businesses and consumers. However, the expansion of e-commerce puts tremendous pressure on retailers, particularly when it comes to powerful companies like Amazon and Walmart Inc. Multiple techniques are being analyzed to enhance sales in retail. Different companies keep track of what their customers buy so they can find the best combinations of products and make connections between cus-

tomers and their buying habits. They look at how customers buy things by seeing which things are bought together, which things are bought with other things, and which things are bought based on the season. This process produces valuable information: if customers buy a set of items, what is the probability they will buy another set of items? For instance, if customers purchase bundles of water bottles in the summer, what is the likelihood they will also buy bottles of juice?.Business administrators are looking for technology that should determine the strong linkages between the items, which ultimately aids them in placing co-occurring products close to one another.

There are different algorithms that are used to identify associations between items. In this study, the apriori and FP growth algorithms have been used. Apriori and FP Growth are two market basket analysis algorithms that are used to find connections between items and groups of items that are often bought together. A lot of studies have been done to see which algorithm is faster at finding associations between items. Computation time is a crucial factor when dealing with big datasets to find relationships between items.

1.3 Research Questions

1. How can market basket analysis be used to enhance sales in retail?

2. Which market basket analysis algorithm is more efficient for mining frequent item-sets?

1.4 Research Aim

The aim of this research is to check whether market basket analysis can help retailers boost their sales. Also, to check which market basket analysis is more efficient at finding associations between items.

1.5 Objectives

1. The primary objective is to find associations between different items in a dataset by using market basket analysis.

2. To check which market basket analysis algorithm takes less computation time to find the frequent items in a data-set.

1.6 Project Plan

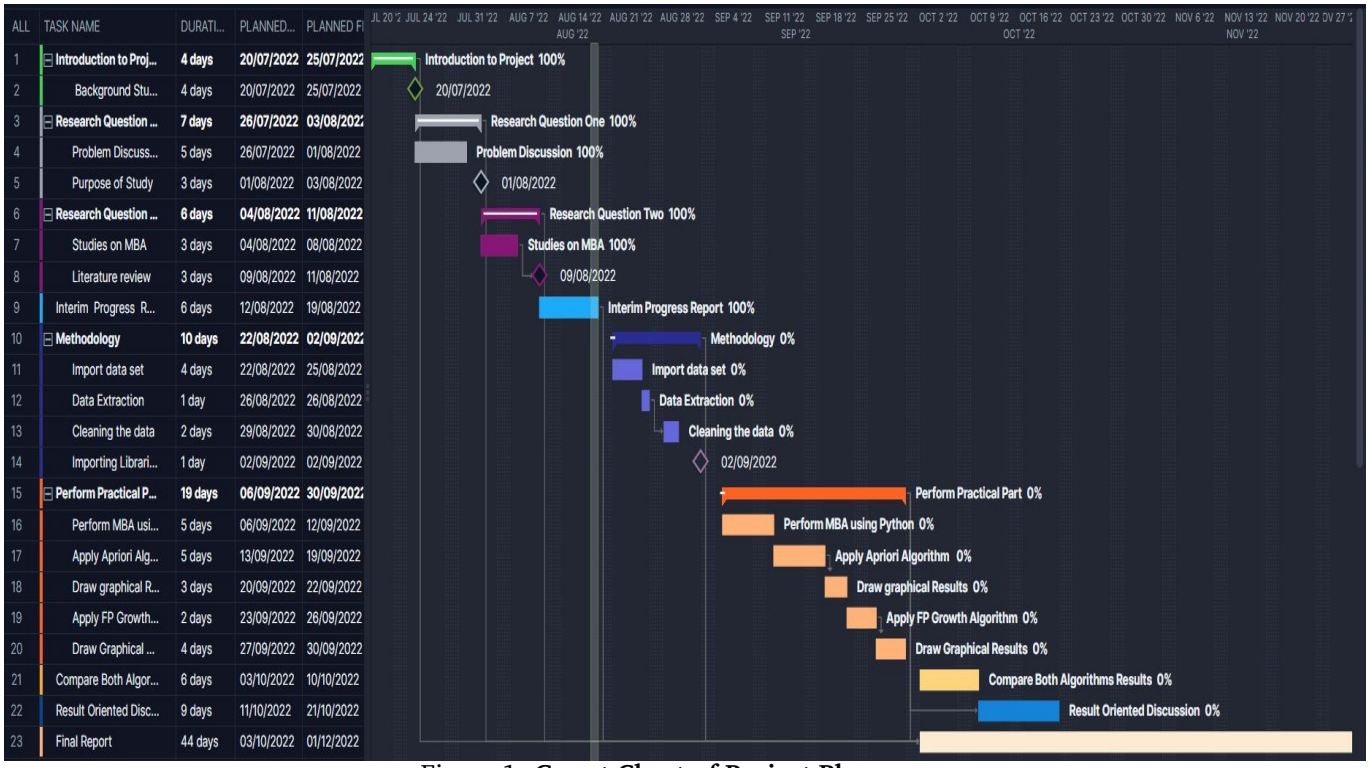


Figure 1: **Gannt Chart of Project Plan**

1.7 Chapter Overview

1.7.1 Chapter 1: Introduction

This chapter gives a quick overview of market basket analysis (MBA) and the role of MBA to boost the profit in retail. Also, this chapter will discuss the work that has already been done using MBA in retail. The chapter will show which MBA algorithms is more efficient regarding time computation.

1.7.2 Chapter II: Literature Review

In this section, the various MBA algorithms used for mining frequent item-sets are reviewed along with the existing studies on MBA.

1.7.3 Chapter III: Research Methodology

Based on the research gap found in the literature review, a proposed technique is to check which MBA algorithm is more efficient for finding frequent items in a dataset. This chapter expands on the full descriptions of all research methodologies, including the extraction and

cleaning of datasets, the exploration of data in Python, the application of data mining algorithms to datasets to find associations between items, and the analysis of the computation time of both algorithms.

1.7.4 Chapter IV: Results and Analysis

The results that were obtained are noted and explained in this section. Comparative analysis has been done to check the efficiency of both algorithms.

1.8 Discussion

Result oriented discussion

1.8.1 Chapter V: Conclusion and Future Work

This chapter concludes the discussion and provides a summary of the whole report. It evaluates if the fundamental objectives were met and provides a study scope for the future.

2 Literature Review

There are numerous market prospects offered by data mining techniques (MBA). For a firm to survive in this tough market, decision-making and comprehending customer behaviour have become crucial and difficult problems. In order to obtain a competitive advantage, the firm faces hurdles in extracting data from its extensive client data-sets.

The goal of data mining, according to Yanthy et al.(2016), to uncover hidden knowledge from data. Various methods have been developed for this purpose, but the issue is that not all generated rules are typically interesting—only a tiny portion of the rules would be of interest to any given user. As a result, many approaches, including confidence, support, and lift, have been suggested to identify the most effective or intriguing rules. Some algorithms, however, are good at producing rules that are high in one metric but poor in another **(Yanthy et al., 2009)**.

The Apriori method was proposed by Agrawal et al..(1994). Apriori was the first associative algorithm to be proposed, and it has since been utilised as a strategy in other associative, classification, and associative classification algorithms. Transactions are counted using the level-wise, breadth-first Apriori technique. The a priori method makes advantage of previously acquired knowledge of frequent item set features. A level-wise search is an iterative procedure used in Apriori to explore (n+1)-item sets using n-item sets. The Apriori property

is used in this case to improve the efficiency of level-wise creation of common item collections. The apriori property states that any non-empty subsets of a frequent item set must also be frequent. The support for an item set never rises above the support for its subsets because to the anti-monotone property of the support measure. A two-step procedure consists of join and prune operations that are carried out iteratively.**(Agrawal et al., 1994)**.

It is one of the algorithms used in data mining to find the most common items or groups of items in a given data repository. There are two steps in the algorithm.

1. Pruning 2. joining

Before using the algorithm, it is crucial to take the Apriori property into account. Apriori property specifies that Support

$$(XUY) = min(Support(X), Support(Y))$$

if an item X is connected with an item Y. In essence, support and confidence are used to quantify the strength of an association rule, or how strong the relationship is between the transaction of the objects. The quantity of transactions that contain an item constitutes the item's support. Items that fall short of the required level of support are not processed any further. How frequently a rule is applied to a certain data collection is determined by support. The conditional likelihood that a transaction containing the LHS will also contain the RHS is known as confidence **(Han et al.,2004**).

(LHS->RHS->P(RHS/LHS)=P(RHSLHS)/P(LHS)=Support(RHSLHS)/Support (LHS)

The frequency with which an item from RHS appears in a transaction containing LHS is determined by confidence. These two factors must be measured since they are crucial to our decision-making. It's possible for a rule with very little support to happen by accident. The dependability of the inference drawn by the rule is measured by confidence, on the other hand.

Han .et al.(2007) introduced a novel candidate rule-free association rule mining method dubbed FP-growth that produces a highly compressed frequent pattern tree (fptree) representation of the transactional database. The tree contains a maximum of one path for each database transaction. FP-tree is a smaller database than the original. Its development needs two database scans; the first scan produces frequent item sets and the support they receive in each transaction, and the second scan builds the FP-tree. Concatenating the patterns with

those generated by the conditional FP-tree is done during the mining phase. The memory requirement for the FP-growth approach, particularly in databases with enormous dimensions, is a limitation **Han .et al.2007**.

For market basket analysis, authors Abdulsalam et al.(2014) employed the Apriori algorithm. Six diverse products were used to symbolise the thirty (30) different transactions that made up the authors' attempt to depict the sales pattern of a supermarket. The Apriori technique in the JAVA programming language was used by the authors to try and identify the itemsets that are frequent under the assumption that the minimum support is 50% **(Abdulsalam et al.,2014)**.

Classification Dependent Predictive Association Rules (CPAR), Associative Classification, Classification Association Rule Mining (CARM), Distributed Apriori Association Rule, Six Sigma Technique, and Apriori algorithm were featured in Dhanabhakyam and Punithavalli, (1970) work. They listed the benefits and drawbacks of each approach and attempted to draw a judgement as to which approach is superior. Apriori algorithm is reportedly the best approach for association out of all others, however it has numerous challenges. The authors suggested using fuzzy logic and the Apriori method to address this, which will produce superior results **(Dhanabhakyam and Punithavalli, 1970)**.

The authors Liu et al. (2009) used FP Growth in their paper once more, which can address Apriori's drawbacks. The authors claim that FP Growth creates an FP tree with extremely compressed information. To determine the relationship between transactions, the authors created the FP tree using five transactions **(Liu et al., 2009)**.

The Customer Purchase Prediction Model (COREL) was developed by authors Liu et al. (2009) to simulate consumer purchasing behaviour in an online setting. Essentially, this paradigm has two stages. In the beginning, a candidate production collection is created by identifying relationships between products, and by doing so, it forecasts the motives of customers. Based on consumer preferences, the second step is utilised to identify the prospective products that are most frequently purchased. "Jingdong" provided the authors with client feedback and product information. Their research's findings demonstrated the importance of client preference in purchasing choices **(Liu et al., 2009)**.

Authors Kaur and Kang. (2016) suggested a technique using association rule mining to identify the morphing market data trends. They described several data mining techniques before attempting to demonstrate the importance of market basket analysis. Using extended bakery datasets, they looked for outliers. The authors suggested using this strategy in other situations **(Kaur and Kang, 2016)**.

By analysing the market basket using annual data from a "Do it yourself" (DIY) retailer, Van den Poel et al. (2004) were able to identify the related product pairs. The impact on these complimentary product pairs of various advertising campaigns were the results were examined, and the required suggestions were made **(Van den Poel et al., 2004)**.

According to Chen et al. (2015), the presumption that goods are always on shop shelves prevents major purchase patterns in chain-store operations from being retrieved. They also added warehouse and time information to the rules created differently in related experiments to get around this issue. In order to build marketing, product supply, inventory, and distribution strategies for the entire shop chain, they sought to employ the association rules **(Chen et al., 2005)**.

The usual association principles of mining, according to **Yun et al. (2006)**, did not yield beneficial results in multistore situations. To analyse the association rules in these repositories, an algorithm resembling the Apriori algorithm was created. The built algorithm's rules also included information about the shop's location and the time of day. With regard to simulation work carried out in stores of various sizes and with continuously changing product mixtures, it was observed that the proposed method was superior to the conventional Apriori algorithm **(Yun et al., 2006)**.

Many authors focused on contrasting various mining procedures in their publications. As an illustration, consider the recent paper by (Khan, Lee and Khattak, (2009). The ARM (association-rule mining) and clustering data mining methodologies. From the transactional data of a supermarket called Sales Day, useful information is derived (which is the same as the current paper). For mining association rules, the apriori algorithm is employed. If related products are found and shelves are rearranged to make them easier to find, sales are more likely to go up. However, clustering is frequently more preferable for use in classification. In the Khan article, K-means clustering is used to categorize clients and the many product kinds they may include in a basket based on their behavior and purchasing power. With the right use of the clustering strategy, retailers could use more ways to promote their products, which would lead to more sales. Clustering techniques are used to pull out data like information about a consumer's age, income, and traffic **(Khattak et al., 2009)**.

Bhargav et al. (2014), they used an artificial neural network, which minimises time complexity, to shorten the computation time. They employed a feed-forward network with just one layer. According to this theory, choosing combinations of various items is necessary for market basket analysis in order to determine which combinations are more frequently purchased. Customers' needs vary depending on their age group, the season, and the time of year, so their buying habits may alter frequently. if there is a sizable database. We must perform calculations for each combination for the same reason. There would be a lot of calculations if the database were really huge. The feed-forward network decreases the time complexity of repeated scanning due to its single layer. As a result, they increased effectiveness and cut down on programme execution time **(Bhargav et al., 2014)**.

Setiabudi et al.(2011) employed an artificial neural network that minimises temporal complexity to shorten the amount of time needed for calculations. Multidimensional data mining was employed. In this case, the dimensions are the things you bought and the time, the rule is information, and the choice is based on how those things are correlated. They employed the multidimensional association rule and the Apriori algorithm. They improved the effec-

tiveness of the apriori algorithm by including this strategy. Hybrid dimensional rule adds more properties, which increases time complexity **(Setiabudi et al., 2011)**.

FP growth and apriori algorithms have been used in a number of studies, but when computation time is taken into account, questions come up, such as which algorithm finds the association rule faster. In this study, it has been checked to see which algorithm is more efficient.

3 Methodology

3.1 Data Preprocessing

In order to stimulate the extraction of meaningful insights from the data, data preparation is a crucial step in the Market Basket Analysis (MBA) process. "Data preparation" in the context of an MBA refers to the process of preparing raw data for model creation and training by cleaning and arranging it. Simply said, data preparation in MBA is a data mining technique that transforms unprocessed data into a format that is readable and understandable.

When it comes to the deployment of MBA, the process begins with data preparation. Real-world data is typically inconsistent, erroneous (including errors or outliers), incomplete, and inconsistent. It also frequently lacks specific attribute values or trends. In this case, data preparation is important since it aids in organising, cleaning, and formatting raw data so that machine learning models can use it. Let's look at the different data pretreatment steps.

3.1.1 Data Collection

We talked about how supermarkets use data to better understand what customers want and, ultimately, to get them to spend more. Market Basket Analysis (MBA), one of the main strategies used by big retailers, looks for groups of products that are often bought together in transactions to find links between products.

I used the dataset from Kaggle https://www.kaggle.com/datasets/irfanasrullah/groceries.

The dataset was downloaded from Kaggle. It is a transactional data, and there are 9835 transactions in the dataset. which represents all the transactions for a UK-based online shop that took place between December 1, 2018, and February 2, 2019.

3.1.2 import all of the necessary libraries

I'll demonstrate how to import Python libraries for data preprocessing in MBA. The main Python packages I utilised for this data preprocessing in MBA are listed below .

Numpy : NumPy is the primary Python package used for performing mathematical operations. It is therefore utilised to incorporate any type of mathematical operation into the code.

Pandas : An excellent open-source Python toolbox for handling and analysing data is called Pandas. Dataset management and import use it regularly. It contains effective and user-friendly Python data analysis tools.

Matplotlib : The 2D charting tool Matplotlib can be used to create any type of chart in Python. It can provide you with figures that are suitable for publication in a range of hardcopy formats and in interactive settings that function across several platforms.

Seaborn: I instruct Python to load the Seaborn library into current environment using the import seaborn portion of the code. The as sns command in the code tells Python to give Seaborn the sns alias.

import apriori: Using the Apriori function, association rule mining can retrieve common itemsets.

Import FP Growth: To extract frequent item sets for association rule mining, use the FP Growth function.

```
import pandas as pd
#Pandas has been one of the most commonly used tools for Data Science and Machine learning
#which is used for data cleaning and analysis.
#Here, Pandas is the best tool for handling this real-world messy data.
#And pandas is one of the open-source python packages built on top of NumPy
import numpy as np
#Numpy is used to perform wide variety of mathematical operations on array.
#It adds powerful data structures to Python that guarantee efficient calculations with arrays and matrices
#It supplies an enormous library of high-level mathematical functions that operate on these arrays and matrices.
from mlxtend.frequent_patterns import apriori
#Apriori function to extract frequent itemsets for association rule mining
from mlxtend.frequent_patterns import association_rules
#Function to generate association rules from frequent itemsets
from mlxtend.frequent_patterns import fpgrowth
#FP Growth function to extract frequent itemsets for association rule mining
import time
#As the name suggests Python time module allows to work with time in Python.
#It allows functionality like getting the current time, pausing the Program from executing, etc.
#So before starting with this module we need to import it.
import matplotlib.pyplot as plt
#Matplotlib is a cross-platform, data visualization and graphical plotting library for Python and its numerical extension NumPy.
import seaborn as sns
#Seaborn is a library for making statistical graphics in Python.
#It builds on top of matplotlib and integrates closely with pandas data structures.
#Seaborn helps you explore and understand your data
```

Figure 2: **Python Libraries**

3.1.3 Import the dataset

One of the crucial phases of data preprocessing in MBA is importing the dataset. But the working directory must be set to the current directory before the dataset can be imported. After setting the working directory where the right dataset is, I use the "read csv()" function of the Pandas library to import it. This function can read a CSV file (either locally or from an URL) and do a number of things with it. The read csv() function is represented by:

$$df = pd.read\_csv('Dataset.csv')$$

```
df = pd.read_csv('Dataset.csv')
# The dataset that i have used is available on kaggle

df.info()

#Pandas dataframe.info() function is used to get a concise summary of the dataframe.
#It comes really handy when doing exploratory analysis of the data.
#To get a quick overview of the dataset we use the dataframe.info() function
```

Figure 3: **Importing Dataset**

3.1.4 Identifying and dealing with the missing values

Identification and proper handling of missing values are crucial steps in data preprocessing; if you don't, you risk drawing from the data incorrect conclusions and inferences.

There are essentially two methods for handling missing data:

Delete specific row: Using this strategy, I eliminated any features or columns where more than 75% of the values were missing by removing any rows that contained null values. It is recommended to only use this strategy, which doesn't always work, when the dataset has enough samples.

Calculate the mean: This approach is valueable for features that provide numerical data, such as number of students, number of employees, year, etc. It is possible to compute the mean, median, or mode of a certain feature, column, or row that has a missing value and utilise the outcome to fill in the blanks. This approach can increase the variety of the dataset and compensate for any data loss.

3.1.5 Fix structural errors

You have created a structural error if you measure or transmit data and discover odd naming practises, errors in grammar, or incorrect capitalization. These discrepancies could lead to incorrectly named classes or categories. When you encounter "N/A" and "Not Applicable," for

example, you should consider them to be part of the same group. Since my data set was not particularly large, Excel was used to solve this kind of problem. Python was used to sanitise the data set in the Jupyter notebook. I started looking through the data. I begin by drawing the data set's first five rows, then the last five.

	Item(s)	Item 1	Item 2	Item 3	Item 4	Item 5	Item 6	Item 7	Item 8	Item 9	...	Item 23	Item 24	Item 25	Item 26	Item 27	Item 28	Item 29	Item 30	Item 31	Item 32
0	4	citrus fruit	semi-finished bread	margarine	ready soups	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
1	3	tropical fruit	yogurt	coffee	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
2	1	whole milk	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
3	4	pip fruit	yogurt	cream cheese	meat spreads	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
4	4	other vegetables	whole milk	condensed milk	long life bakery product	NaN	NaN	NaN	NaN	NaN	...	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Figure 4: **Display First Five Rows of the datset**

After that, I showed the top 25 frequently bought items.

```
plt.rcParams['figure.figsize']=25,9
#matplotlib.rcParams contains some properties in matplotlibrc file
#We can use it to control the defaults of almost every property in Matplotlib
#figure size and DPI, line width, color and style, axes, axis and grid properties, text and font properties and so on.
#In order to use matplotlib.rcParams, we should know what properties are stored in it, these properties can be foud in matplotlil
sns.countplot(data=df, x=df['Item 1'],order = df['Item 1'].value_counts().head(25).index)
sns.set_theme(style="whitegrid")
plt.xticks(rotation=75)
plt.xlabel('Product')
plt.title('Top 25 frequently bought products')
plt.show()
```

Figure 5: **Calling most frequently bought items**

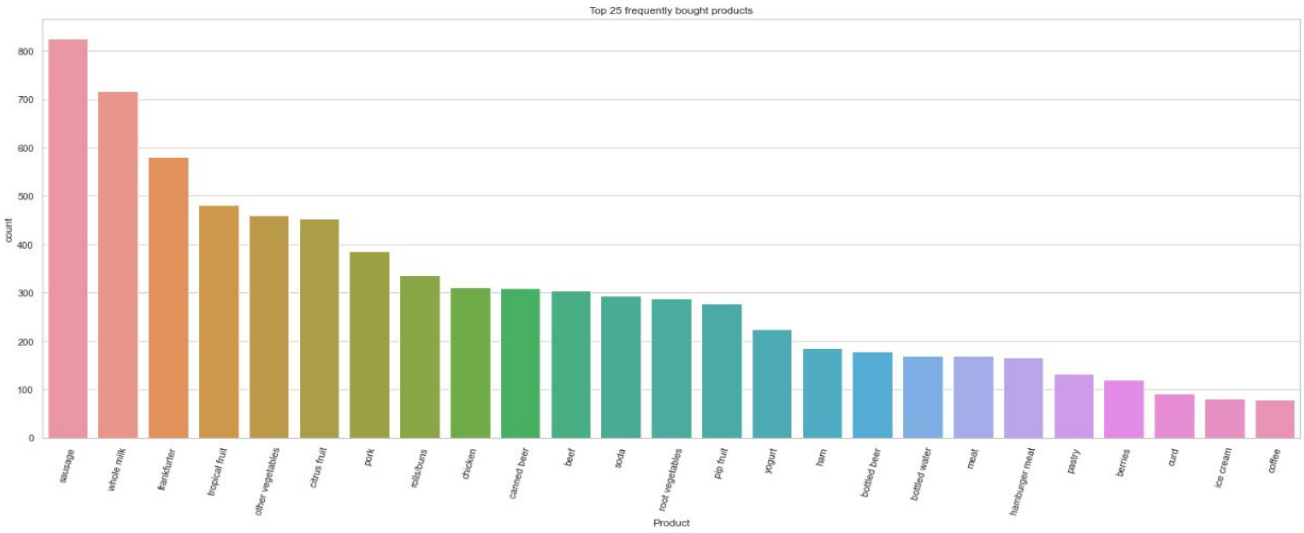


Figure 6: **Display twenty five bought items**

After preprocessing the data, I had a clear, clean, and useful dataset, so I began to use the MBA algorithm to design the association rule.

3.2 Implementation of MBA Algorithm

In this project, I am using two MBA algorithms, which are Apriori and FP Growth, both are used for mining frequent item sets.

Frequent Itemset : A group of items is referred to as "frequent" if it meets a predetermined threshold for confidence and support. Support shows transactions for items that were purchased in several transactions. Deals in which the items are purchased one at a time are displayed with confidence. For the frequent item mining approach, only transactions that satisfy the minimal threshold support and confidence requirements are taken into account. These mining algorithms' insights can help you save money and provide you an advantage over your rivals, among many other advantages. The frequent item sets discovered by Apriori and FP Growth have a wide range of applications in data mining tasks. Mining "association rules", locating sequences, and spotting intriguing patterns in databases are a few of the more significant ones. **association rules** are used to supermarket transaction data in order to analyse consumer behaviour in connection to the products they have purchased. The frequency with which the items are bought together is specified in the rules of association.

Association Rules: he definition of association rule mining is:

Let 'n' binary attributes collectively be referred to as 'items' in the statement 'let I= '. Let F=... be a database, which is a collection of transactions. Each transaction in F, each of which has its unique transaction ID, contains a fraction of the items in I. An implication of the form X->Y, where X, Y? I, and X? Y=?, constitutes the definition of a rule. The rule's antecedent and consequent are the groups of objects X and Y, respectively. It is possible to find associations between attributes in large databases by learning association rules. An association rule C=>D will say that "given a series of transactions, some value of itemset A causes the values of itemset D" if the minimal support and confidence requirements are satisfied.

For Instance:

Bread=> butter [support=1.9%, confidence-59%]

An illustration of an association rule is the phrase above. This indicates that 59% of customers purchased both butter and bread, with 1.9% of transactions combining the two purchases.

Support, Confidence, and Lift are the three fundamental metrics that are used in association rule learning, and we may make use of them.

Support: Support is simply the probability that an event will occur. The percentage of transactions that contain an item set is used to calculate it. To put it another way, Support(A)

is equal to the total number of transactions divided by the number of transactions that include A.

Formula:

$$\text{Support (C)} : \frac{\text{Number of Tranction in which A appear}}{\text{Total Number of Tranction}}$$

Confidence: Conditional probability can be used to express the confidence in an event given its antecedent. It is, to put it simply, the likelihood that event A will occur given that event B has already occurred. This can be used to explain the likelihood that an item will be bought when another one is already in the shopping cart. It is calculated by dividing the percentage of transactions that include both item X and Y by the percentage of transactions that only include Y.

Formula:

$$\text{Confidence (C} \rightarrow \text{D)} : \frac{\text{Support}(C \cup D)}{\text{Support(C)}}$$

Lift: The observed to expected ratio is known as lift. Lift accounts for the popularity of the two things in its measurement of the likelihood that one will be purchased when the other is. It can be determined by multiplying the likelihood that both of the events will occur separately by the likelihood that both events will occur independently as if there were no correlation between them.

3.2.1 The implementation of Apriori Algorithm

The first algorithm to be suggested for frequent itemset mining was the apriori algorithm. Later, it was enhanced by R Agarwal and R Srikant, and the result was known as Apriori. The "join" and "prune" steps of this algorithm are used to condense the search space. Finding the most common item sets is done iteratively.The Apriori Algorithm, an MBA algorithm, is used to understand the hierarchical relationships between the various things involved. The algorithm's most common practical use is to make purchase recommendations based on the items the customer already has in their shopping cart. Many businesses have been able to suggest products to their users by using the algorithm.

Before beginning to implement Apriori , it is important to comprehend the fundamental ideas behind it. The apriori approach is reliant on the item set's frequency distribution. It produces several tables with various item combinations. It searches the primary dataset, which displays all transactions, and determines frequencies by taking into account how frequently these combinations appear in the primary dataset. Support values are the term used for these frequencies in the Apriori method. The maximum item set length also affects how many tables are used. There will be 5 support value tables if the item-set length is allowed to be no more than 5. But a minimum support value is utilised to determine which item-sets

should remain in the database. Let's assume that the minimum support values were set at 2. Therefore, we should eliminate any item set from the table if it appears in the primary data set fewer than two times.

Let's look at the following illustration to see how Apriori functions.

The probability that item A is not frequent is if:

1. If P(A) is less than the minimal support criterion, A is not common.

2. If P (A+B) is less than the minimal support level and B is a member of the itemset, A+B are not frequent.

3. If the value of an itemset set is less than the minimum support, then all of its supersets will also be less than the minimum support and can be disregarded **(Says: et al., 2022)**.

Steps in Apriori:

1. In the first iteration of the algorithm, each item is considered a potential member of a 1-itemset. The algorithm will tally the frequency with which each item appears.

2. Permit a minimum amount of support, min sup (eg 2). It is determined which 1-item sets have occurrences that fulfil the min sup. The remaining candidates are narrowed down, and only those whose scores are greater than or equal to min sup go to the following stage.

3. Next, frequent two-itemset items with min-sup are discovered. In the join step, two items are joined with one another to form the 2-itemset for this purpose.

4. The minimal support threshold value is used to trim the 2-itemset candidates. Only two item sets with min-sup will remain in the table going forward.

5. The connect and prune process will be used in the subsequent iteration to produce three itemsets. The 3-item set subsets or the 2-item set subsets of each group in this iteration must adhere to the antimonotone property wherever they fall in min sup. If all 2-item subgroups are frequent, the superset will be frequent; otherwise, it will be reduced **(Says: et al., 2022)**.

6. A 3-itemset will be merged with itself and pruned if its subset does not meet the min sup requirement before becoming a 4-itemset. The procedure ends when the most popular itemset is obtained.

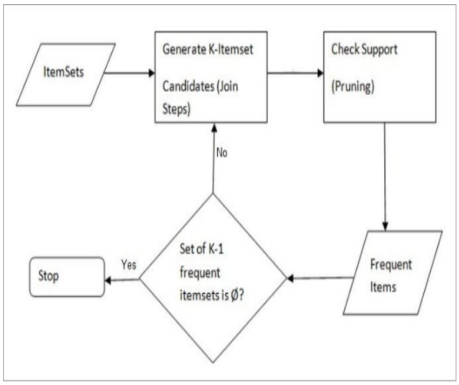


Figure 7: **Display twenty five bought items**

Implementing Apriori Algorithm in Python:

Before using the Apriori machine learning technique, various adjustments to the dataset are required. We can convert this dataset into a logical data frame using **Transactionencoder**. Each row represents a record or a transaction for a single purchase, and each column represents an item.

data = pd.get_dummies(df.drop(columns=['Item(s)'], axis=1), prefix="i")

pandas.get_dummies() is used for data manipulation. It converts categorical data into dummy or indicator variables.

Data = data.groupby(level=0, axis=1).sum()

The data must be transformed into a particular format in order to use this technique. The only thing this format is is a Boolean matrix with products as columns and transaction ids as rows. If a product is present in the transaction, the columns in that row are marked as 1, otherwise they are marked as 0. As a result, before the data is sent to the algorithm, it is first translated into this format.

	i_Instant food products	i_UHT-milk	i_abrasive cleaner	i_artif. sweetener	i_baby cosmetics	i_baby food	i_bags	i_baking powder	i_bathroom cleaner	i_beef	...	i_turkey	i_vinegar	i_waffles	i_whipped/sour cream	i_wh
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
9830	0	0	0	0	0	0	0	0	0	1	...	0	0	0	1	
9831	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	
9832	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	
9833	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	
9834	0	0	0	0	0	0	0	0	0	0	...	0	1	0	0	

9835 rows × 169 columns

Figure 8: **Boolean form dataset**

Now dataset is ready to use for the Apriori algorithm.

Set Support Value for **Apriori Algorithm**

freq_itemsets = apriori(data,min_support=0.05,use_colnames=True)

We have a minimum support value that has been set as 0.05. To get a more consistent result, we can set min support as 0.5. It means that if any item or item set occurs in a transaction less than 0.5 times, It won't take it into account. But at the moment, I set 0.05.

	support	itemsets
0	0.052466	(i_beef)
1	0.080529	(i_bottled beer)
2	0.110524	(i_bottled water)
3	0.064870	(i_brown bread)
4	0.055414	(i_butter)
5	0.077682	(i_canned beer)
6	0.082766	(i_citrus fruit)
7	0.058058	(i_coffee)
8	0.053279	(i_curd)
9	0.063447	(i_domestic eggs)
10	0.058973	(i_frankfurter)
11	0.072293	(i_fruit/vegetable juice)
12	0.058566	(i_margarine)
13	0.052364	(i_napkins)
14	0.079817	(i_newspapers)
15	0.193493	(i_other vegetables)
16	0.088968	(i_pastry)
17	0.075648	(i_pip fruit)
18	0.057651	(i_pork)
19	0.183935	(i_rolls/buns)
20	0.108998	(i_root vegetables)
21	0.093950	(i_sausage)

Figure 9: **Itemset According to different Support**

Adjust the support above 0.1

freq_itemsets[(freq_itemsets['support']>0.1)]

Frequent itemstes		
Numbers	Support	Itemsets
2	0.11067	Water Bottles
15	0.1167	Others Vegetable
19	0.110789	Rolls/buns
20	0.1001	Vegetables
23	0.11056	Tropical Fruits
26	0.110678	Whole milk
27	0.10560	Yogurt

Adjust Confidence

Set Confidence minimum 0.1

rules = association_rules(freq_itemsets,metric='confidence',min_threshold=0.1)
rules.sort_values(by='confidence',ascending=False)

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
4	(i_yogurt)	(i_whole milk)	0.139502	0.255516	0.056024	0.401603	1.571735	0.020379	1.244132
0	(i_other vegetables)	(i_whole milk)	0.193493	0.255516	0.074835	0.386758	1.513634	0.025394	1.214013
3	(i_rolls/buns)	(i_whole milk)	0.183935	0.255516	0.056634	0.307905	1.205032	0.009636	1.075696
1	(i_whole milk)	(i_other vegetables)	0.255516	0.193493	0.074835	0.292877	1.513634	0.025394	1.140548
2	(i_whole milk)	(i_rolls/buns)	0.255516	0.183935	0.056634	0.221647	1.205032	0.009636	1.048452
5	(i_whole milk)	(i_yogurt)	0.255516	0.139502	0.056024	0.219260	1.571735	0.020379	1.102157

Design a rule in which Confidence should be greater than 0.3 and lift should be greater than one

rules[(rules['confidence']>0.3) & (rules['lift']>1)]

3.2.2 The implementation of FP Growth Algorithm

A tree-based technique called the FP-growth algorithm is used for market basket analysis to mine common itemsets or frequent patterns. The FP-tree, a tree structure used by the method to describe the data, preserves the association data between the frequent items. The method uses the database to group similar things into an FP tree while keeping the rules of association. Then, it splits the data into several conditional databases, which are a type of projected database. Each conditional database is linked to a single frequent data item.

FP-tree: The FP tree is the fundamental idea behind the FP-growth algorithm. The database itemset is stored in memory as part of the FP-tree, which also keeps track of the relationships

between the items. Each item set is added as a subtree and used to build the tree. The fundamental tenet of the FP tree is that items will be more likely to be shared if they occur more frequently.

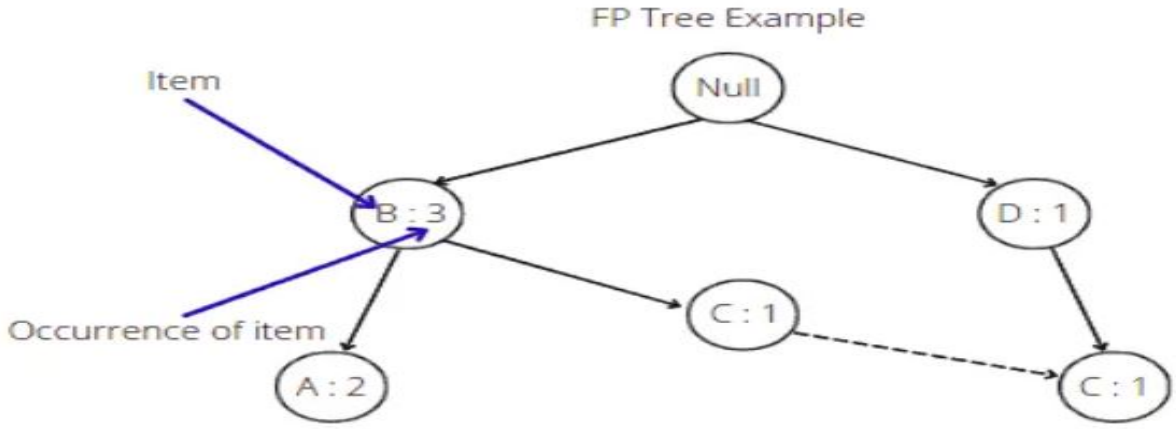


Figure 10: **FP Growth working diagram**

Adjust the support above 0.1

freq_itemsets[(freq_itemsets['support']>0.1)]

The FP-root tree's node is null. The item name and the support (or item occurrence) number are stored in each node of the subtree at the very least. A link to a node with the same name from another subtree may also be present in the node (representing alternative item-sets which are available in data-set).

Building FP-tree: The following steps are used by the FP-growth algorithm to construct the FP-tree from the database.

1. First-time scanning of itemsets from the database

2. Find frequent things (single item patterns) and arrange them in decreasing order of frequency in a list L. As an illustration, L = X:6, Y:4, Z;3, W:2

3. Order the frequent things in each transaction in accordance with L's hierarchy.

4. After a second database scan, create an FP-tree by stacking each frequency-ordered transaction on top of it.

Let's take a sample database and make an FP-tree out of it to demonstrate algorithm steps:

FP Growth Algorithm	
ID	Items Bought
100	f,a,c,d,g,i,m,p
200	a,b,c,f,l,m,o
300	b,f,h,j,o
400	b,c,k,s,p
500	a,f,c,e,l,p,m,n

First, the dataset needs to be scanned. Then, each item in the database needs to be turned into a frequency table and put in order of how often it appears. Additionally, we must define the minimal level of support (number of item occurrences). Items with a support value below the minimal support will not be included by the algorithm to the frequency table. As an illustration, let's set the minimum support value to be 3. We will then receive the frequency table shown below:

FP Growth Algorithm	
Items	Frequency
f	4
c	3
a	3
b	3
m	3
p	3

All items with frequency less than the minimum support value in the frequency table were not taken into account by the algorithm.

The next step is to perform a second database search and organize the pieces according to the frequency table:

Items with a higher frequency number are prioritized.

In the event that two things have the same frequency number, they will be listed alphabetically.

FP Growth Algorithm	
ID	Items Bought
100	f,a,c,d,g,i,m,p
200	a,b,c,f,l,m,o
300	b,f,h,j,o
400	b,c,k,s,p
500	a,f,c,e,l,p,m,n

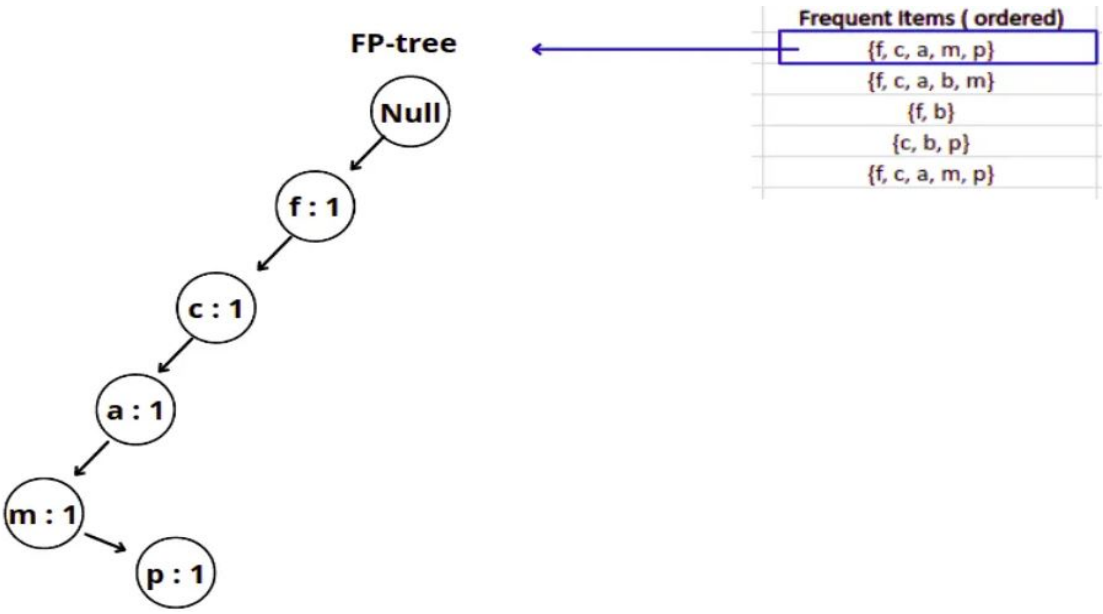
$\rightarrow$

Frequent items
f,c,a,m,p
f,c,a,b,m
f,b
c,b,p
f,c,a,m,p

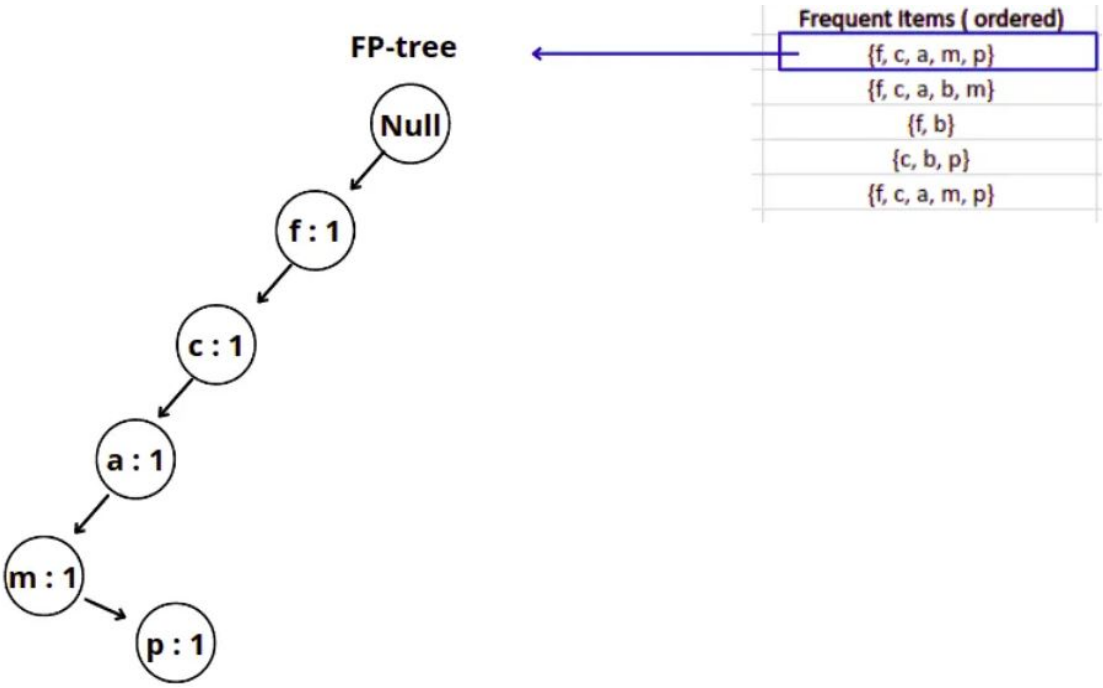
We are now ready to construct the FP tree.

Starting with a null node, we will add nodes to the tree depending on the frequent items table.

For the first frequent item from the list, for instance, this is how the FP-growth will build the FP-tree:



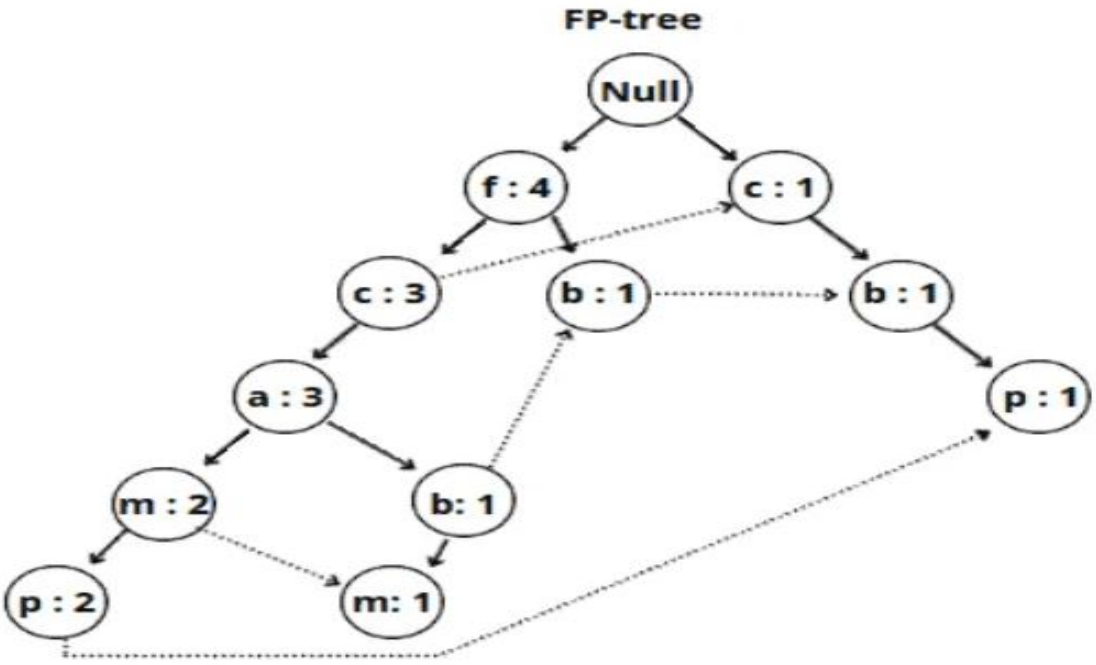
Similar to that, the following item from the list will be added to the FP-tree in the following stage, as seen below:



As we keep adding the same element to the tree, the support will grow. However, since item b was missing from our initial tree after item a, we had to add a new node for it. Due to the fact that this is the same element that can be found in several subtrees, we have linked items together.

In addition, I include elements from the FP-tree's table of frequently occurring objects.

FP tree is finished here:



3.2.3 Implementation of FP-growth algorithm using Python

Now I will use Python to implement the FP-growth algorithm. To install the necessary libraries, enter the following instructions in the cell of the Jupyter Notebook:

First I imported the library pandas as pd

I imported another library import numpy as np

The most important python library from mlxtend.frequent_patterns import association_rules.

Then I Imported from mlxtend FP growth library

Importing and Exploring dataset:

df = pd.read_csv('Dataset.csv')

The dataset that i have used is available on kaggle **(Dedhia, 2020)**.

The data must be transformed into a particular format in order to use this technique. The only thing this format is is a boolean matrix with products as columns and transaction ids as rows. If a product is present in the transaction, the columns in that row are marked as 1,

otherwise they are marked as 0. As a result, before the data is sent to the algorithm, it is first translated into this format.

data = pd.get_dummies(df.drop(columns=['Item(s)'], axis=1), prefix="i")

To convert catagorical data into dummy or indicator variable I used the command pandas.get dummies(). Which is also called data manipulation.

data = data.groupby(level=0, axis=1).sum()

	i_Instant food products	i_UHT-milk	i_abrasive cleaner	i_artif. sweetener	i_baby cosmetics	i_baby food	i_bags	i_baking powder	i_bathroom cleaner	i_beef	...	i_turkey	i_vinegar	i_waffles	i_whipped/sour cream	i_wh
0	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	
3	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
9830	0	0	0	0	0	0	0	0	0	1	...	0	0	0	1	
9831	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	
9832	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	
9833	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	
9834	0	0	0	0	0	0	0	0	0	0	...	0	1	0	0	

9835 rows × 169 columns

Now I have dataset in binary form so I can implement FP Growth algorithm

Set the Support and apply FP algorith

res=fpgrowth(data,min_support=0.05, use_colnames=True)

	support	itemsets
0	0.082766	(i_citrus fruit)
1	0.058566	(i_margarine)
2	0.139502	(i_yogurt)
3	0.104931	(i_tropical fruit)
4	0.058058	(i_coffee)
5	0.255516	(i_whole milk)
6	0.075648	(i_pip fruit)
7	0.193493	(i_other vegetables)
8	0.055414	(i_butter)
9	0.183935	(i_rolls/buns)
10	0.080529	(i_bottled beer)
11	0.110524	(i_bottled water)
12	0.053279	(i_curd)
13	0.052466	(i_beef)
14	0.174377	(i_soda)
15	0.058973	(i_frankfurter)
16	0.079817	(i_newspapers)
17	0.072293	(i_fruit/vegetable juice)
18	0.088968	(i_pastry)
19	0.108998	(i_root vegetables)
20	0.077682	(i_canned beer)
21	0.093950	(i_sausage)
22	0.098526	(i_shopping bags)
23	0.064870	(i_brown bread)
24	0.052364	(i_napkins)

Print first five rows

res. head ()

Adjust Support	
Support	Frequent Itemset
0.82766	Fruits
0.52766	Margarine
0.132766	Yougurt
0.102766	Tropical
0.058058	Coffee

Print last five rows

res. head ()

Last Five Rows	
Support	Frequent Itemset
0.058058	Pork
0.068058	eggs
0.05789	Whole milk
0.078058	Vegetables and milk
0.058058	Milk and buns

Adjust Support

res = fpgrowth(data,min_support=0.05,use_colnames=True)

FP Growth	
Support	Frequent Itemset
0.13952	Yougurt
0.10952	Fruits
0.23952	Whole milk
0.19952	Vegetables and milk
0.18952	Milk and buns
0.11952	Water Bottles
0.17952	Soda
0.10952	Vegetables

Adjust Confidence

res = association_rules(res,metric='confidence',min_threshold=0.1)

res.sort_values(by='confidence',ascending=False)

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(i_yogurt)	(i_whole milk)	0.139502	0.255516	0.056024	0.401603	1.571735	0.020379	1.244132
2	(i_other vegetables)	(i_whole milk)	0.193493	0.255516	0.074835	0.386758	1.513634	0.025394	1.214013
4	(i_rolls/buns)	(i_whole milk)	0.183935	0.255516	0.056634	0.307905	1.205032	0.009636	1.075696
3	(i_whole milk)	(i_other vegetables)	0.255516	0.193493	0.074835	0.292877	1.513634	0.025394	1.140548
5	(i_whole milk)	(i_rolls/buns)	0.255516	0.183935	0.056634	0.221647	1.205032	0.009636	1.048452
1	(i_whole milk)	(i_yogurt)	0.255516	0.139502	0.056024	0.219260	1.571735	0.020379	1.102157

Adjust lift and confidence

res[(res['confidence']>0.2) & (rules['lift']>1)]

3.3 Comparative analysis of MBA algorithm (Apriori vs Fp Growth) with respect to excution time

The experiments performed for this study are described in depth in this section. The Market Basket Analysis methods lack precise and well stated evaluation metrics. Therefore, it is crucial to conduct numerous tests, study the association rules that are produced, and determine **how long it takes to create these relationships**.To check which algorithm took more time to creat association rule.Here I will check which MBA algorithm is more efficient.

I used the same dataset that i used for association rule designing. The FP Growth and Apriori were given the pre-processed and converted data as input. The threshold parameters were specified to have minimum support values of 0.5 and minimum confidence values of 0.7 The experiment is run multiple times with different numbers of input transactions, and each time, **the execution time is recorded.**

I used my own laptop. With Following Specification

1. Device name DESKTOp- Ziafat Shehzad
2. Processor Intel(R) Core(TM) i7-8550U CPU @ 1.80GHz 1.99 GHz
3. Installed RAM 12.0 GB (11.9 GB usable)
4. System type 64-bit operating system, x64-based processor
5. Name Intel(R) UHD Graphics 620
6. Adapter Type NVIDIA GeForce 940MX, NVIDIA compatible
7. Adapter Description NVIDIA GeForce 940MX

4 Results and Analysis

Results Using Apriori Algorithm

So after adjusting the values of three metrics (Support, Confidence and Lift). Now we can design associations according to antecedents and consequent. According to the results, if Someone buys yoghurt, the most probability is he/she will also purchase whole milk.

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(i_other vegetables)	(i_whole milk)	0.193493	0.255516	0.074835	0.386758	1.513634	0.025394	1.214013
3	(i_rolls/buns)	(i_whole milk)	0.183935	0.255516	0.056634	0.307905	1.205032	0.009636	1.075696
4	(i_yogurt)	(i_whole milk)	0.139502	0.255516	0.056024	0.401603	1.571735	0.020379	1.244132

FP Growth algorithm results.

consequently, following modifying the values of three measures (Support, Confidence and Lift). We can now create associations based on antecedents and consequences. The findings indicate that there is a high likelihood that someone who buys yoghurt will also buy whole milk.

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(i_yogurt)	(i_whole milk)	0.139502	0.255516	0.056024	0.401603	1.571735	0.020379	1.244132
1	(i_whole milk)	(i_yogurt)	0.255516	0.139502	0.056024	0.219260	1.571735	0.020379	1.102157
2	(i_whole milk)	(i_other vegetables)	0.255516	0.193493	0.074835	0.292877	1.513634	0.025394	1.140548
3	(i_other vegetables)	(i_whole milk)	0.193493	0.255516	0.074835	0.386758	1.513634	0.025394	1.214013
4	(i_whole milk)	(i_rolls/buns)	0.255516	0.183935	0.056634	0.221647	1.205032	0.009636	1.048452
5	(i_rolls/buns)	(i_whole milk)	0.183935	0.255516	0.056634	0.307905	1.205032	0.009636	1.075696

Both algorithm have quite similar results at the same support, confidence and lift values.

According to the above results of both algorithm, **I can answer my first research question (RQ 1)** easily: Based on the above results, we know that if a customer buys i_rolls/buns, there is more probability that the customer will purchase i_whole milk.

Using the above results I can design association rule which will help retailers to enhance sales in retail.

Using the association rule, the retailer can **Create product bundles**

Online retail stores frequently offer bundled deals, which pair two or more related products together, occasionally at a price but frequently purely for convenience. If you are purchasing

yoghurt? The association rule data may reveal your associations with whole milk, butter, and cream. Sales can be increased by creating a bundle that combines all three of these into one item and enables customers to buy it with just one click.

campaigns and promotions: By using association rule retailers can do promotions and run campaign. This analysis aids in comprehending complementary items and those that serve as pillars of the product line.

Customer behaviour analysis: By analyzing association rule retailers can judge their customer behaviours, which will help them to optimize their catalog according to customer preferences. Market basket analysis enables you to comprehend how customers behave so that UX/UI and catalogue design can be more effective. So, this whole process helps the retailers to enhance sales in retail.

In-store operations optimisations:

MBA finds excellent application in optimising the inventory by determining the popularity of the products in a store.So, after these results it is clear that which items are more popular.

Comparative results

In association rule mining and data mining, I have compared and contrasted two popular pattern mining algorithms called Apriori and FP Growth. To understand these algorithms' edge features, I compared them using the same database transactions. Apriori Algorithm Execution time is higher then FP Growth.

```
apriori_matrix, apriori_exec_time = perform_rule_calculation(data, rule_type="apriori")

print("Apriori Execution took:  seconds".format(apriori_exec_time))
```

Computed Apriori!

```
Apriori Execution took: 0.13272048950195312 seconds
```

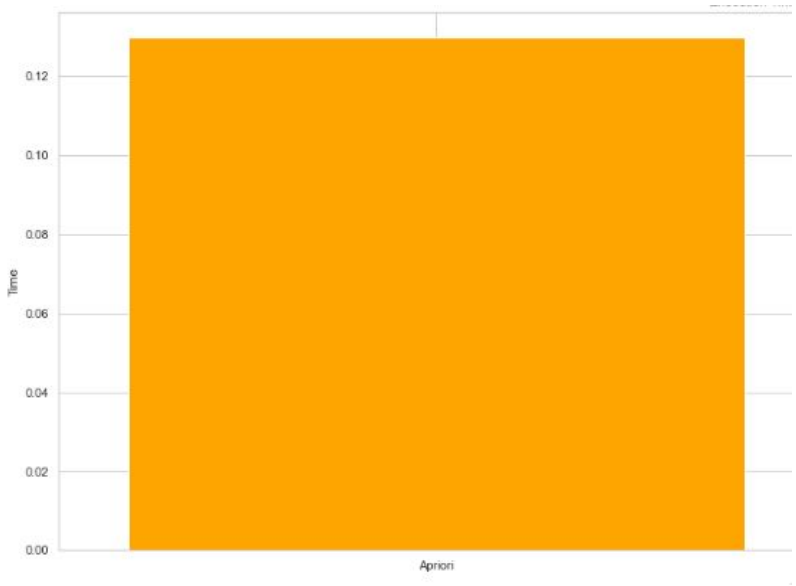


Figure 11: **Apriori Algorithm Execution Time**

```
fpgrowth_matrix, fp_growth_exec_time = perform_rule_calculation(data)
print("Fp Growth execution took:  seconds".format(fp_growth_exec_time)
```

FP Growth Execution took: 0.03790163993835449 seconds

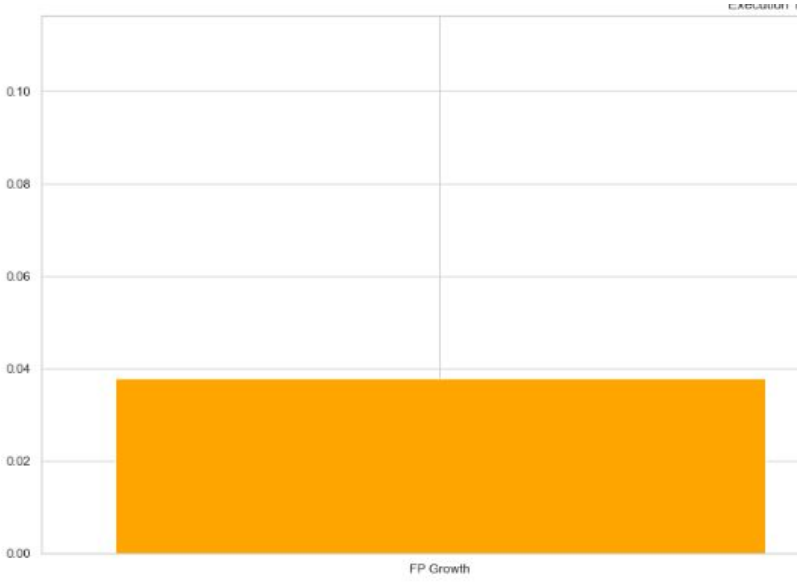


Figure 12: **Apriori Algorithm Execution Time**

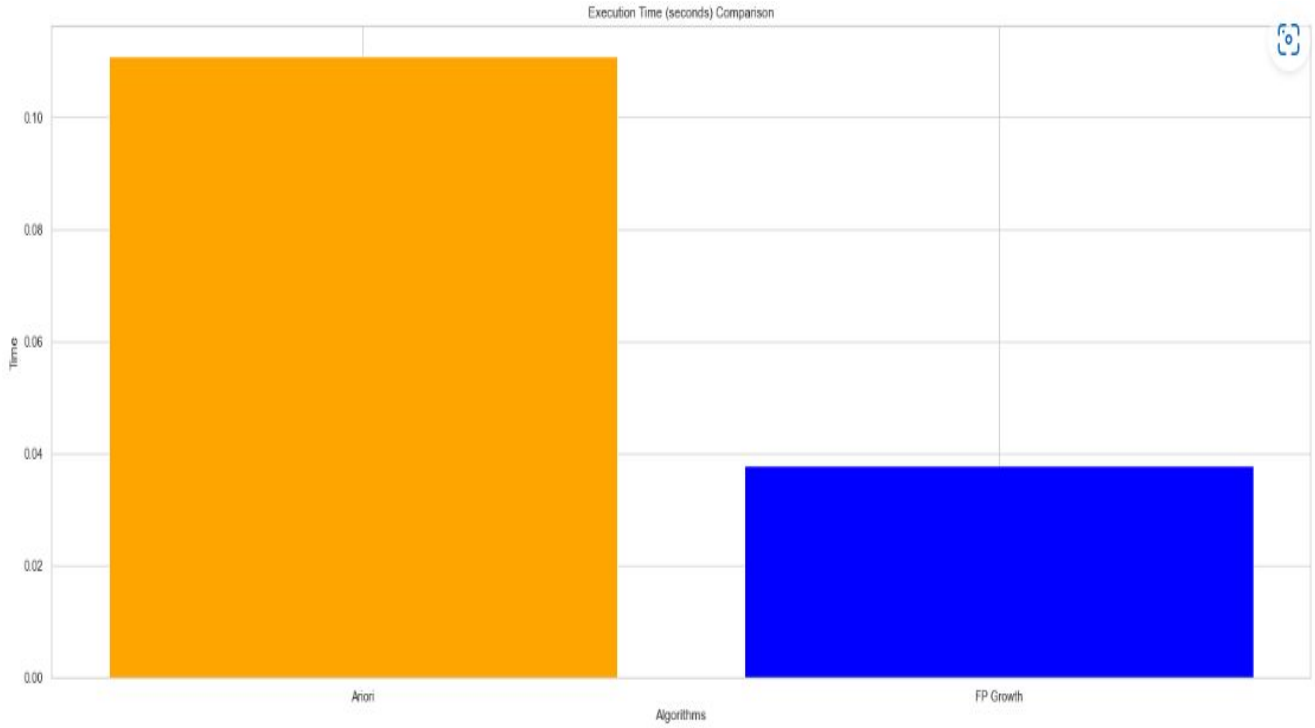


Figure 13: **Apriori Compution time VS FP Growth Compution time**

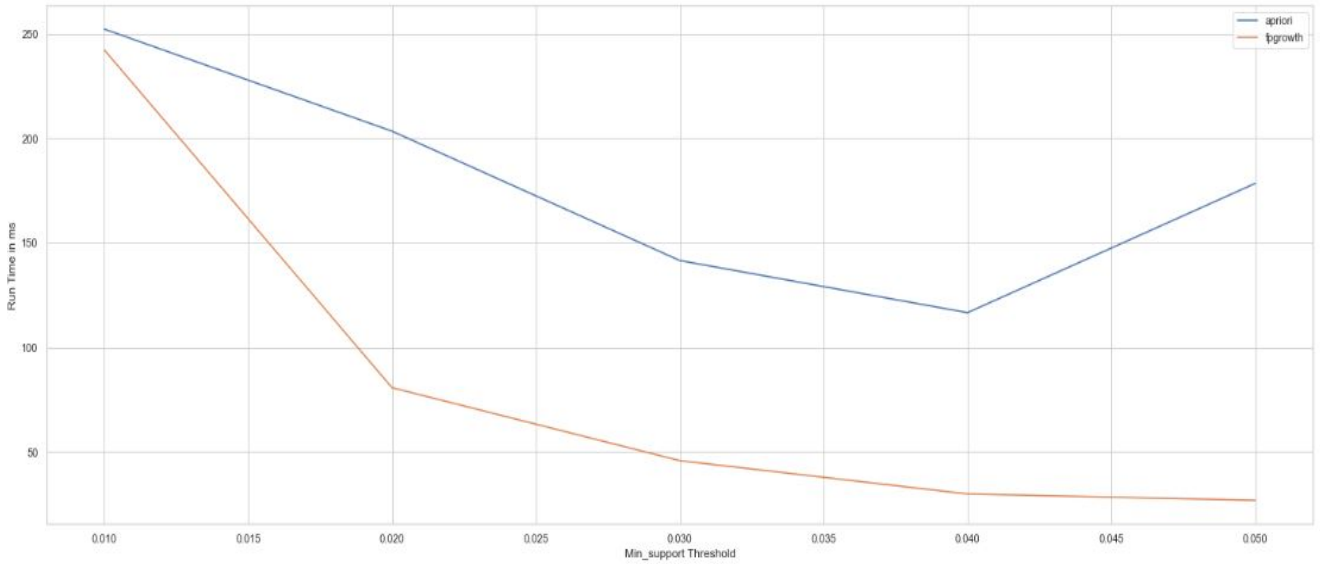


Figure 14: **Apriori Compution time VS FP Growth Compution time**

According to the aforementioned results, apriori takes the longest to compute compared to the other two algorithms. This is because each time the a priori algorithm iterates, it must scan the database, increasing the computation time. In comparison to apriori algorithms, FP-Growth performs better in terms of computing time.

So, I can answer my second reserach question that Fp Growth algorithm is more efficient in contrast to apriori algorithm.

5 Discussion

We'll talk about market basket analysis in this project and how it may be used to figure out which of your consumers' favourite goods is. We'll also offer advice on how to add these items to your inventory in order to boost sales. Read on for advice that can help your retail operation reach new heights, whether you're just starting out or have been in the industry for a while. Market basket analysis can be an effective approach in the retail industry for determining which products are frequently bought together. Customers can identify the products they need more easily by using this information to develop tailored marketing. Retailers can improve their financial situation and increase their profitability by growing sales.

Consequently, we can help you employ market basket analysis to grow your retail business's sales.

We must comprehend what our customers are purchasing in order to provide them with the greatest possible service. We need to understand what drives them, what obstacles they confront, and what qualities they seek in a good or service. This knowledge enables us to create goods and services that better satisfy their demands and to offer the most satisfying user experience. We can establish lasting relationships that are advantageous to both our business and our clients by really getting to know them.

Examining clients' intentions is one method to analyse what they are purchasing. What want or need are they attempting to fulfil? Customers might purchase a product, for instance, because it is reasonably priced or because it is the best alternative. They might also purchase a product because they find it useful or fashionable. You can decide what characteristics to highlight in your marketing and how to position your product in the market by understanding the driving forces behind consumer behaviour. Consideration of customers' values is another technique to consider what they are purchasing. What do they hold dear, and how does it affect the things they buy?

For instance, some clients may prioritise social responsibility while others may value environmental sustainability. Understanding customer values can help you develop a brand that more deeply connects with them. Lastly, it's crucial to comprehend the environment in which clients are making their purchases. What elements play a role in their decision-making?

For instance, if they are purchasing a gift for someone else, they can be swayed by the preferences or interests of the recipient. When purchasing anything for themselves, consumers might be motivated by time or financial restraints. You can offer a more individualised and pertinent shopping experience by comprehending the context of your customers' buying selections.

Market basket analysis can be used to determine which products are most popular with your target market once we have a solid understanding of what customers are purchasing. Then, it will be simpler to build targeted marketing using this information.

We must examine data from a variety of sources, including our point-of-sale systems, online sales data, consumer surveys, and more, to gain a thorough picture of what they are purchasing. This information enables us to make well-informed choices about everything from the merchandise we offer in our stores to the sites of new store openings. Most importantly, it assists us in making sure that we are meeting the demands of our clients and giving them the finest shopping experience possible.

Based on the analysis's findings utilising the apriori and FP-Growth algorithms, it is clear that when creating association rules for grocery store datasets, I used both of these techniques to create frequent itemsets. Based on the findings of each algorithm, I was able to determine the combination pattern of buying foodstuffs with a confidence value of 0.05% for each algorithm. I discovered the combination pattern of buying yoghurt and whole milks. The a priori approach has the longest calculation time when compared to the FP Growth algorithms, according to the average performance evaluation for each algorithm. This is because the a priori algorithm must scan the database for each iteration, increasing the computation time.

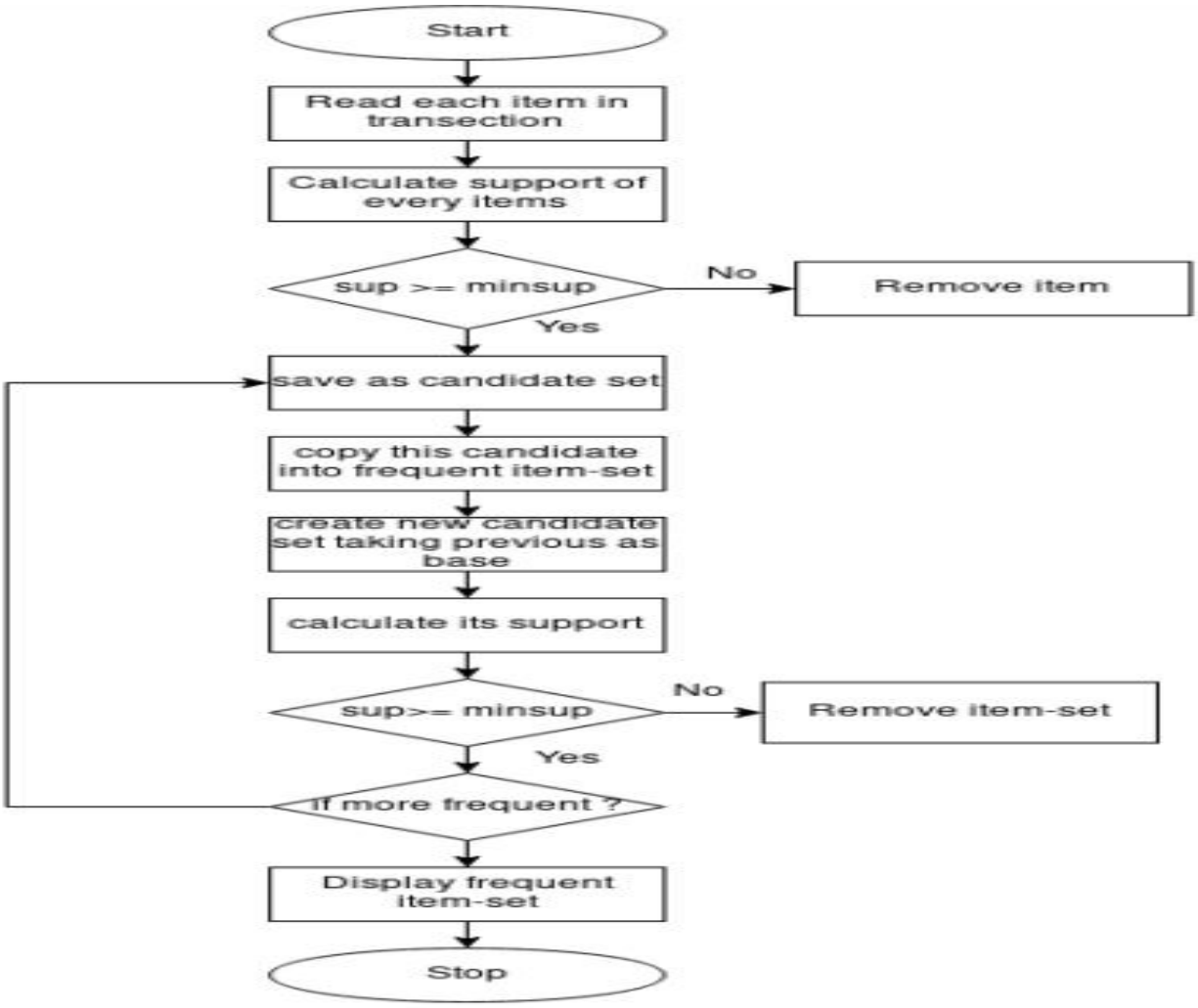


Figure 15: **Apriori Algorithm Flow Diagarm**

While FP-Growth outperforms apriori algorithms in terms of computational time, apriori algorithms often consume more memory on average. Although it is lighter than FP-Growth, the Apriori method requires roughly the same amount of RAM as other market basket algorithms. FP Growth, however, moves more quickly than apriori. Because FP-tree only needs to scan the database once in the first stages, it takes less time.

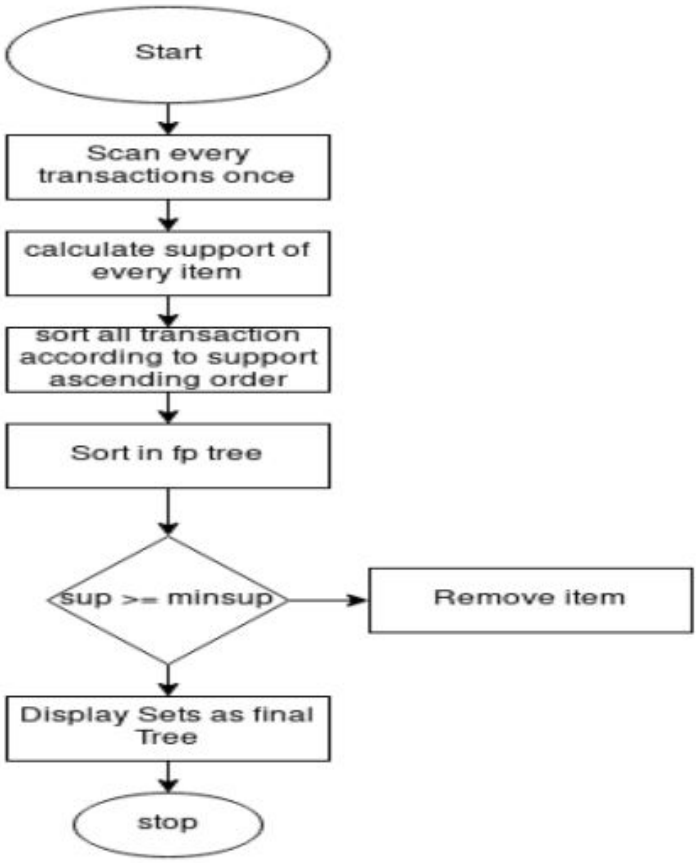


Figure 16: **FP Growth Algorithm Flow Diagarm**

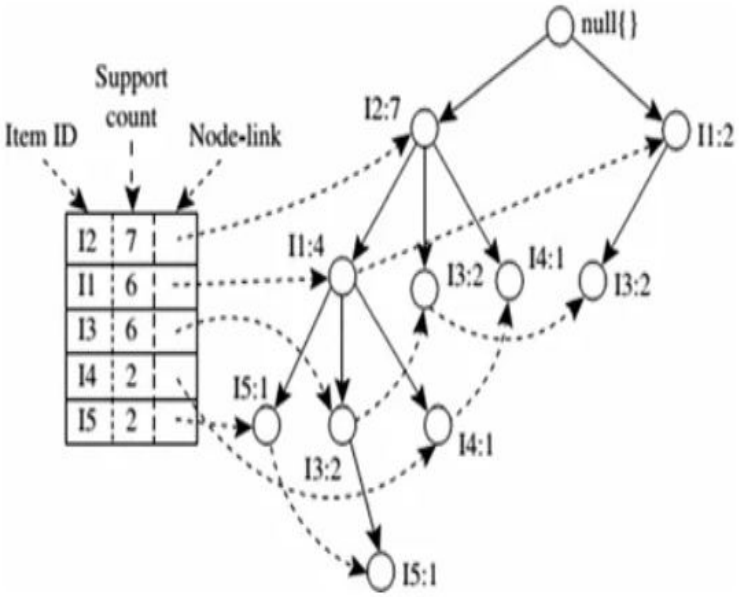


Figure 17: **FP Growth Tree Diagarm**

6 Conclusion

In this Master Thesis report, a Market Basket Analysis project in Retail was described. Results of the study showed that the MBA is most use-full tool that can be used to enhance sales in retail.According to above results in methdology, FP growth is a superior method to Apriori. The time needed to search for a mining dataset is less in FP Growth than it is in Apriori algorithms. This is because item sets are created for the given item depending on the number of transactions, and it all depends on the minimum support count value, which is a key factor in determining the time needed for data mining. So, I have met my all objectives that i defined at the start of the project. Through the project, creating frequent item patterns, association rules and measurements of interest have been seen. The retail industry's unspoken motto of cross-selling and upselling is what encourages customers to make larger purchases. For such companies, using market basket analysis in data mining to identify patterns and extract customer insights to improve the performance of their brands, has become a thriving component. According to an urban tale, a grocery store saw an increase in sales after grouping beer and diapers together since a market basket analysis revealed that both products were frequently purchased by men. Different Companies are making significantly good profit by strategically using this tactic to influence customer mental processes. It is a practical strategy to increase sales without spending more time or money on marketing that won't yield the same amazing results.

7 Future Recommendation

This study's primary objective is be to compare the performance of the FP Growth and Apriori algorithms. In order to detect links between frequent item sets that satisfy a given minimum confidence, we first identify all frequent item sets that satisfy a predefined minimum support for both techniques. The findings that are listed in the results section are then marked out after comparing the execution time to the transaction. When the database is vast, Apriori creates a lot of candidate sets, making it impossible for the FP Growth algorithm to build a main memory-based FPtree. Therefore, it will be beneficial if we can lower our computation in the future. Reducing the dataset items with the best-selling products is the upcoming method that has been suggested. So, using the goods that customers purchased the most of, we reshaped the datasets. A crucial consideration is, however, how many best-selling products will be appropriate for the suggested strategy. We can do this by computing all of the products in the databases, then taking the 25%, 45%, 55%, and 55% top-selling products and comparing the findings against frequent itemsets and association rules.

8 Refrences

[1] Agrawal, R. and Srikant, R., 1994, September. Fast algorithms for mining association rules. In Proc. 20th int. conf. very large data bases, VLDB (Vol. 1215, pp. 487-499).

[2] Abdulsalam, S.O., Adewole, K.S., Akintola, A.G. and Hambali, M.A., 2014. Data mining in market basket transaction: An association rule mining approach. International Journal of Applied Information Systems (IJAIS), 7(10), pp.15-20.

[4] Bhargav, A., Mathur, R.P. and Bhargav, M., 2014, April. Market basket analysis using artificial neural network. In International Conference for Convergence for Technology-2014 (pp. 1-6). IEEE.

[5] Anggraeni, S., Iha, M.A., Erawati, W. and Khairunnas, S., 2019. The Analysis of Sales by Using Apriori and FP-Growth at PT. Panca Putra Solusindo. REMIK: Riset dan E-Jurnal Manajemen Informatika Komputer, 3(2), pp.41-46.

[6] Chen, Y.L., Tang, K., Shen, R.J. and Hu, Y.H., 2005. Market basket analysis in a multiple store environment. Decision support systems, 40(2), pp.339-354.

[7] Cavique, L., 2007. A scalable algorithm for the market basket analysis. Journal of Retailing and Consumer Services, 14(6), pp.400-407.

[8] Dhanabhakyam, M. and Punithavalli, M., 2011. A survey on data mining algorithm for market basket analysis. Global Journal of Computer Science and Technology.

[9] Dedhia, H. (2020) Groceries dataset, Kaggle. Available at: https://www.kaggle.com/datasets/heeral dataset (Accessed: December 2, 2022)

[10] Gunadi, G. and Sensuse, D.I., 2016. Penerapan metode data mining market basket analysis terhadap data penjualan produk buku dengan menggunakan algoritma apriori dan frequent pattern growth (fp-growth): studi kasus percetakan pt. Gramedia. Telematika MKOM, 4(1), pp.118-132.

[11] Han, J., Cheng, H., Xin, D. and Yan, X., 2007. Frequent pattern mining: current status and future directions. Data mining and knowledge discovery, 15(1), pp.55-86.

[12] Han, J., Pei, J., Yin, Y. and Mao, R., 2004. Mining frequent patterns without candidate generation: A frequent-pattern tree approach. Data mining and knowledge discovery, 8(1), pp.53-87.

[13] Liu, X., Guan, J. and Hu, P., 2009. Mining frequent closed itemsets from a landmark window over online data streams. Computers & Mathematics with Applications, 57(6), pp.927-936.

[14] Liu, Y. and Guan, Y., 2008, November. Fp-growth algorithm for application in research of market basket analysis. In 2008 IEEE International Conference on Computational Cybernetics (pp. 269-272). IEEE.

[15] Kaur, M. and Kang, S., 2016. Market Basket Analysis: Identify the changing trends of market data using association rule mining. Procedia computer science, 85, pp.78-85.

[16] Khattak, A.M., Khan, A.M., Rasheed, T., Lee, S. and Lee, Y.K., 2009, December. Comparative analysis of xlminer and weka for association rule mining and clustering. In International Conference on Database Theory and Application (pp. 82-89). Springer, Berlin, Heidelberg.

[17] Nengsih, W., 2015, May. A comparative study on market basket analysis and apriori association technique. In 2015 3rd international conference on information and communication technology (ICoICT) (pp. 461-464). IEEE.

[18] Qiu, J., Lin, Z. and Li, Y., 2015. Predicting customer purchase behavior in the e-commerce context. Electronic commerce research, 15(4), pp.427-452.

[19] Rong, Z., Xia, D. and Zhang, Z., 2013, May. Complex statistical analysis of big data: implementation and application of apriori and FP-growth algorithm based on MapReduce. In 2013 IEEE 4th International Conference on Software Engineering and Service Science (pp. 968-972). IEEE.

[20] Singh, A.K., Kumar, A. and Maurya, A.K., 2014, May. An empirical analysis and comparison of apriori and FP-growth algorithm for frequent pattern mining. In 2014 IEEE International Conference on Advanced Communications, Control and Computing Technologies (pp. 1599-1602). IEEE.

[21] Sagin, A.N. and Ayvaz, B., 2018. Determination of association rules with market basket analysis: application in the retail sector. Southeast Europe Journal of Soft Computing, 7(1).

[22] Surjandari, I. and Seruni, A., 2010. Design of product placement layout in retail shop using market basket analysis. Makara Journal of Technology, 9(2), pp.43-47.

[23] Setiabudi, D.H., Budhi, G.S., Purnama, I.W.J. and Noertjahyana, A., 2011, August. Data mining market basket analysis' using hybrid-dimension association rules, case study in Minimarket X. In 2011 International Conference on Uncertainty Reasoning and Knowledge Engineering (Vol. 1, pp. 196-199). IEEE.

[24] Says:, S.Y., says:, S.S.A. and Says:, P. (2022) Data science apriori algorithm in Python - Market Basket Analysis, Intellipaat Blog. Available at: https://intellipaat.com/blog/data-science-apriori-algorithm/ (Accessed: December 2, 2022).

[25] Van den Poel, D., De Schamphelaere, J. and Wets, G., 2004. Direct and indirect effects of retail promotions on sales and profits in the do-it-yourself market. Expert Systems with Applications, 27(1), pp.53-62.

[26] Yun, C.H., Chuang, K.T. and Chen, M.S., 2006. Adherence clustering: an efficient method for mining market-basket clusters. Information systems, 31(3), pp.170-186

[27] Yanthy, W., Sekiya, T. and Yamaguchi, K., 2009, December. Mining interesting rules by association and classification algorithms. In 2009 Fourth International Conference on Frontier of Computer Science and Technology (pp. 177-182). IEEE.

.

A Appendix: Sample source code

```
1
2
3 import pandas as pd
4
5 import numpy as np
6
7 from mlxtnd.frequent_patterns import apriori
8
9 from mlxtnd.frequent_patterns import association rules
10
11 from mlxtnd.frequent_patterns import FP Growth
12
13 import matplotlib.pyplot as plt
14
15 import seaborn as sns
16
17
18 df = pd.read_csv('Dataset.csv')
19 # The dataset that i have used is available on kaggle
20
21 df.info()
22
23 df.head()
24
25 df.tail()
26
27 plt.rcParams['figure.figsize']=25,9
28
29 sns.countplot(data=df, x=df['Item 1'],order = df['Item1'].value_counts
      ().head(25).index)
30
31 sns.set_theme(style="whitegrid")
32 plt.xticks(rotation=75)
33 plt.xlabel('Product')
34 plt.title('Top 25 frequently bought products')
35 plt.show()
36
37 data = pd.get_dummies(df.drop(columns=['Item(s)'], axis=1), prefix="i")
38
39
40
41 data = data.groupby(level=0, axis=1).sum()
42
```

```
43
44 frquently_bought_itemsets = apriori_(data,min_support=0.04,
     use_colnames=True)
45
46 frquently_bought_itemsets
47
48 frquently_bought_itemsets[(frquently_bought_itemsets['support']>0.1)]
49 rules = design_Association_Rules(frquently_bought_itemsets,parameter='
     confidence',minimum_threshold=0.1)
50 Rules.sort_values(by='confidence',ascending=False)
51
52 rules =  design_Association_Rules ( frquently_bought_itemsets,metric='
     lift', minimum_threshold=1)
53
54 Rule.sort_values(by='confidence',)
55 # ascending order shpould be false
56
57 Rule[(rules['confidence']>0.4) & (rules['lift']>1.1)]
58
59 RES=fpgrowth(frquently_bought_itemsets,minimum_support=0.05,
     use_colnames=True)
60 RES
61 RES.head()
62 RES.tail()
63 RES = fpgrowth(frquently_bought_itemsets,min_support=0.05,use_colnames=
     True)
64 RES
65 RES = design_Association_Rules(RES,metric='confidence',min_threshold
     =0.1)
66 RES.sort_values(by='confidence',ascending=False)
67
68
69 RES=design_Association_Rules(RES, parameter = "lift", minimum_threshold
     =1)
70
71
72 RES.sort_values("confidence",ascending=False)
73
74 RES[(RES['confidence']>0.2) & (rules['lift']>1.1)]
75
76 def perform_rule_calculation(transact_items_matrix, rule_type="fpgrowth
     ",
77 min_support=0.05):
78
79     start_time = 0
80
```

```
    total_execution = 0

    if(not rule_type=="fpgrowth"):

        start_time = time.time()

        rule_items = apriori(transact_items_matrix,  min_support=
   min_support,
 use_colnames=True)

total_execution = time.time() - start_time

  print("The compution time of Apriori Algorithm!")

    else:

        Starting_time_of_Apriori_Algorithm = time.time()

        rule_items = fpgrowth(transact_items_matrix,     min_support=
   min_support,
     use_colnames=True)

        total_time_took_to_Execute = time.time() -------Starting_time


        print("The compution time of FP Growth")

    rule_items['number_of_items'] = rule_items['itemsets'].apply(lambda
    x: len(x))

    return rule_items, total_execution

    def compute_association_rule(rule_matrix, metric="lift", min_thresh
   =1):


    @desc: Compute the final association rule
    @params:
        - rule_matrix: the corresponding algorithms matrix

        - metric: the metric to be used (default is lift)

        - min_thresh: the minimum threshold (default is 1)


    @returns:
```

```
123
124         - rules: all the information for each transaction satisfying
    the given metric & threshold
125
126     rules = association_rules(rule_matrix,   metric=metric,
127     min_threshold=min_thresh)
128
129     return rules
130
131     # Plot Lift Vs Coverage(confidence)
132 def plot_metrics_relationship(rule_matrix, col1, col2):
133     """
134     desc: shows the relationship between the two input columns
135     @params:
136         - rule_matrix: the matrix containing the result of a rule (
    apriori or Fp Growth)
137
138
139
140
141     ........
142     .......
143
144
145
146     execution_times = [Algo1[1], Algo2[1]]
147     algo_names = (algo1[0], algo2[0])
148
149     y=np.arange(len(algo_names))
150
151     plt.bar(y,execution_times,color=['orange', 'blue'])
152     plt.xside(x,algo_names)
153
154     plt.xlabel('Algorithms')
155
156     plt.ylabel('time')
157
158     plt.title("The comparision of execution time in seconds")
159     plt.show()
160
161     fpgrowth_matrix, fp_growth_exec_time = perform_rule_calculation(
    data)
162
163 print("The execution time of FP growth algorithm: {} SECONDS".format(
    fp_growth_exec_time))
164
```

```
fp_growth_rule_lift = compute_association_rule(fpgrowth_matrix)
plot_metrics_relationship(fp_growth_rule_lift, col1='lift', col2='
    confidence')

apriori_matrix, apriori_exec_time = perform_rule_calculation(data,
    rule_type="apriori")

print("Apriori Execution took: {} seconds".format(apriori_exec_time))

algo1 = ['Ariori', fp_growth_exec_time]
algo2 = ['FP Growth', apriori_exec_time]

compare_time_exec(algo1, algo2)
import time
l=[0.01,0.02,0.03,0.04,0.05]
t=[]
for i in l:
    t1=time.time()
    apriori(data,min_support=i,use_colnames=True)
    t2=time.time()
    t.append((t2-t1)*1000)

    l=[0.01,0.02,0.03,0.04,0.05]
f=[]
for i in l:
    t1=time.time()
    fpgrowth(data,min_support=i,use_colnames=True)
    t2=time.time()
    f.append((t2-t1)*1000)

    sns.lineplot(x=l,y=f,label="apriori")
sns.lineplot(x=l,y=t,label="fpgrowth")
plt.xlabel("Min_support Threshold")
plt.ylabel("Run Time in ms")
sns.lineplot(x=l,y=f,label="fpgrowth")
sns.lineplot(x=l,y=t,label="apriori")
plt.xlabel("Min_support Threshold")
plt.ylabel("Run Time in ms")
```

B Appendix: Sample figure

This appendix demonstrates how to include a figure with a caption, label and reference. See `https://www.overleaf.com/learn/latex/Inserting_Images#Captioning.2C_labelling_and_referencing`. The caption should explain what is presented in the figure. As always, link the figure into the main body of text as follows: Figure **??** shows a modern poster of Oscar Wilde's "Ìmportance of Being Earnest".