1. **INTRODUCTION**

Market Basket Analysis most popularly known as Association Rules. Most business organizations produce huge data from their daily transactions. Similarly, customer purchase data is collected in grocery stores. But not all the information is useful. It is important to extract useful information from the given data. The process of extracting useful information is called data mining. This process involves steps like selection, preprocessing, transformation, data mining, and interpretation. The retailers are interested in finding the purchase pattern of the customers. This will help the retailers to analyze business-related data and make decisions in the organization.

The Association analysis is very much helpful in finding the hidden patterns in the large datasets. These patterns or relations are represented in the form of association rules or frequent item sets. The association analysis can also be applied in bioinformatics, medical diagnosis, web mining, and scientific data analysis.

The market basket data can be represented in binary format. Considering the transactional data, each row is corresponding to the transaction and the column refers to the items.

1. **LITERATURE SURVEY**

Data mining has taken an important part in marketing literature in the last few decades. Market basket analysis is also one of the oldest areas of data mining and is most commonly used for mining association rules. Various algorithms are developed by researchers in the field of Association Rule Mining (ARM) and Clustering to help users achieve their objectives.

The market Basket Analysis technique is used to find the items that are bought together and how buying one product influence to purchase another product. MBA is a tool, which can keep track of buying pattern of customers.

Two main researchers named Ramakrishnan Srikant and Rakesh Agrawal developed the apriori algorithm for finding frequent patterns in large datasets and analyzes the customers buying the products and the percentage of total sales of the product. Such association rules are helpful to find the leading products.

**2.1 Existing System**

The existing algorithms work on static data and they do not capture changes in data with time. But proposed algorithm not only mine static data but also provides a new way to take into account changes happening in data. This will be helpful to examine the customer behavior and assists in increasing the sales.

**Disadvantages of existing system**

1. The existing system records the data in ledgers or Excel sheet.
2. Customer data id not maintained properly.
3. Does not simulate any report.
4. Does not identify frequent item set.
   1. **Proposed System**

The proposed system works on Association Rules and Apriori algorithm to mine the data. Apriori algorithm uses frequent item-sets to generate association rules. It is based on the concept that a subset of a frequent item-set must also be a frequent item-set. Frequent Item-set is an item-set whose support value is greater than a threshold value that is nothing but support value.

**Association rule mining:**

Association Rule Mining is used to identify the association between items from the dataset. Association analysis is helpful for discovering relations present in large data set. The rule says that a strong relationship should exist between the items. There are two issues to be discussed are discovering the patterns in the large data set and some of the patterns may be spurious because they may occur by chance.

The association rule is expressed as X → Y, where X and Y are taken as disjoint item sets (i.e. X ∩ Y = ∅). The strength of association rule can be expressed using Support and Confidence.

**Support:**

The number of transactions that contain all the items over the total number transactions. If support value is high, it means that items are more frequently to occur.

The support gives an idea that how many times an item set has occurred in the overall transactions.

Support 𝑆𝑢𝑝𝑝𝑜𝑟𝑡 𝐴 =

**Confidence:**

Confidence is a measure of the likelihood that customer buy product A will buy product B as well. A rule of association is therefore a remark of the form (item set A) ⇒ (item-set B) where

A is the precedent and B is the consequence. Confidence gives the probability of consequence occurring on the cart provided with pre-existing antecedents. For frequently appearing Consequent, it doesn’t matter what the customer have it in the Antecedent. The confidence of an Association rule, which results very often, will always be of greater value.

Confidence 𝐴 =

**Lift**

Given that different items are bought at different frequencies, we want to know that 2-items really do have a strong association.

There are multiple ways to express the formula to calculate lift. Let me first show what the formulas look like, and then I will describe an intuitive way for you to think about it.

Lift (A→B) = Probability (A & B) / [ Support (A) \* Support (B) ]

Lift (A→B) = Confidence (A & B) / Support (B)

* Lift (A => B) = 1 means that there is no correlation within the item-set.
* Lift (A => B) > 1 means that there is a positive correlation within the item-set,

i.e., Products in the item-set, A, and B, are more likely to be bought together.

* Lift (A => B) < 1 means that there is a negative correlation within the item-set, i.e., products in item-set, A, and B, are unlikely to be bought together.

**Conviction**

Conviction is another way of measuring association, although it is a bit harder to get your head around. It compares the probability that A appears without B if they were independent with the actual frequency of the appearance of A without B. Let’s take a look at the general formula:

Conviction (A→B)= (1 - Support (B)) / (1 - Confidence (A→B))

* 1. **Feasibility Study**

Market basket analysis is one possible way to find out which items can be put together. 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.

* 1. **Tools and technologies used**

**Python**

Besides web and software development, Python is used for data analytics, machine learning, and even design. We take a closer look at some of the uses of Python, as well as why it's such a popular and versatile programming language.

**Numpy**

NumPy is an array-processing application for general purposes. It stands for 'Numerical Python'. It is a library of multidimensional array objects, and a set of array processing routines. NumPy has functions built in for linear algebra and the generation of random numbers. NumPy is very useful for performing mathematical and logical operations on Arrays.

**Pandas**

Pandas is an open source, Python licensed library that offers high-performance, easy-to-use data structures, anddata analysis tools to the Python programming language. The Data Frame is the core data structure. Data frame allows tabular data to be stored and manipulated in. Pandas library is mainly used for data analysis. It allows importing data from various file formats such as comma-separated values, JSON, SQL database tables or queries, and Microsoft Excel.

**Matplotlib**

It is the art of displaying data through charts, icons, presentations and more. It is most common to translate complex data for a non-technical audience into comprehensible insights. Matplotlib is one of the most powerful Data Visualization Python packages used. This is a cross - platform framework designed to make Two dimensional graphs from records in arrays. This also provides an object-oriented API which helps, for example, to embed plots into implementations using Python GUI toolkits such as PyQt. This library is essential for drawing the graphs that helps to show the visualization of item-sets.  It helps to understand the huge amount of data through different visualizations.

**MLxtend**

The primary goal of MLxtend is to create widely used tools to focus solely consistency with existing machine learning libraries on user-friendly and intuitive APIs. While MLxtend enforces a wide range of functions, highlights include sequential selection feature algorithms, stacked generalization implementations for classification and regression, and frequent pattern mining algorithms. MLxtend offers a variety of utilities that draw on Python’s scientific computing stack and increasing its capabilities. This library helps to generate the rules common task in the mining of frequent patterns and for frequent pattern mining.

1. **SYSTEM REQUIREMENTS**

**Hardware Requirements**

* Laptop or PC
* Memory: 4GB RAM
* Hard disk: 10GB HDD

**Software Requirements**

* Operating system platform: Windows 10
* Language: Python, Machine Learning
* Software: Jupyter Notebook

1. **SOFTWARE REQUIREMENTS SPECIFICATIONS**

**Functional Requirements:**

The system shall provide insight into customer behavior patterns A two pronged approach was taken to executing this requirement Frequent item set generation Association rule generation based on frequent item sets. The system shall display the insights of rule generated as output.

**Non Functional Requirements:**

**Availability:**

Market basket analysis is a 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.

**Efficiency:**

Market basket analysis enables retailers to set up effective joint promotions by helping them accurately see customer purchasing patterns.

**Flexibility:**

Since it is built on Python, there are chances to add new techniques in future, which is absolutely easy with the python without affecting the existing data.

**Integrity:**

Market basket analysis is a 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.

**Performance:**

Market Basket Analysis takes data at transaction level, which lists all items bought by a customer in a single purchase. The technique determines relationships of what products were purchased with which other product(s). These relationships are then used to build profiles containing If-Then rules of the items purchased.

**5. SYSTEM DESIGN**

**5.1 Data Flow Diagram**

Collect the External Dataset

Data Preprocessing

Apply Support and Confidence

(Building the Model)

Generate Frequent Item sets

(Learn the Model)

Apply FP-Growth Algorithm

Filter records based on minSupport & minConfidence

Final Item-set

Fig 5.1 Data Flow Diagram

A data flow diagram (DFD) maps out the flow of information for any process or system. It uses defined symbols like rectangles, circles and arrows, plus short text labels, to show data inputs, outputs, storage points and the routes between each destination

**Data Preprocessing:** It is a technique that is used to convert the raw data into clean data.

**Support and Confidence:** Support represents the popularity of that product of all the product transactions. ... Confidence can be interpreted as the likelihood of purchasing both the products A and B. Confidence is calculated as the number of transactions that include both A and B divided by the number of transactions includes only product A.

**Frequent Item-set:** An item-set is frequent if it appears in more than a minimum number of transactions. The number of transactions containing an item-set is known as its “support”, and the minimum support (as a percentage of transactions) is a control parameter in the algorithm.

**FP Growth Algorithm:** FP-growth is an improved version of the Apriori Algorithm which is widely used for frequent pattern mining (AKA Association Rule Mining). The Apriori Algorithm produces frequent patterns by generating item-sets and discovering the most frequent item-set over a threshold “minimal support count”.

**minSupport and minConfidence**: Minimum support is applied to find all frequent item-sets in a data set. These frequent item-sets and the minimum confidence constraint are used to compose the rules.

**5.2 Use case Diagram**

Shop Owner

(Apriori)

Fig 5.2 Use caseDiagram

A use case diagram is a graphical depiction of a user's possible interactions with a system. A use case diagram shows various use cases and different types of users the system has and will often be accompanied by other types of diagrams as well.

* **Association rule:** Association rule mining is the data mining process of finding the rules that may govern associations and causal objects between sets of items. So in a given transaction with multiple items, it tries to find the rules that govern how or why such items are often bought together.
* **Data Preprocessing:** It is a technique that is used to convert the raw data into clean data.
* **Support and Confidence:** Support represents the popularity of that product of all the product transactions. Confidence can be interpreted as the likelihood of purchasing both the products A and B. Confidence is calculated as the number of transactions that include both A and B divided by the number of transactions includes only product A.

**5.3 Activity Diagram**

Read the uploaded Data

File Not Found

Data Preprocessing

No Yes

Data Cleaning

Provide minSupport & minConfidence

Generate Frequent Itemset

Draw a graph for visualization

Final Itemset

Fig 5.3 Activity Diagram

Activity diagram is basically a flowchart to represent the flow from one activity to another activity. The activity can be described as an operation of the system. The control flow is drawn from one operation to another. This flow can be sequential, branched, or concurrent.

* + **Data Preprocessing:** It is a technique that is used to convert the raw data into clean data.
  + **Support and Confidence:** Support represents the popularity of that product of all the product transactions. ... Confidence can be interpreted as the likelihood of purchasing both the products A and B. Confidence is calculated as the number of transactions that include both A and B divided by the number of transactions includes only product A.
  + **Frequent Item-set:** An item-set is frequent if it appears in more than a minimum number of transactions. The number of transactions containing an item-set is known as its “support”, and the minimum support (as a percentage of transactions) is a control parameter in the algorithm.
  + **FP Growth Algorithm:** FP-growth is an improved version of the Apriori Algorithm which is widely used for frequent pattern mining (AKA Association Rule Mining). The Apriori Algorithm produces frequent patterns by generating item-sets and discovering the most frequent item-set over a threshold “minimal support count”.
  + **minSupport and minConfidence**: Minimum support is applied to find all frequent item-sets in a dataset. These frequent item-sets and the minimum confidence constraint are used to compose the rules.

**6. IMPLEMENTATION**

**6.1 Source Code:**

**Reading the Dataset:**

os.listdir("mba")

data **=** pd.read\_excel("mba/Dataset.xlsx", header**=**None)

**min Support and min confidence:**

N**=** df.shape[0]

min\_support **=** 1

min\_confidence **=** 35

**Generate frequent 1-itemsets:**

c1 **=** df.sum()

l1 **=** (c1**/** N) **\*** 100

l1 **=** l1[l1 **>=** min\_support]

**Generate 2-itemsets:**

counts **=** []

**for** item **in** k2\_items:

counts.append((df[item[0]]**\***df[item[1]]).sum())

### Generate 3-itemsets

c3 **=** pd.DataFrame()

c3['itemset'] **=** k3\_items

c3['counts'] **=** counts3

c3['support'] **=** (c3.counts**/**N) **\*** 100

**def** find\_itemset(itemset):

**for** i, r **in** l2.iterrows():

**if** len(set(itemset) **-** set(r['itemset'])) **==** 0:

**return** r['counts']

**else**:

**return** 0

**break**

len(counts)

xyz\_conf **=** []

yzx\_conf **=** []

zxy\_conf **=** []

**for** i, r **in** c3.iterrows():

**if** find\_itemset(r['itemset'][0:2]) **==** 0:

xyz\_conf.append(0)

**else**:

xyz\_conf.append((r['counts']**/**find\_itemset(r['itemset'][0:2]))**\***100)

**if** find\_itemset(r['itemset'][1:]) **==** 0:

yzx\_conf.append(0)

**else**:

yzx\_conf.append((r['counts']**/**find\_itemset(r['itemset'][1:]))**\***100)

**if** find\_itemset(r['itemset'][::2]) **==** 0:

zxy\_conf.append(0)

**else**:

zxy\_conf.append((r['counts']**/**find\_itemset(r['itemset'][::2]))**\***100)

**MLXtend library to generate and evaluate rules**

from mlxtend.preprocessing import TransactionEncoder

from mlxtend.frequent\_patterns import apriori, fpmax, fpgrowth, association\_rules

**Converting transformed values to pandas DataFrame**

encoded\_df = pd.DataFrame(encoded\_data, columns=encoding.columns\_)

**Filter the relevent rules based on min\_support and min\_confidence**

rules[ (rules['confidence'] > 0.35) &

(rules['support'] > 0.01) ].reset\_index()

rules

**Generating association rules with updated support and confidence**

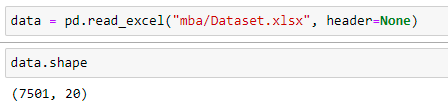
rules1 = association\_rules(frequent\_itemsets3, metric="lift", min\_threshold=1)

print("Number of rules: ", rules1.shape)

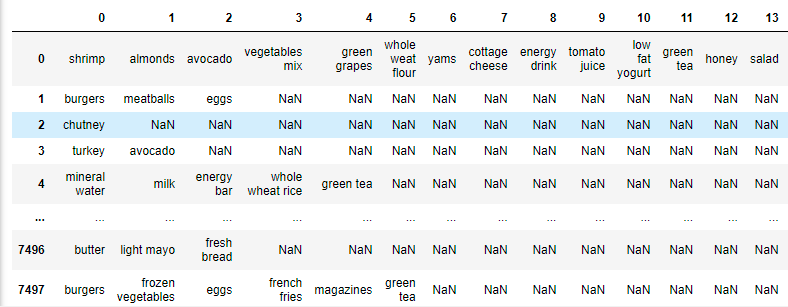
rules1[ (rules1['confidence'] > 0.4) &

(rules1['support'] > 0.03) ].reset\_index()

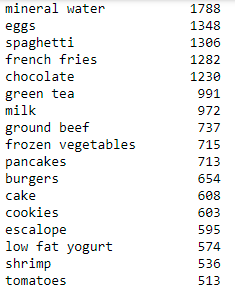
**Screenshots:**

****

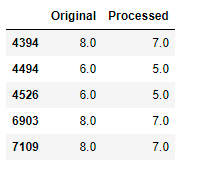
Reading the Dataset



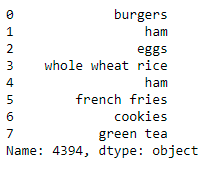
One-hot encoding



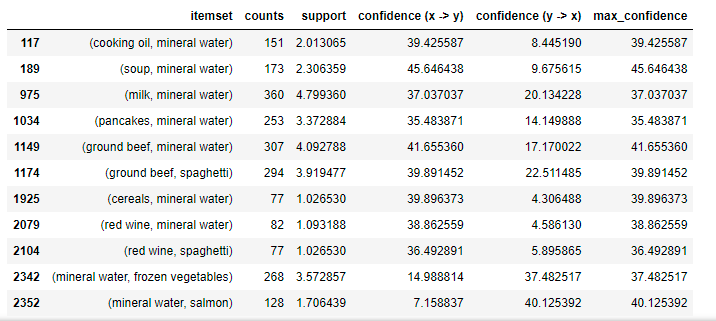
Count the total number of each items



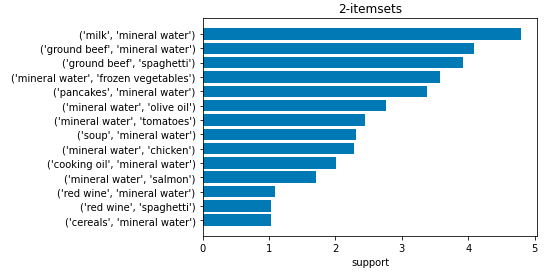
Check for duplicate items and filter

****

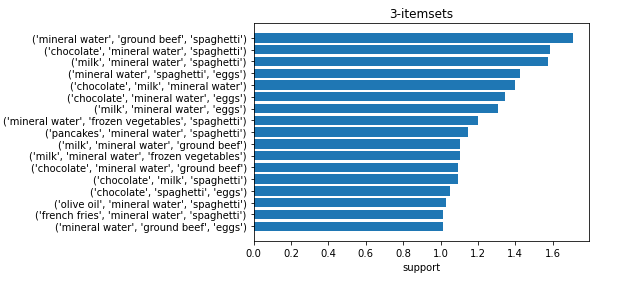
Remove duplicate items

****

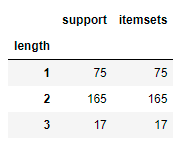
Calculating Support and Confidence for 2-Item-sets

****

Visualize the 2-itemsets sorted based on Support measure

****

Visualize the 3-itemsets sorted based on Support measure

****

Verify Support and count of item-sets with manual and Mlxtend library calculated value

**7. TESTING**

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| **Test Case ID** | **Test Scenario** | **Test Steps** | **Expected Results** | **Actual Results** | **Pass/Fail** |
| TC01 | Reading Dataset | Read Excel Dataset | It should read the dataset | As Expected | Pass |
| TC02 | Show Shape of Dataset | Read Dataset | Display Count of total rows and columns | As Expected | Pass |
| TC03 | Draw Word Cloud figure | Read Dataset to Draw figure | Word Cloud figure by listing all items | Word Cloud figure with items | Pass |
| TC04 | Data Preprocessing | Remove Duplicates and NULL values | Remove duplicates and null values | Preprocessed Data frame | Pass |
| TC05 | Perform One-hot encoding | Assign 0/1 to items | 1-item purchased  0-item not purchased | As expected | Pass |
| TC06 | Generate frequent itemset-1,2,3 | (x)  (x, y)  (x, y, z) | Item set’s | As expected | Pass |
| TC07 | Provide support and confidence | Values for support and confidence | Filtered item set’s based on support and confidence | As Expected | Pass |

**Data Preprocessing:**

orignial\_count = [data.iloc[x, :].dropna().shape[0] for x in range(data.shape[0])]

unique\_count = [data.iloc[x, :].dropna().unique().shape[0] for x in range(data.shape[0])]

orignial\_count1 = pd.Series(orignial\_count)

unique\_count1 = pd.Series(unique\_count)

duplicates = orignial\_count1.compare(unique\_count1)

duplicates.columns =['Original', 'Processed']

duplicates

**Removing Duplicates:**

for val in duplicates.index.values:

    data.iloc[val, :] = pd.Series(data.iloc[val, :].unique()).reindex(range(20))

data.iloc[4394, :].dropna()

c1 = df.sum()

l1 = (c1/ N) \* 100

l1 = l1[l1 >= min\_support]

**Applying MLxtend library to evaluate Rules:**

rom mlxtend.preprocessing import TransactionEncoder

from mlxtend.frequent\_patterns import apriori, fpmax, fpgrowth, association\_rules

dataset = data.fillna("ZERO")

encoding = TransactionEncoder()

encoding.fit(dataset.values)

encoded\_data = encoding.transform(dataset.values)

encoded\_df = pd.DataFrame(encoded\_data, columns=encoding.columns\_)

encoded\_df.drop('ZERO', axis=1, inplace=True)

frequent\_itemsets = apriori(encoded\_df, min\_support=0.01, use\_colnames=True)

frequent\_itemsets['length'] = frequent\_itemsets['itemsets'].apply(lambda x: len(x))

frequent\_itemsets.groupby("length").count()

**Applying FP Growth Algorithm:**

frequent\_itemsets = fpgrowth(encoded\_df, min\_support=0.01, use\_colnames=True)

frequent\_itemsets['length'] = frequent\_itemsets['itemsets'].apply(lambda x: len(x))

frequent\_itemsets.groupby("length").count()

**8. FUTURE ENHANCEMENTS**

Project can be improved by implementing new and advanced mining algorithms along with apriori, FP growth for better performance and fast results for sparse dataset .In the current approach, we only use association rules to exploit the collective information i.e. building an model by finding similarity between customers’ products associations and recommending an similar associated item to another customer to purchase. In future work, association rules can also be used to exploit the content-based information i.e. finding a similarity between products, and recommending an products based on interest of a similar products. Content based recommendation system is not based on a lot of user data since the calculation of similarities takes place at the product level. Perhaps we can build recommendation system in future work, incorporating the two approaches into a hybrid approach that can benefit from the strengths of both item-based and customer-based approaches. This application can be extended to other areas such as: sales tracking, product tracking, discount and calculation of prices etc. This method can be applied in future to very large databases where memory space is valuable and needs enhancement. It can be further tuned for improved efficiency and performance.

**9. CONCLUSION**

At present many 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 x 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. x Predicting future association rules as well as gives us right methodology to find out outliers. Authors suggested that, some areas are still there which need to be focused on. Firstly, results have influenced greatly by the manual threshold values for score, so it is needed to automate the threshold values for better recognition of outliers. Secondly, this approach is specifically targeted at Market Basket Data, it may perhaps be extended to other areas.

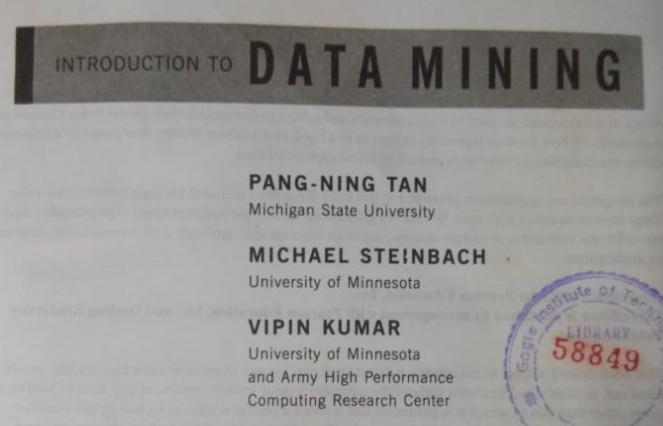
**10. BIBLOIGRAPHY**

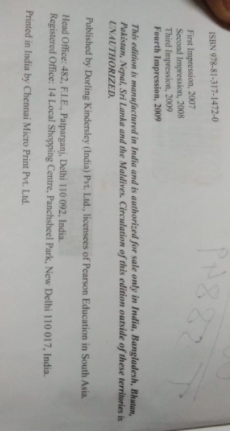
**Textbook References:**

1. Pang-Ning Tan/Michael Steinbach/Vipin Kumar, Introduction To Data Mining, Chennai Micro Print Pvt. Ltd, 4th edition.

**Website References:**

1. <https://www.sciencedirect.com/science/article/pii/S1877050916305208>
2. <https://www.diva-portal.org/smash/get/diva2:1031325/FULLTEXT02>
3. <https://towardsdatascience.com/association-rules-2-aa9a77241654>
4. <http://rasbt.github.io/mlxtend/user_guide/frequent_patterns/apriori/>
5. [https://towardsdatascience.com/apriori-algorithm-for-association-rule-learning- how-to-find-clear-links-between-transactions-bf7ebc22cf0a](https://towardsdatascience.com/apriori-algorithm-for-association-rule-learning-%20%20%20%20how-to-find-clear-links-between-transactions-bf7ebc22cf0a)
6. <https://www.researchgate.net/publication/343484851_Market_Basket_Analysis_Recommendation_System_Using_Association_Rules>

****

**See the textbook references and refer these images**

****

**author / title / publisher / edition**

sequence of papers

certificate

decl

ack

abstract

table of content