ONLINE SHOPPING CART ANALYSIS TO UNDERSTAND THE CUSTOMER'S ONLINE EXPENDITURE PATTERN

A PROJECT REPORT

for

DATA MINING TECHNIQUES (ITE2006)

in

B.Tech – Information Technology and Engineering

by

MADHUR PATIDAR (19BIT0059)

ROHAN PAL (19BIT0211)

NIBRAS IBRAHIM MOHAMMED ALI (19BIT0351)

Under the Guidance of

Dr. SENTHILKUMAR N C

Associate Professor, SITE

VIT®

Vellore Institute of Technology

(Deemed to be University under section 3 of UGC Act, 1956)

School of Information Technology and Engineering

December, 2021

DECLARATION BY THE CANDIDATE

We hereby declare that the project report entitled **"ONLINE SHOPPING CART ANALYSIS TO UNDERSTAND THE CUSTOMER'S ONLINE EXPENDITURE PATTERN"** submitted by us to Vellore Institute of Technology University, Vellore in partial fulfillment of the requirement for the award of the course **Data Mining Techniques (ITE2006)** is a record of bonafide project work carried out by us under the guidance of **Dr. Senthilkumar N C.** We further declare that the work reported in this project has not been submitted and will not be submitted, either in part or in full, for the award of any other course.

Place : Vellore

Signature

Date : 24/11/21

Rohan Pal

VIT®
Vellore Institute of Technology
(Deemed to be University under section 3 of UGC Act, 1956)

School of Information Technology & Engineering [SITE]

CERTIFICATE

This is to certify that the project report entitled **“ONLINE SHOPPING CART ANALYSIS TO UNDERSTAND THE CUSTOMER’S ONLINE EXPENDITURE PATTERN”** submitted by **MADHUR PATIDAR (19BIT0059), ROHAN PAL(19BIT0211), NIBRAS IBRAHIM MOHAMMED ALI(19BIT0351)** to Vellore Institute of Technology University, Vellore in partial fulfillment of the requirement for the award of the course **Data Mining Techniques (ITE2006)** is a record of bonafide work carried out by them under my guidance.

Dr. Senthilkumar N C

GUIDE

Associate Professor, SITE

ONLINE SHOPPING CART ANALYSIS TO UNDERSTAND THE CUSTOMER'S ONLINE EXPENDITURE PATTERN

Abstract

In day-to-day activities huge amounts of data are generated, as a result the volume of data is increasing dramatically. Mining information from this explosive growth of data has become one of the major challenges for data management and mining communities. Moreover, the majority of the recognized organizations collect and store massive amounts of customer transaction data. However, having these massive data do not mean the organizations had rich commercial information. The business industries need to discover valuable information and knowledge from this vast quantity of data. This leads us to online shopping cart analysis. Shopping cart analysis aims at finding out purchasing patterns by discovering important associations among the products which they place in their online shopping carts. It not only assists in decision making process but also increases sales in many e-commerce websites like amazon, flipkart etc... Apriori and FP Growth are the most common algorithms for mining frequent itemsets. For both of these algorithms predefined minimum support is needed to satisfy for identifying the frequent itemsets. But when the minimum support is low, a huge number of candidate sets will be generated which requires large computation. In this project, we plan to follow an approach that has been proposed to avoid this large computation by reducing the items of dataset with top selling products. The top selling products will be marketed more with the help of suggestions to the customers. Various percentages of top selling products like 30%, 40%, 50%, 55% have been taken and for both algorithms frequent itemsets and association rules are generated. The results show that if top selling items are used, it is possible to get almost same frequent itemsets and association rules within a short time comparing with those outputs which are derived by computing all the items. From time comparison it is also found that FP Growth algorithm takes smaller time than Apriori algorithm

Keywords – Market Basket Analysis (MBA), Data Mining, Association Rule Mining (ARM), Product Recommendation system.

I. INTRODUCTION

Market basket Analysis is a method of data analysis based on Association data mining technique used by retailers to increase sales by better understanding customer purchasing patterns. It involves analyzing large data sets, such as purchase history, to reveal product groupings, as well as products that are likely to be purchased together. Market basket analysis finds out customers' purchasing patterns by discovering important associations among the products which they place in their shopping baskets. It not only assists in decision making process but also increases sales in many business organizations. Apriori and FP Growth are the most common algorithms for mining frequent itemset. Market Basket Analysis is an important part of the analytical system in the retail organization to determine the placement of goods, designing sales promotion for different segments of customers to improve customer satisfaction and hence the profit of the supermarkets.

II. BACKGROUND

We have used the concept of Association Rule Mining in our project. Association rule mining, at a basic level, involves the use of machine learning models to analyze data for patterns, or co-occurrences, in a dataset. It identifies frequent if-then associations, which themselves are the association rules. An association rule has two parts: an antecedent (if) and a consequent (then). An antecedent is an item found within the data. A consequent (X) is an item found in combination with the antecedent (Y). Association rules are created by searching data for frequent if-then patterns and using the criteria support and confidence to identify the most important relationships. Support is an indication of how frequently the items appear in the data. Confidence indicates the number of times the if-then statements are found true. Lift, can be used to compare confidence with expected confidence, or how many times an if-then statement is expected to be found true. Leverage, computes the difference between the observed frequency of two items X and Y appearing together and the frequency that we would expect if these items are independent. Conviction can be interpreted as the ratio of the expected frequency that X occurs without Y (that is to say, the frequency that the rule makes an incorrect prediction) if X and Y were independent divided by the observed frequency of incorrect predictions. Measures of the effectiveness of association rules: The strength of a given association rule is measured by two main parameters: support and confidence. Support refers to how often a given rule appears in the database being mined. Confidence refers to the amount of times a given rule turns out to be true in practice. A rule may show a strong correlation in a data set because it appears very often but may occur far less when applied. This would be a case of high support, but low confidence. Conversely, a rule might not particularly stand out in a data set, but continued analysis shows that it occurs very frequently. This would be a case of high confidence and low support. Using these measures helps analysts separate causation from correlation, and allows them to properly value a given rule. A third value parameter, known as the lift value, is the ratio of confidence to support. If the lift value is a negative value, then there is a negative correlation between datapoints. If the value is positive, there is a positive correlation, and if the ratio equals 1, then there is no correlation. Example: A supermarket has 200,000 customer transactions. About 4,000 transactions, or about 2% of the total number of transactions, include the purchase of bread. About 5,500 transactions (2.75%) include the purchase of butter. Of those, about 3,500 transactions, 1.75%, include both the purchase of bread and butter. Based on the percentages, that large number should be much lower. However, the fact that about 87.5% of bread purchases include the purchase of butter indicates a link between bread and butter. Association rule algorithms:

1) Apriori Algorithm Apriori algorithm is a classical algorithm in data mining. It is used for mining frequent itemset and relevant association rules. It is devised to operate on a dataset containing a lot of transactions, for instance, items brought by customers in a store. Association rule learning is a prominent and a well-explored method for determining relations among variables in large databases.

2) FP GROWTH (Frequent Pattern Growth): This algorithm is an improvement to the Apriori method. A frequent pattern is generated without the need for candidate generation. FP growth algorithm represents the dataset in the form of a tree called a frequent pattern tree or FP tree. This tree structure will maintain the association between the itemset. The database is fragmented using one frequent item. This fragmented part is called “pattern fragment”. The itemset of these fragmented patterns are analyzed. Thus, with this method, the search for frequent itemset is reduced comparatively.

III. Literature Survey

[1] This study aims to examine how Market Basket Analysis (MBA) is useful in finding the association in consumer's buying behavior and giving the Product recommendation to consumers based on consumer's past purchases. In this paper they have focused on mainly two algorithms: Apriori Algorithm to find out the frequent buying pattern of the consumer and Collaborative Filtering algorithm to give the Product Recommendations. They have conducted the comparative study of these two algorithms to find out the differences and similarities between the product association and product recommendation.

[2] This paper aims to presents a basic Concepts of some of the algorithms (FPGrowth, COFI-Tree, CT-PRO) based upon the FP- Tree like structure for mining the frequent item sets along with their capabilities and comparisons. Data mining implementation on medical data to generate rules and patterns using Frequent Pattern (FP)-Growth algorithm is the major concern of this research study. This paper presents how data mining can apply on medical data.

[3] Study explores gender wise determinants of online shopping intentions amongst existing online shoppers in India. Research framework was developed by integrating Perceived Trust into TAM construct and was tested on samples of male (n=432) and female (n=275) respondents. With statistical tools such as EFA, CFA and SEM, Perceived Usefulness was validated as antecedent of male respondent's intention to continue with online shopping while Perceived Ease of Use and Perceived Trust determines intention to continue amongst Female counterparts.

[4] This study aimed to evaluate Fuzi and Fuzi-based formulas in modern clinical practices using artificial intelligence and data mining methods. This nationwide descriptive study with market basket analysis used a cohort selected from the Taiwan National Health Insurance database that contained one million national representatives between 2003 and 2010 used for our analysis. Descriptive statistics were performed to demonstrate the modern clinical indications of Fuzi. Market basket analysis was calculated by the Apriori algorithm to discover the association rules between Fuzi and other TCM herbs.

[5] This study aims to apply association rule mining (ARM) to uncover specific associations between operating components of a chiller system and improve its coefficient of performance (COP), hence reducing the electricity use of buildings with central air conditioning. First, 13 operating variables were identified, comprising measures of temperatures and flow rates of system components and their switching statuses. The variables were grouped into four bins before carrying out ARM. Strong rules were produced to associate the variables and switching statuses with different COP classes.

[6] This research is conducted to analyze the shopping basket by using association rules in the retail area, more specifically in a home goods sales company such as appliances, computer items, furniture, and sporting goods, among others. With the rise of globalization and the advancement of technology, retail companies are constantly struggling to maintain and raise their profits, as well ordering the products and services that the customer wants to obtain. In this sense, they need a new approach to identify different objectives in order to be more competitive and successful, looking for new decision-making strategies. To achieve this goal, and to obtain clear and efficient strategies, by providing large amounts of data collected in business transactions, the need arises to intelligently analyze such data in order to extract useful knowledge that will support decision-making and, an understanding of the association patterns that occur in sales‐customer behavior. Predicting which product will make the most profit, products that are sold together, this type of information is of great value for storing products in inventory. Knowing when a product is out of fashion can support inventory management effectively. In this sense, this work presents the rules of association of products obtained by analyzing the data with the FPGrowth algorithm using the Orange tool.

[7] In this model, a five-step procedure is used to solve the problem of predicting the Sales revenue for different products at different outlet locations for Big Mart Companies. The processes includes: data being acquired, collected and divided into training and test label (This data undergoes a preliminary analysis which includes univariate and bivariate analysis), Data pre-processing (for missing and erroneous values), Feature Selection and Modification, feature transformation, model building (Linear regression, XGBoost, and evaluation and finally result analysis was done based on mean, variance and covariance.

[8] E-commerce platforms such as Taobao can collect massive users 'shopping behavior data, which makes it possible to grasp users' shopping preferences. However, the current research methods of user operation behavior prediction usually only analyze a certain type of user's operation behavior, which cannot fully reflect the overall characteristics of user behavior. Based on the shopping behavior data of Alibaba's e-commerce platform, this article mines user characteristics, product characteristics, product category characteristics, user-product characteristics, and user-product category characteristics from a large amount of online shopping behavior data.

[9] The paper examines the rise of e-commerce trends in the past few years, with a specific focus on online purchasing behaviour in the Republic of North Macedonia. Ecommerce has seen dramatic increase in 2020, especially with the emergence of the COVID-19 pandemic which influenced consumers towards online shopping. Statistics demonstrate that in the following years we can expect this rate of growth to continue, making e‐commerce an option for all types of businesses and industries. These trends are also evident in the Republic of North Macedonia, where e-commerce sales have increased by more than 50% comparing 2019 to 2020.

[10] This study examines how consumer demographics and psychographics may influence their online shopping patronage (i.e., the frequency of online purchases). Findings show that younger people and individuals with a higher income and education level are likely to shop online more frequently, while gender has no effect on the frequency of online shopping. This study also finds that propensity to trust, variety seeking, and impulsive buying are positively related to the frequency of online shopping, while risk aversion is negatively related to the frequency of online shopping. Theoretical and managerial implications are discussed.

[11] In this paper, it is observed from the analysis that the data mining tools can be effectively used for optimizing the patterns associated with dynamic behaviors of the transactions which were made by the customers in purchasing some specific products. Using the Market Basket algorithm, the frequent transactions made by the customers have been analyzed using the support and confidence of the customers in buying associated items. By using this methodology, it is seen that there exists certain association between the products at the time of purchasing the products by the customers.

[12] The FP-Growth algorithm is a development of Apriori, the deficiency of the Apriori algorithm improved by the FP-Growth algorithm. In Apriori a generate candidate is required to get frequent itemset. However, FP-Growth generating candidate algorithm is not done because FP-Growth uses the concept of tree development in search of the frequent itemset. This is why, the FP�Growth algorithm is faster than the Apriori algorithm.

[13] The E-commerce zone is crowded with many Internet users. Medical E-commerce has had significant growth in part because of a great deal of growth in the Indian E-commerce field. Medical E-commerce sites use cloud computing to guarantee a high quality of service anywhere and anytime in the world. For online access, the customer's expectations are very high. Medical E-commerce retailers are directed towards cloud service providers based on their quality of service. During online shopping, impatient customers may abandon a specific medical E�commerce shopping cart due to slow response. This is quite difficult to endure for a medical E�commerce firm. The research described herein observed the effect of shopping cart abandonment on medical E-commerce websites deployed in cloud computing. The impact of the idle virtual machine on customer impatience during medical E-commerce shopping was also studied. The ultimate aim of this study was to propose a stochastic queueing model and to yield results through probability generating functions. The results of the model may be highly useful for a medical E-commerce firm facing customer impatience, so as to design its service system to offer satisfactory quality of service.

[14] Apriori and FP Growth are two most basic algorithms for finding frequent itemsets and discovering associations among products . In this paper, They have used Apriori and FP Growth algorithms for discovering popular items in transactional datasets and obtaining relations among those items. They have also proposed a new approach for mining association rules by selecting a specific percentage of frequent items from our dataset and have performed many tests to support our proposal.

[15] Shopping behavior data is of great importance in understanding the effectiveness of marketing and merchandising campaigns. Online clothing stores are capable of capturing customer shopping behavior by analyzing the click streams and customer shopping carts. Retailers with physical clothing stores, however, still lack effective methods to comprehensively identify shopping behaviors. In this paper, we show that backscatter signals of passive RFID tags can be exploited to detect and record how customers browse stores, which garments they pay attention to, and which garments they usually pair up. The intuition is that the phase readings of tags attached to items will demonstrate distinct yet stable patterns in a time-series when customers look at, pick out, or turn over desired items. We design ShopMiner, a framework that harnesses these unique spatial-temporal correlations of time-series phase readings to detect comprehensive shopping behaviors. We have implemented a prototype of ShopMiner with a COTS RFID reader and four antennas, and tested its effectiveness in two typical indoor environments. Empirical studies from two-week shopping-like data show that ShopMiner is able to identify customer shopping behaviors with high accuracy and low overhead, and is robust to interference.

[16] This paper explained regarding all the activities and processes that being done by the researcher in order to achieve all the objectives that have been set at the early stage of the project. The main objective of this research to predict the items that will be sold the highest for every month and to understand the correlations between the purchased items by the customers. Even there was some limitations for this project, it still can successfully be done according to the current needs and requirements.

[17] This paper is focused on the customer attitude towards online shopping with reference to cart abandonment, here in this paper it’s also mentioned to find out the various demographic variables which are responsible for the actual purchase. The study based on identification and investigation of potential demographic factors responsible for shopping cart abandonment.

[18] This paper describes the way of Market Basket Analysis implementation to Six Sigma methodology. Data Mining methods provide a lot of opportunities in the market sector. Basket Market Analysis is one of them. Six Sigma methodology uses several statistical methods. With implementation of Market Basket Analysis (as a part of Data Mining) to Six Sigma (to one of its phase), we can improve the results and change the Sigma performance level of the process. In our research we used GRI (General Rule Induction) algorithm to produce association rules between products in the market basket. These associations show a variety between the products.

[19] The Apriori algorithm is divided into several stages called narrative which consists of: The establishment of candidate items, Calculation of support of each k-itemset candidate, Set the high frequency pattern and if no new high frequency pattern is obtained the whole process is stopped. If not, then k plus one and return part 1. In this paper conducted by this author has been able to produce the pattern of selling the most products of interest by customers by applying data mining system on each transaction data. By using a priori Algorithm as the basis of which there are methods of association rules and CRISP-DM in this system can determine the value of support and confidence of the implementation of data mining based on the rules of the combination of products as input value.

[20] The aim of this paper was to analyze the sales of a big superstore, and predict their future sales. The proposed approach was organized into three stages, first is data collection, which includes collecting data and transforming it into processed data. Then, it includes modeling the data for predictions using machine learning techniques (focusing on linear regression algorithm) and, finally validating and implementation of our results using precision and accuracy techniques. Algorithms used: Naïve Bayes classifier, K Means Clustering Algorithm, SVM, Linear Regression, Logistics Regression, Decision tree, Random forest, nearest neighbor.

[21] In this Paper, the data will be used to determine the efficient proposed layout to reduce the total people walking distance. After comparing the total people walking distance of the Initial Layout and Proposed Layout by using 7 data transactions as a sample, there is reduced length of people walking distance from 179.38 meters for initial layout to 64.38 for proposed layout. As a result, the improvement in percentage is about 64.97%. It means that the proposed layout consumes less walking distance if compared to the previous layout. Displaying products in certain level of shelf also has significant influence on customers' buying behavior.

[22] Online shopping has taken off as an increasing number of consumers purchase increasingly diversified products on the Internet. Given that how to attract and retain consumers is critical to the success of online retailers, research on the antecedents of consumer acceptance of online shopping has attracted widespread attention. There has yet to be a holistic view of online shopping acceptance from the perspective of consumers. This research paper has conducted an extensive survey of extant related studies and synthesized their findings into a reference model called OSAM (Online Shopping

Acceptance Model) to explain consumer acceptance of online shopping. The literature survey reveals that a myriad of factors have been examined in the context of online shopping and mixed results on those factors have been reported. The proposed model helps reconcile conflicting findings, discover recent trends in this line of research, and shed light on future research directions.

[23] In this paper, a forecasting model that combines the Bass/Norton model and sentiment analysis techniques isproposed. In contrast to the extant literature that uses online ratings,this paper extends the Bass model by analysing sentiments expressed in online reviews. In contrast to the original Bass model, both historical sales and online review data are directly used in the extended model. The NB method is adopted to calculate the sentiment index and conduct polarity classifications for each online review, and the extracted sentiment index is used to expand the imitation coefficient in the Bass model. The same method is used to expand the Norton model. Sentiment information is rarely used to extend the Bass model in existing studies.

[24] This Paper describes MBA and that it allows for inductive theorizing; can address contingency (i.e., moderated) relationships; does not rely on assumptions such as linearity, normality, and residual equal variance, which are often violated when using general linear model–based techniques; allows for the use of data often considered “unusable” and “messy” in management research (e.g., data not collected specifically for research purposes); can help build dynamic theories (i.e., theories that consider the role of time explicitly); is suited to examine relationships across levels of analysis; and is practitioner friendly.

[25] The authors examine consumers’ information channel usage during the customer journey by employing a hedonic and utilitarian (H/U) perspective, an important categorization of consumption purpose. Taking a retailer-category viewpoint to measure the H/U characteristics of 20 product categories at 40 different retailers, this study combines large-scale secondary clickstream and primary survey data to offer actionable insights for retailers in a competitive landscape. The data reveal that, when making hedonic purchases (e.g., toys), consumers employ social media and on-site product pages as early as two weeks before the final purchase. By contrast, for utilitarian purchases (e.g., office supplies), consumers utilize third-party reviews up to two weeks before the final purchase and make relatively greater usage of search engines, deals, and competitors’ product pages closer to the time of purchase.

[26] A prediction model is made up of a number of predictors, which are variable factors that are likely to influence future behavior or results. Market Basket Analysis is one of the most popular types of data analysis used in the marketing world. The purpose of Market Basket Analysis is to determine what products are most commonly purchased or used by consumers. This MBA is analyzing consumer buying habits by finding associations between different products that consumers place in shopping basket. Any classifier that can be used for determining a particular class for a test object is part of predictive modeling. Therefore market basket analysis can be used for prediction modeling Predictive modeling offers the potential for firms to be proactive instead of receptive. Predictive modeling using transactional data create particular challenges which need to be carefully addressed to develop valuable models.

[27] This study has provided an empirical glimpse into the minds of consumers as to what factors are perceived differently by consumers who prefer online shopping and those who prefer offline shopping. A direct survey was used to collect the data for this study. The results revealed that five factors were perceived differently between these two groups of consumers. These factors were: (1) perceived risk with online shopping, (2) past experience with online shopping, (3) perceived benefits of online shopping, (4) perceived ease of online shopping, and (5) perceived uncertainty of online shopping. Determining such factors may provide online businesses with a base level awareness of online consumers' perceptions and of what factors into preferences for online or offline shopping avenues. This awareness could provide insights into what needs to be done to attract and retain more online customers.

[28] In this paper data mining algorithms have been developed and applied on variety of practical problems. However periodic mining is a new approach in data mining which has gained its significance these days. This field is evolving due to needs in different applications and limitations of data mining. This would enhance the power of existing data mining techniques. Finding out the patterns due to changes in data is in itself an interesting area to be explored. It may helpful in find out interesting patterns from large amount of data, automatically track the changes in facts from previous data; due to this feature it may be helpful in fraud detection. And predicting future association rules as well as gives us right methodology to find out outliers.

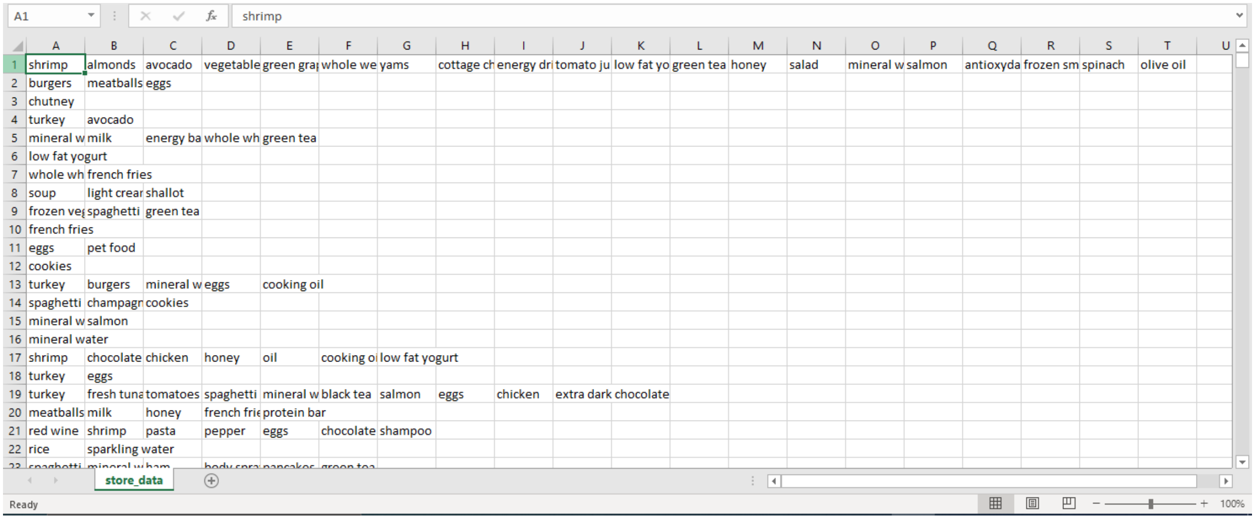
[29] ARM is a procedure for finding relationships between items of a defined dataset and searching for and finding relationships between items in a dataset Implementation of data mining with the rules of association aims to find information of items that are alternating since in the form of rules. The rules of association are data mining techniques for finding association rules among a combination of items.

[30] This paper presents the application of various machine learning models, hybrid models and decomposition technique to forecast the sales of Rossmann Store. The various prediction models here involves: Forecasting models-ARIMA (ACF, PACF plots), Auto Regressive Neural Network (ARNN has been used), XGBoost (Training loss-MSE, Regularization-prevent overfitting), SVM.Hybrid approach-hybrid approach takes into account a linear and a nonlinear model for better results.

IV. DATASET DESCRIPTION & SAMPLE DATA

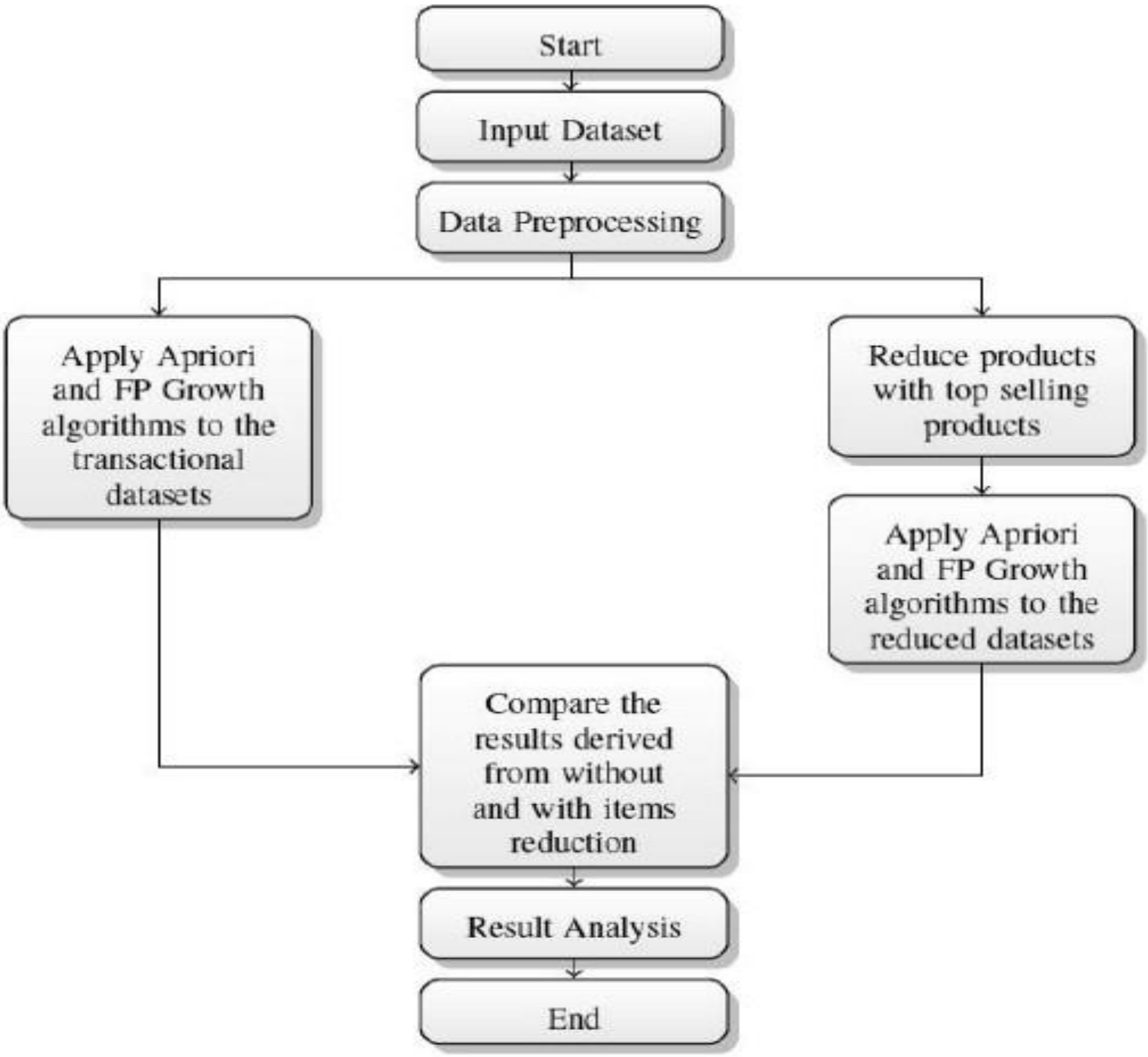
The data involved in any sale transaction in e-commerce websites such as amazon, such as data of items purchased, time of purchase, total sales volume, item price. E-commerce companies require additional data for managers to make strategic decisions that can increase company profits, such as the most sold product information, slightly sold products, and rarely sold products. To maintain inventory, it is essential to know the pattern of consumer spending that often occurs at these websites by analyzing the data of sales transactions. The placement of the product layout is still less accurate and optimal because it is only based on management's perception by categorizing the existing products and has not been reviewed from the consumer's point of view. So that, the researcher's initiative to try to provide solutions in the placement of the product layout.

Sample Data:



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U
1	shrimp	almonds	avocado	vegetable	green gra|	whole we	yams	cottage ch	energy dri	tomato ju	low fat yo	green tea	honey	salad	mineral w	salmon	antioxyda	frozen sm	spinach	olive oil	
2	burgers	meatballs	eggs																		
3	chutney																				
4	turkey	avocado																			
5	mineral w	milk	energy ba	whole wh	green tea																
6	low fat yogurt																				
7	whole wh	french fries																			
8	soup	light crear	shallot																		
9	frozen ve|	spaghetti	green tea																		
10	french fries																				
11	eggs	pet food																			
12	cookies																				
13	turkey	burgers	mineral w	eggs	cooking oil																
14	spaghetti	champagn	cookies																		
15	mineral w	salmon																			
16	mineral water																				
17	shrimp	chocolate	chicken	honey	oil	cooking oi	low fat yogurt														
18	turkey	eggs																			
19	turkey	fresh tuna	tomatoes	spaghetti	mineral w	black tea	salmon	eggs	chicken	extra dark chocolate											
20	meatballs	milk	honey	french frie	protein bar																
21	red wine	shrimp	pasta	pepper	eggs	chocolate	shampoo														
22	rice	sparkling water																			
23	spaghetti	mineral w	ham	body spra	pancakes	green tea															

V. PROPOSED ALGORITHM WITH FLOWCHART


Start
Input Dataset
Data Preprocessing
Apply Apriori and FP Growth algorithms to the transactional datasets
Reduce products with top selling products
Apply Apriori and FP Growth algorithms to the reduced datasets
Compare the results derived from without and with items reduction
Result Analysis
End

Initially the dataset is taken as input and then the pre processing of the data in the dataset takes place. Then, two different steps occur parallelly:-

1. Apirori and FP growth algorithms are applied to the interested section of the dataset.

2. Before applying the mining algorithms the number of entries in the existing dataset is reduced by checking which of them are top selling products. After this is done the algorithms are applied on the resulting reduced dataset.

Once both the above results are obtained the results are compared and then analyzed.

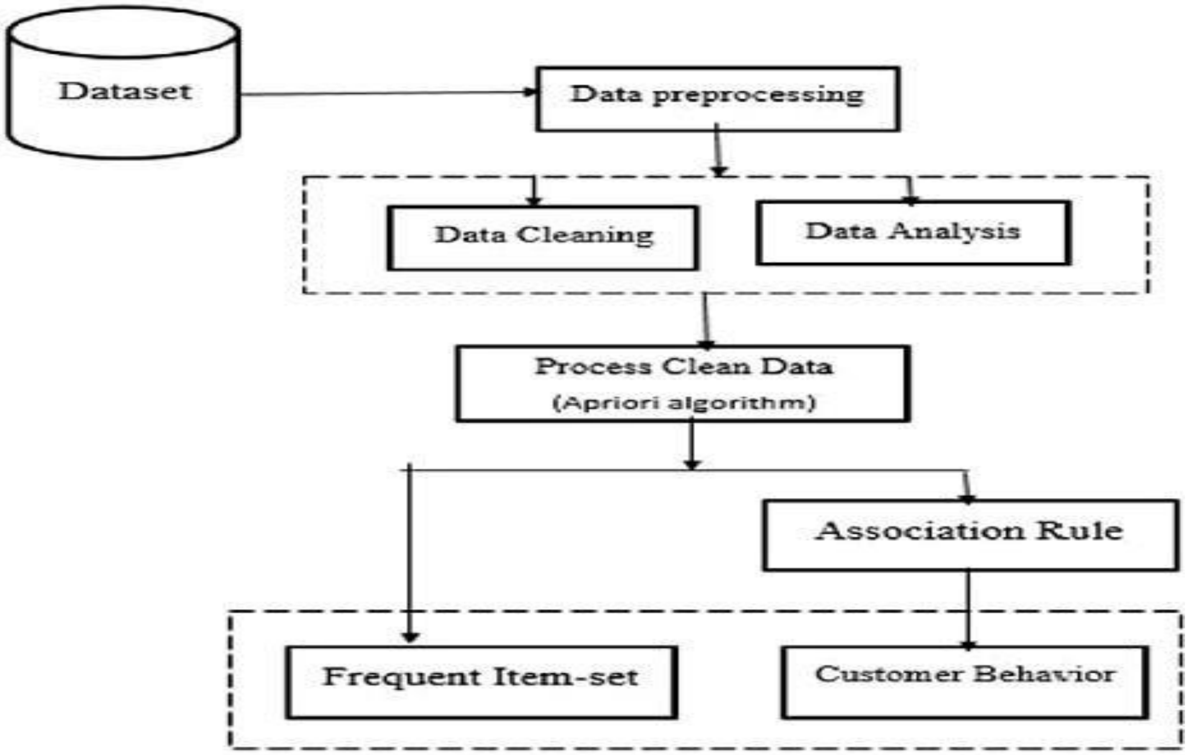
Apriori algorithm

Association rule mining finds interesting associations and/or correlation relationships among large set of data items. Association rules shows attribute value conditions that occur frequently together in a given dataset. A typical andwidely used example of association rule mining is Market Basket Analysis. Forexample, data are collected from the supermarkets. Such market basket databases consist of a large number of transaction records. Each record lists allitems bought by a customer on a single purchase transaction. Association rules provide information of this type in the form of “IF-THEN” statements. The rules are computed from the data, an association rule has two numbers that express the degree of uncertainty about the rule.

FP-GROWTH (Frequent Pattern Growth):

This algorithm is an improvement to the Apriori method. A frequent pattern is generated without the need for candidate generation. FP growth algorithm represents the database in the form of a tree called a frequent pattern tree or FPtree. This tree structure will maintain the association between the itemset. The database is fragmented using one frequent item. This fragmented part is called “pattern fragment”. The itemset of these fragmented patterns are analyzed.

Thus, with this method, the search for frequent itemset is reducedcomparatively.



VI. EXPERIMENTS RESULTS

The above process generates Association Rules with various fixed metrics such as Support, Confidence, lift, etc. which are used to analyze retail basket or transaction data. These metrics help us understand the strength of association between antecedent and consequent**.**

In our experiment we have made 2 Association Rule with the help of Apriori and FpGrowth algorithm.

- In the first Association rule we have taken minimum support value of frequent item set is 0.05 and confidence of 0.3. For an example when antecedent is chocolate and consequent is mineral water then that means that 30% of the transactions containing chocolate and also contain mineral water.
- In the Second Association rule we have taken minimum support value of frequent item set is 0.05 and minimum lift value of 1.3. If the lift value is greater than 1that means the probability of occurrence of the antecedent and that of the consequent are greater.

VII. COMPARATIVE STUDY / RESULTS AND DISCUSSION

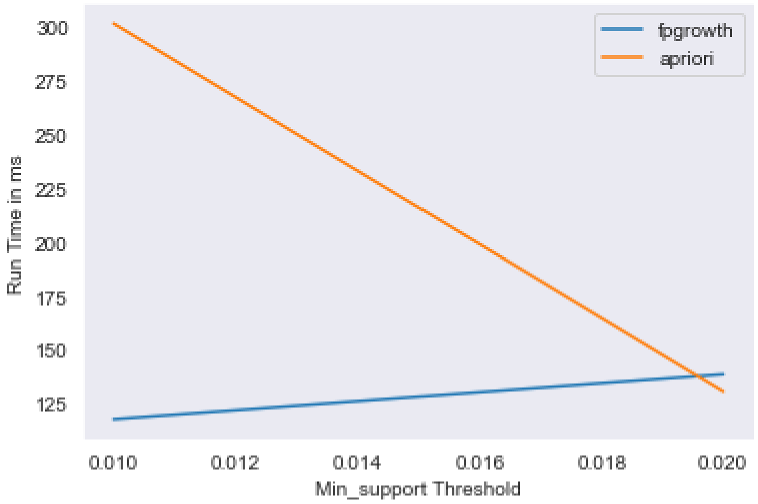
Comparative Study: Apriori Vs FP-Growth:

Apriori scans the dataset in each of its steps, so it becomes time-consuming for data where the number of items is larger.

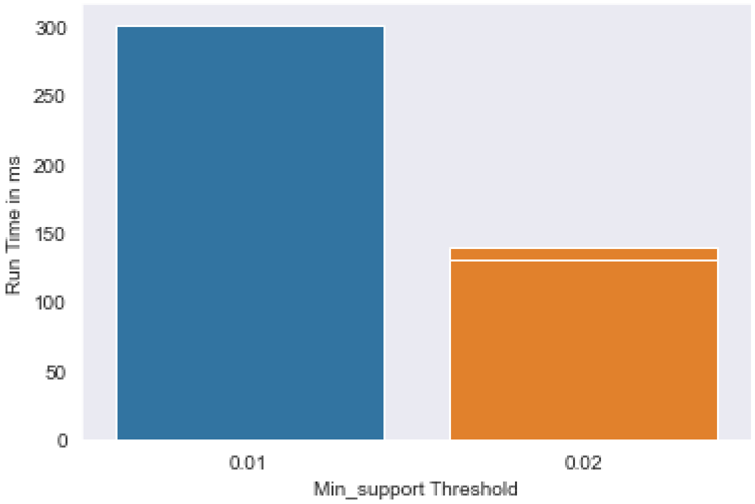
FP-Growth requires only one scan of the dataset in its beginning steps so it consumes less time.

Run Time Comparision of both the algorithm:

Line plot for Apriori Vs FP-Growth



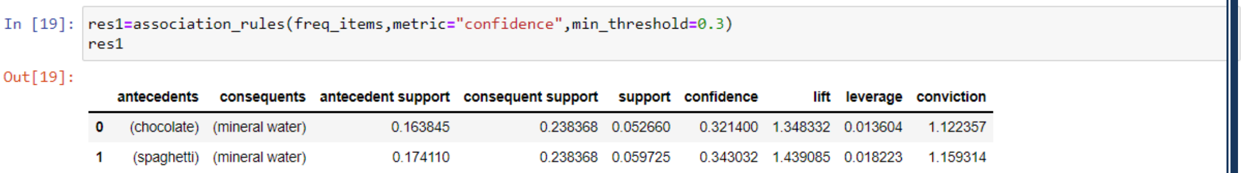
Bar plot for Apriori Vs FP-Growth



As we can see clearly in the diagram that the FP growth algorithm takes much lesser time than Apriori.

Results: Association Rules are generated for Apriori and Fpgrowth algorithm for antecedent and consequent items in our dataset.

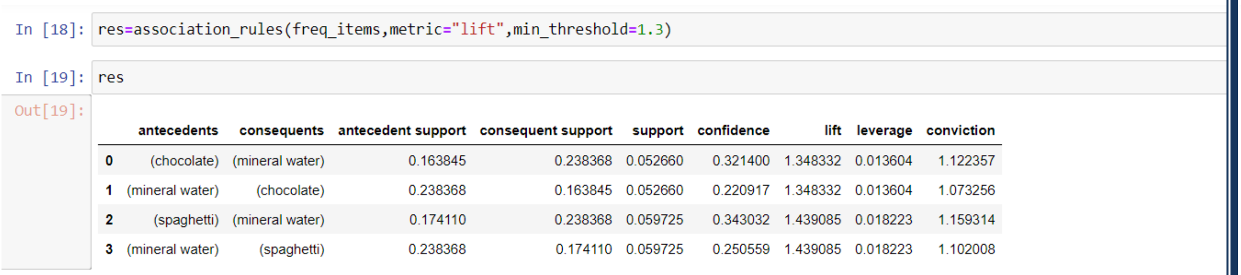
Association rule of the freq_items where min_support= 0.05 and min_confidence =0.3 :



```
In [19]: res1=association_rules(freq_items,metric="confidence",min_threshold=0.3)
         res1
Out[19]:
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(chocolate)	(mineral water)	0.163845	0.238368	0.052660	0.321400	1.348332	0.013604	1.122357
1	(spaghetti)	(mineral water)	0.174110	0.238368	0.059725	0.343032	1.439085	0.018223	1.159314

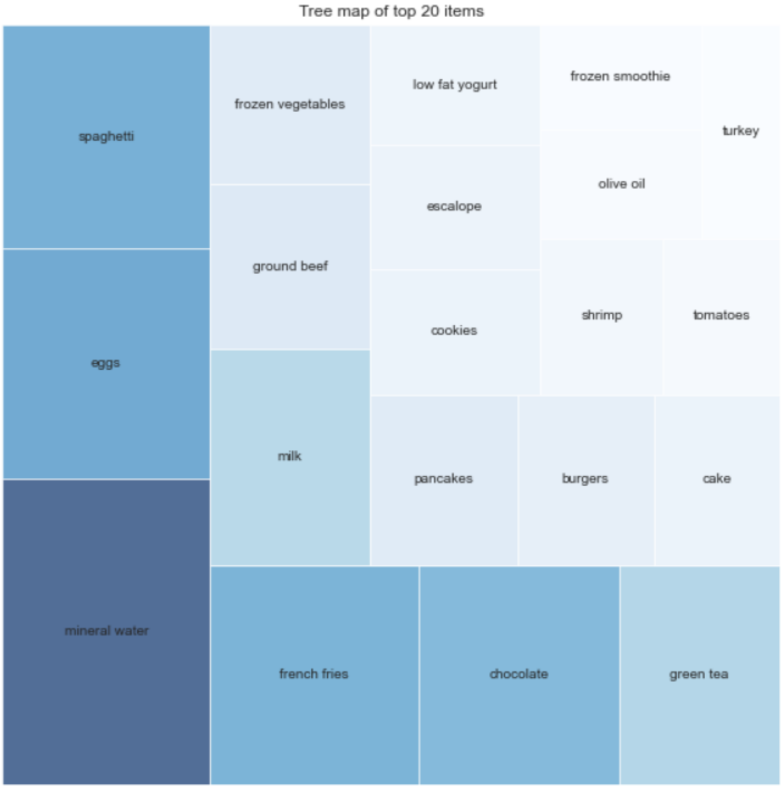
Association rule of the freq_items where min_support= 0.05 and min_threshold for lift =1.3 :



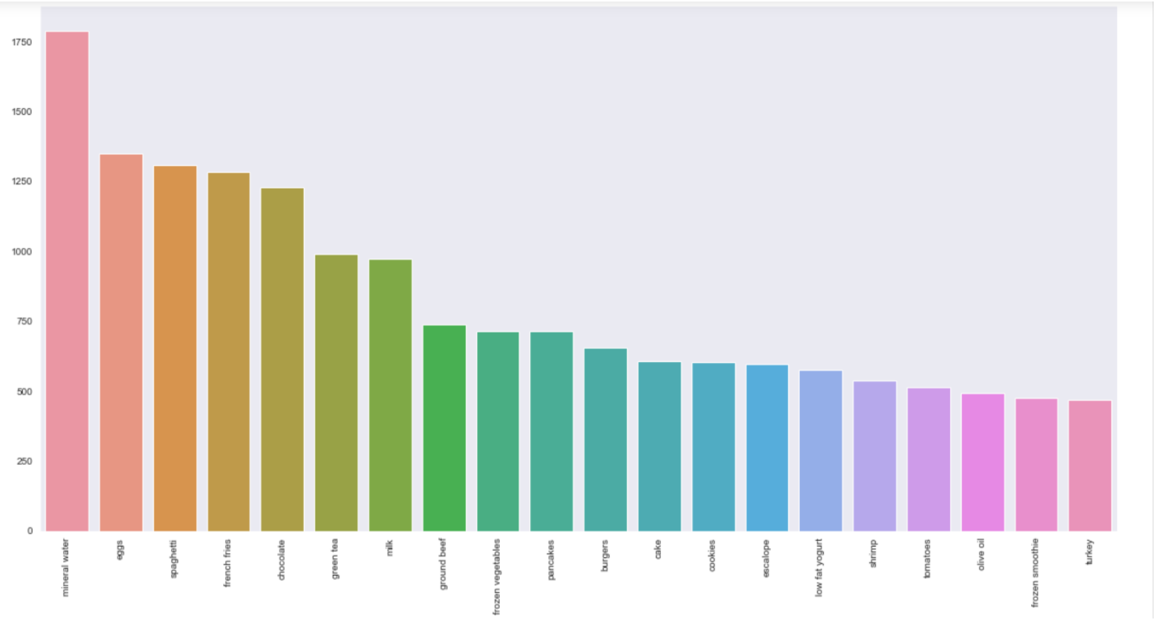
```
In [18]: res=association_rules(freq_items,metric="lift",min_threshold=1.3)

In [19]: res

Out[19]:
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(chocolate)	(mineral water)	0.163845	0.238368	0.052660	0.321400	1.348332	0.013604	1.122357
1	(mineral water)	(chocolate)	0.238368	0.163845	0.052660	0.220917	1.348332	0.013604	1.073256
2	(spaghetti)	(mineral water)	0.174110	0.238368	0.059725	0.343032	1.439085	0.018223	1.159314
3	(mineral water)	(spaghetti)	0.238368	0.174110	0.059725	0.250559	1.439085	0.018223	1.102008

Tree Map of the 20 most selling items:



Bar Plot of 20 top selling items in our dataset:



VIII. CONCLUSION AND FUTURE WORK

Conclusion: From the output above, we see that the top associations are not surprising, with one item for a specific purpose gets purchased along with another flavour required for the same purpose: E.g.: - Bread and Butter are strongly associated as they are consumed together. Similarly, Toothbrush and Toothpaste, Chocolate and Chips, Soap and Shampoo also have strong associations as they are frequently purchased together and used for similar purposes.

Future Work: In the future, once common application of association rules mining is in the domain of recommender systems and item pairs have been identified as having positive relationship, recommendations can be made to customers in order to increase sales. Also, there is a possibility of introducing customers to items they never would have tried before.

IX. REFERENCES

1. Dubey, Shish Kumar Mittal, Sonu Chattani, Seema Shukla, Vinod Kumar, *Comparative Analysis of Market Basket Analysis through Data Mining Techniques.*
2. P. Pravallika and K. Narendra, *Analysis on Medical Data sets using Apriori Algorithm Based on Association Rules.*
3. Lele, Sachin,Maheshkar, Snehal, *Online Shopping: Do Men Behave Differently than Women?*
4. Chi-Jung Tai Mohamed El-Shazly Yi-Hong Tsai Dezső Csupor Judit Hohmann Yang-Chang Wu Tzyy-Guey Tseng, Fang-Rong Chang Hui-Chun Wang, *Uncovering Modern Clinical Applications of Fuzi and Fuzi-Based Formulas: A Nationwide Descriptive Study With Market Basket Analysis.*
5. Wai Tung Ho Fu Wing Yu, *Chiller system performance management with market basket analysis.*
6. Garcia-Diaz Maria-Elena Marcos Martinez Bel´en Escobar , Diego P. Pinto-Roa, *Market basket analysis with association rules in the retail sector using Orange. Case Study: Appliances Sales Company.*
7. Purvika Bajaj, Renesa Ray, Shivani Shedge, Prof.Dr.Nikhilkumar Shardoor, *Sales Prediction System using Machine Learning.*
8. Hu Xin, Yang Yanfei, Chen Lanhua, Zhu Siru, *Research on a Prediction Model of Online Shopping Behavior Based on Deep Forest Algorithm.*
9. BOCEVA, Brankica, KISELICKI, Martin, *Online buying habits in the republic of north Macedonia.*
10. Jianwei Hou1, Online Shopping Patronage: *Do Demographics and Psychographics Really Matter.*
11. A.A. Raorane, R.V. Kulkarni and B.D. Jitkar, *Market Basket Analysis using association rule mining.*
12. M. Kavitha and M.S.T.T. Selvi, *Performance comparison of Apriori and FP-Growth algorithms in generating association rules.*
13. Vedhanayagam Priya S. Subha Balusamy Balamurugan, *Analysis of performance measures to handle medical E-commerce shopping cart abandonment in cloud.*

14. Hossain, Maliha Sattar, A H M Sarowar Paul, Mahit Kumar, *Market Basket Analysis Using Apriori and FP Growth Algorithm.*
15. Zhou, Zimu Shangguan, Longfei Zheng, Xiaolong Yang, Lei Liu, Yunhao, *Design and Implementation of an RFID-Based Customer Shopping Behavior Mining System.*
16. Norulhidayah Isa, Nur Syuhada Mohd Yusof, Muhammad Atif Ramlan, *The Implementation of Data Mining Techniques for Sales Analysis using Daily Sales Data.*
17. Dr. Murli Dhar Panga, Mr. Arpan Shrivastava and Mr. Akhilesh Dubey, *A STUDY ON CUSTOMER ATTITUDE TOWARDS ONLINE SHOPPING AND SHOPPING CART ABANDONMENT IN INDORE REGION.*
18. A. Trnka, *Market basket analysis with Six sigma methodology improvement.*
19. F. Alfiah, B. W. Pandhito, A. T. Sunarni, D. Muharam, and P. R. Matusin, *Data Mining Systems to Determine Sales Trends and Quantity Forecast using Association Rule.*
20. Gopalakrishnan T, Ritesh Choudhary and Sarada Prasad, *Prediction of Sales Value in online shopping using Linear Regression.*
21. M.R. Purnomo, B.I. Mulkan, *An Approach to Improve Layout Store and Product Placement.*
22. Lina Zhou, Liwei Dai, Dongsong Zhang, *ONLINE SHOPPING ACCEPTANCE MODEL — A CRITICAL SURVEY OF CONSUMER FACTORS IN ONLINE SHOPPING.*
23. Zhi-Ping Fan, Yu-jie Che, Zhen-Yu chen, *Product sales forecasting using online reviews and historical sales data: A method combining the Bass model and sentiment analysis.*
24. H. Aguinis, L.E. Forcum, H. Joo, *Market Basket Analysis in Management Research.*
25. Li Jingjing, Abbasi Ahmed, Cheema Amar, Abraham Linda B, *Path to Purpose? How Online Customer Journeys Differ for Hedonic Versus Utilitarian Purchases.*
26. R. Gangurde, D. B. Kumar, and D. S. D. Gore, *Building Prediction Model using Market Basket Analysis.*
27. Chuleeporn Changchit, Texas A&M University - Corpus Christi, *CONSUMER PERCEPTIONS OF ONLINE SHOPPING.*
28. Manpreet Kaur, Shivani kang, *Market Basket Analysis: Identify the changing trends of market data using association rule mining.*
29. M. Berry, G. Linoff, *Market Basket Analysis using Association Rule Mining.*

30. Mohit Gurnani, Yogesh Korke, Prachi Shah, Sandeep Udmale, Vijay Sambhe, *Forecasting of sales by using fusion of Machine Learning techniques.*

Appendix

```
In [1]:
import pandas as pd
import numpy as np
%matplotlib inline
import matplotlib.pyplot as plt
import seaborn as sns
sns.set_style("dark")
import matplotlib
from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules
from mlxtend.preprocessing import TransactionEncoder
from IPython.core.interactiveshell import InteractiveShell
InteractiveShell.ast_node_interactivity = "all"
In [2]:
data = pd.read_csv('D:\VIT ALL FILES\Fall Sem 2021-22 (5th Sem)\DMT Project\datasets/ecommerce_data.csv',header=None)
In [3]:
data
In [4]:
data.shape
In [5]:
data.head()
data.tail()
In [6]:
trans=[]
for i in range(len(data)):
  trans.append([str(data.values[i,j]) for j in range(0,20)])
trans=np.array(trans)
print(trans.shape)
```

In [7]:

```
trans
```

In [8]:

```
t=TransactionEncoder()
data=t.fit_transform(trans)
data=pd.DataFrame(data,columns=t.columns_,dtype=int)
data.shape
```

In [9]:

```
data.drop('nan',axis=1,inplace=True)
```

In [10]:

```
data.shape
'nan' in data.columns
```

In [11]:

```
data.head()
```

In [12]:

```
r=data.sum(axis=0).sort_values(ascending=False)[:20]
plt.figure(figsize=(20,10))
s=sns.barplot(x=r.index,y=r.values)
s.set_xticklabels(s.get_xticklabels(), rotation=90)
```

In [13]:

```
import squarify
```

In [14]:

```
my_values=r.values
cmap = matplotlib.cm.Blues
mini=min(my_values)
maxi=max(my_values)
norm = matplotlib.colors.Normalize(vmin=mini, vmax=maxi)
colors = [cmap(norm(value)) for value in my_values]
plt.figure(figsize=(10,10))
squarify.plot(sizes=r.values, label=r.index, alpha=.7,color=colors)
plt.title("Tree map of top 20 items")
plt.axis('off')
```

In [15]:

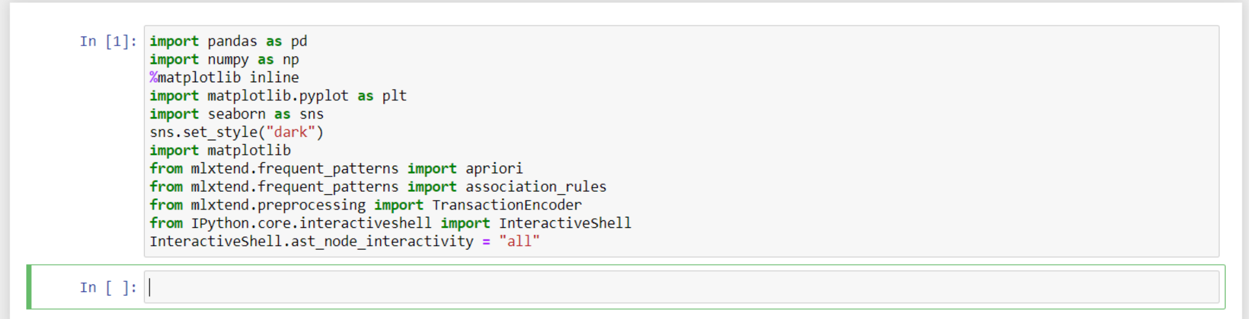
```
freq_items=apriori(data,min_support=0.05,use_colnames=True)
```

```
In [16]:
freq_items
In [17]:
res=association_rules(freq_items,metric="lift",min_threshold=1.3)
In[18]:
res
In [19]:
res1=association_rules(freq_items,metric="confidence",min_threshold=0.3)
res1
In [20]:
res2 = res[['antecedents','consequents','support','confidence','lift']]
res2
In [21]:
res3= res[res['confidence']>=0.1]
res3
In [22]:
frequent_itemsets = apriori(data, min_support = 0.05, use_colnames=True)
frequent_itemsets['length'] = frequent_itemsets['itemsets'].apply(lambda x: len(x))
frequent_itemsets
In [23]:
frequent_itemsets[ (frequent_itemsets['length'] == 2) &
(frequent_itemsets['support'] >= 0.05) ]
In [24]:
frequent_itemsets[ (frequent_itemsets['length'] == 1) &
(frequent_itemsets['support'] >= 0.05) ]
In [25]:
from mlxtend.frequent_patterns import fpgrowth

In [26]:
freq_items=fpgrowth(data,min_support=0.05,use_colnames=True)
In [27]:
freq_items
In [28]:
res=association_rules(freq_items,metric="lift",min_threshold=1)
```

```
res
In [29]:
res1 = association_rules(freq_items,metric="confidence",min_threshold=0.3)
res1
In[30]:
import time
l=[0.01,0.02]
t=[]
for i in l:
    t1=time.time()
    apriori(data,min_support=i,use_colnames=True)
    t2=time.time()
    t.append((t2-t1)*1000)
In[31]:
l=[0.01,0.02]
f=[]
for i in l:
    t1=time.time()
    fpgrowth(data,min_support=i,use_colnames=True)
    t2=time.time()
    f.append((t2-t1)*1000)
In[32]:
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")
In[33]:
sns.barplot(x=l,y=f,label="fpgrowth")
sns.barplot(x=l,y=t,label="apriori")
plt.xlabel("Min_support Threshold")
plt.ylabel("Run Time in ms")
```

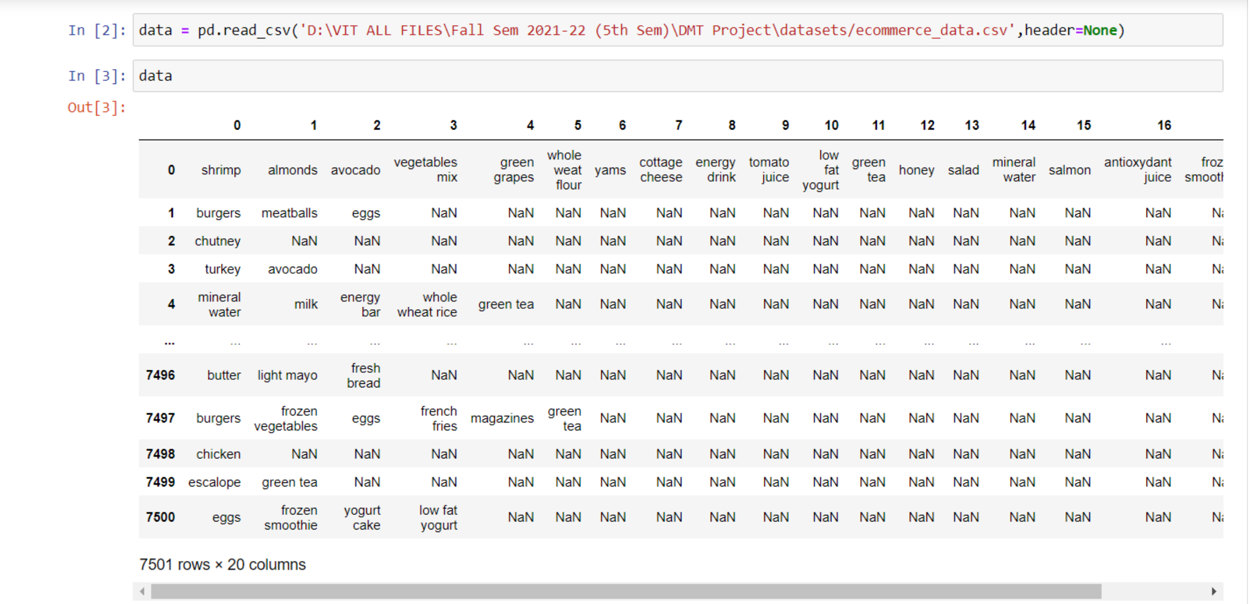
Code Output Screenshots:

Importing Libraries:



```
In [1]: import pandas as pd
        import numpy as np
        %matplotlib inline
        import matplotlib.pyplot as plt
        import seaborn as sns
        sns.set_style("dark")
        import matplotlib
        from mlxtend.frequent_patterns import apriori
        from mlxtend.frequent_patterns import association_rules
        from mlxtend.preprocessing import TransactionEncoder
        from IPython.core.interactiveshell import InteractiveShell
        InteractiveShell.ast_node_interactivity = "all"

In [ ]: |
```

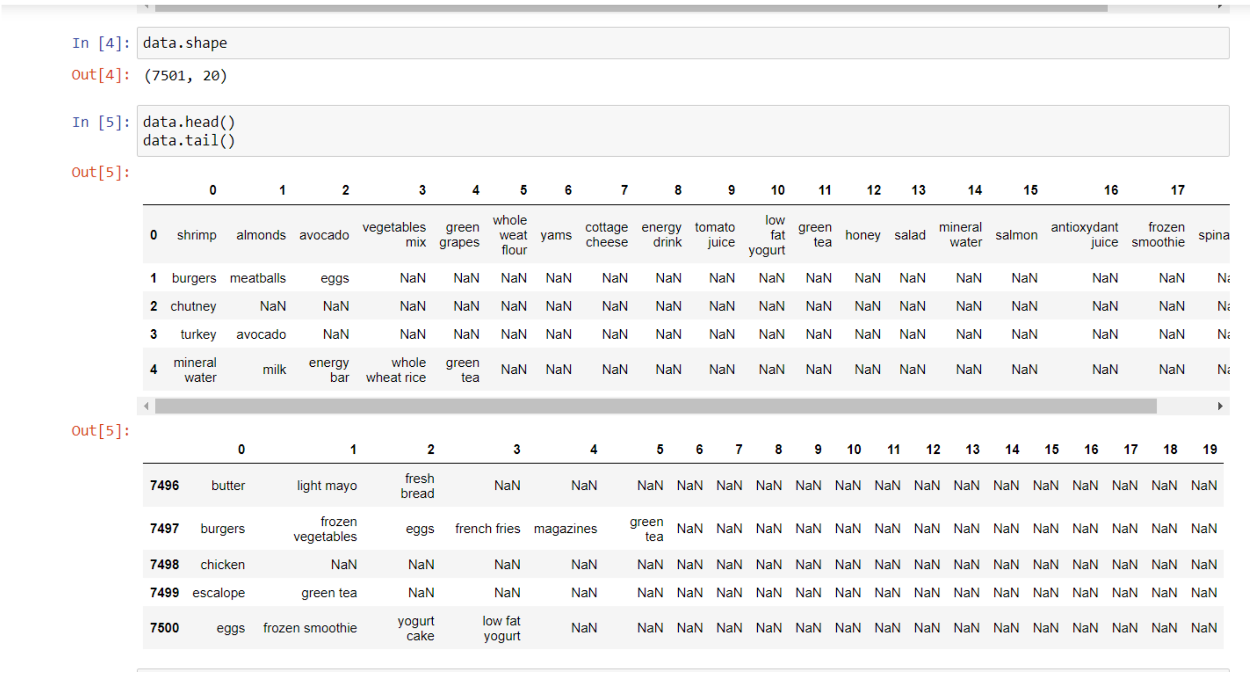
Reading Data Sets:



```
In [2]: data = pd.read_csv('D:\VIT ALL FILES\Fall Sem 2021-22 (5th Sem)\DMT Project\datasets/ecommerce_data.csv',header=None)

In [3]: data
Out[3]:
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
0	shrimp	almonds	avocado	vegetables mix	green grapes	whole weat flour	yams	cottage cheese	energy drink	tomato juice	low fat yogurt	green tea	honey	salad	mineral water	salmon	antioxydant juice	froz smoot
1	burgers	meatballs	eggs	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
2	chutney	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
3	turkey	avocado	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
4	mineral water	milk	energy bar	whole wheat rice	green tea	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	...	
7496	butter	light mayo	fresh bread	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
7497	burgers	frozen vegetables	eggs	french fries	magazines	green tea	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
7498	chicken	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
7499	escalope	green tea	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
7500	eggs	frozen smoothie	yogurt cake	low fat yogurt	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na

7501 rows × 20 columns



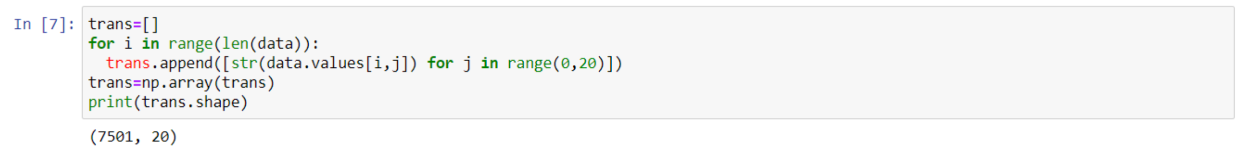
```
In [4]: data.shape
Out[4]: (7501, 20)

In [5]: data.head()
        data.tail()
Out[5]:
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	
0	shrimp	almonds	avocado	vegetables mix	green grapes	whole weat flour	yams	cottage cheese	energy drink	tomato juice	low fat yogurt	green tea	honey	salad	mineral water	salmon	antioxydant juice	frozen smoothie	spina
1	burgers	meatballs	eggs	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
2	chutney	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
3	turkey	avocado	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na
4	mineral water	milk	energy bar	whole wheat rice	green tea	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	Na

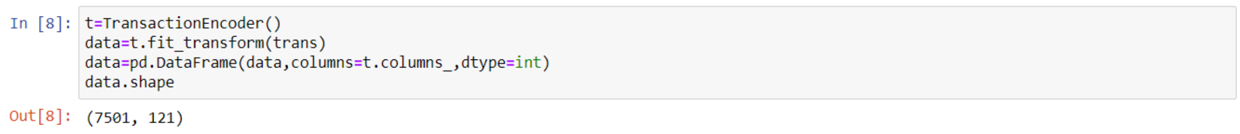
```
Out[5]:
```

	0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19
7496	butter	light mayo	fresh bread	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
7497	burgers	frozen vegetables	eggs	french fries	magazines	green tea	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
7498	chicken	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
7499	escalope	green tea	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN
7500	eggs	frozen smoothie	yogurt cake	low fat yogurt	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN	NaN

Append the dataset values into an array:

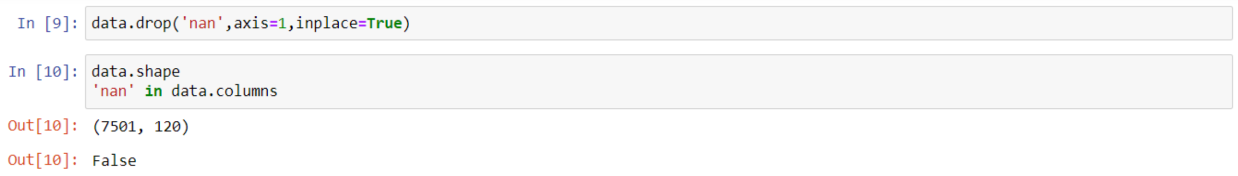


```
In [7]: trans=[]
        for i in range(len(data)):
          trans.append([str(data.values[i,j]) for j in range(0,20)])
        trans=np.array(trans)
        print(trans.shape)

        (7501, 20)
```

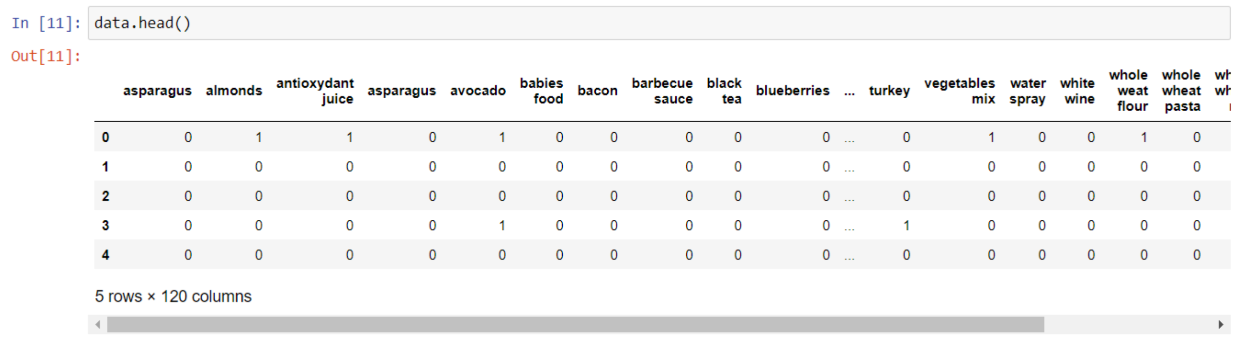


```
In [8]: t=TransactionEncoder()
        data=t.fit_transform(trans)
        data=pd.DataFrame(data,columns=t.columns_,dtype=int)
        data.shape
Out[8]: (7501, 121)
```

Drop the 'nan' values:



```
In [9]: data.drop('nan',axis=1,inplace=True)

In [10]: data.shape
         'nan' in data.columns
Out[10]: (7501, 120)
Out[10]: False
```

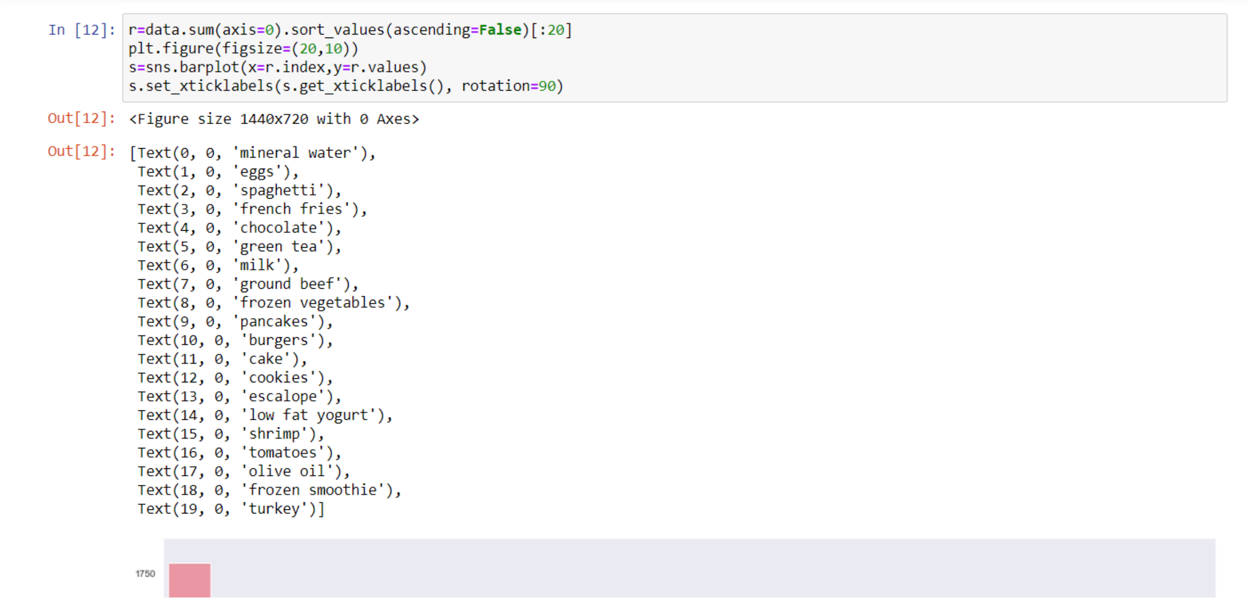


```
In [11]: data.head()
Out[11]:
```

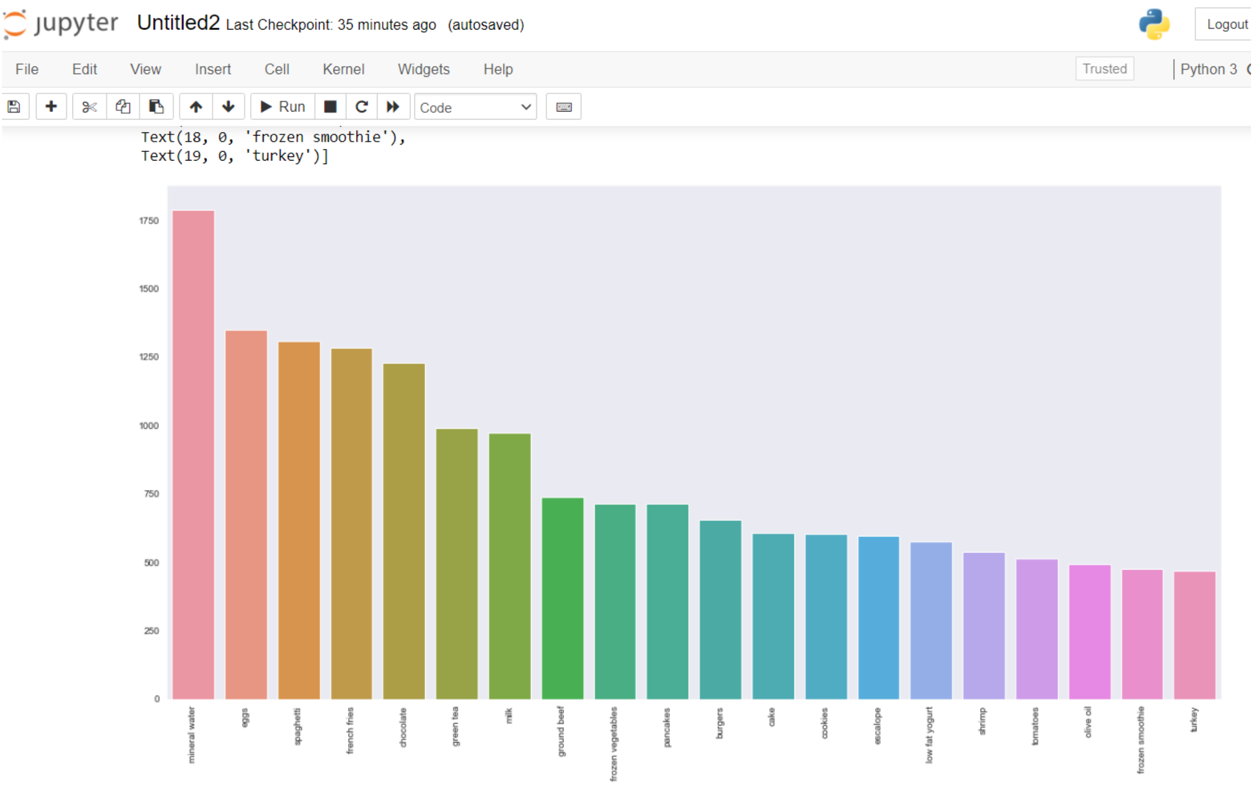
	asparagus	almonds	antioxydant juice	asparagus	avocado	babies food	bacon	barbecue sauce	black tea	blueberries	...	turkey	vegetables mix	water spray	white wine	whole weat flour	whole wheat pasta	wh wh
0	0	1	1	0	1	0	0	0	0	0	...	0	1	0	0	1	0	
1	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	
2	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	
3	0	0	0	0	1	0	0	0	0	0	...	1	0	0	0	0	0	
4	0	0	0	0	0	0	0	0	0	0	...	0	0	0	0	0	0	

5 rows × 120 columns

Exploratory Data Analysis (EDA) of 20 top selling items:

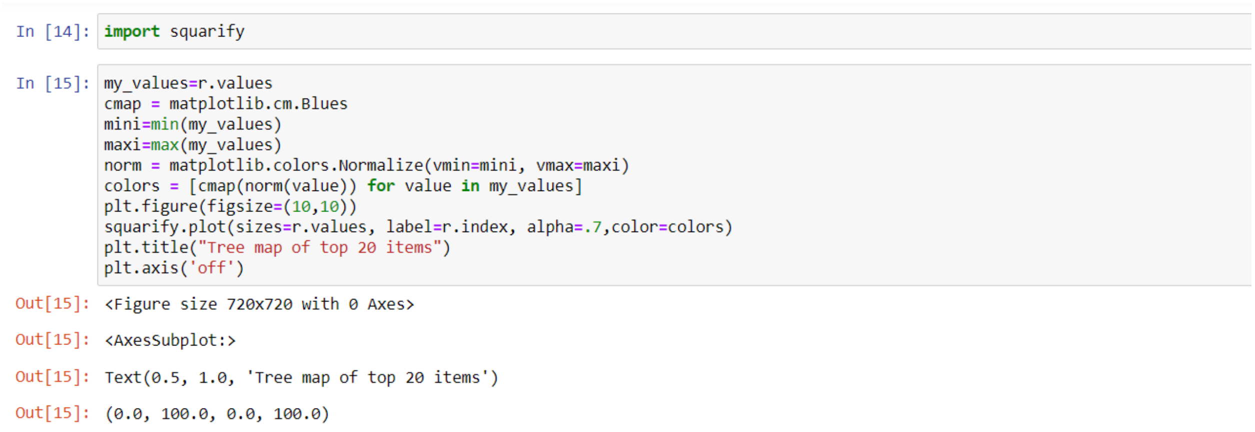


```
In [12]: r=data.sum(axis=0).sort_values(ascending=False)[:20]
         plt.figure(figsize=(20,10))
         s=sns.barplot(x=r.index,y=r.values)
         s.set_xticklabels(s.get_xticklabels(), rotation=90)

Out[12]: <Figure size 1440x720 with 0 Axes>

Out[12]: [Text(0, 0, 'mineral water'),
          Text(1, 0, 'eggs'),
          Text(2, 0, 'spaghetti'),
          Text(3, 0, 'french fries'),
          Text(4, 0, 'chocolate'),
          Text(5, 0, 'green tea'),
          Text(6, 0, 'milk'),
          Text(7, 0, 'ground beef'),
          Text(8, 0, 'frozen vegetables'),
          Text(9, 0, 'pancakes'),
          Text(10, 0, 'burgers'),
          Text(11, 0, 'cake'),
          Text(12, 0, 'cookies'),
          Text(13, 0, 'escalope'),
          Text(14, 0, 'low fat yogurt'),
          Text(15, 0, 'shrimp'),
          Text(16, 0, 'tomatoes'),
          Text(17, 0, 'olive oil'),
          Text(18, 0, 'frozen smoothie'),
          Text(19, 0, 'turkey')]
```



```
          Text(18, 0, 'frozen smoothie'),
          Text(19, 0, 'turkey')]
```

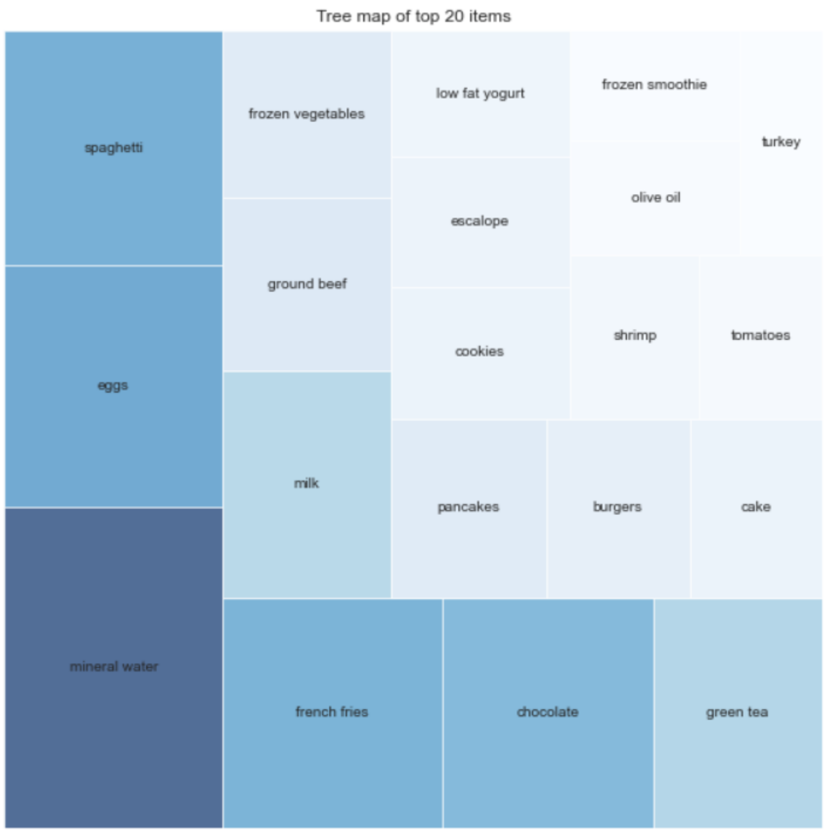
Tree Map of the 20 most selling items:



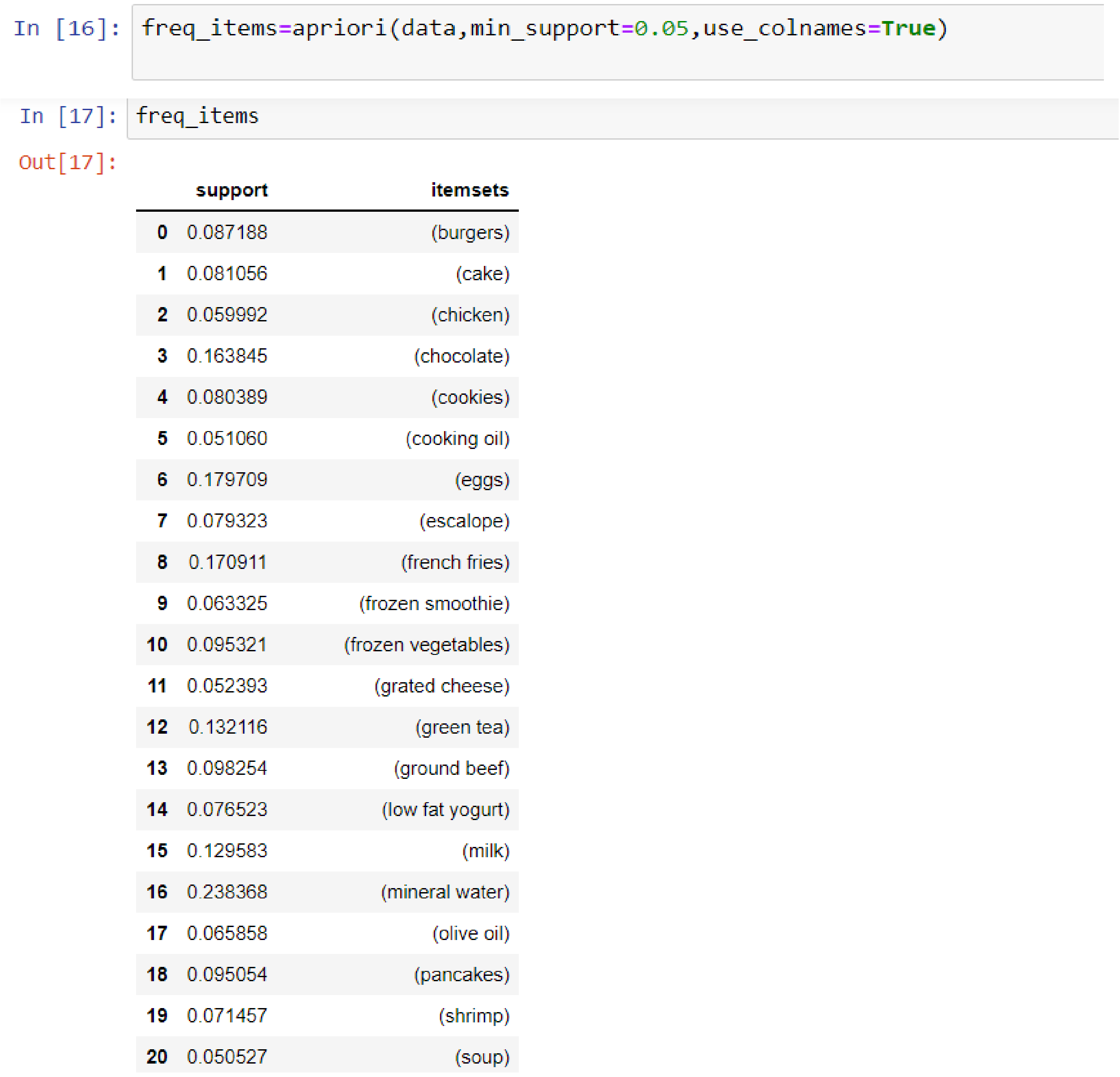
```
In [14]: import squarify
```

```
In [15]: my_values=r.values
         cmap = matplotlib.cm.Blues
         mini=min(my_values)
         maxi=max(my_values)
         norm = matplotlib.colors.Normalize(vmin=mini, vmax=maxi)
         colors = [cmap(norm(value)) for value in my_values]
         plt.figure(figsize=(10,10))
         squarify.plot(sizes=r.values, label=r.index, alpha=.7,color=colors)
         plt.title("Tree map of top 20 items")
         plt.axis('off')
```

```
Out[15]: <Figure size 720x720 with 0 Axes>

Out[15]: <AxesSubplot:>

Out[15]: Text(0.5, 1.0, 'Tree map of top 20 items')

Out[15]: (0.0, 100.0, 0.0, 100.0)
```



Making frequent items lists using Apriori where minimum support= 0.05:

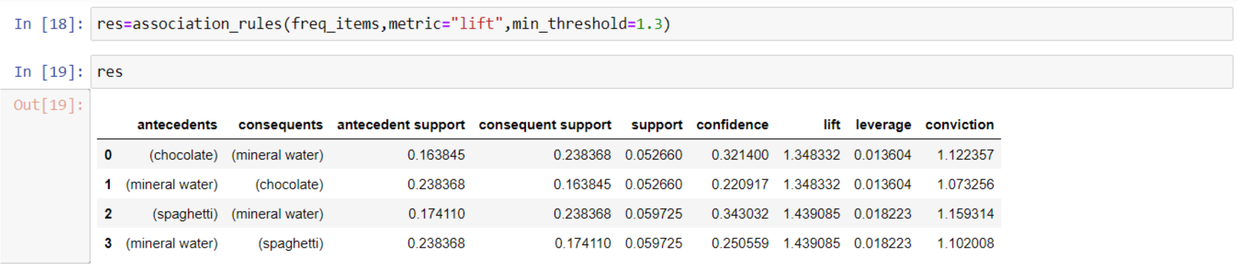


```
In [16]: freq_items=apriori(data,min_support=0.05,use_colnames=True)
```

```
In [17]: freq_items
Out[17]:
```

	support	itemsets
0	0.087188	(burgers)
1	0.081056	(cake)
2	0.059992	(chicken)
3	0.163845	(chocolate)
4	0.080389	(cookies)
5	0.051060	(cooking oil)
6	0.179709	(eggs)
7	0.079323	(escalope)
8	0.170911	(french fries)
9	0.063325	(frozen smoothie)
10	0.095321	(frozen vegetables)
11	0.052393	(grated cheese)
12	0.132116	(green tea)
13	0.098254	(ground beef)
14	0.076523	(low fat yogurt)
15	0.129583	(milk)
16	0.238368	(mineral water)
17	0.065858	(olive oil)
18	0.095054	(pancakes)
19	0.071457	(shrimp)
20	0.050527	(soup)

Association rule1 of the freq_items where min_threshold for lift =1.3 :

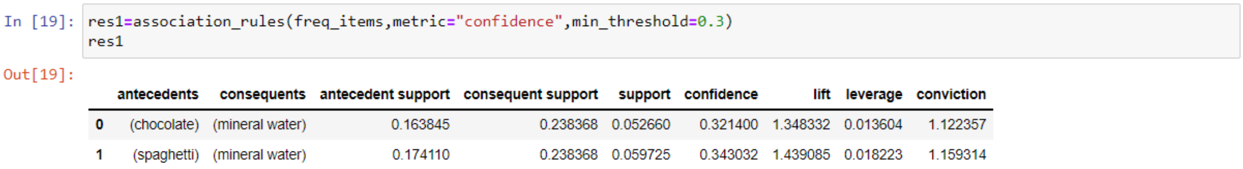


```
In [18]: res=association_rules(freq_items,metric="lift",min_threshold=1.3)
```

```
In [19]: res
Out[19]:
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(chocolate)	(mineral water)	0.163845	0.238368	0.052660	0.321400	1.348332	0.013604	1.122357
1	(mineral water)	(chocolate)	0.238368	0.163845	0.052660	0.220917	1.348332	0.013604	1.073256
2	(spaghetti)	(mineral water)	0.174110	0.238368	0.059725	0.343032	1.439085	0.018223	1.159314
3	(mineral water)	(spaghetti)	0.238368	0.174110	0.059725	0.250559	1.439085	0.018223	1.102008

Association rule2 of the freq_items where min_threshold for confidence =0.3 :

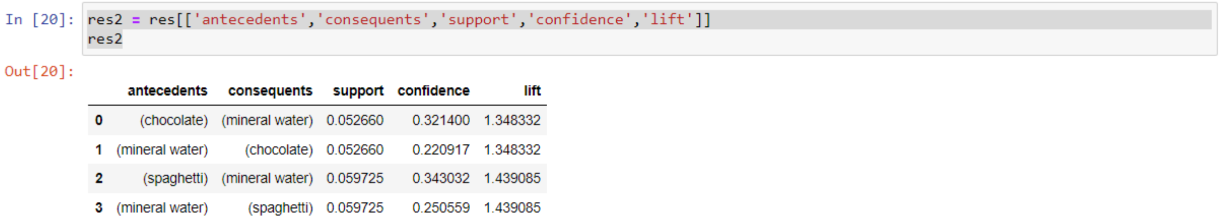


In [19]:
```
res1=association_rules(freq_items,metric="confidence",min_threshold=0.3)
res1
```

Out[19]:

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(chocolate)	(mineral water)	0.163845	0.238368	0.052660	0.321400	1.348332	0.013604	1.122357
1	(spaghetti)	(mineral water)	0.174110	0.238368	0.059725	0.343032	1.439085	0.018223	1.159314

Association rule3 of the freq_items where some selected attributes are displayed:

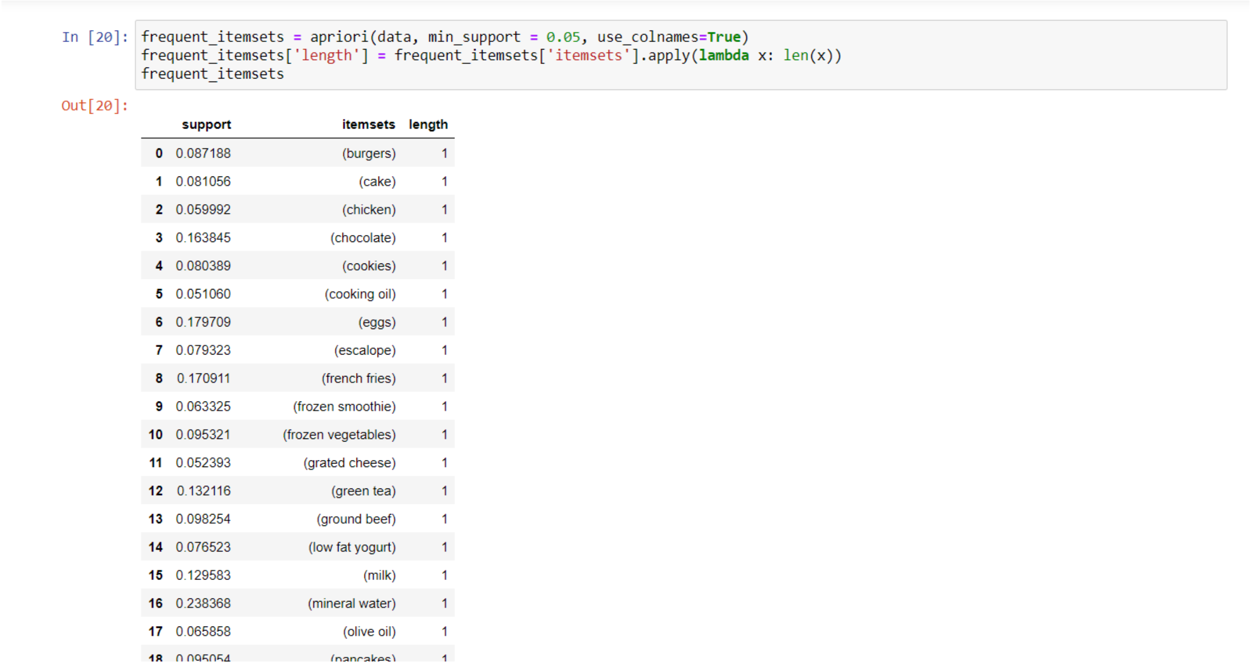


In [20]:
```
res2 = res[['antecedents','consequents','support','confidence','lift']]
res2
```

Out[20]:

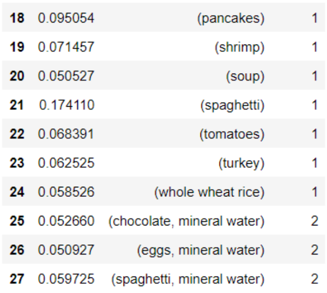
	antecedents	consequents	support	confidence	lift
0	(chocolate)	(mineral water)	0.052660	0.321400	1.348332
1	(mineral water)	(chocolate)	0.052660	0.220917	1.348332
2	(spaghetti)	(mineral water)	0.059725	0.343032	1.439085
3	(mineral water)	(spaghetti)	0.059725	0.250559	1.439085

Listing frequent_itemsets where minimum support = 0.05:



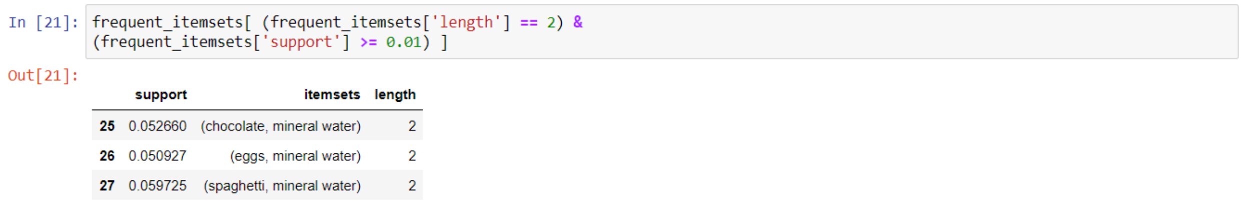
In [20]:
```
frequent_itemsets = apriori(data, min_support = 0.05, use_colnames=True)
frequent_itemsets['length'] = frequent_itemsets['itemsets'].apply(lambda x: len(x))
frequent_itemsets
```

Out[20]:



	support	itemsets	length
0	0.087188	(burgers)	1
1	0.081056	(cake)	1
2	0.059992	(chicken)	1
3	0.163845	(chocolate)	1
4	0.080389	(cookies)	1
5	0.051060	(cooking oil)	1
6	0.179709	(eggs)	1
7	0.079323	(escalope)	1
8	0.170911	(french fries)	1
9	0.063325	(frozen smoothie)	1
10	0.095321	(frozen vegetables)	1
11	0.052393	(grated cheese)	1
12	0.132116	(green tea)	1
13	0.098254	(ground beef)	1
14	0.076523	(low fat yogurt)	1
15	0.129583	(milk)	1
16	0.238368	(mineral water)	1
17	0.065858	(olive oil)	1
18	0.095054	(pancakes)	1
19	0.071457	(shrimp)	1
20	0.050527	(soup)	1
21	0.174110	(spaghetti)	1
22	0.068391	(tomatoes)	1
23	0.062525	(turkey)	1
24	0.058526	(whole wheat rice)	1
25	0.052660	(chocolate, mineral water)	2
26	0.050927	(eggs, mineral water)	2
27	0.059725	(spaghetti, mineral water)	2

Listing items where length of item sets are 2 and support is greater than 0.01 :

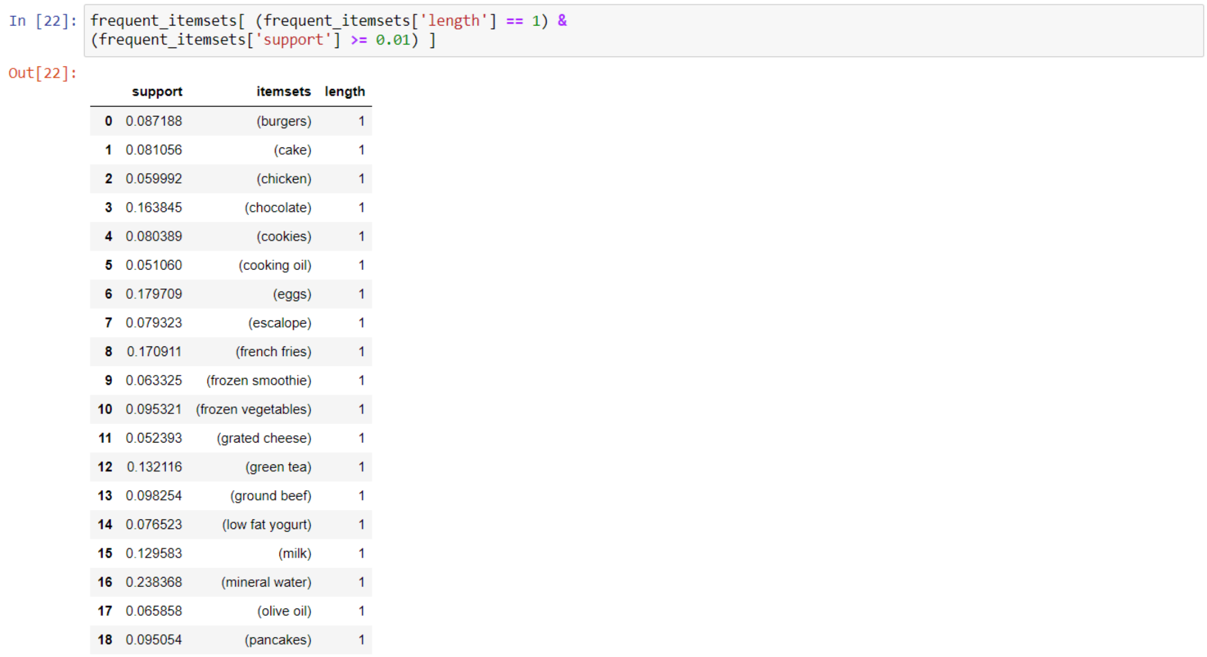


```
In [21]: frequent_itemsets[ (frequent_itemsets['length'] == 2) &
         (frequent_itemsets['support'] >= 0.01) ]
```

Out[21]:

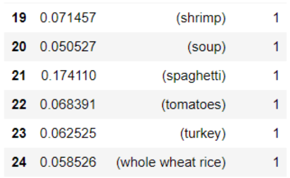
	support	itemsets	length
25	0.052660	(chocolate, mineral water)	2
26	0.050927	(eggs, mineral water)	2
27	0.059725	(spaghetti, mineral water)	2

Listing items where length of item set is 1 and support is greater than 0.01 :



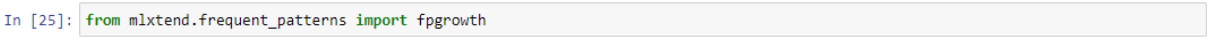
```
In [22]: frequent_itemsets[ (frequent_itemsets['length'] == 1) &
         (frequent_itemsets['support'] >= 0.01) ]
```

Out[22]:



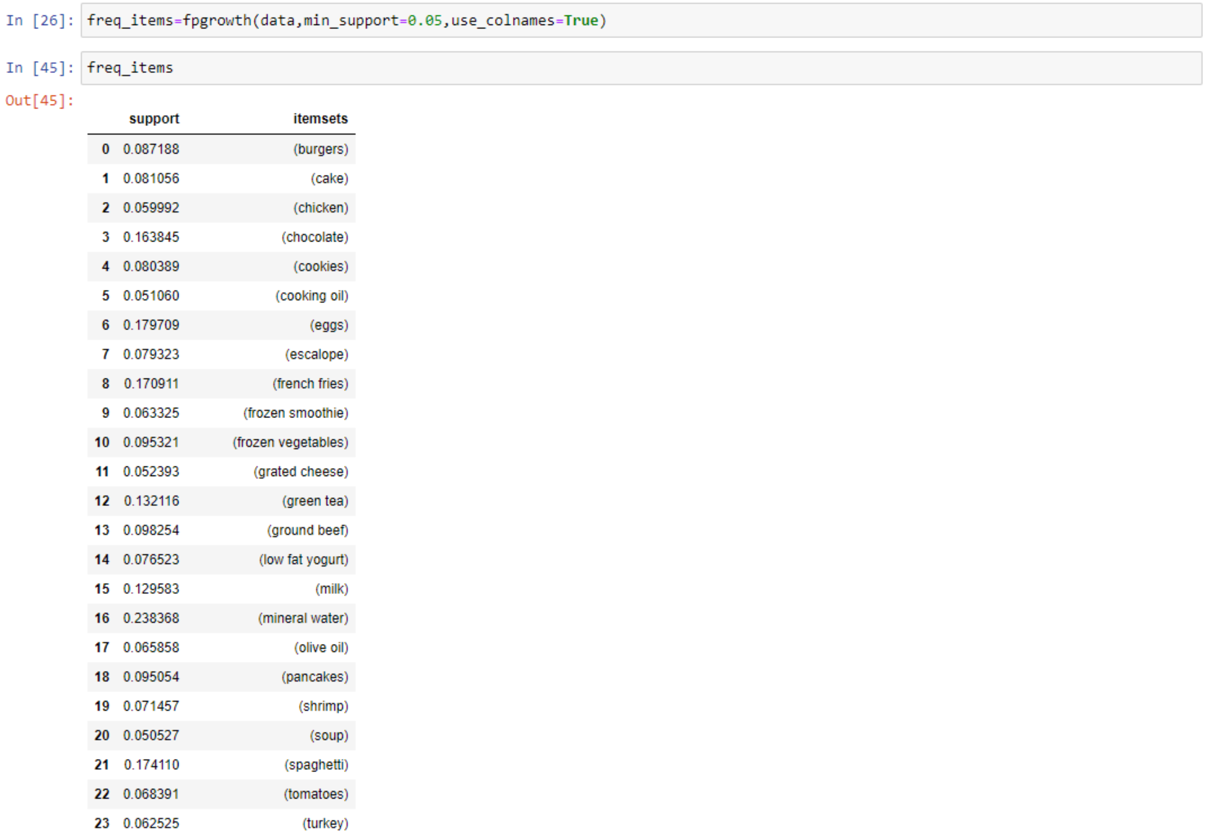
	support	itemsets	length
0	0.087188	(burgers)	1
1	0.081056	(cake)	1
2	0.059992	(chicken)	1
3	0.163845	(chocolate)	1
4	0.080389	(cookies)	1
5	0.051060	(cooking oil)	1
6	0.179709	(eggs)	1
7	0.079323	(escalope)	1
8	0.170911	(french fries)	1
9	0.063325	(frozen smoothie)	1
10	0.095321	(frozen vegetables)	1
11	0.052393	(grated cheese)	1
12	0.132116	(green tea)	1
13	0.098254	(ground beef)	1
14	0.076523	(low fat yogurt)	1
15	0.129583	(milk)	1
16	0.238368	(mineral water)	1
17	0.065858	(olive oil)	1
18	0.095054	(pancakes)	1
19	0.071457	(shrimp)	1
20	0.050527	(soup)	1
21	0.174110	(spaghetti)	1
22	0.068391	(tomatoes)	1
23	0.062525	(turkey)	1
24	0.058526	(whole wheat rice)	1

Importing fpgrowth from mlxtend.frequent_patterns:



```
In [25]: from mlxtend.frequent_patterns import fpgrowth
```

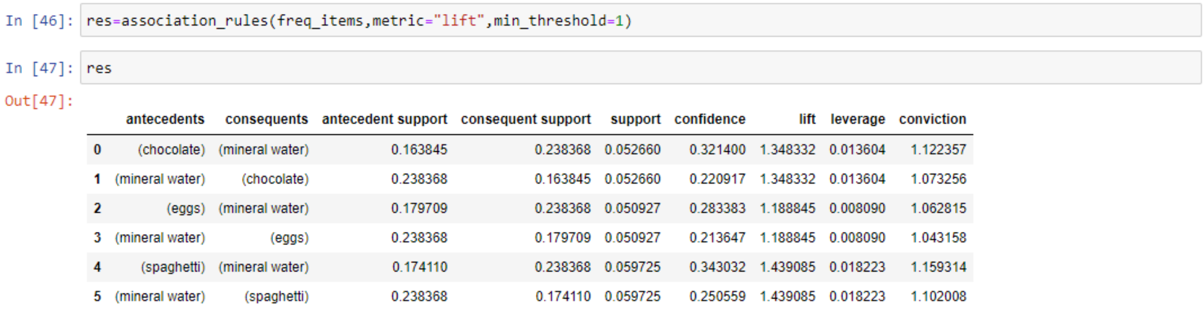
Making frequent item sets using fpgrowth where minimum support= 0.05:



```
In [26]: freq_items=fpgrowth(data,min_support=0.05,use_colnames=True)

In [45]: freq_items
Out[45]:
```

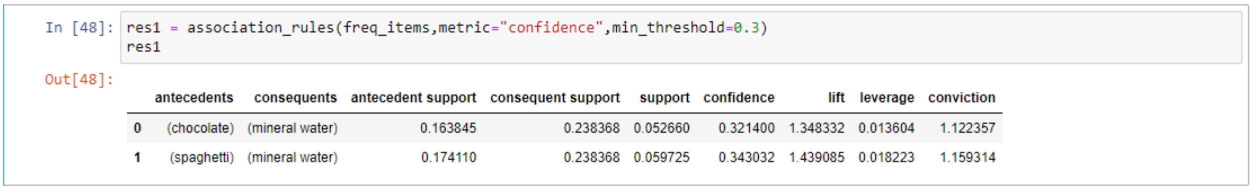
	support	itemsets
0	0.087188	(burgers)
1	0.081056	(cake)
2	0.059992	(chicken)
3	0.163845	(chocolate)
4	0.080389	(cookies)
5	0.051060	(cooking oil)
6	0.179709	(eggs)
7	0.079323	(escalope)
8	0.170911	(french fries)
9	0.063325	(frozen smoothie)
10	0.095321	(frozen vegetables)
11	0.052393	(grated cheese)
12	0.132116	(green tea)
13	0.098254	(ground beef)
14	0.076523	(low fat yogurt)
15	0.129583	(milk)
16	0.238368	(mineral water)
17	0.065858	(olive oil)
18	0.095054	(pancakes)
19	0.071457	(shrimp)
20	0.050527	(soup)
21	0.174110	(spaghetti)
22	0.068391	(tomatoes)
23	0.062525	(turkey)

Association rules of the freq_items where min_threshold for lift is 1 and confidence =0.3 :



```
In [46]: res=association_rules(freq_items,metric="lift",min_threshold=1)

In [47]: res
Out[47]:
```

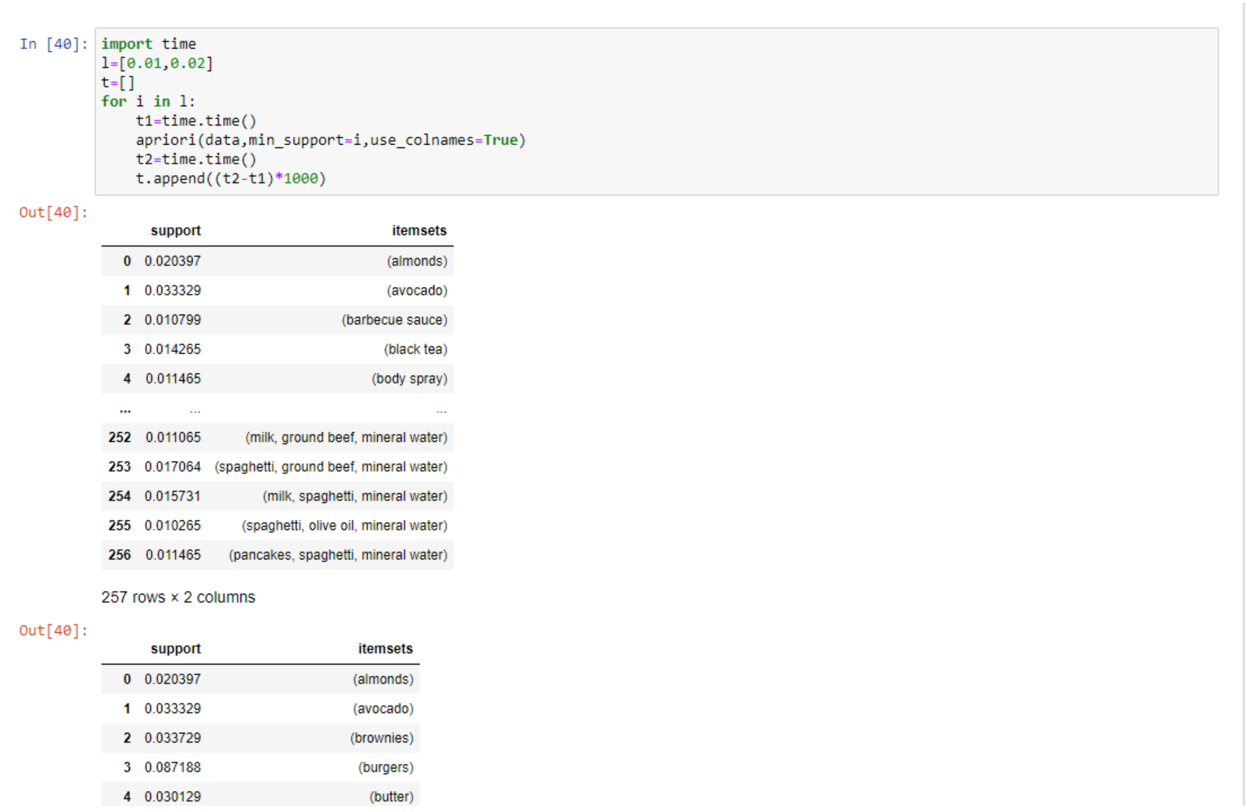
	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(chocolate)	(mineral water)	0.163845	0.238368	0.052660	0.321400	1.348332	0.013604	1.122357
1	(mineral water)	(chocolate)	0.238368	0.163845	0.052660	0.220917	1.348332	0.013604	1.073256
2	(eggs)	(mineral water)	0.179709	0.238368	0.050927	0.283383	1.188845	0.008090	1.062815
3	(mineral water)	(eggs)	0.238368	0.179709	0.050927	0.213647	1.188845	0.008090	1.043158
4	(spaghetti)	(mineral water)	0.174110	0.238368	0.059725	0.343032	1.439085	0.018223	1.159314
5	(mineral water)	(spaghetti)	0.238368	0.174110	0.059725	0.250559	1.439085	0.018223	1.102008



```
In [48]: res1 = association_rules(freq_items,metric="confidence",min_threshold=0.3)
         res1
Out[48]:
```

	antecedents	consequents	antecedent support	consequent support	support	confidence	lift	leverage	conviction
0	(chocolate)	(mineral water)	0.163845	0.238368	0.052660	0.321400	1.348332	0.013604	1.122357
1	(spaghetti)	(mineral water)	0.174110	0.238368	0.059725	0.343032	1.439085	0.018223	1.159314

Comparison of RunTime between Apriori and FPGrowth using Barplot and Lineplot:



```
In [40]: import time
         l=[0.01,0.02]
         t=[]
         for i in l:
             t1=time.time()
             apriori(data,min_support=i,use_colnames=True)
             t2=time.time()
             t.append((t2-t1)*1000)
```

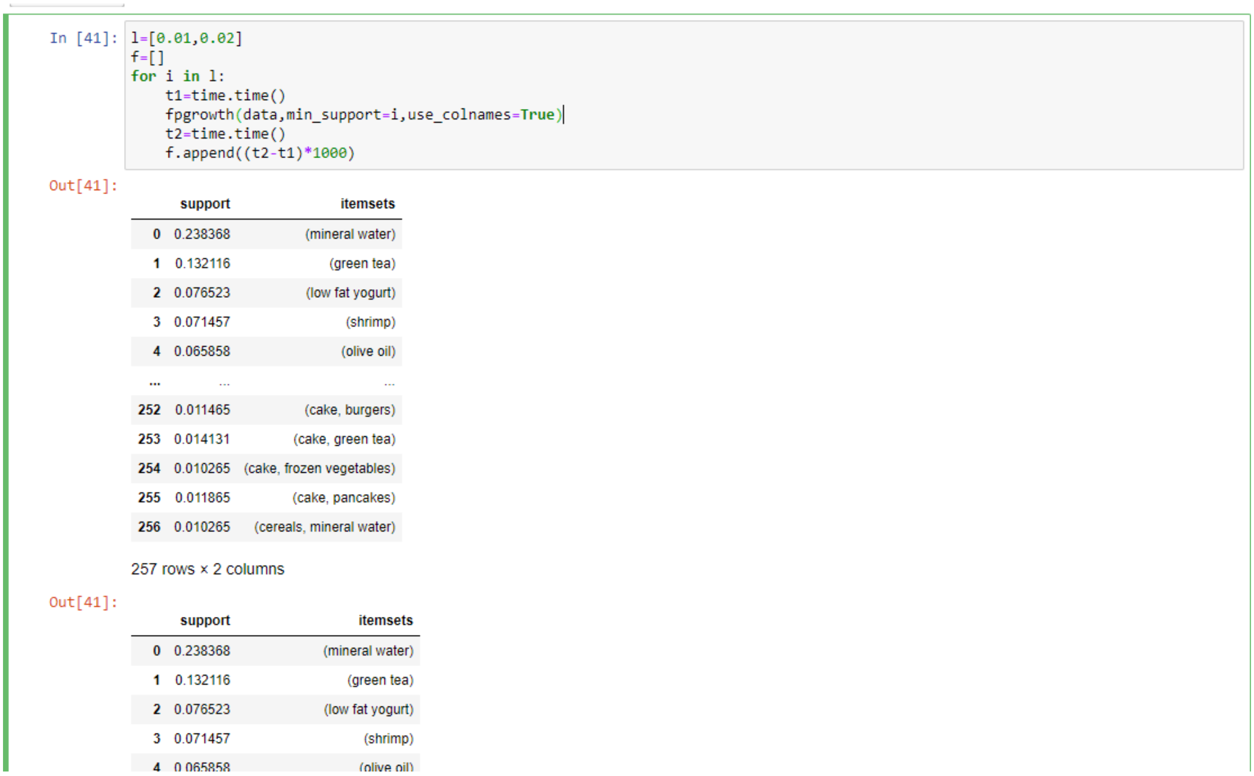
Out[40]:

	support	itemsets
0	0.020397	(almonds)
1	0.033329	(avocado)
2	0.010799	(barbecue sauce)
3	0.014265	(black tea)
4	0.011465	(body spray)
...	...	...
252	0.011065	(milk, ground beef, mineral water)
253	0.017064	(spaghetti, ground beef, mineral water)
254	0.015731	(milk, spaghetti, mineral water)
255	0.010265	(spaghetti, olive oil, mineral water)
256	0.011465	(pancakes, spaghetti, mineral water)

257 rows × 2 columns

Out[40]:

	support	itemsets
0	0.020397	(almonds)
1	0.033329	(avocado)
2	0.033729	(brownies)
3	0.087188	(burgers)
4	0.030129	(butter)



```
In [41]: l=[0.01,0.02]
         f=[]
         for i in l:
             t1=time.time()
             fpgrowth(data,min_support=i,use_colnames=True)
             t2=time.time()
             f.append((t2-t1)*1000)
```

Out[41]:

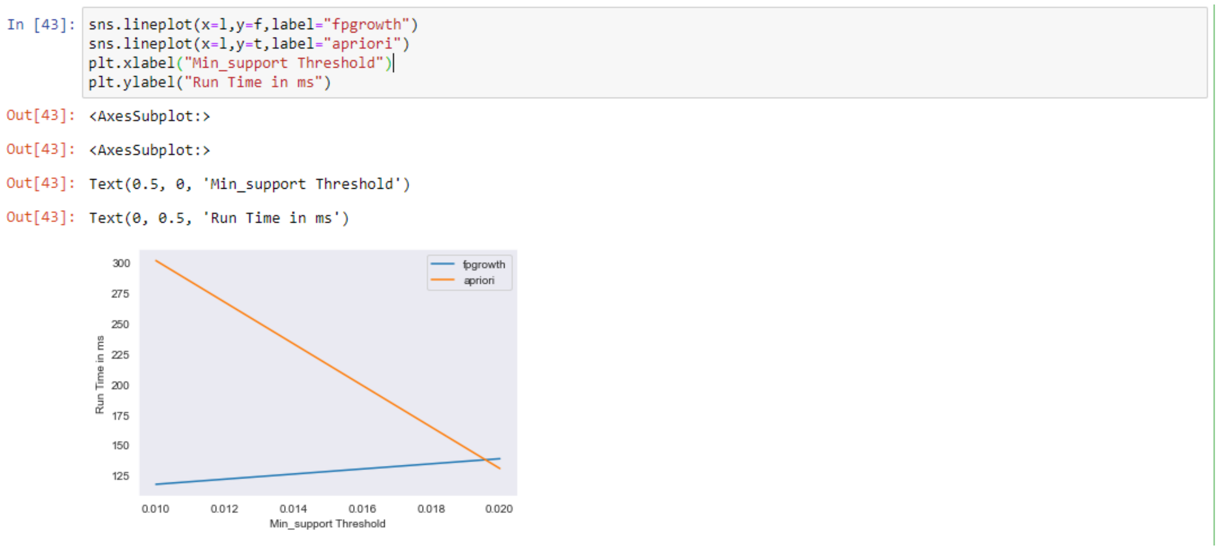
	support	itemsets
0	0.238368	(mineral water)
1	0.132116	(green tea)
2	0.076523	(low fat yogurt)
3	0.071457	(shrimp)
4	0.065858	(olive oil)
...	...	...
252	0.011465	(cake, burgers)
253	0.014131	(cake, green tea)
254	0.010265	(cake, frozen vegetables)
255	0.011865	(cake, pancakes)
256	0.010265	(cereals, mineral water)

257 rows × 2 columns

Out[41]:

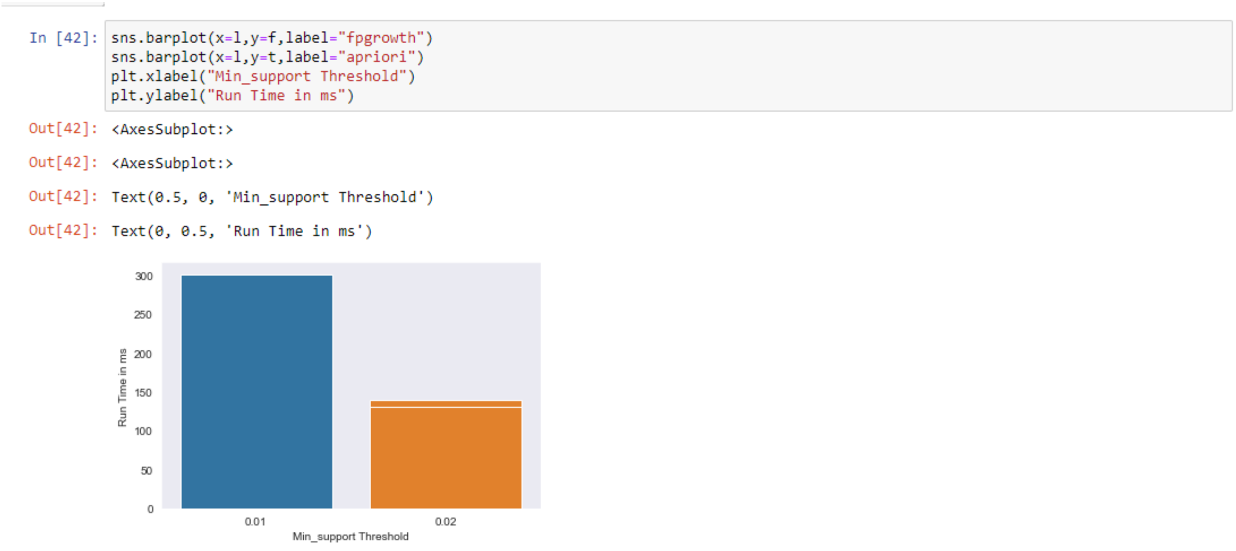
	support	itemsets
0	0.238368	(mineral water)
1	0.132116	(green tea)
2	0.076523	(low fat yogurt)
3	0.071457	(shrimp)
4	0.065858	(olive oil)

LinePlot:



```
In [43]: 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")

Out[43]: <AxesSubplot:>

Out[43]: <AxesSubplot:>

Out[43]: Text(0.5, 0, 'Min_support Threshold')

Out[43]: Text(0, 0.5, 'Run Time in ms')
```

Barplot:



```
In [42]: sns.barplot(x=l,y=f,label="fpgrowth")
         sns.barplot(x=l,y=t,label="apriori")
         plt.xlabel("Min_support Threshold")
         plt.ylabel("Run Time in ms")

Out[42]: <AxesSubplot:>

Out[42]: <AxesSubplot:>

Out[42]: Text(0.5, 0, 'Min_support Threshold')

Out[42]: Text(0, 0.5, 'Run Time in ms')
```