Market Basket Analysis

A report submitted for the course of

Application Development­\_Machine Learning Explore

III B. Tech I Semester

by

V.Sriman- 2011CS030164

K.Thanshith – 2011CS030177

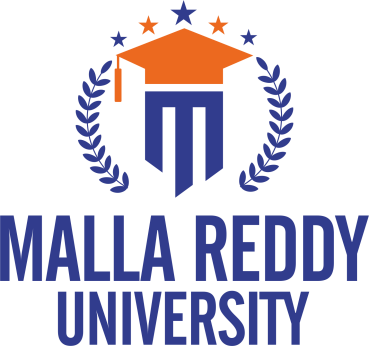
SK.Ansar Ali- 2011CS030148

*(Times New Roman, 12-point size, Bold, Centered)*

Under the esteemed guidance of

Ms.M.Shailaja

(Designation)



Department of Computer Science & Engineering (Data Science)

Malla Reddy Univeristy

Maisammaguda, Dulapally,

Hyderabad, Telangana 500100  
www.mallareddyuniversity.ac.in

2022-23

## DECLARATION

We certify that

1. The work contained in this report is original and has been done by me under the guidance of my supervisor(s).
2. The work has not been submitted to any other Institute for any degree or diploma.
3. I have followed the guidelines provided by the Institute in preparing the report.
4. I have conformed to the norms and guidelines given in the Ethical Code of Conduct of the Institute.
5. Whenever I have used materials (data, theoretical analysis, figures, and text) from other sources, I have given due credit to them by citing them in the text of the report and giving their details in the references. Further, I have taken permission from the copyright owners of the sources, whenever necessary.
6. V.Sriman 2011CS030164
7. K.Thanshith 2011CS030177
8. SK.Ansar Ali 2011CS0300164

## CERTIFICATE

This is to certify that the project report entitled Market Basket Analysis submitted by Mr.V.Sriman, Mr.K.Thanshith, Mr.SK.Ansar.Ali to the Malla Reddy University, Hyderabad in partial fulfillment of the requirements for the award of the Degree Bachelor of Technology in Cse(DataScience)is a bonafide record of work carried out by him/her under my/our guidance and supervision. The contents of this report, in full or in parts, have not been submitted to any other Organization for the award of any Degree.

INTERNAL GUIDE APP DEVELOPMENT CONVENER HEAD OF THE DEPARTMENT

Dr. K Suresh Kumar Dr.GS Naveen Kumar

Designation Associate Professor CSE (Data Science)

CSE (Data Science)

Date:

External Examiner

## ACKNOWLEDGEMENT

My journey through this dissertation and this degree has required more effort than I was able to

deliver on my own. For this reason, I would like to acknowledge all the people who supported me throughout the process.

Firstly, I would to thank my supportive supervisor Mrs.M.Shailaja without whose guidance and knowledge I would not have completed this journey. He has encouraged me to learn new concepts and new tools which will be highly useful in building a successful career in future.

I would like to thank all the experts that participated in the evaluation of this and guided me with their knowledge and expertise no matter the distance and time.

At last, I would like to thank my family and friends for being supportive and loving at every step and for being there for me to give me a push whenever I needed it.

## ABSTRACT

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

MBA is well known activity of ARM ultimately used for business intelligent decisions. Mining frequent item sets and hence deduce rules to build classifiers with good accuracy is essential for efficient algorithm.

The issues for a leading supermarket are addressed here using frequent item set mining. The project uses file as database. Here, the itemsets and transactions of items are kept in a matrix form representing rows as list of items and column as transactions.

The frequent item sets are mined from database using the Apriori algorithm and then the association rules are generated.

The project is beneficial for supermarket managers to determine the relationship between the items that are purchased by their customers.

**Keywords**: Market Basket Analysis, Association Rule Mining, Apriori Algorithm

## CONTENTS

|  |  |  |
| --- | --- | --- |
| Title Page |  | I |
| Certificate | | II |
| Declaration |  | III |
| Acknowledgement | | IV |
| Abstract |  | V |
| Contents |  | VI |
| List of Figures | | VII |
| List of Tables | | VII |
| Abbreviations | | X |
| Chapter 1 | Introduction | 1 |
|  | 1.1 Section | 1 |
|  | 1.2 Section | … |
|  | 1.3 Section | … |
| Chapter 2 | Review of Relevant Literature | … |
| Chapter 3 | Methodology | … |
| Chapter 4 | Results and Discussions | … |
| Chapter 5 | Conclusions and Future Scope of Study | … |
| References |  | … |
|  | | … |

LIST OF TABLES

|  |  |  |
| --- | --- | --- |
| Table | Title | Page |
| 1.1 | Table Name1 | 5 |
| 1.2 | Table Name2 | 8 |
| 1.3 | Table Name3 | 9 |

## LIST OF FIGURES

|  |  |  |
| --- | --- | --- |
| Figure | Title | Page |
| 1.1 | Fig Name1 | 3 |
| 1.2 | Fig Name2 | 4 |
| 1.3 | Fig Name3 | 8 |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |
|  |  |  |

## LIST OF ABBREVIATIONS

ANNs Artificial Neural Networks

BM Bayesian Models

DM Deep Learning

DR Dimensionality Reduction

DT Decision Trees

EL Ensembles Learning

IBM Instance Based Models.

**Chapter 1: INTRODUCTION**

* 1. **Background**

Market Basket analysis is a data mining method focusing on discovering purchase patterns of the customers by extracting association or co-occurrences from a store’s transactional data. For example, when the person checkout items in a supermarket all the details about their purchase goes into the transaction database. Later, this huge data of many customers are analyzed to determine the purchasing pattern of customers. Also decisions like which item to stock more, cross selling, up selling, store shelf arrangement are determined.

Association rule mining (ARM) identifies the association or relationship between a large set of data items and forms the base for market basket analysis. Association rule mining has been widely used in various industries besides supermarkets, such as mail order, telemarketing production, fraud detection of credit card and e-commerce.

One of the challenges for companies that have invested heavily in customer data collection is how to extract important information from their vast customer databases and product feature databases, in order to gain competitive advantage.

Market basket analysis has been intensively used in many companies as a means to discover product associations.A retailer must know the needs of customers and adapt to them. Market basket analysis is one possible way to find out which items can be put together.



Figure1

Market Basket Analysis helps to identify the purchasing behavior of the customer. By mining the data from the huge transaction database shop managers can study the behavior or buying habits of the customer to increase the sale. In Market Basket Analysis, you look to see if there are combinations of products that frequently co-occur in a transaction.

**1.2Problem Statement:**

Nowadays people buy daily goods from super market nearby. There are many supermarkets that provide goods to their customer. The problem many retailers face is the placement of the items. They are unaware of the purchasing habits of the customer so they don’t know which items should be placed together in their store. With the help of this application shop managers can determine the strong relationships between the items which ultimately helps them to put products that co-occur together close to one another. Also decisions like which item to stock more, cross selling, up selling, store shelf arrangement are determined.

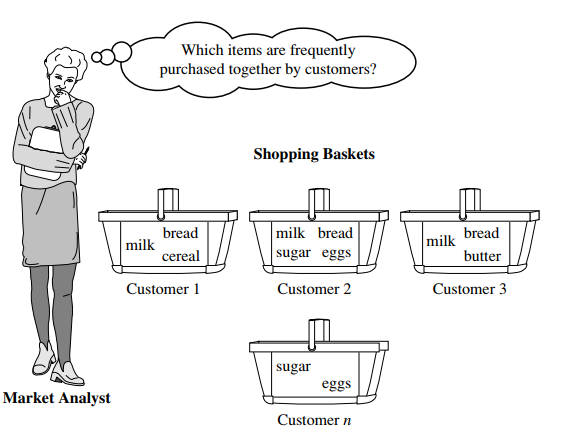


Figure 1.2

**1.3 Objectives:**

* To identify the frequent items from the transaction on the basis of support and confidence
* To generate the association rule from the frequent item sets.

**1.4 Scope:**

The scope of the application is limited to desktop application right now. The application is targeted towards a supermarket.

**1.5 Limitations:**

* The application will be desktop and will not be available online.
* Input to the application will be a file which contains integer values representing the list of items, the integer values will be mapped manually.

**1.6 Life cycle of Market Basket Analysis:**



Figure 1.3

**CHAPTER 2: REVIEW OF RELEVANT LITERATURE**

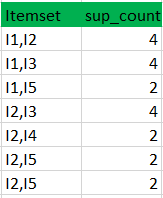
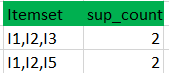
Data Mining provides a lot of opportunities in the market sector. Decision making and understanding the behavior of the customer has become vital and challenging problem for the organization in order to sustain in this competitive environment.

The challenges that the organization faces is to extract the information from their vast customer databases, in order to gain competitive advantage.Yanthy et al [1] in this paper author states about the important goal in data mining is to reveal hidden knowledge from data and various algorithms have been proposed for, but the problem is that typically not all rules are interesting –only small fraction of the generated rules would be of interest to any given users.

Hence numerous methods such as confidence, support, and lift have been proposed to determine the best or most interesting rules. However some algorithms are good at generating rules high in one measure but bad in other.It is one of the Data Mining Algorithm which is used to find the frequent items/item set from a given data repository.

The algorithm involves 2 steps

a. Pruning

b. Joining

**figure 2.1**: pruning and joining again until there are no more frequent items left*.*

We follow the same procedure again. First, we do the join step and we cross join each itemset with one another. In out example the first two elements of the item set should match.

After, check all subsets of these item sets are frequent or not. In our example the itemset formed after join step is {I1, I2, I3, I5}. So, one of the subsets of this itemset is {I1, I3, I5} which is not frequent. Therefore, there is no itemset left anymore.

The Apriori property is the important factor to be consider before proceeding with the algorithm Apriori property states that If an item X is joined with item Y,

It has support, confidence, lift.

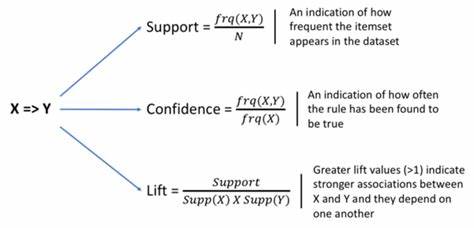


Figure2.1

Basically when we are determining the strength of an association rule i.e. how string the relationship is between the transaction of the items we measure through the use of the support and confidence.

The support of an item is the number of transaction containing the item. Those items that do not meet the minimum support are excluded from the further processing.

Support determines how often a rule is applicable to a given data set. Confidence is defined as the conditional probability that a transaction containing the LHS will also contain the RHS. Confidence(LHS->RHS->

P(RHS/LHS)=P(RHS∩LHS)/P(LHS)=support(RHS∩LHS)/support(LHS)

Confidence determines how frequently item in RHS appears in the transaction that contain LHS.

While determining the rules we must measure these two components as it is very important to us.

A rule that has very low support may occur simply by chance. Confidence on the other hand, measures the reliability of the inference made by the rule. Han [4, 5] presented a new association rule mining approach that does not use candidate rule generation called FP-growth that generates a highly condensed frequent pattern tree (fptree) representation of the transactional database. Each database transaction is represented in the tree by at most one path. FP-tree is smaller in size than the original database the construction of it requires two database scans, where in the first scan, frequent item sets along with their support in each transaction are produced and in the second scan, FP-tree is constructed.

The mining process is performed by concatenating the patterns with the ones produced from the conditional FP-tree. One constraint of FP-growth method is that memory may not fit FP-tree especially in dimensionally large database.

Liu [6] proposed CBA the first Associative Classification (AC) algorithm. CBA implements the famous Apriori algorithm[3] in order to discover frequent rule items. The Apriori algorithm consists of three main steps.

a. Continuous attribute in the training data set gets discredited.

b. Frequent rule items discovery

c. Rule generation CBA selects high confidence rules to represent the classifier.

Finally, to predict a test case CBA applies the highest confidence rule whose body matches the test case. Experimental result designated that CBA drives higher quality classifiers with regards to accuracy that rule induction and decision tree classification approaches.

Association Rules and existing data mining algorithms usage for market basket analysis but focuses on Apriori algorithm and concludes that the algorithm can be modified and it can be extended in the future work which also decrease the time complexity. Author also clearly states the De-merits of the algorithm but claims that there is the way to improve the efficiency of the algorithm

**CHAPTER 3: METHODOLOGY**

Methodology are the guidelines or path on how to proceed in validating knowledge on your subject matter. Different areas of science have developed very different bodies of methodology based on which to conduct their research (Little, 2012).

**3.1. RESEARCH PURPOSE:**

The ultimate purpose of every business is to find better ways to improve the profit for a long run. But for this research the aim would be to encountering actual case of dependencies among products chosen by customer.

Though several different products could be bought in a single visit to a mega store like, groceries, pillowcase, furniture, an electric toaster, etc. However, we believe that there are no coincidences for these choices. These decisions from several categories results in forming customer’s shopping basket. Which with-holds the collection of categories that customer purchased on a specific shopping trip. (Manchanda, Ansari, & Gupta, 1999).

**3.2. RESEARCH APPROACH**

The implementation of Market Basket Analysis results into inventory management, marketing and promotion strategies, etc. Therefore, this a comparative analysis thesis, which displays the relevance of time. This analysis would be done by implementing the idea on a dataset and check the results. Exploratory data mining techniques are used followed by association rules, or pattern recognition, or ‘if then’ statements. Also, a comparative analysis is done on the data for three months (i.e. April, May and June) all together and individually.

**3.3. RESEARCH STRATEGY**

There are three main types of research strategies that exists namely quantitative, qualitative and mixed (Creswell, 2013). Researches can also be experimental and non-experimental which in some books are falling into another category while Carswell believes they are under three main mentioned categories. In the book research design by John W. Creswell he mentions that these three categories are not discrete, and they actually are at the ends on a continuum (Creswell, 2013). The research tends to be more qualitative than quantitative and vice versa so as a result all other types can be fitted in between this continuum. Therefore, he adds that the mixed methods are in 14 middle since it contains the elements of both these types. The methods alongside their types are presented in the following tables below explaining the strategies more clearly.

The data was collected from <http://www.salemmarafi.com/wp> content/uploads/2014/03/groceries.csv due to the unavailability of data from the supermarkets.

Table 4- Sample data

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Transaction | Items | | | | |
| 1 | Fruit | Bread | Butter | Soups |  |
| 2 | Fruit | Yogurt | Coffee |  |  |
| 3 | Milk |  |  |  |  |
| 4 | Fruit | Yogurt | Cheese | Meat |  |
| 5 | Vegetables | Milk | Bakery |  |  |
| 6 | Milk | Butter | Yogurt | Rice | cleaner |
| 7 | Rolls/bun |  |  |  |  |

**3**.**3Data preprocessing:**

The data collected was mapped manually as integer values as shown in Figure 4. For

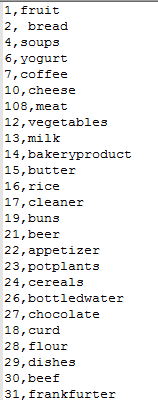
example the “Fruit” was labeled as 1, “Bread” as 2 “Soups” as 4 and so on.

Figure 3.1 - Mapped to integers

The mapped integer’s values were then saved in a text file and given as the input to the system. Figure 5 shows the input file that is given to the system

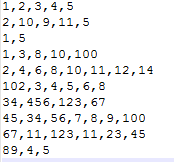


Figure 3.2- Input file to system

The Apriori algorithm was used for processing the input data and result was produced as the list of rules that are strongly associated with each other.

**3.4 Apriori algorithm:**

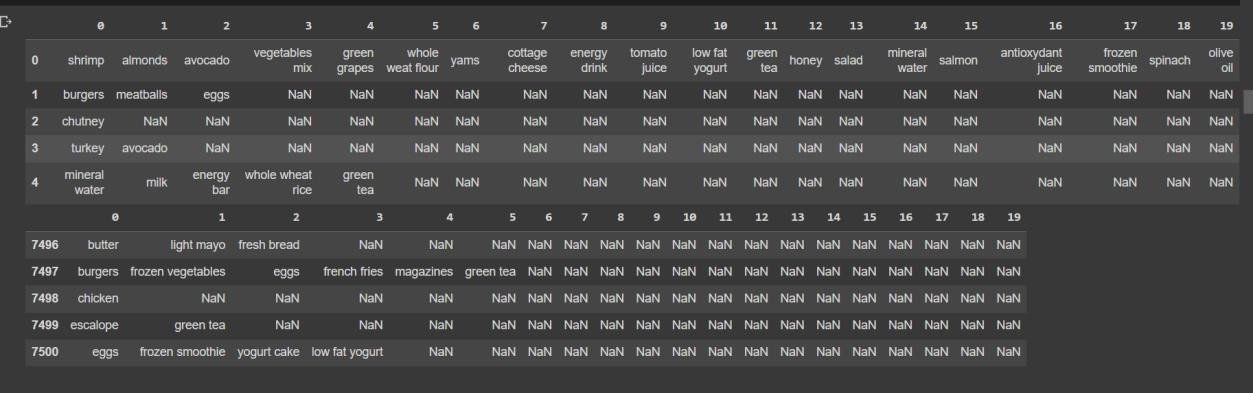
Association rule mining finds interesting associations and/or correlation relationships among large set of data items. Association rules shows attribute value conditions that occur frequently together in a given dataset. A typical and widely used example of association rule mining is Market Basket Analysis. For example, data are collected from the supermarkets. Such market basket databases consist of a large number of transaction records. Each record lists all items 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.

1. Support
2. Confidence
3. lift

###### Chapter:4 Result and Discussions

Association rules are found to be functional if minimum support threshold and minimum confidence threshold meet the threshold set by the user or consultant. Market basket analysis rules can be written as if **{A} then {B} i.e. {A} => {B}**.

The store data that we have collected is of the size 7501 x 20. Each transaction in the dataset is the combination of products bought together by a customer. Figure 1 shows the head and tail of the dataset.



**Figure 4.1** Head and tail of the dataset

This data needs a lot of pre-processing. Transaction Encoder() of mlxtend. preprocessing does the pre-processing for us. Transaction Encoder() helps find the different products in transaction and give each transaction a list that contains a binary array where True represents a purchased product. The pre-processed data is visualised as bar charts and each column represent distinct item. Figure 4.1 shows sales of unique item

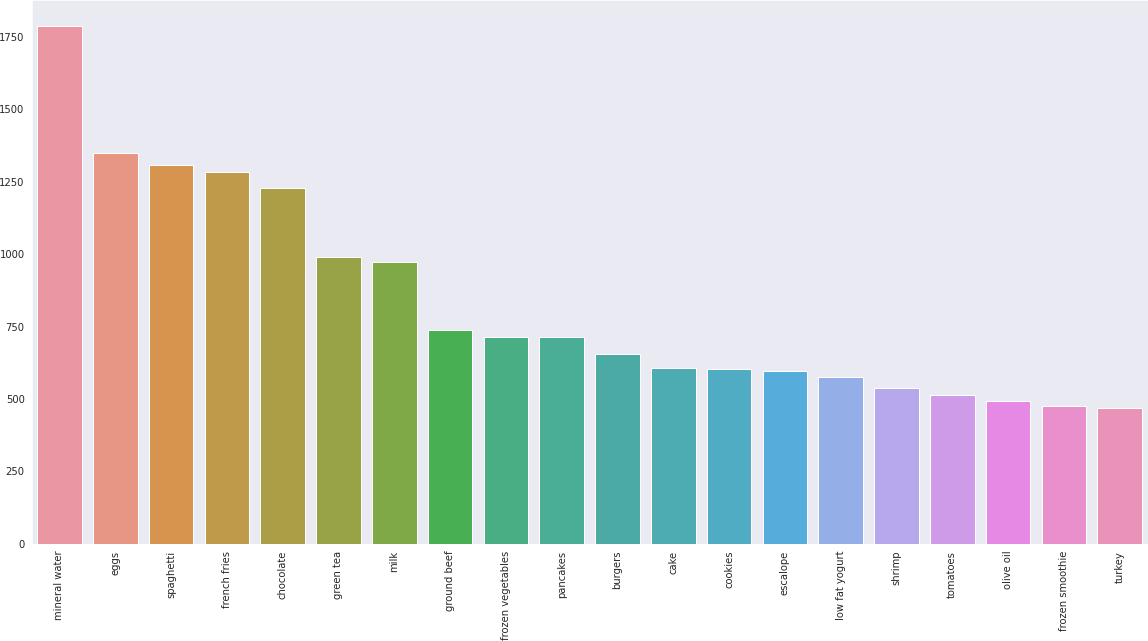


Figure 4.2.Sales of unique items

The most popular item in the store, as shown in figure 4.2 above, is mineral water. Rule generation involves two steps in the process. The first rule is to create a list of repeating items, and the second is to create a list of rules that are acceptable given the observed items. One approach is to look at all potential subsets of the item set under consideration, look at the item sets' support values, and then only consider the item sets with values higher than the minimal threshold support value. The apriori algorithm is a support measure that opposes monotonicity. As a result, the search space is smaller, which speeds up the construction of repeating item sets. Apriori cuts back the supersets of an itemset that does not meet the

To generate all these rules requires relatively large number of steps. Apriori breaks down this process. From all the workable rules we recognize the ones they are above the minimum confidence level. Confidence of rules from the same itemset obey the anti-monotone property.

Conf (A, B, C -> D) >= Conf (B, C-> A, D) >= Conf (C-> A, B, D)

Apriori requires numerous candidate item sets creation to examine the support of every itemset created and this is computationally expensive. This limitation can be overcome by FP Growth which is an advancement of Apriori algorithm. A pattern is created without candidate generation. FP tree is a database representation in form of a tree. The association between item sets is preserved in the tree. The database is unusable due to one common item. The term "pattern fragment" refers to the broken or fractured portion. In contrast to the Apriori algorithm, the itemsets of the fragmented patterns are examined, which decreases the temporal complexity.

The FP tree is built using the itemsets that are located at the very beginning of the database. Every node in the FP tree represents a different itemset in some way. The value null is represented at the root node, while the itemsets are represented at the child nodes in the hierarchy. During the process of tree formation, the connection between the nodes is maintained.

Steps in FP Growth to mine frequent pattern.

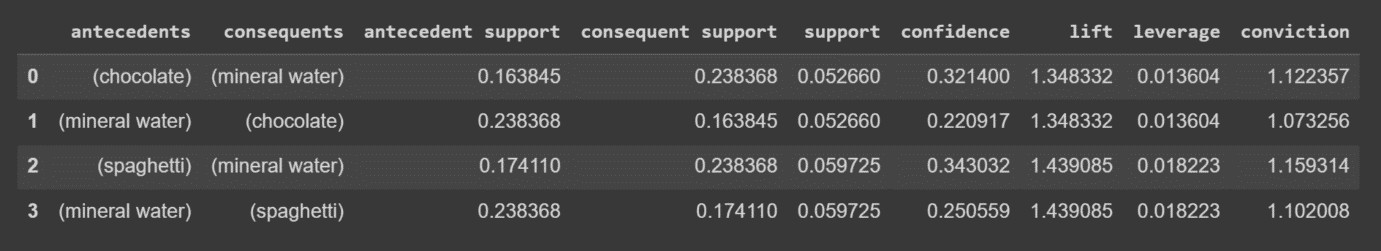
1. Scan the database as in the Apriori.
2. To build the FP tree from the tree's root. Root is represented by Null.
3. Scan the database to study transactions. The itemset with maximum transactions is placed at the top. The item sets are arranged in descending order of number of transactions.
4. Common itemsets are linked to the nodes of another itemset.444444444

The count of the common node and new node is increased by 1 as the

1. nodes are constructed and joined.
2. node is analysed frist along the links. The travesed path is called conditional pattern base.
3. Tree construction for conditional FP. The conditional FP tree is made up of the itemsets that satisfy the threshold support..

# **4.1** **RESULTS AND DISCUSSION**

From Figure 3 we can conclude that mineral water must be in stock at all times.



**Figure 4.3** Apriori algorithm with lift greater than 1.3

Figure 4.3 shows the results of Apriori algorithm for which the lift is greater than 1.3. From this we can conclude that 22% of transactions containing mineral water contains chocolate.

32% of transactions containing chocolate contains mineral water. From the comparison of lift, leverage and conviction of {spaghetti and mineral water} and {chocolate and mineral water} we observe that the chances of transaction of {spaghetti and mineral water} is more than {chocolate and mineral water}.

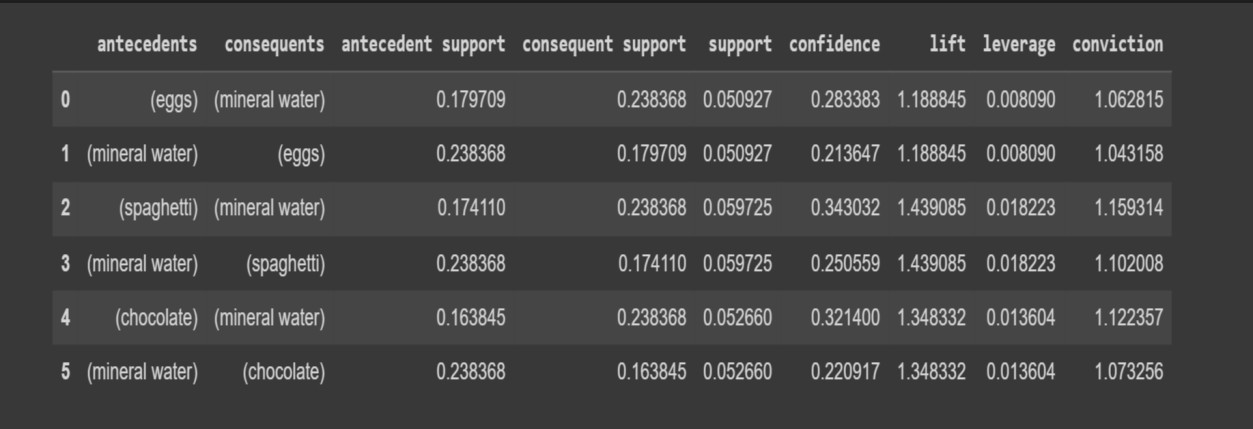
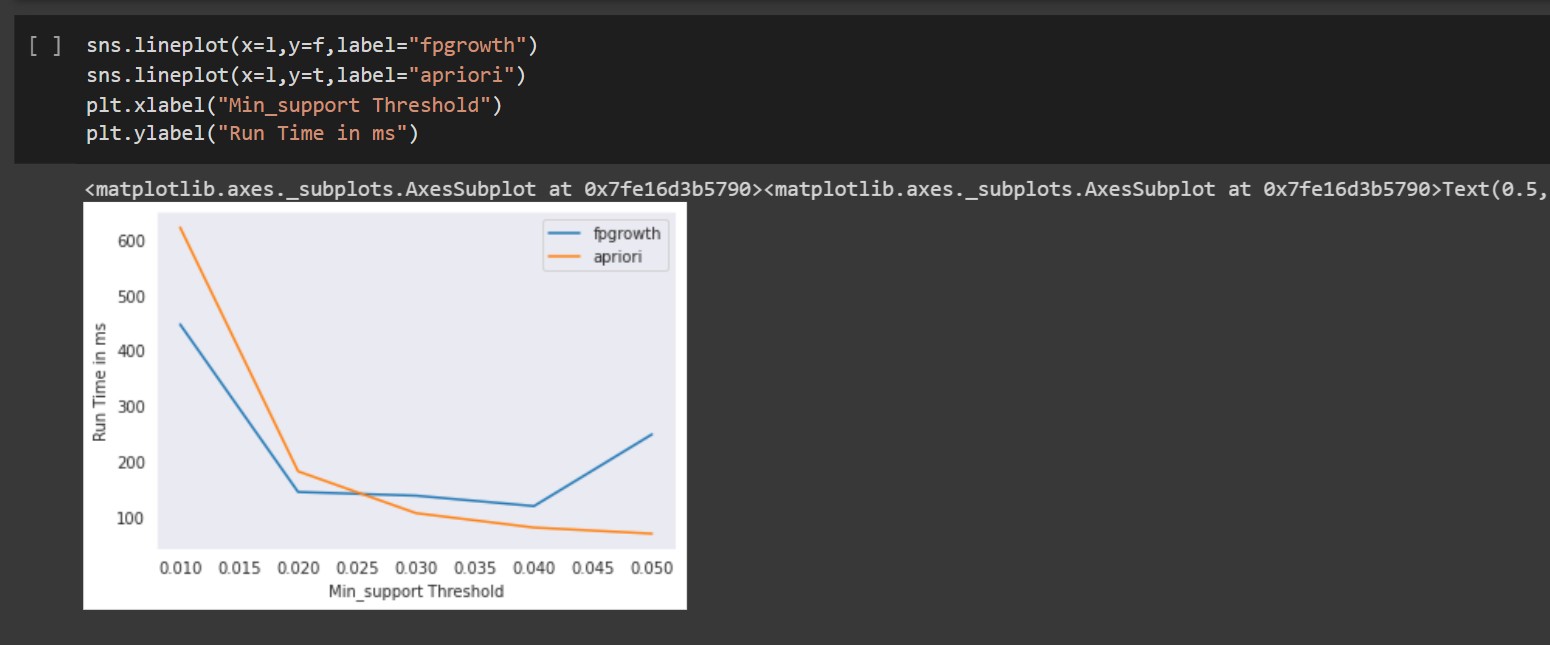


Figure 4.4 FP Growth algorithm

From the result we observe that spaghetti and mineral water are most likely to occur together.



**Figure 4.5** Run time comparison – Apriori and FP Growth

From Figure 5 we get the insights about the run time comparison between Apriori and FP Growth. Apriori algorithm requires creation of candidate sets, therefore it is much slower than FP Growth. FP Growth is five times faster.

**4.2 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 approach is 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 pattern in which event A occurs before event B, which occurs before event C (Han, Kim, & Sohn, 2009).

MBA can be used to access multilevel relationships. It can be applied across all level of analysis ranging from an individual level to firm level, industry level and country level contexts.

### 4.1 TOOLS USED FOR ANALYSIS

* Jupyter notebook
* Python Libraries
  + Pandas
  + Datetime
  + Matplotlib
  + Mlxtend
  + Random
  + xlsxwriter

MS Excel

**Chapter 5 Conclusions and Future Scope of Study References**

Based on the findings, the FP Growth algorithm can be considered not only more sophisticated than the Apriori algorithm, but also significantly quicker. This data shows that mineral water is the most commonly purchased product. Thus, there must always be a supply of mineral water for purchase.

A predictive form of market basket analysis is gaining traction across various industries in an effort to pinpoint sequential purchases as industry leaders continue to investigate the technique's use.

### SUMMARY & FUTURE WORK

This research shows that time is a very important factor which should be kept in mind while performing any market basket analysis. It can help reveal very interesting information / insights about the customers which contribute in profit maximization. For example, the discovery of complementary or supplementary products can lead to cross-selling or promotional opportunities.

This research is a superior and cheaper approach than the traditional customer surveys, which are usually very costly and are time consuming. They also include errors at every step of the survey research.

In this research we have eliminated less frequent items from the dataset by setting minimum threshold. The results which we got after detecting the purchase behaviour of the customer can be used in cross selling recommendation and to enhance their marketing strategies while deciding for promotions.

This research also shows that how market basket analysis also has its applicability in many domains and applying market basket analysis considering time as an important factor will be able to solve several problems in much effective and efficient way.

We used graphs to see the pattern of the association rules in 3 months periods, which is a good way to visualize top association rules and compare them at different time periods. With more data, we will be able to easily identify which association rules are more active during certain time periods. This insight will be very useful for the super markets to preplan their marketing strategies for that period.

We not only did comparison of association rules but also found top association rules during the 3 months of period which can help the company is inventory management as well as which categories of products should be kept close to each other to upscale the purchase. Mining into their data provides managers with a unique over view of what is happening with their business so that they can implement strategies efficiently and can move faster than their competitors.

Market basket analysis can also focus just on one or more than one Category/ categories which will give us insight within some important categories based on the requirements of the marketing team.

In our research we used the monthly time period. But MBA can give more in-depth insights if the analysis was done on a daily or weekly period. The association rules that we will be getting will display more rapid fluctuation in confidence and lift. It will be interesting to get more insights at a shorter period basis.

In future research it will be very interesting to do in-depth understanding of the association rules by evaluating the changes in the lift and confidence values, which can be made possible by calculating the standard deviation. This way we will be able to witness the evolution of association rules.

Future work would also be to design and develop an intelligent prediction model to generate the association rules that can be adopted on a recommendation system to make the functionality more operational.

References :

1. Liao, S. H., Chu, P. H., & Hsiao, P. Y. (2012). Data mining techniques and applications - A decade review from 2000 to 2011. *Expert Systems with Applications*. <https://doi.org/10.1016/j.eswa.2012.02.063>
2. Mushtaq, A. (2015). Data Mining For Marketing. *Journal of Marketing and Consumer Research*, *9*, 17–29.
3. Ngai, E. W. T., Xiu, L., & Chau, D. C. K. (2009). Application of data mining techniques in customer relationship management: A literature review and classification. *Expert Systems with Applications*, *36*(2 PART 2), 2592–2602. <https://doi.org/10.1016/j.eswa.2008.02.021>
4. 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>

1. Gangurde, R., Kumar, B., & Gore, D. (2017). Building Prediction Model using Market Basket Analysis. *International Journal of Innovative Research in Computer and Communication Engineering*, *5*(2). <https://doi.org/10.15680/IJIRCCE.2017.0502136>
2. Hibino, A., & Niwa, Y. (2008). Graphical representation of nuclear incidents/accidents by associating network in nuclear technical communication. *Journal of Nuclear Science and Technology*, *45*(5), 369–377. <https://doi.org/10.1080/18811248.2008.9711446>
3. Gangurde, R., Kumar, B., & Gore, D. (November 2017). Optimized Predictive Model using Artificial Neural Network for Market Basket Analysis. *International Journal of Computer Science & Communication, 9(1).* <https://doi.org/10.090592/IJCSC.2017.014>
4. Liao, S. H., Chu, P. H., & Hsiao, P. Y. (2012). Data mining techniques and applications - A decade review from 2000 to 2011. *Expert Systems with Applications*. <https://doi.org/10.1016/j.eswa.2012.02.063>
5. 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>
6. 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>
7. Kumar, K. Suresh, T. Ananth Kumar, S. Sundaresan, and V. Kishore Kumar. "Green IoT for Sustainable Growth and Energy Management in Smart Cities." In Handbook of Green Engineering Technologies for Sustainable Smart Cities, pp. 155-172. CRC Press, 2021.
8. <https://www.analyticsvidhya.com/blog/2021/10/a-comprehensive-guide-on-market-basket-analysis/>
9. <https://raw.githubusercontent.com/CSIT-GUIDE/FYP-2016/master/1813_Sanjeev_MarketBasketAnalysis.pdf>
10. <https://towardsdatascience.com/market-basket-analysis-using-associative-data-mining-and-apriori-algorithm-bddd07c6a71a>
11. https://www.javatpoint.com/apriori-algorithm

**Source code**

## Imports

**import** **numpy** **as** **np**

**import** **pandas** **as** **pd**

**import** **matplotlib.pyplot** **as** **plt**

**import** **seaborn** **as** **sns**

**from** **mlxtend.frequent\_patterns** **import** apriori

**from** **mlxtend.frequent\_patterns** **import** association\_rules

## Dataset

The contains information about customers buying different grocery items.

data = pd.read\_csv('Market\_Basket.csv', header = **None**)

data.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 7501 entries, 0 to 7500

Data columns (total 20 columns):

0 7501 non-null object

1 5747 non-null object

2 4389 non-null object

3 3345 non-null object

4 2529 non-null object

5 1864 non-null object

6 1369 non-null object

7 981 non-null object

8 654 non-null object

9 395 non-null object

10 256 non-null object

11 154 non-null object

12 87 non-null object

13 47 non-null object

14 25 non-null object

15 8 non-null object

16 4 non-null object

17 4 non-null object

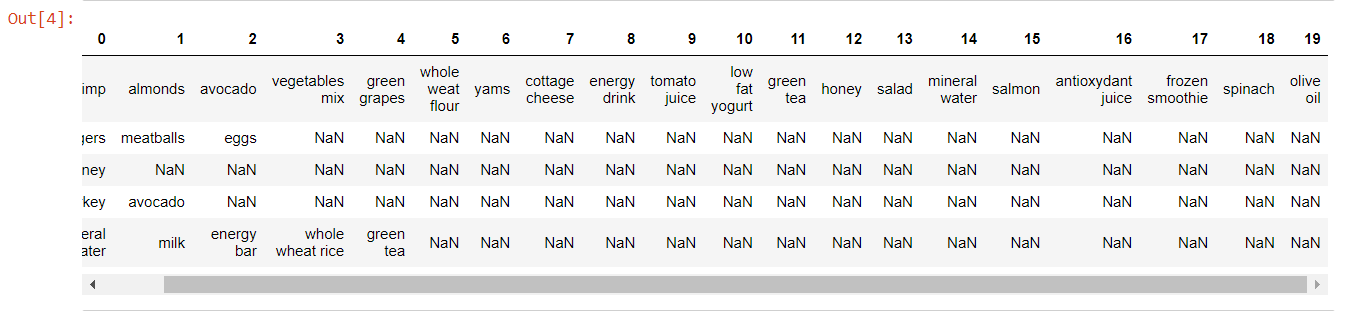
18 3 non-null object

19 1 non-null object

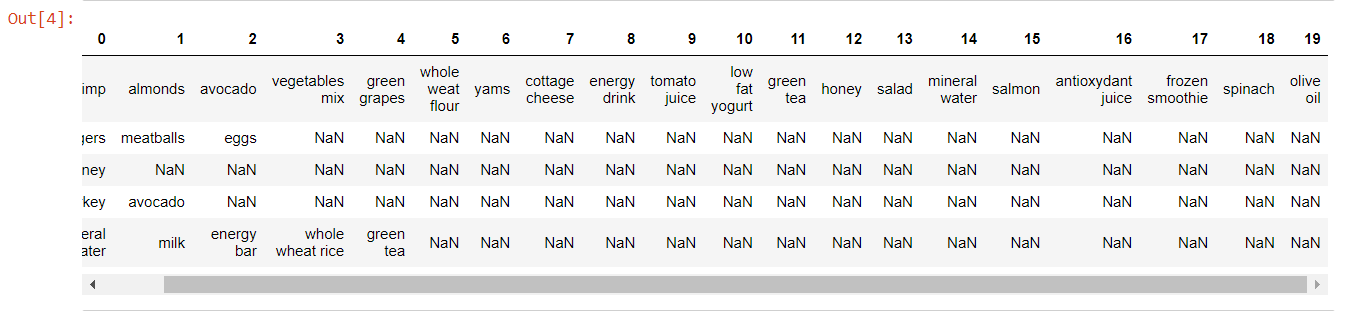
dtypes: object(20)

memory usage: 1.1+ MB

data.head()



data.describe()



**EDA**

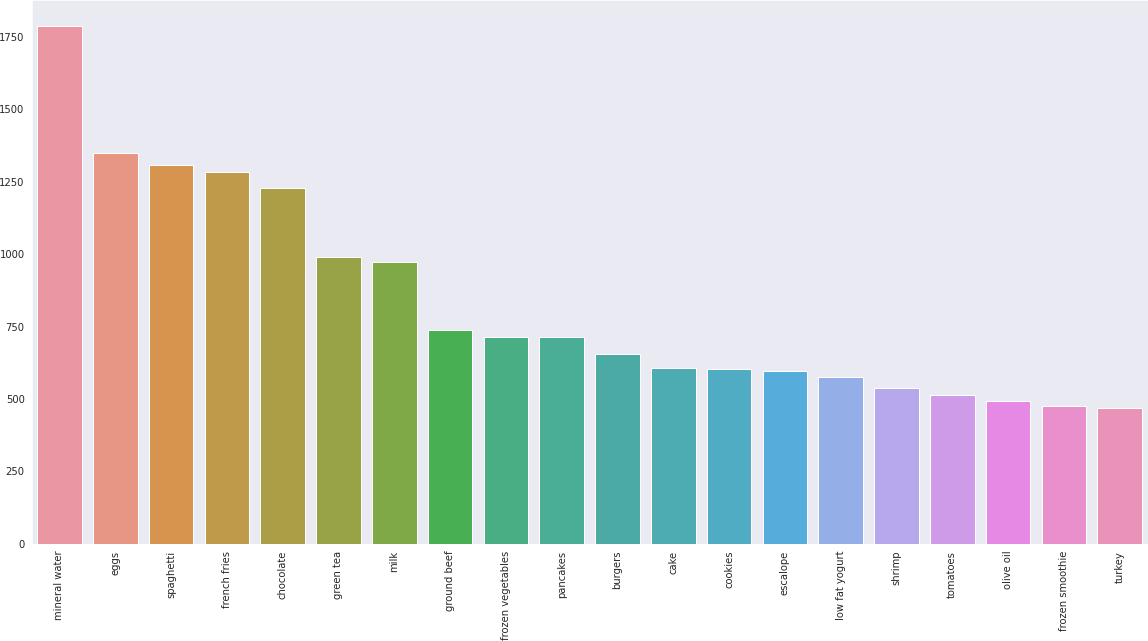
color = plt.cm.rainbow(np.linspace(0, 1, 40))

data[0].value\_counts().head(40).plot.bar(color = color, figsize=(13,5))

plt.title('frequency of most popular items', fontsize = 20)

plt.xticks(rotation = 90 )

plt.grid()

plt.show()

## Getting the list of transactions

Once we have read the dataset, we need to get the list of items in each transaction. SO we will run two loops here. One for the total number of transactions, and other for the total number of columns in each transaction. This list will work as a training set from where we can generate the list of association rules.

*# Getting the list of transactions from the dataset*

transactions = []

**for** i **in** range(0, len(data)):

transactions.append([str(data.values[i,j]) **for** j **in** range(0, len(data.columns))])

In [10]:

transactions[:1]

output:

[['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',

'spinach',

'olive oil',

'Food']]

## **Association rules**

* ****Association rule****
  + Contains antecedent and consequent
    - {health} → {cooking}
* ****Multi-antecedent rule****
  + {humor, travel} → {language}
* ****Multi-consequent rule****
  + {biography} → {history, language}
* ****Multi-antecedent and consequent rule****
  + {biography, non-fiction} → {history, language}

**from** **itertools** **import** permutations

*# Extract unique items.*flattened = [item **for** transaction **in** transactions **for** item **in** transaction]items = list(set(flattened))

In [12]:

print('# of items:',len(items))print(list(items))

# of items: 122

['ham', 'champagne', 'red wine', 'asparagus', 'burgers', 'protein bar', 'spaghetti', 'cereals', 'hand protein bar', 'shrimp', 'flax seed', 'mineral water', 'grated cheese', 'pet food', 'mashed potato', 'cider', 'oatmeal', 'body spray', 'honey', 'shampoo', 'strawberries', 'salad', 'milk', 'chutney', 'bramble', 'cottage cheese', 'strong cheese', 'cauliflower', 'parmesan cheese', 'chocolate', 'whole weat flour', 'Food', 'escalope', 'babies food', 'pasta', 'vegetables mix', 'gluten free bar', 'tea', 'sandwich', 'whole wheat rice', 'light mayo', 'bacon', 'energy bar', 'sparkling water', 'low fat yogurt', 'cream', 'toothpaste', 'chicken', 'nan', 'soup', 'frozen smoothie', 'ketchup', 'olive oil', 'magazines', 'soda', 'eggplant', 'barbecue sauce', 'hot dogs', 'chocolate bread', 'yams', 'herb & pepper', 'carrots', 'butter', 'pepper', ' asparagus', 'rice', 'energy drink', 'candy bars', 'cookies', 'water spray', 'black tea', 'oil', 'muffins', 'meatballs', 'cooking oil', 'mushroom cream sauce', 'light cream', 'whole wheat pasta', 'brownies', 'burger sauce', 'mint green tea', 'melons', 'cake', 'dessert wine', 'almonds', 'mint', 'fresh bread', 'avocado', 'spinach', 'mayonnaise', 'tomatoes', 'shallot', 'salmon', 'french wine', 'corn', 'blueberries', 'pancakes', 'fresh tuna', 'clothes accessories', 'antioxydant juice', 'white wine', 'chili', 'frozen vegetables', 'nonfat milk', 'pickles', 'salt', 'green grapes', 'turkey', 'french fries', 'eggs', 'yogurt cake', 'zucchini', 'fromage blanc', 'ground beef', 'gums', 'bug spray', 'green beans', 'green tea', 'napkins', 'tomato juice', 'tomato sauce', 'extra dark chocolate']

In [13]:

**if** 'nan' **in** items: items.remove('nan')print(list(items))

['ham', 'champagne', 'red wine', 'asparagus', 'burgers', 'protein bar', 'spaghetti', 'cereals', 'hand protein bar', 'shrimp', 'flax seed', 'mineral water', 'grated cheese', 'pet food', 'mashed potato', 'cider', 'oatmeal', 'body spray', 'honey', 'shampoo', 'strawberries', 'salad', 'milk', 'chutney', 'bramble', 'cottage cheese', 'strong cheese', 'cauliflower', 'parmesan cheese', 'chocolate', 'whole weat flour', 'Food', 'escalope', 'babies food', 'pasta', 'vegetables mix', 'gluten free bar', 'tea', 'sandwich', 'whole wheat rice', 'light mayo', 'bacon', 'energy bar', 'sparkling water', 'low fat yogurt', 'cream', 'toothpaste', 'chicken', 'soup', 'frozen smoothie', 'ketchup', 'olive oil', 'magazines', 'soda', 'eggplant', 'barbecue sauce', 'hot dogs', 'chocolate bread', 'yams', 'herb & pepper', 'carrots', 'butter', 'pepper', ' asparagus', 'rice', 'energy drink', 'candy bars', 'cookies', 'water spray', 'black tea', 'oil', 'muffins', 'meatballs', 'cooking oil', 'mushroom cream sauce', 'light cream', 'whole wheat pasta', 'brownies', 'burger sauce', 'mint green tea', 'melons', 'cake', 'dessert wine', 'almonds', 'mint', 'fresh bread', 'avocado', 'spinach', 'mayonnaise', 'tomatoes', 'shallot', 'salmon', 'french wine', 'corn', 'blueberries', 'pancakes', 'fresh tuna', 'clothes accessories', 'antioxydant juice', 'white wine', 'chili', 'frozen vegetables', 'nonfat milk', 'pickles', 'salt', 'green grapes', 'turkey', 'french fries', 'eggs', 'yogurt cake', 'zucchini', 'fromage blanc', 'ground beef', 'gums', 'bug spray', 'green beans', 'green tea', 'napkins', 'tomato juice', 'tomato sauce', 'extra dark chocolate']

## **One-hot encoding transaction data**

Throughout we will use a common pipeline for preprocessing data for use in market basket analysis. The first step is to import a pandas DataFrame and select the column that contains transactions. Each transaction in the column will be a string that consists of a number of items, each separated by a comma. The next step is to use a lambda function to split each transaction string into a list, thereby transforming the column into a list of lists. Then we will transform the transactions into a one-hot encoded DataFrame, where each column consists of TRUE and FALSE values that indicate whether an item was included in a transaction.

In [15]:

*# Import the transaction encoder function from mlxtend*

**from** **mlxtend.preprocessing** **import** TransactionEncoder

*# Instantiate transaction encoder and identify unique items*

encoder = TransactionEncoder().fit(transactions)

*# One-hot encode transactions*

onehot = encoder.transform(transactions)

*# Convert one-hot encoded data to DataFrame*

onehot = pd.DataFrame(onehot, columns = encoder.columns\_).drop('nan', axis=1)

*# Print the one-hot encoded transaction dataset*

onehot.head()

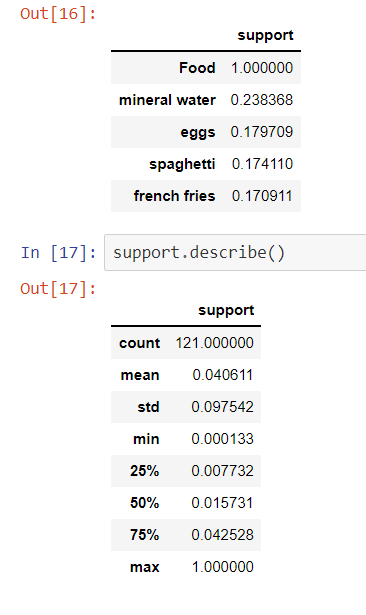


*# Compute the support*

support = onehot.mean()support = pd.DataFrame(support, columns=['support']).sort\_values('support',ascending=**False**)

*# Print the support*

support.head()

**

## **Recommending food with support**

A grocery-store wants to get members to eat more and has decided to use market basket analysis to figure out how. They approach you to do the analysis and ask that you use the five most highly-rated food items.

In [18]:

*# Compute support for burgers and french fries*

supportBF = np.logical\_and(onehot['burgers'], onehot['french fries']).mean()

*# Compute support for burgers and mineral water*

supportBM = np.logical\_and(onehot['burgers'], onehot['mineral water']).mean()

*# Compute support for french fries and mineral water*

supportFM = np.logical\_and(onehot['french fries'], onehot['mineral water']).mean()

*# Print support values*

print("burgers and french fries: **%.2f**" % supportBF)

print("burgers and mineral water: **%.2f**" % supportBM)

print("french fries and mineral water: **%.2f**" % supportFM)

**burgers and french fries: 0.02**

**burgers and mineral water: 0.02**

**french fries and mineral water: 0.03**

## **Computing the support metric**

Previously we one-hot encoded a small grocery store's transactions as the DataFrame onehot. In this exercise, we'll make use of that DataFrame and the support metric to help the store's owner. First, she has asked us to identify frequently purchased items, which we'll do by computing support at the item-level. And second, she asked us to check whether the rule {mineral water} → {french fries} has a support of over 0.050.05.

In [19]:

*# Add a mineral water+french fries column to the DataFrame onehot*

onehot['mineral water+french fries'] = np.logical\_and(onehot['mineral water'], onehot['french fries'])

*# Compute the support*

support = onehot.mean()val = support.loc['mineral water+french fries']

*# Print the support values*

print(f'mineral water+french fries support = **{val}**')

mineral water+french fries support = 0.03372883615517931

## **Refining support with confidence**

After reporting your findings from the previous exercise, the store's owner asks us about the direction of the relationship. Should they use mineral water to promote french fries or french fries to promote mineral water?

We decide to compute the confidence metric, which has a direction, unlike support. We'll compute it for both {mineral water} → {french fries} and {french fries} → {mineral water}.

In [20]:

*# Compute support for mineral water and french fries*

supportMF = np.logical\_and(onehot['mineral water'], onehot['french fries']).mean()

*# Compute support for mineral water*

supportM = onehot['mineral water'].mean()

*# Compute support for french fries*

supportF = onehot['french fries'].mean()

*# Compute confidence for both rules*

confidenceMM = supportMF / supportMconfidenceMF = supportMF / supportF

*# Print results*

print('mineral water = **{0:.2f}**, french fries = **{1:.2f}**'.format(confidenceMM, confidenceMF))

**mineral water = 0.14, french fries = 0.20**

Even though the support is identical for the two association rules, the confidence is much higher for french fries -> mineral water, since french fries has a higher support than mineral water.

## **Further refinement with lift**

Once again, we report our results to the store's owner: Use french fries to promote mineral water, since the rule has a higher confidence metric. The store's owner thanks us for the suggestion, but asks us to confirm that this is a meaningful relationship using another metric.

You recall that lift may be useful here. If lift is less than 11, this means that mineral water and french fries are paired together less frequently than we would expect if the pairings occurred by random chance.

In [21]:

*# Compute lift*

lift = supportMF / (supportM \* supportF)

*# Print lift*

print("Lift: **%.2f**" % lift)

**Lift: 0.83**

## **Computing association and dissociation**

The store's owner has returned to you once again about your recommendation to promote french fries using burgers. They're worried that the two might be dissociated, which could have a negative impact on their promotional effort. They ask you to verify that this is not the case.

You immediately think of Zhang's metric, which measures association and dissociation continuously. Association is positive and dissociation is negative.

In [26]:

*# Compute the support of burgers and french fries*

supportT = onehot['burgers'].mean()supportP = onehot['french fries'].mean()

*# Compute the support of both food items*

supportTP = np.logical\_and(onehot['burgers'], onehot['french fries']).mean()

*# Complete the expressions for the numerator and denominator*

numerator = supportTP - supportT\*supportPdenominator = max(supportTP\*(1-supportT), supportT\*(supportP-supportTP))

*# Compute and print Zhang's metric*

zhang = numerator / denominatorprint(zhang)

**0.3533836982354581**

Once again, the association rule if burgers then french fries proved robust. It had a positive value for Zhang's metric, indicating that the two food items are not dissociated.

## **Generating association rules**

Previously we computed itemsets for the novelty gift store owner using the Apriori algorithm. You told the store owner that relaxing support from 0.05 to 0.04 increased the number of itemsets from 5050 to 8787. Satisfied with the descriptive work we've done, the store manager asks us to identify some association rules from those two sets of frequent itemsets we computed.

Our objective is to determine what association rules can be mined from these itemsets.

In [50]:

*# Import the association rule function from mlxtend***from** **mlxtend.frequent\_patterns** **import** association\_rules

*# Compute all association rules for frequent\_itemsets\_1*rules\_1 = association\_rules(frequent\_itemsets\_1,

metric = "support",

min\_threshold = 0.001)

*# Compute all association rules for frequent\_itemsets\_2*rules\_2 = association\_rules(frequent\_itemsets\_2,

metric = "support",

min\_threshold = 0.002)

*# Print the number of association rules generated*print(len(rules\_1), len(rules\_2))

6 2

## **Pruning with lift**

Once again, we report back to the novelty gift store manager. This time, we tell her that we identified 22 rules when you used a higher support threshold for the Apriori algorithm and only 66 rules when you used a lower threshold. She commends us for the good work, but asks you to consider using another metric to refine the two rules.

You remember that lift had a simple interpretation: values greater than 11 indicate that items co-occur more than we would expect if they were independently distributed across transactions. We decide to use lift, since that message will be simple to convey.

In [51]:

*# Import the association rules function*

**from** **mlxtend.frequent\_patterns** **import** association\_rules

*# Compute frequent itemsets using the Apriori algorithm*

frequent\_itemsets = apriori(onehot, min\_support = 0.03, max\_len = 2, use\_colnames = **True**)

*# Compute all association rules for frequent\_itemsets*

rules = association\_rules(frequent\_itemsets,

metric = "lift",

min\_threshold = 1.0)

*# Print association rules*

rules.info()

<class 'pandas.core.frame.DataFrame'>

RangeIndex: 34 entries, 0 to 33

Data columns (total 9 columns):

antecedents 34 non-null object

consequents 34 non-null object

antecedent support 34 non-null float64

consequent support 34 non-null float64

support 34 non-null float64

confidence 34 non-null float64

lift 34 non-null float64

leverage 34 non-null float64

conviction 34 non-null float64

dtypes: float64(7), object(2)

memory usage: 2.5+ KB

In [52]:

rules.head()

