Engineer to Excel

SAVEETHA
SCHOOL OF ENGINEERING
Approved by AICTE | IET-UK Accreditation

PROJECT TITTLE

MARKET BASKET ANALYSIS

A PROJECT REPORT

Submitted to

Department of Computer Science and Information Technology

SAVEETHA SCHOOL OF ENGINEERING

In partial fulfillment of the requirements for the Bachelor's Degree in Computer Science and Information Technology

Submitted by

K.vishnu vardhan(192224265)

R.Tharun (192210662)

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

TABLE OF CONTENTS

LIST OF FIGURES

LIST OF TABLES

CHAPTER 1: INTRODUCTION

1.1 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.

.

1.2 Problem 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

1.3 Objectives

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

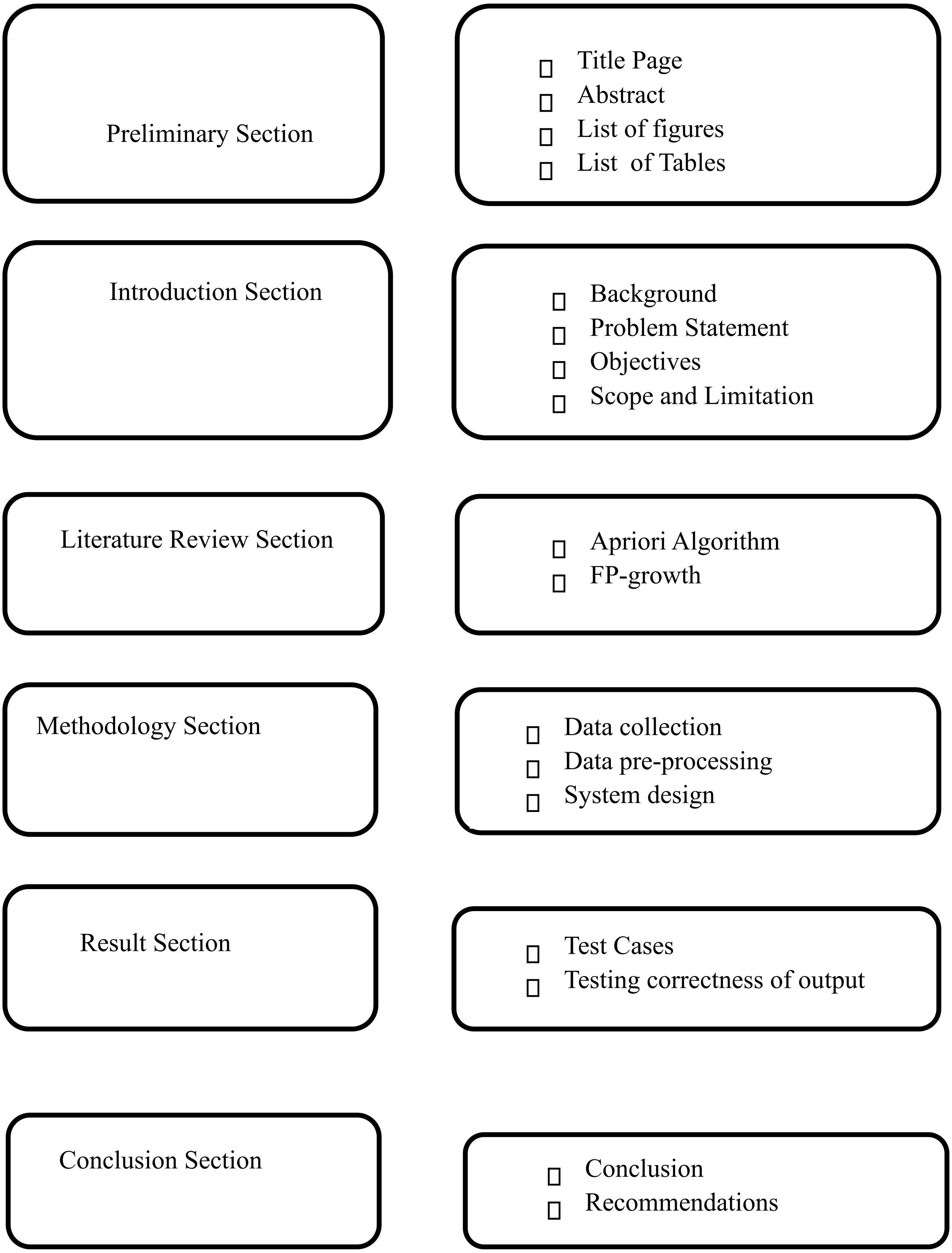
1.4 Scope

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

1.5 Limitations

a. The application will be desktop and will not be available online.
b. 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 Report Organization



Preliminary Section

- Title Page
- Abstract
- List of figures
- List of Tables

Introduction Section

- Background
- Problem Statement
- Objectives
- Scope and Limitation

Literature Review Section

- Apriori Algorithm
- FP-growth

Methodology Section

- Data collection
- Data pre-processing
- System design

Result Section

- Test Cases
- Testing correctness of output

Conclusion Section

- Conclusion
- Recommendations

CHAPTER 2: REQUIREMENT ANALYSIS AND FEASIBILITY

2.1 Literature Review

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.

Rakesh Agarwal [2] proposed the Apriori algorithm. Apriori was the first associative algorithm proposed and future development in assocuaton,classification,associative classification algorithms have used apriori as part of the technique. Apriori algorithm is a level-wise, breadth-first algorithm which counts transactions Apriori algorithm uses prior knowledge of frequent item set properties. Apriori uses an iterative approach known as a level-wise search, in which n-item sets are used to explore (n+1) - item sets. To improve the efficiency of the level-wise generation of frequent item sets Apriori property is used here. Apriori property insists that all non-empty subsets of a frequent item set must also be frequent. This is made possible because of the anti-monotone property of support measure

- the support for an item set never exceeds the support for its subsets. A two-step process consists of join and prune actions are done iteratively

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

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,

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

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. Phani Prasad J, Murlidher Mourya [7] in this paper author states that there are lots of case studies about the a ssociation 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.

Demerits

a. It scans the database lot of times i.e. every time it runs it scans database every time, this results in shortage of memory to store the data.

b. The I/O load is not sufficient therefore it takes a lot of time to process and exhibits low efficiency.

c. The time complexity is very high.

Solution to improve efficiency

a. Group items into higher conceptual groups e.g. white and brown bread become "bread".

b. Reduce the number of scan of the entire database.

2.2 Requirement Analysis

2.2.1 Functional requirement

Table 1- Functional requirement

Functional Requirements	Description	Comments
FR1	The system shall take input as a file of data containing the transaction data	Data exploration was done manually. Data had to be mapped to integer values to give input to the system.
FR2	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.
FR3	The system shall display the insights of rule generated as output.	Report can be available in the file.

2.2.2 Non-functional requirement

Table 2- Non-functional requirement

Non-Functional Requirements	Category	Description
NFR1	System Availability	The system shall be available more than 99% of the time
NFR2	Maintainability	The system shall be easy to maintain.
NFR3	Responsiveness	The system shall respond in a timely fashion to user’s requests.
NFR4	Performance	The system shall respond in a timely fashion and will not consume inordinate amounts of system resources.
NFR5	Correctness	The system shall return valid and correct data and results to user requests.
NFR6	Supportability	The system must be easy to support.

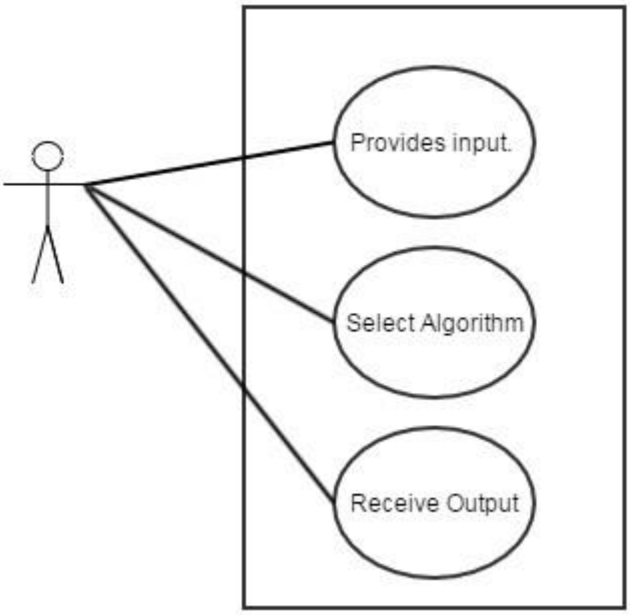


Figure 1- Use case diagram

The user will provide input to the application .The input is a text file where items are mapped into integer's value. The row represents the items that were purchased in one single transaction and column represents the transaction.

The user then can select the algorithm to run and provide the necessary parameters i.e. confidence and support.

After processing the user will receive output in to the desired path that the user wants.

The output will be a text file containing association rules.

2.3 Feasibility Analysis

The feasibility of the project is analyzed in this phase and business proposal is put forth with a very general plan for the project and some cost estimates. During system analysis the feasibility study of the proposed system is to be carried out. This is to ensure that the proposed system is not a burden to the company. Three key considerations involved in the feasibility analysis are

2.3.1 Technical feasibility

This study is carried out to check the technical feasibility, that is, the technical requirements of the system. Any system developed must not have a high demand on the available technical resources. This will lead to high demands on the available technical resources. This will lead to high demands being placed on the client. The developed system must have a modest requirement, as only minimal or null changes are required for implementing this system.

2.3.2 Operational feasibility

The project is operationally feasible as the user having basic knowledge about computer and Internet can use. Furthermore the project can be easily used if the computers have no internet access.

2.3.3 Schedule feasibility

The schedule feasibility analysis is carried out using the CPM method. CPM was used to identify critical tasks and calculate the interrelationship between tasks. The plan was carried out which defined critical and non-critical tasks with the goal of preventing timeframe problems and process bottlenecks. The CPM analysis was carried out as shown in

Figure 2.

Table 3- Activities

Activity	**Time (weeks)**	**Predecessor**
Data Collection and Mapping(A)	1	-
Data Preprocessing (B)	1	A
Implement Apriori Algorithm (C)	4	A,B
User Interface Design (D)	2	C
Testing (E)	5	D
Documentation (F)	9	B

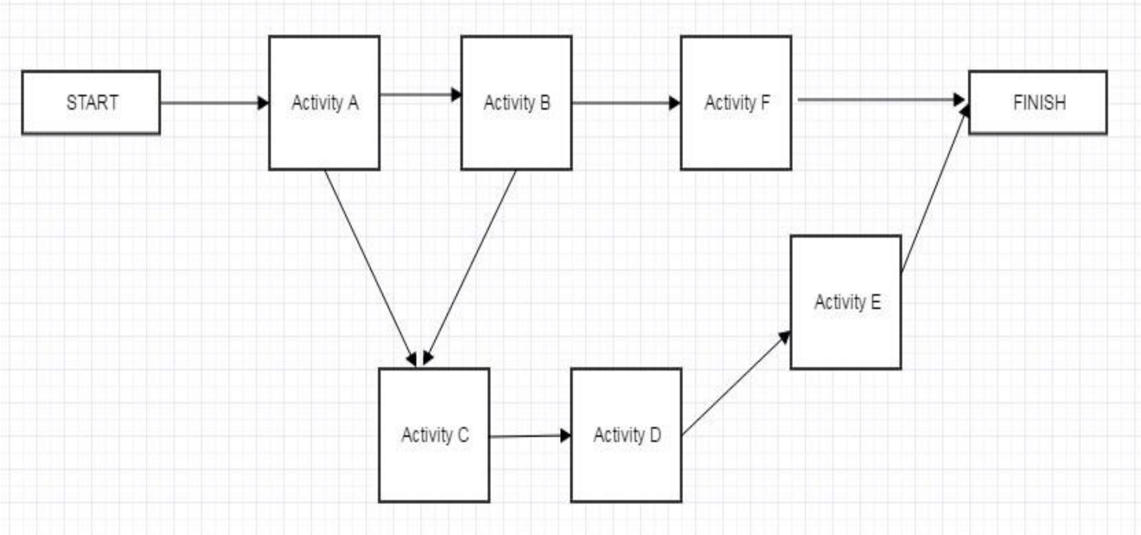


Figure 2- Critical path

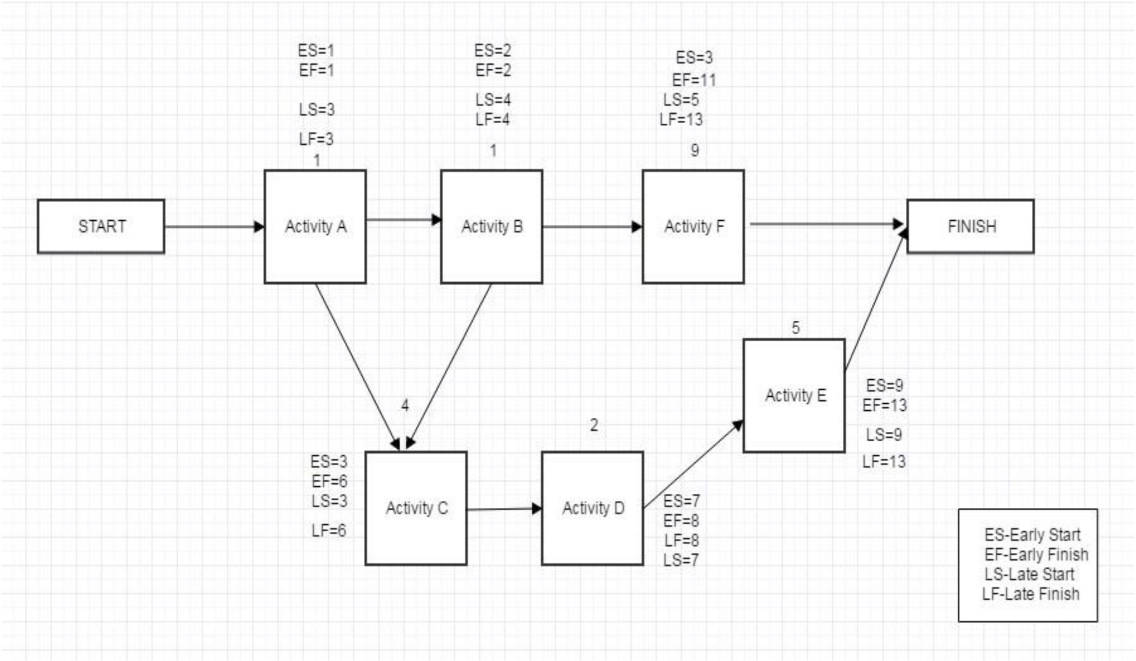


Figure 3- Critical path schedule

From Figure 2 and Figure 3 we can see that the application was completed in 13 weeks which is within 15 weeks of a semester. Hence, the project was determined to be feasible in terms of schedule

CHAPTER 3: SYSTEM DESIGN

3.1 Methodology

3.1.1 Data collection

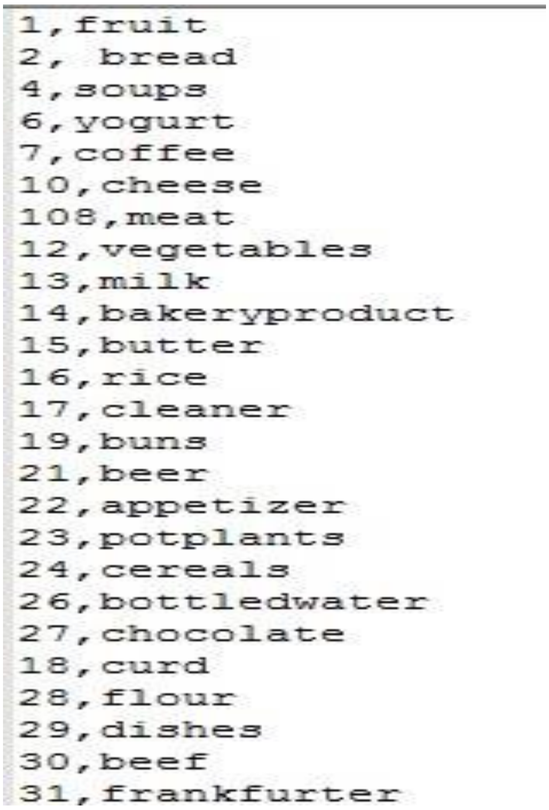
The data was collected from http://www.salemmarafi.com/wp content/uploads/2014/03/groceries.csv due to the unavailability of data from the supermarkets.

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				

3.1.2 Data preprocessing

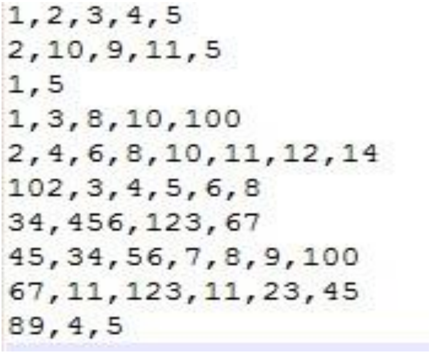
The data collected was mapped manually as integer values as shown in Figure 4. For example the “Fruit” was labeled as 1, “Bread” as 2 “Soups” as 4 and so on.



```
1,fruit
2, bread
4,soups
6,yogurt
7,coffee
10,cheese
108,meat
12,vegetables
13,milk
14,bakeryproduct
15,butter
16,rice
17,cleaner
19,buns
21,beer
22,appetizer
23,potplants
24,cereals
26,bottledwater
27,chocolate
18,curd
28,flour
29,dishes
30,beef
31,frankfurter
```

Figure 4- 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.



```
1,2,3,4,5
2,10,9,11,5
1,5
1,3,8,10,100
2,4,6,8,10,11,12,14
102,3,4,5,6,8
34,456,123,67
45,34,56,7,8,9,100
67,11,123,11,23,45
89,4,5
```

Figure 5- 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.

3.1.3 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.

a. Support
b. Confidence

Support

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.

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

Confidence

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).

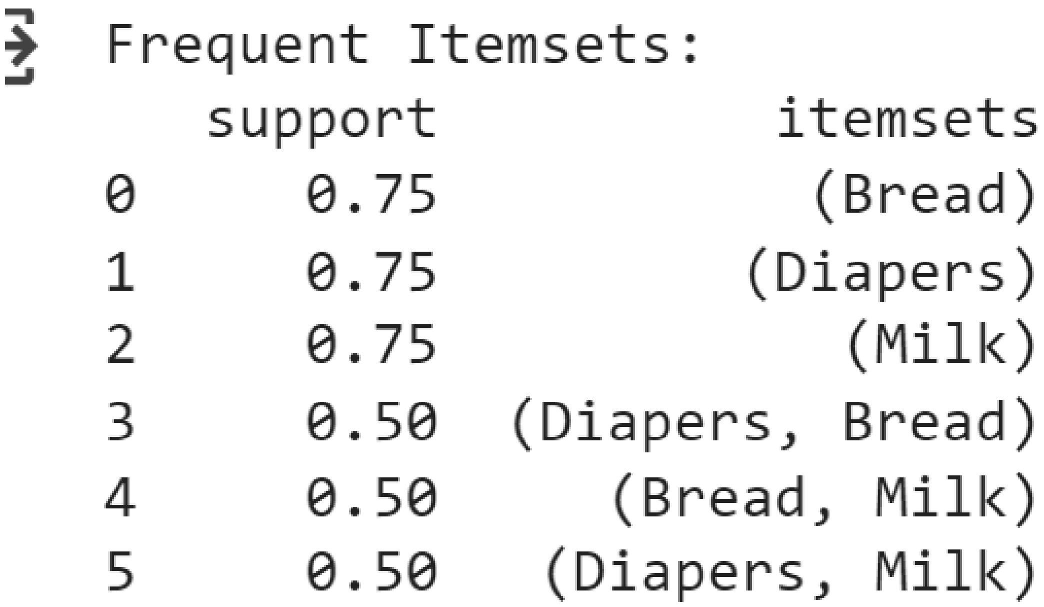
Confidence determines how frequently item in RHS appears in the transaction that Contain LHS. While determining the rules we must measure these two components as it is very important to us. A rule that has very low support may occur simply by chance.

Code

```
# Install mlxtend library if not already installed
# pip install mlxtend

from mlxtend.frequent_patterns import apriori
from mlxtend.frequent_patterns import association_rules
import pandas as pd

# Sample transaction data
data = {'TransactionID': [1, 1, 2, 2, 2, 3, 3, 4, 4, 4],
        'Item': ['Milk', 'Bread', 'Milk', 'Diapers', 'Beer',
                 'Bread', 'Diapers', 'Milk', 'Bread', 'Diapers']}
df = pd.DataFrame(data)

# Convert transaction data to one-hot encoded format
basket = (df
          .groupby(['TransactionID', 'Item'])['Item']
          .count().unstack().reset_index().fillna(0)
          .set_index('TransactionID'))
```

```
# Convert item counts to binary values

basket_encoded = basket.applymap(lambda x: 1 if x > 0 else 0)


# Generate frequent itemsets using Apriori algorithm

frequent_itemsets = apriori(basket_encoded, min_support=0.5, use_colnames=True)


# Generate association rules

rules = association_rules(frequent_itemsets, metric="lift", min_threshold=1)


print("Frequent Itemsets:")

print(frequent_itemsets)

print("\nAssociation Rules:")

print(rules)
```

OUTPUT



```
Frequent Itemsets:
   support          itemsets
0     0.75           (Bread)
1     0.75         (Diapers)
2     0.75            (Milk)
3     0.50  (Diapers, Bread)
4     0.50     (Bread, Milk)
5     0.50   (Diapers, Milk)
```

3.2 System Design

3.2.1 Class diagram

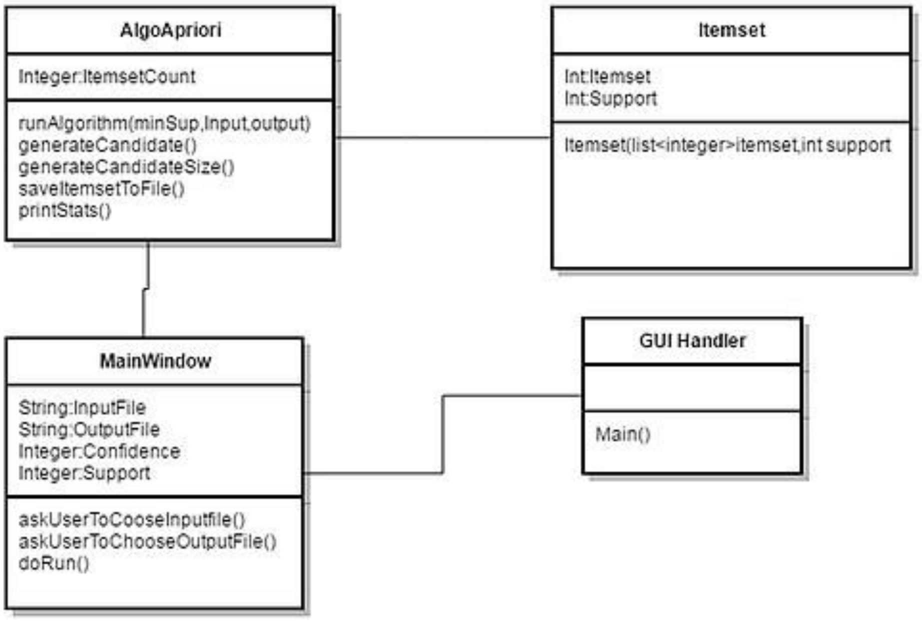


Figure 6- Class diagram

As shown in Figure 6, there are three main classes used in the application

The mainWindow class is used to present the user interface for choosing the input file and output file as desired by the user.

The AlgoApriori is the class that performs all the calculations once the data is provided by the user.It generates the candidate item sets and determines the size of the item sets. Finally the statistics are provide to the user in the same GUI and output is written to the desired file.

The item set class stores the items as the array of integer and provides the support of the respective item from the given input data

3.2.2 Sequence diagram

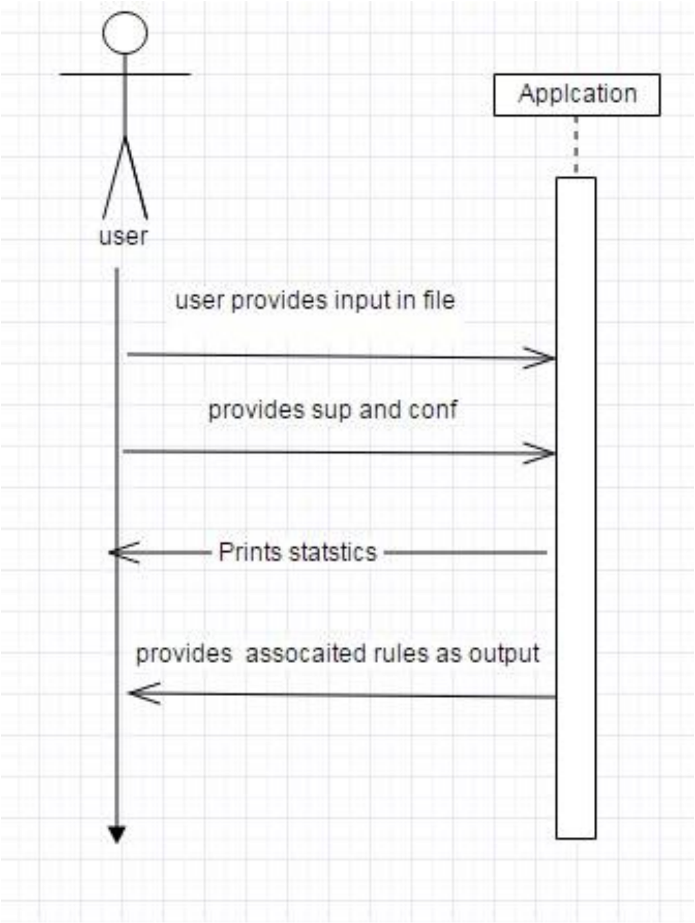

Applcation
user
user provides input in file
provides sup and conf
Prints statstics
provides assocaited rules as output

Figure 7- Sequence diagram Figure 7 shows the sequence diagram of this application.

The user needs to choose the input file that is going to be processed. The file should contain the data in integer where the row represents the items and column represents transactions.

Confidence and support should be provided by the user. After all the input is given the application process the data and provide the output to the user.

CHAPTER 4: IMPLEMENTATION AND TESTING

4.1 Implementation

This project is implemented in java. For the user interface to provide the input data java swing is used to design the interface. The apriori algorithm is used to process the data and generated the association rule as a output in a file.

4.1.1 Tools used

Intellij: Intellij was used as a IDE to develop this application

Java swing: Java swing was used for designing the user interface. **Java**:

java programming language was used to implement the algorithm **File**:

File is used as the database to process the data.

4.1.2 Description of major classes and methods.

MainWindow

This class is used to display the user interface for providing the input to the system. This class extends the JFrame class

Some of the important methods of this class are:

ActionPerformed() askUserToChooseInputFile()

askUserToChooseOutputFile()

AlgoApriori

This is the main class that executes the algorithm after user presses the run algorithm button.This class performes the processiong of the data from user input and display output to the user decided output file.

Some of the important methods of this class are:

runAlgorithm(double minsup, String input, String output)

generateCandidate2(List<Integer> frequent1)

generateCandidateSizeK(List<Itemset> levelK_1) printStats()

Itemset

This class stores the items as the array of integer and provides the support of the respective item from the given input data.

Some of the important methods of this class are:

getAbsoluteSupport() getItems()

4.2 Testing

The purpose of testing is to discover errors. Testing is the process of trying to discover every conceivable fault or weakness in a work product. It provides a way to check the functionality of components, subassemblies, assemblies and/or a finished product. It is the process of exercising software with the intent of ensuring that the Software system meets its requirements and user expectations and does not fail in an unacceptable manner.

Unit testing was performed to test correctness of different modules.

Test case 1: Correctness of the output.

4.2.1 Testing correctness of the output.

In this test approach sample input was given to the system and different support and confidence was provided. At first large sample input were given to the system with different support and confidence value. Then few sample input were given to the system with different support and confidence. The result of the test is shown in Table 5 and Table 6.

Table 5- Test result for large sample input

No of transaction	Support	Confidence	No of rules generated	No rules with confidence=100%
24	10%	40%	21	12
	20%	40%	2	1
	30%	40%	0	0

As show in the table 5 sample data of 24 transaction was taken as input to the system and when the support was 10% and confidence 40% 21 rules were generate from which 12 rules were found to be strong with 100% confidence. When the support was 20% with same confidence then 2 rules was generated from which 1 rule was found o be strong and when support was 30 % no rule was generated.

Table 6- Test result for small sample input

No of transaction	Support	Confidence	No of rules generated	No rules with confidence=100%
7	10%	50%	62	31
	10%	60%	31	31
	20%	70%	10	10
	30%	70%	0	0

As show in the Table 6 sample data of 7 transaction was taken as input to the system and when the support was 10% and confidence 50% 62 rules was generated from which 31 rules was found to be strong with 100% confidence. When the support was 10% with

60% confidence then 31 rules was generated from which 31 rules was found to be strong. When support was 20% and confidence was 70% 10 rules was generated from which 10 rules was found to be strong and when support was 30 % no rule was generated.

CHAPTER 5: MAINTENANCE AND SUPPORT

Table 7 represents the maintenance and support plan that is carried out for the application.

Table 7- Maintenance and support plan

Category	**Activity**
Adaptive	• Accommodation of changes to data and to hardware and software • Changes in the external environment
Perfective	• Emergency fixes • User enhancement • Improved documentation • Recording of computation efficiency • User recommendations for new capabilities

CHAPTER 6: CONCLUSION

6.1 Conclusion

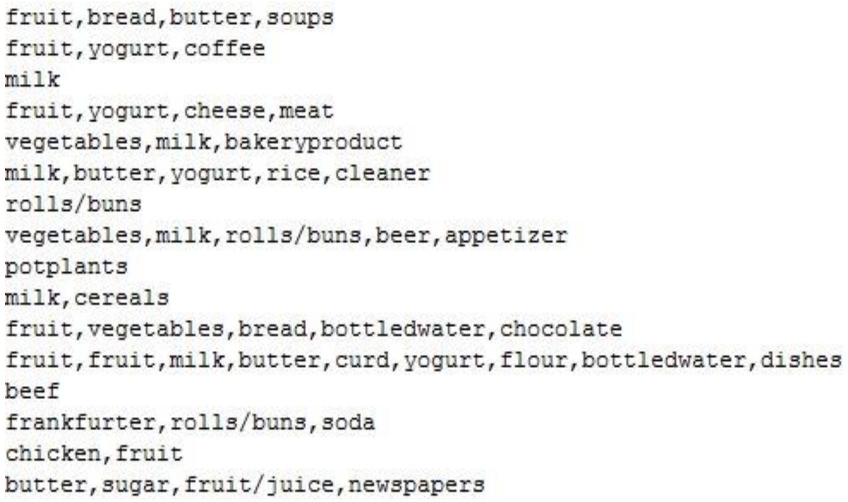
The Apriori algorithm effectively generates highly informative frequent itemsets and association rules for the data of the supermarket. The frequent data items are generated from the given input data and based on the frequent item stets strong association rules were generated.

6.2 Recommendations

The input data given to the application is used as the integer value mapped from the transaction database. The mapping is done manually. If database converter is made then the system will work effectively for any format of data. The application can be efficiently used by using more efficient algorithm rather that Apriori Algorithm in future.

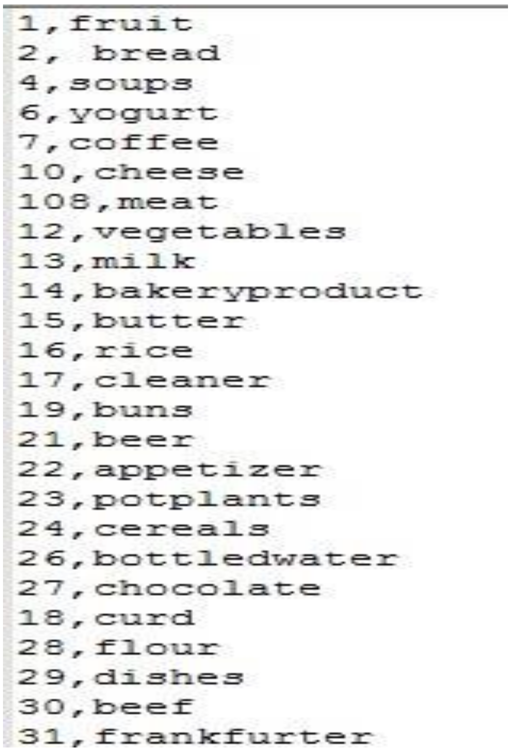
APPENDIX

Sample Data



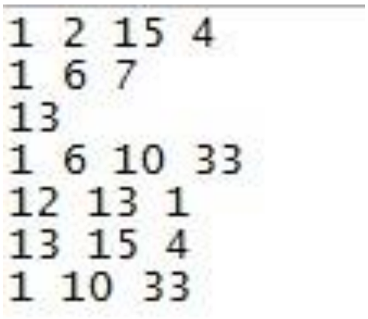
```
fruit,bread,butter,soups
fruit,yogurt,coffee
milk
fruit,yogurt,cheese,meat
vegetables,milk,bakeryproduct
milk,butter,yogurt,rice,cleaner
rolls/buns
vegetables,milk,rolls/buns,beer,appetizer
potplants
milk,cereals
fruit,vegetables,bread,bottledwater,chocolate
fruit,fruit,milk,butter,curd,yogurt,flour,bottledwater,dishes
beef
frankfurter,rolls/buns,soda
chicken,fruit
butter,sugar,fruit/juice,newspapers
```

Data mapped from sample data



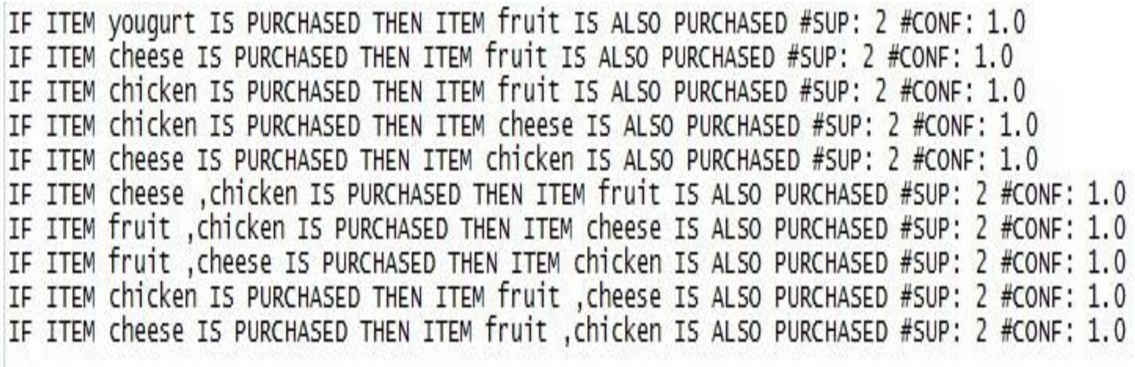
```
1,fruit
2,  bread
4,soups
6,yogurt
7,coffee
10,cheese
108,meat
12,vegetables
13,milk
14,bakeryproduct
15,butter
16,rice
17,cleaner
19,buns
21,beer
22,appetizer
23,potplants
24,cereals
26,bottledwater
27,chocolate
18,curd
28,flour
29,dishes
30,beef
31,frankfurter
```

Input data to the system



```
1 2 15 4
1 6 7
13
1 6 10 33
12 13 1
13 15 4
1 10 33
```

Output rules generated



```
IF ITEM yougurt IS PURCHASED THEN ITEM fruit IS ALSO PURCHASED #SUP: 2 #CONF: 1.0
IF ITEM cheese IS PURCHASED THEN ITEM fruit IS ALSO PURCHASED #SUP: 2 #CONF: 1.0
IF ITEM chicken IS PURCHASED THEN ITEM fruit IS ALSO PURCHASED #SUP: 2 #CONF: 1.0
IF ITEM chicken IS PURCHASED THEN ITEM cheese IS ALSO PURCHASED #SUP: 2 #CONF: 1.0
IF ITEM cheese IS PURCHASED THEN ITEM chicken IS ALSO PURCHASED #SUP: 2 #CONF: 1.0
IF ITEM cheese ,chicken IS PURCHASED THEN ITEM fruit IS ALSO PURCHASED #SUP: 2 #CONF: 1.0
IF ITEM fruit ,chicken IS PURCHASED THEN ITEM cheese IS ALSO PURCHASED #SUP: 2 #CONF: 1.0
IF ITEM fruit ,cheese IS PURCHASED THEN ITEM chicken IS ALSO PURCHASED #SUP: 2 #CONF: 1.0
IF ITEM chicken IS PURCHASED THEN ITEM fruit ,cheese IS ALSO PURCHASED #SUP: 2 #CONF: 1.0
IF ITEM cheese IS PURCHASED THEN ITEM fruit ,chicken IS ALSO PURCHASED #SUP: 2 #CONF: 1.0
```

The output is generated when a support of 22% and confidence of 70% is provided to the system. Here, the first rule states that when a person buys item yogurt then item fruit is also purchased. The item yogurt and fruit are correlated with each other.

REFERENCES

[1] W. Yanthy, T. Sekiya, K. Yamaguchi,"Mining Interesting Rules by association and Classification Algorithms", FCST 09.

[2] R. Agrawal and R. Srikant," Fast Algorithms for Mining Association Rules in Large Databases", Journal of Computer Science and Technology, vol. 15

[3] X. Yin, J. Han, "CPAR: Classification based on Predictive Association Rules", Proceedings of the Third SIAM International Conference on Data Mining, pp 331-335, 2003. 9.Gourab Kundu , Sirajum Munir, Md. Faizul Bari, Md. Monirul Islam, and K. Murase, "A Novel Algorithm for Associative Classification",14thInternational Conference, ICONIP 2007, Kitakyushu, Japan, pp 453-459 , November 13-16, 2007.

[4] J. Han, J. Pei, and Y. Yin, "Mining frequent patterns without candidate generation", Proc of the ACM SIGMOD International Conference on, vol. 1, , pp. 1-12, 2000. 5.

[5] J. Han, J. Pei, Y. Yin, and R. Mao, "Mining Frequent Patterns without Candidate Generation: A Frequent-Pattern Tree Approach"

[6] Phai Prasad J, Murlidher Mourya, "A Study On Market basket Analysis Using Data