République Algérienne Démocratique et Populaire
الجمهوريــة الجزائريــة الديمقراطيــة الشـعبية
Ministère de l'Enseignement Supérieur et de la Recherche Scientifique
وزارة التعليـم العـالي و البحــث العلمــي

المدرسة العليا للإعلام الآلي - سيدي بلعباس

Ecole Supérieure en Informatique-Sidi Bel Abbes

Machine Learning Report

Option : Artificial Intelligence & Data Science(IASD)

Theme: Market basket analysis (MBA)

Realised by:

Atek Abdeljalil

Djebarra Rabeh Abderrezak

Meharzi Slimane

Messioud Mohammed Amir

Supervised by:

Dr. Dif Nassima

Promotion : 2023/2024

TABLE OF CONTENTS :

1. MARKET BASKET ANALYSIS

Market basket is defined as an itemset bought together by a customer on a single visit to a store. On our visit to the supermarket we tend to buy a lot of products from different categories and put them all together in one single basket. Which is considered to be a single transaction. Market basket analysis is the analysis of those baskets all together. Market basket analysis encompasses a broad set of analytic techniques aimed at uncovering the associations and connections between specific objects, discovering customer behaviors and relation between items. In retail, it is used based on the following idea: if a customer buys a certain group of items, is more (or less) likely to buy another group of items. For example, it is known that when a customer buys cheese, in most cases, buys chips as well. These behaviors produced in purchases is something that the companies selling their products are interested in. The seller's/supermarkets are interested in analyzing which items are purchased together in order to create new marketing/sales strategies that can be helpful in improving the benefits of the company as well as customer experiences.
The market basket analysis is a powerful tool for the implementation of up-selling, cross-selling, inventory management strategies (Chen, Tang, Shen, & Hu, 2005).
Market Basket Analysis is also known as association rule mining or affinity analysis, which have been used to understand consumer behavior regarding the types of the purchases they make. It is a Data Mining technique that originated in the field of marketing and was initially used to understand purchase patterns of the customers by extracting associations and co-occurrence from a transactional database (i.e. market basket data). For example, when shopping in a supermarket, consumers rarely buy one product, they are far more likely to purchase an entire basket of products, mostly from different product categories. This allows us to uncover nonobvious, usually hidden and counterintuitive associations between items, products, or categories. We are also able to extract products and product categories which are purchased together, and these associations can be represented in the form of association rules. These association rules enable managers to develop marketing strategies like developing interventions, promoting specific product categories, offering promotions, etc.which eventually leads customers to spend more money based on two different principles:

- Upselling, which consists in buying a large quantity of the same product or adding new features.
- Cross-selling, which consists in adding more products from various categories.

Market Basket Analysis is also very much useful in stock management and placement of items.

2.STRATEGY FOR MARKET BASKET ANALYSIS

In this section we describe the entire research process. Before getting into the steps of the analysis.

First, we clear some of the concepts that we will be coming across in our analysis.

2.1. KEY TERMS AND CONCEPTS

2.1.1. Association rules

Association analysis is also known as affinity analysis or association rule mining (ARM), a method commonly used for market basket analysis. ARM is currently the most suitable method for analysis of big market basket data but when there is a large volume of sales transactions with a high number of products, the data matrix to be used for association rule mining usually ends up large and sparse,resulting in longer time to process data.

Association rules provide information of this type in the form of **"IF-THEN"** statements.

There are three indexes which are commonly used to understand the presence, nature and strength of an association rule. (Berry & Linoff, 2004; Larose, 2005; Zhang & Zhang, 2002) Lift is obtained first because it provides information on whether an association exists or not or if the association is positive or negative. If the value for lift suggests that there is an existence of association rule, then we obtain the value for support.

$$\textit{Lift} = P(A \cap B) \,/\, P(A) * P(B)$$

If lift is greater than 1, it suggests that the presence of the items on the LHS has increased the probability that the items on the RHS will occur on this transaction. If the lift is below 1, it suggests that the presence of the items on the LHS make the probability that the items on the RHS will be part of the transaction lower. If the lift is 1, it suggests that the presence of items on the LHS and RHS are independent: knowing that the items on the LHS are present makes no difference to the probability that items will occur on the RHS.

Support of an item or itemset is the fraction of transactions in our dataset that contain that item or itemset. It is an important measure because a rule that has low support may occur simply by chance. A low support rule may also be uninteresting from a business perspective because it may not be profitable to promote items that are seldom bought together. For these reasons, support is often used to eliminate uninteresting rules.

$$\textbf{\textit{Support}} = \boldsymbol{P(A \cap B)/N}$$

Confidence is defined as the conditional probability that shows that the transaction containing the Left Hand Side (LHS) will also contain the Right Hand Side (RHS).

$$\textbf{\textit{Confidence}} = \boldsymbol{P(A \cap B) \,/\, P(A)}$$

Confidence and support measure the strength of an association rule. Since the transactional database is quite large, there is a higher risk of getting too many unimportant rules which may not be of our interest. To avoid these kinds of errors we commonly define a threshold of support and confidence prior to the analysis, so that only useful and interesting rules are generated in our result.

2.1.2. Antecedent and Consequent

In every association rule we have an antecedent and a consequent, also called rule body and rule head accordingly. The generated association rule relates the rule body with the rule head. LHS is the Antecedent and RHS is the Consequent.

2.1.3. Frequent Itemset:

In many (but not all) situations, we only care about association rules or casualties involving sets of items that appear frequently in baskets. For example, we cannot run a good marketing strategy involving items that no one buys anyway. Thus, much data mining starts with the assumption that we only care about sets of items with high support; i.e., they appear together in many baskets. We then find association rules or causalities only involving a high-support set of items. The consequent must appear in at least a certain percent of the baskets, called the support threshold.

2.1.4. Transaction

In the field of marketing, a typical transaction consists of a set of products purchased by a customer at a retail store or on a website. These transactions contain all the information about each specific transaction which make up the data entered into the database. These can include information on the customer, information of what products were purchased in what quantity, information on time of purchases, information on if the companies marketing strategies are attracting customers or not, etc.

Also, a transaction can take place at one point in time or over time and could involve a day, a quarter, a fiscal year, or even longer periods. Because they are not limited to an event.

2.1.5. Long Tail Effect

This term often refers to data products purchased in supermarkets describing their distribution as a long tail in which a small number of products is purchased more frequently whereas a large number of products is purchased less frequently. This phenomenon creates a data sparsity problem and worsens even more their elaboration.
Studying such data with MBA would be practically helpful because its transactional data and best approach to work with transactional data is to do market basket analysis.

2.2.1.ADVANTAGES OF USING MARKET BASKET ANALYSIS

• It allows researchers to make use of data from stores which is in abundance to build their theory. This capability of MBA was first highlighted by Locke in 2007. He said using MBA has the potential to lead to important contributions by allowing researchers to implement an inductive approach to theory building, which, despite its advantages, is currently underutilized. Indeed, it has led to insights in marketing and other fields. For example, Russell et al. (1999) pointed out multiple-category decision making i.e. theoretical models of purchasing decisions involving products in more than one category. Apart from marketing, other researches like patients with food allergies allowed Kanagawa et al. (2009) to build models regarding which allergens are related to which.

• MBA allows the researchers to use the data which appears to be messy and unusable. Given the affordability and availability of data storage systems, the organizations collect data every day on employees (their performance, training, skills, etc.), customers (frequency of visits, purchases, expenditure, etc.) and many other issues. Most of that data is collected in a very unsystematic and unorganized format without any specific study in mind. MBA is ideally suited to be used inductively with such datasets to uncover association rules that may not be readily apparent (Hafley & Lewis, 1963; Shmueli, Patel, & Bruce, 2010). Messy data often involves dirty data with lots of missing values and outliers. MBA is not immune to the problem of messy data, it just allows the interpretation of missing data as indicating that no option was selected, and association rules are less influenced by outliers compared to other traditional data analytic approaches. In the context of MBA, outliers result in infrequently occurring associations (He, Xu, Huang, & Deng, 2004)

• MBA can help in building dynamic theories, which states how important is the role of time in theory building. There are mainly two approaches of building dynamic theories via MBA i.e. multiple MBA and sequential MBA. When the available data include transactions as they have occurred overtime, multiple MBA approaches are used (Tang et al. 2008). Sequential MBA is used when the available data describes individual events as they have occurred over time. It may uncover the presence of a pattern in which event A occurs before event B, which occurs before event C (Han, Kim, & Sohn, 2009).
• MBA can be used to access multi level relationships. It can be applied across all levels of analysis ranging from an individual level to firm level, industry level and country level contexts.

2.2.3. APRIORI ALGORITHM

This part will explain the algorithm that will be running behind the python libraries for Market Basket Analysis. This will help the companies to understand their clients more and analyze their data more closely and attentively. Rakesh Agrawal proposed the Apriori algorithm which was the first associative algorithm proposed and future developments in association, classification, associative classification algorithms have used it as a part of the technique.

Association rule mining is seen as a two-step approach:

1. Frequent Itemset Generation

Find all frequent item-sets with support >= predetermined minimum support count. In frequent mining usually the interesting associations and correlations between item sets in transactional and relational databases are found. In short, Frequent Mining shows which items appear together in a transaction or relation. The discovery of frequent itemsets is accomplished in several iterations. Counting new candidate item-sets from existing item sets requires scanning the entire training data. In short it involves only two important steps:

a. Pruning
b. Joining

Frequent Itemset Generation scans the whole database and finds the frequent itemset with a threshold on support. Since it scans the whole database, it is the most computationally expensive step. In the real-world, transaction data for retail can exceed Gigabytes and Terabytes of data for which an optimized algorithm is needed to exclude item-sets that will not help in later steps. For this, the Apriori algorithm is used.

2. Rule Generation

List all association rules from frequent item-sets. Calculate Lift and Confidence for all the rules. Prune rules which fail minimum support and minimum confidence thresholds.
The main idea of the Apriori Algorithm is “Any subset of a frequent itemset must also be frequent”. In other words, no superset of an infrequent itemset must be generated or tested.

In the image below, which is a graphical representation of the Apriori algorithm principle. It consists of a k-item-set node and relation of subsets of the k-item-set. You can notice in the figure that in the bottom is all the items in the transaction data and then you start moving up creating subsets till it reaches to the null set.

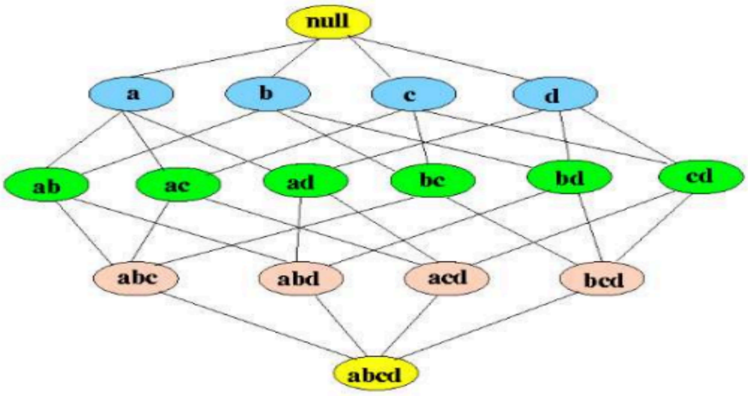


figure : All possible subsets

This shows that it will be difficult to generate frequent item-set by finding support for each combination. Therefore, in the figure below we can notice that Apriori algorithm helps to reduce the number of sets to be generated.

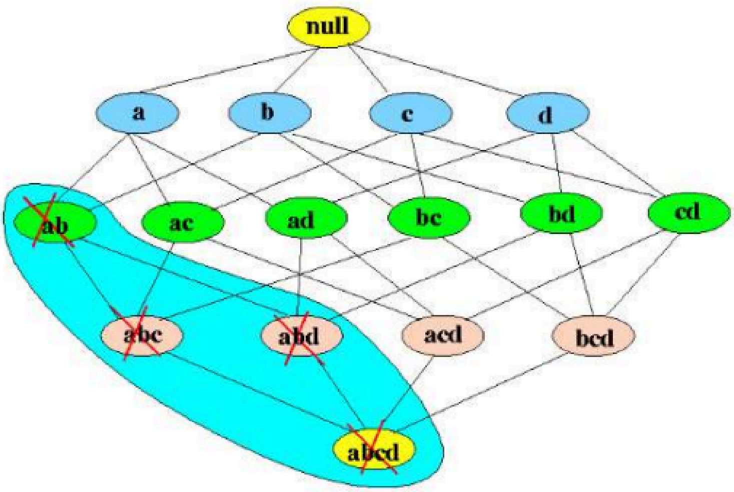


Figure: if an itemset is infrequent, we do not consider its super sets

steps

Step 1. Computing the support for each individual item

The algorithm is based on the notion of **support**. The support is simply **the number of transactions in which a specific product (or combination of products) occurs**.

The first step of the algorithm is to compute the support of each individual item. This basically just comes down to counting, for each product, in how many transactions it occurs.

Step 2. Deciding on the support threshold

Now that we have the support for each of the individual products, we will use that to filter out some of the products that are not frequent. To do so, we need to decide on a support threshold. For our project , we tested multiple possible support thresholds and generated multiple models from which we chose the best one.

Step 3. Selecting the frequent items

Any product that does not meet the minimum threshold will be eliminated and not be considered during the next steps of the Apriori Algorithm.

Step 4. Finding the support of the frequent itemsets

The next step is to do the same analysis, but now using pairs of products instead of individual products. As you can imagine, the number of combinations can quickly become large here, especially if you have a large number of products.

The great ‘invention’ behind the Apriori algorithm is that we will directly ignore all pairs that contain any of the non-frequent items (pruning). Thanks to this, we will have a lot fewer item pairs to scan.

Step 5. Generate Association Rules

Now that you have the largest frequent itemsets, the next step is to convert them into **Association Rules.** Association Rules go a step further than just listing products that occur together frequently.

Association Rules are written in the format: X => Y where X and Y are sets of products. This means that you obtain a rule that tells you that if you buy products in X, you are also likely to buy products in Y.

The confidence tells you a percentage of cases in which this rule is valid. 100% confidence means that this association always occurs; 50% for example means that the rule only holds 50% of the time.

Step 6. Compute lift

Once you have obtained the rules, the last step is to compute the lift of each rule. According to the definition, the lift of a rule is a performance metric that indicates the strength of the association between the products in the rule.

This means that lift basically compares the improvement of an association rule against the overall dataset. If “any product => X” in 10% of the cases whereas “A => X” in 75% of the cases, the improvement would be of 75% / 10% = 7.5.

If the lift of a rule is 1, then the products are independent of each other. Any rule that has a lift of 1 can be discarded.

If the lift of a rule is higher than 1, the lift value tells you how strongly the right hand side product depends on the left-hand side.

This basically gives us three metrics to interpret:

- support (the number of times, or percentage, that the products co-occur)
- confidence (the number of times that a rule occurs, also the conditional probability of the right-hand side given the left-hand side)
- lift (the strength of association)

Those three metrics all have their own validity. It is therefore hard to choose between them. For example, if you have a rule that has a higher lift but lower confidence than another rule, it would be difficult to state that one rule is ‘better than another‘. At this point, you may just want to keep both rules or try to find a reason to prefer one metric over the other in your specific use case.

3.PRACTICAL IMPLEMENTATION OF MARKET BASKET ANALYSIS ON THE DATASET

3.1. DATA COLLECTION

This is a transnational data set which contains all the transactions occurring between 01/12/2010 and 09/12/2011 for a UK-based and registered non-store online retail.The company mainly sells unique all-occasion gifts. Many customers of the company are wholesalers.

3.2. DATASET DESCRIPTION

from the dataset's source website

Dataset Characteristics: Multivariate, Sequential, Time-Series
Subject Area: Business
Associated Tasks: Classification, Clustering
Feature Type: Integer, Real
Instances: 541909
Features: 6

3.2.1. ATTRIBUTES

Attribute	Type	Description	Missing values
InvoiceNo	Categorical	a 6-digit integral number uniquely assigned to each transaction.	no
StockCode	Categorical	a 5-digit integral number uniquely assigned to each distinct product	no
Description	Categorical	product name	no
Quantity	Integer	the quantities of each product (item) per transaction	no
InvoiceDate	Date	the day and time	no

		when each transaction was generated	
UnitPrice	Continuous	product price per unit	no
CustomerID	Categorical	a 5-digit integral number uniquely assigned to each customer	no
Country	Categorical	the name of the country where each customer resides	no

3.3. TOOLS USED FOR ANALYSIS

- Jupyter notebook
- Python Libraries
  - Pandas
  - Datetime
  - Matplotlib
  - Mlxtend
  - Random
  - xlsxwriter
- MS Excel

3.4. Data EXPLORING & TRANSFORMATION

Before doing anything with the data as soon as we read the data in a jupyter notebook, we start exploring it. Data exploring is the most important step for any analysis. While, exploring the data the issues that needed fixing were as follows:

The wide lines of the data exploring and transformation phase are:
- Clean the original dataset
- extract the transactions dataset from the original one by grouping together transactions made by the same customer within a predefined time interval (we'll only consider the product name denoted by 'Description')

some data transformation and cleaning steps we conducted:

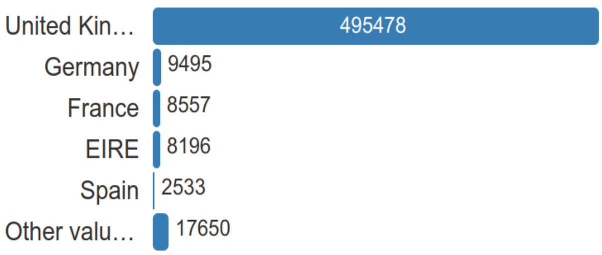
1. An important thing to note is that the majority of transactions were originated from the United Kingdom (UK), exactly 91.4%, therefore only transaction from the UK are considered in this study

Country

Categorical

IMBALANCE

Distinct	38
Distinct (%)	< 0.1%
Missing	0
Missing (%)	0.0%
Memory size	4.1 MiB



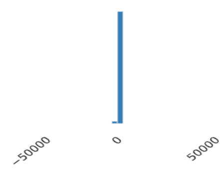
ustomerId' and applying the time interval filter.

2. As this study only concerns patterns between different products, returned items will not be of importance, therefore, the removal of rows with negative quantities (returns)

Quantity

Real number (ℝ)

Distinct	722	Minimum	-80995
Distinct (%)	0.1%	Maximum	80995
Missing	0	Zeros	0
Missing (%)	0.0%	Zeros (%)	0.0%
Infinite	0	Negative	10624
Infinite (%)	0.0%	Negative (%)	2.0%
Mean	9.5522495	Memory size	4.1 MiB



3. A report generated by the famous *ydata_profiling.ProfileReport* library (found in process.ipynb) shows that three columns are of unsupported type: 'StockCode', 'InvoiceNo' and 'Description':

InvoiceNo is an unsupported type, check if it needs cleaning or further analysis	Unsupported
StockCode is an unsupported type, check if it needs cleaning or further analysis	Unsupported
Description is an unsupported type, check if it needs cleaning or further analysis	Unsupported

which is a result of the fact that these columns contain data of different data types.

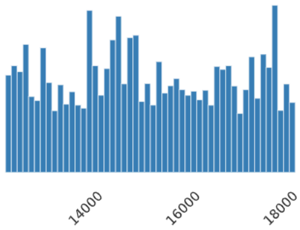
4. the 'InvoiceNo' is not needed, so it was deleted, the other two columns('StockCode' and 'Description') are converted into String data type.
5. Before converting data types, another look into the report tells us that the Description column has 0.3% missing data, if the description of the product is missing, the row will not provide any value for the study, therefore rows with missing Description were dropped.

6. Negative quantity values signifies returned items, which are not a subject for this study, therefore they are to be removed.
7. Another remark is that more than 17% of rows are duplicated rows, which are to be dropped.
8. Next, the final transactions dataset is created by dropping 'InvoiceNo', 'Quantity', 'UnitPrice' and 'Country' columns from the original dataset.
9. Another problem we see is that the CustomerId field has a lot of missing values (24.9%):

CustomerID
Real number (ℝ)

MISSING

Distinct	4372	**Minimum**	12346
Distinct (%)	1.1%	**Maximum**	18287
Missing	135080	**Zeros**	0
Missing (%)	24.9%	**Zeros (%)**	0.0%
Infinite	0	**Negative**	0
Infinite (%)	0.0%	**Negative (%)**	0.0%
Mean	15287.691	**Memory size**	4.1 MiB



To address this we'll split the logic into two loops, one to extract items from the subset with known CustomerID, and another to extract from the subset with missing CustomerID.
For the anonymous users, one solution is to consider each transaction in the original dataset as a basket, the problem with this solution is that it generates a lot of unary transactions (transactions with only one item) which adds little value to the analysis, because in reality clients generally buy a lot of products together, not a single one, therefore, we used another technique which depends on the InvoiceDate column, in summary, all transactions made within a predefined time interval will be considered as a single transaction, this same technique was used to extract transactions made by registered clients after grouping by CustomerID, and then applying the time interval filter

After applying steps 1 to 8 we obtained a cleaner dataset, which is described in great detail in the report generated by the *ydata_profiling.ProfileReport* python library, which is available in the *process.ipynb* notebook.

The obtained dataset will be used in the last step: extracting transactions.

3.5. ANALYSIS & DISCUSSION

3.5.1. Association rules from dataset:

In case that we want to see the rules , we can check the rules json file with the path `Apriori\results\rules.json` , let's take an example about the first 10 rules with their confidences and lifts :

RULES	confidence	lift
DOLLY GIRL LUNCH BOX ========> ROUND SNACK BOXES SET OF 4 WOODLAND	0.4444444444444447	3.5623409669211169
REGENCY MILK JUG PINK ========> ROSES REGENCY TEACUP AND SAUCER	0.5625	19.39655172413793
PICNIC BOXES SET OF 3 RETROSPOT ========> POSTAGE	0.4444444444444444	1.019367991845056
SET OF 60 I LOVE LONDON CAKE CASES ========> POSTAGE	0.5	1.146788990825688
PLASTERS IN TIN WOODLAND ANIMALS ========> PLASTERS IN TIN SPACEBOY	0.46987951807228917	6.43670572701766
TRADTIONAL ALPHABET STAMP SET ========> MINI LIGHTS WOODLAND MUSHROOMS	0.5714285714285714	18.433179723502302
PLASTERS IN TIN CIRCUS PARADE ========> PLASTERS IN TIN WOODLAND ANIMALS	0.5189873417721519	6.2528595394235165
RED DINER WALL CLOCK ========> BLUE DINER WALL CLOCK	0.6428571428571429	35.71428571428572
FELTCRAFT DOLL ROSIE ========> FELTCRAFT DOLL MOLLY	0.47058823529411764	29.41176470588235
WHITE SPOT BLUE CERAMIC DRAWER KNOB ========> RED STRIPE CERAMIC DRAWER KNOB	0.4166666666666667	20.833333333333332

We've obtained multiple results by changing the parameters of our model:
We tried the following combinations of parameters:

- Support : 5 , Minimum Confidence : 0.4 , Time : 10 , Basket size : 50
  We got 678 Rules
- Support : 4 , Minimum Confidence : 0.3 , Time : 5, Basket size : 30
  We got 2527 Rules
- Support : 5 , Minimum Confidence : 0.4 , Time : 10 , Basket size : 30
  We got 2068 Rules
- Support : 4 , Minimum Confidence : 0.2 , Time : 3, Basket size : 20
  We got more than 3000 Rules
  Therefore we chose to deploy the last model.

	antecedent	consequent	confidence	lift
0	DOLLY GIRL LUNCH BOX	ROUND SNACK BOXES SET OF4 WOODLAND	0.466667	3.562341
1	REGENCY MILK JUG PINK	ROSES REGENCY TEACUP AND SAUCER	0.562500	19.396552
2	PICNIC BOXES SET OF 3 RETROSPOT	POSTAGE	0.444444	1.019368
3	SET OF 60 I LOVE LONDON CAKE CASES	POSTAGE	0.500000	1.146789
4	PLASTERS IN TIN WOODLAND ANIMALS	PLASTERS IN TIN SPACEBOY	0.469880	6.436706
5	TRADTIONAL ALPHABET STAMP SET	MINI LIGHTS WOODLAND MUSHROOMS	0.571429	18.433180
6	PLASTERS IN TIN CIRCUS PARADE	PLASTERS IN TIN WOODLAND ANIMALS	0.518987	6.252860
7	RED DINER WALL CLOCK	BLUE DINER WALL CLOCK	0.642857	35.714286
8	FELTCRAFT DOLL ROSIE	FELTCRAFT DOLL MOLLY	0.470588	29.411765
9	WHITE SPOT BLUE CERAMIC DRAWER KNOB	RED STRIPE CERAMIC DRAWER KNOB	0.416667	20.833333

4. Integration into Website:

In the context of this project, A website application was developed in order to showcase our model.
As known that website applications have two essential components Backend and Frontend, all of this is explained in detail below in the upcoming sections.

4.1. System Architecture

The backend or another famous term is the **API** which stands for Application Programming Interface.
FastAPI is one of the technologies which provides cutting-edge, high-performance web tools for creating APIs, and built to manage challenging jobs effectively and dependably.

The website's development requires the usage of various well-known FastAPI / Python modules, which are the following :

- Pydantic : gives a declarative syntax for constructing data models and serves as a framework for data validation and processing. Its main function is to impose restrictions and type annotations on data structures, making sure that they follow predetermined patterns.
  In situations requiring data validation, serialization, and deserialization, Pydantic is very helpful as it improves code dependability and lowers the possibility of runtime mistakes.
- CORSMiddleware : online browsers use a security feature called Cross-Origin Resource Sharing (CORS) to limit and regulate how resources from various domains may be accessed on online pages. Unauthorized cross-origin requests are stopped, which might jeopardize the security of web apps.The CORSMiddleware module integrates CORS functionality into FastAPI. CORSMiddleware helps with CORS policy management when it is added to the FastAPI app and imported. The parameters that developers may set to determine which external domains are authorized to access resources from the FastAPI server include allowed origins, methods, headers, and credentials. By doing this, secure communication between client-side browser programs and the FastAPI application is guaranteed.

Most backend Technologies use a technique called Routes, which specify the HTTP methods and related functions or processes that define an API's endpoints. These endpoints serve as the application's entry points for client requests, enabling it to process and react to different actions. A defined URL path and HTTP method are linked to each route, offering an ordered and systematic approach of exposing functionality.

- Predict Route: The user adds the necessary products to his cart, and the API keeps trying to guess what will come next as a product. Behind the scenes our api is just comparing the user basket of products with our rules generated by the model.
- Get Products Route : This Route is designed to return the product list to the frontend in order to let the user search for his desired products.

5. Conclusion

5.1. summary

As we come to the end of our Market Basket Analysis adventure, we can say that it has been an exciting voyage into the realm of transactional data-driven consumer behavior. Understanding fundamental ideas such as association rules and the Apriori algorithm was similar to cracking the code of consumer preferences. The thrill of discovering patterns in our dataset led to the practical implementation. Every stage of the process, from gathering data to using the magic wand of the Apriori algorithm, felt like removing layers to uncover the tales concealed inside shopping trolleys. The association rules that were produced proved to be a treasure map, providing important information about which goods go well together. This study demonstrated the effectiveness of market basket analysis as a narrative tool as well as a tool for corporate planning.

6. References:

1. Agrawal, R., Imielinski, T., & Swami, A. (1993). Mining Association in Large Databases. Proceedings of the 1993 ACM SIGMOD International Conference on Management of Data - SIGMOD '93, 207–216 https://doi.org/10.1145/170036.170072
2. Solnet, D., Boztug, Y., & Dolnicar, S. (2016). An untapped gold mine? Exploring the potential of market basket analysis to grow hotel revenue. International Journal of Hospitality Management, 56, 119–125 https://doi.org/10.1016/j.ijhm.2016.04.013.
3. Kamley, S., Jaloree, S., & Thakur, R. S. (2014). An Association Rule Mining Model for Finding the Interesting Patterns in Stock Market Dataset. International Journal of Computer Applications, 93(9), 975–8887. Retrieved from http://research.ijcaonline.org/volume93/number9/pxc3895792.pdf
4. [the original dataset]. Retrieved from https://archive.ics.uci.edu/dataset/352/online+retail

5. [Figure 3, 4] Hafsa Jabeen (August 21st, 2018). Retrieved from https://www.datacamp.com/community/tutorials/market-basket-analysis-r