Customer Satisfaction
using
Market Basket Analysis

A

Project Report

Submitted in partial fulfillment of the

Requirements for the award of the degree of

BACHELOR OF TECHNOLOGY

IN

COMPUTER SCIENCE & ENGINEERING

Specialization in

Business Analytics and Optimization

BY

Name	Roll No.	SAPID
UTKARSH SANDEEP SINGH	R103216110	500053648
TUSHAR SINGH	R103216109	500054191
SHANKEY GUPTA	R103216089	500054226
ADESH KUMAR GUPTA	R103216006	500053000

Under the guidance of

Dr. Hitesh Kumar Sharma
AP & PIC

Department of Informatics

School of Computer Science

University of Petroleum & Energy Studies

Bidholi, via Prem Nagar, Dehradun, UK

September 2018

CANDIDATES DECLARATION

We hereby certify that the project work entitled **Customer Satisfaction using Market Basket Analysis** in partial fulfilment of the requirements for the award of the Degree of Bachelor of Technology in Computer Science And Engineering with Specialization in Business Analytics and Optimization and submitted to the Department of Informatics at School of Computer Science, University of Petroleum And Energy Studies, Dehradun, is an authentic record of our work carried out during a period from **September, 2018 to December, 2018** under the supervision of **Dr Hitesh Kumar Sharma**, **AP & PIC.**

The matter presented in this project has not been submitted by us for the award of any other degree of this or any other University.

(Name of Student(s))

Utkarsh, Tushar, Shankey, Adesh

Roll No.

R103216110, R103216109,

R103216089, R103216006

This is to certify that the above statement made by the candidate is correct to the best of my knowledge.

(Date: 17 December 2018)

(Name of Guide)
Project Guide

Dr. T.P Singh

Head
Department of Informatics
School of Computer Science
University of Petroleum And Energy Studies
Dehradun - 248 001 (Uttarakhand)

ACKNOWLEDGEMENT

We wish to express our deep gratitude to our guide Name, for all advice, encouragement and constant support he has given us throughout our project work. This work would not have been possible without his support and valuable suggestions.

We sincerely thank to our Head of the Department, Dr. T.P Singh, for his great support in doing our project name at SoCS.

We are also grateful to Dr. Manish Prateek Professor and Director SoCS, UPES for giving us the necessary facilities to carry out our project work successfully.

We would like to thank all our friends for their help and constructive criticism during our project work. Finally we have no words to express our sincere gratitude to our parents who have shown us this world and for every support they have given us.

Name	Shankey Gupta	Tushar Singh	Utkarsh	Adesh
Roll No.	R103216089	R103216109	R103216110	R103216006

Abstract

A data mining approach used in markets like super markets are targeting more customers by using Market Basket Analysis using Apriori Algorithm. By the help of this technique, the store manager will able to fulfill the requirements more as per need of the customer by showing him other set of products that are closely related to a specific product known as combinations. This in turn will help in reducing the efforts of the customers by overcoming the problem of increased lookouts for other products.

We will be using this technique of Data Mining to overcome the challenges of most of the business of marts and retailers as in todays' generation the major factor is "Population" which is increasing day by day globally. As more population results in more production of consumer goods as well as leading to more retailers and businesses. With all these the major issue is high competition among business organizations.

So, to remain standing in a challenge we came up with a solution. That is "Customer Satisfaction using Market Basket Analysis". This technique is already present in market but we are implementing it in a procedural language to provide a cost-efficient product and can be used by small scale retailers as well. Many marts and super markets are using these techniques to implement the same and target more customers. This will prevent the loss of small businesses, if a mart or supermarket is established near to it.

TABLE OF CONTENTS

Contents

1. Introduction:

As in today's generation many marts and wholesalers are competing with each other to provide more satisfaction to their customers. As if the needs of the customer are fulfilled by their retailers along with their less efforts will create a chance of more shopping of the different products by the customer. To understand how to analyze this pattern we are going to implement Market Basket Analysis -a technique of Data Mining by the help of Apriori Algorithm.

By this technique, we are able to understand how the products can be arranged in a mall or a store so that very less effort applied by the customer can give their products as per their needs even if they are not planned to buy it.

Taking an example of milk, if a customer goes to a mall to buy a packet of milk, then if the stock of bread is kept along with the milk then there will be a very high probability of that customer he will buy a packet of bread also along with the milk. To understand this only we are implementing Market Basket Analysis, that how this actually works.

This technique will be very useful for the customer as well as the shopkeeper as well because, talking for the customer, very less effort will be applied by him to reach the desired product and it also results in increase of chance to buy a product which he hadn't planned earlier for it. Now talking by the point of shopkeeper it will increase in their sale of products if the similar categories of products are kept along with it which will results in more profit and to challenge other shopkeeper in the market.

2. Problem Statement:

Nowadays people buy daily goods from super market and marts. The problem which many retailers face is the arrangement of items in their store. They don't know what item should be put at what place in their store so that all the associated products are available nearly.

3. Literature Review:

Today, the large amount of data [1] is being maintained in the databases in various fields, like retail markets, banking sector, medical field etc. But it is not necessary that the whole information is useful for the user. That is why, it is very important to extract the useful information from large amount of data. This process of extracting useful data is known as data mining or A Knowledge Discovery and Data (KDD) Process. The overall process of finding and interpreting patterns from data involves many steps such as selection, preprocessing, transformation, data mining and interpretation.

Frequent itemset mining [2] leads to the discovery of associations and correlations among items in large transactional or relational datasets. With massive amounts of data continuously being collected and stored, many industries are becoming interested in mining such patterns from their databases. The discovery of interesting correlation relationships among huge amount of business transaction records can help in many business decision-making processes such as catalog design, cross-marketing, and customer shopping behavior analysis.

Data Mining [3] 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 are to extract the information from their vast customer databases, in order to gain competitive advantage.

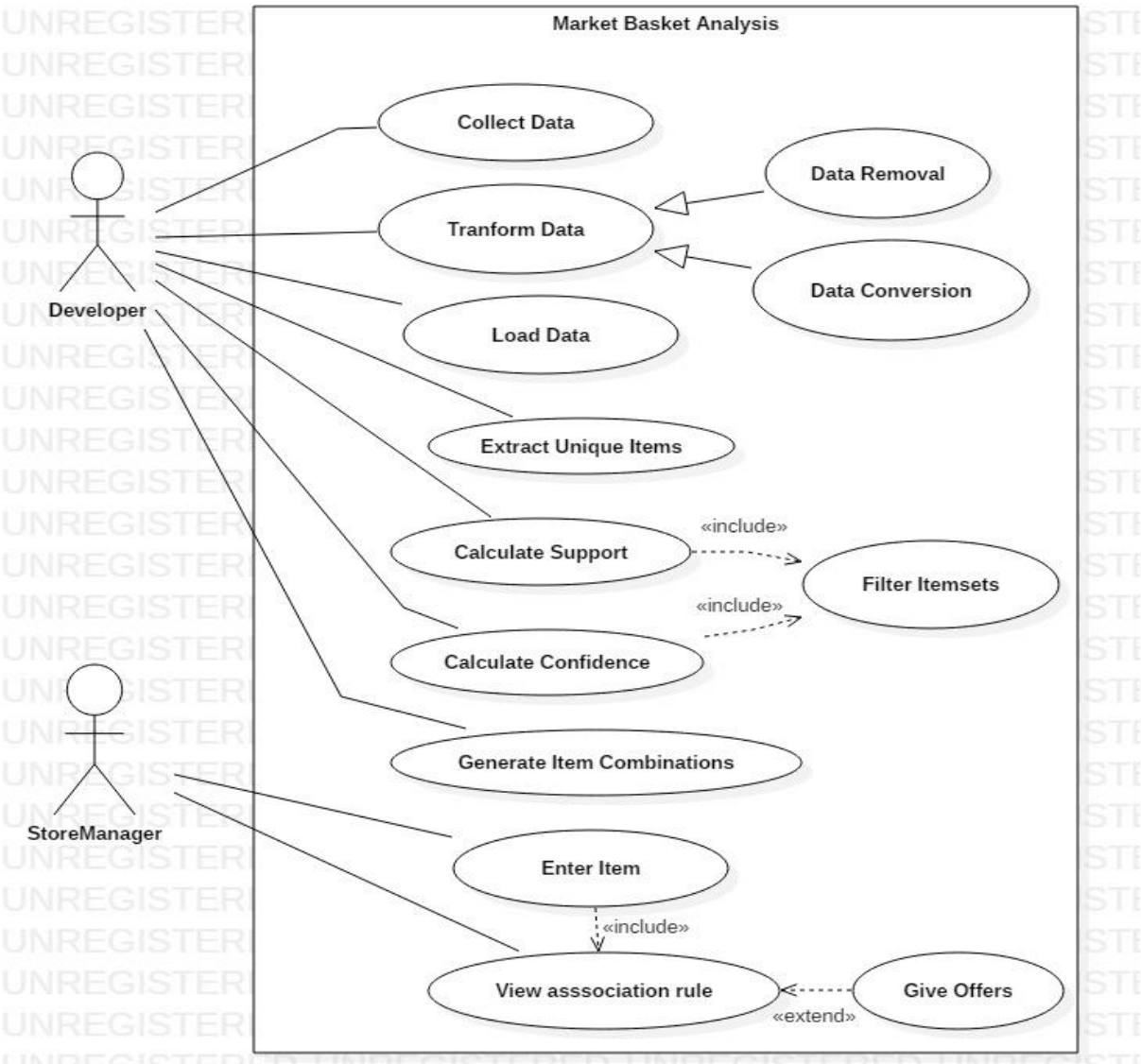
Apriori Algorithm is [4] also called *level-wise algorithm*. It was proposed by Agrawal and Srikant in 1994. It is the most popular algorithm to find all the frequent sets. It makes use of the downward-closure property. As the name suggests, the algorithm is a bottom-up search, moving upward level-wise in the lattice. However, the nicety of the method is that before reading the database at every level, it graciously prunes many of the sets which are unlikely to be frequent sets.

4. Objectives:

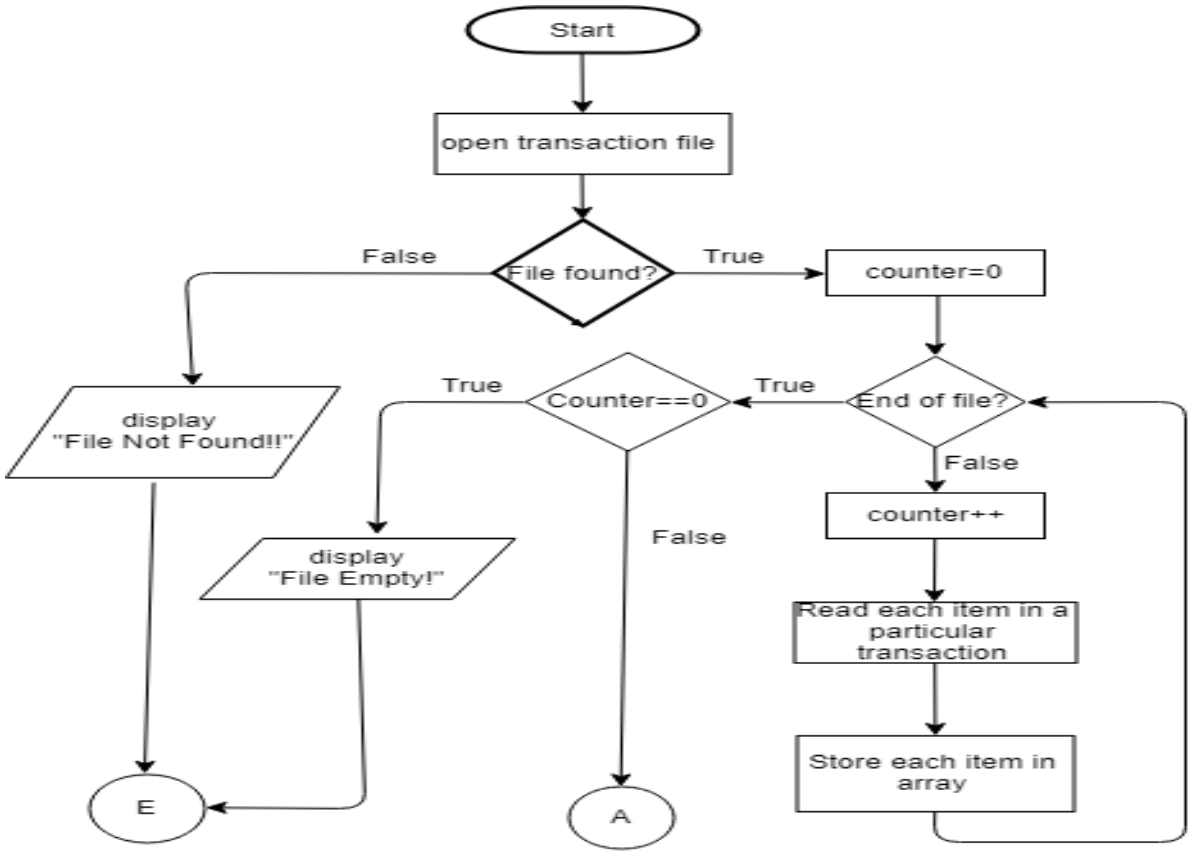
a) To identify the frequent items from the transaction on the basis of support and confidence.
b) To generate the association rule from the frequent item sets.

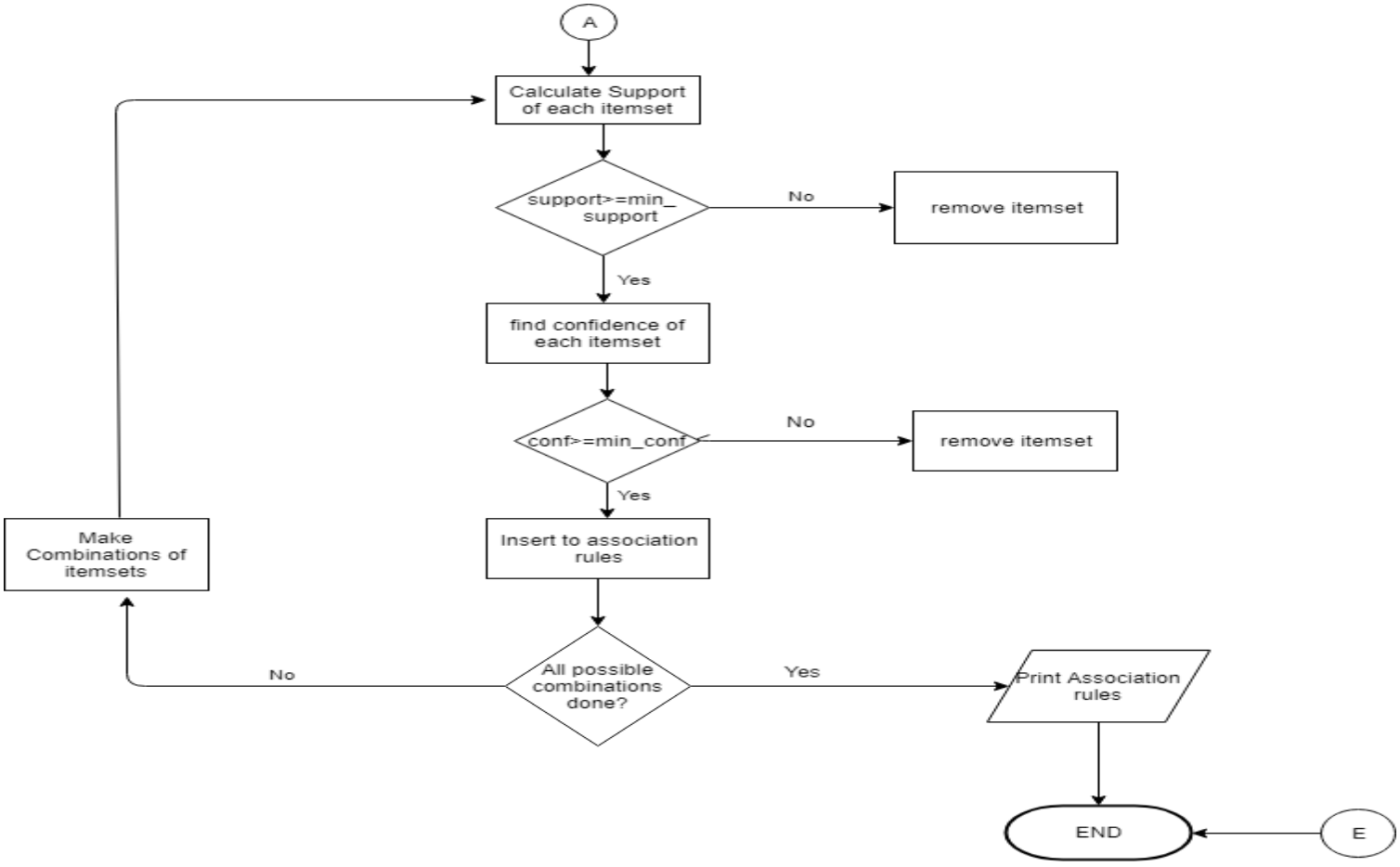
5. Design:

5.1 Use Case Diagram



5.2 Flowchart





6. Methodology:

6.1 Theory:

Market Basket Analysis is a technique which works on the principle that if a customer buys a product, he is more likely to buy some more specific set of products along with it.

For the implementation of this technique, we are need to have some sort of requirements:

1- **DATASET:**
   From Market (Real Time Data)
2- **ALGORITHM:**
   Apriori Algorithm

Apriori Algorithm

By applying Apriori Algorithm onto a dataset, we will be having 2 factors:

1- **Support:**
   A support is a parameter which is used in Apriori Algorithm to determine how frequent the item appears in dataset.
2- **Confidence:**
   Confidence is also a parameter which will be looking for combinations in products and reject those having less confidence.

6.2 Mathematical Model:

- Calculate Support/frequency of individual item from total set of items
- Reject those items, whose support<2
- Combine it further into duplets
- Calculate support/frequency
- Reject those group of items, whose support<2
- Do it again by combining it to triplets, quadruplets and so on…..
- Again, reject those group of items, whose support<2
- At the last, we will get those set of items, which are having high support or which is bought more frequently.
- To calculate Confidence,
  - Confidence=$\frac{Support\ of\ Combined\ Products}{Support\ of\ Prescribed\ Product}$

6.3 Diagramatic Representation:

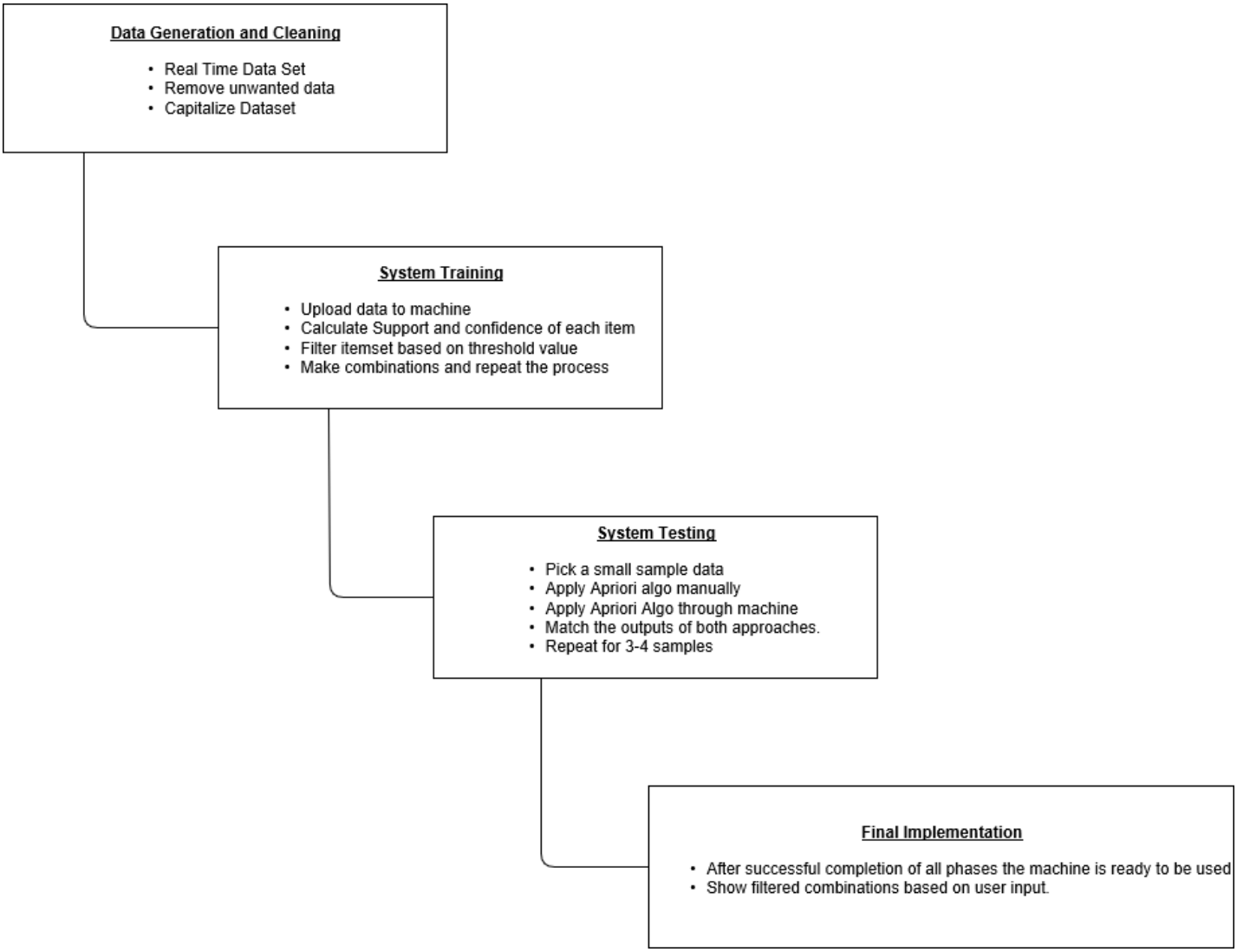


Fig 1.1 Market Basket Analysis

System Requirements: (Software/Hardware)

Hardware:
Ram- 4GB
Processors- Intel Core i3/i5/i7
HDD- 100GB
Software:
Operating System- Windows 10/8.1/8/7/XP |Ubuntu
Programming Language- Procedural Language-C

7. Implementation:

7.1 Algorithm

1. Start
2. Read Transactional Data
3. Store unique items
4. Input minimum support and minimum confidence
5. Calculate support for itemsets
6. Remove itemsets whose support < minimum support
7. Calculate confidence of remaining itemsets.
8. Remove itemsets whose confidence < minimum confidence
9. Add remaining itemsets to association rule.
10. Make combination of itemsets.
11. Repeat from step 5 to step 11 until all possible combinations of items are Formed.

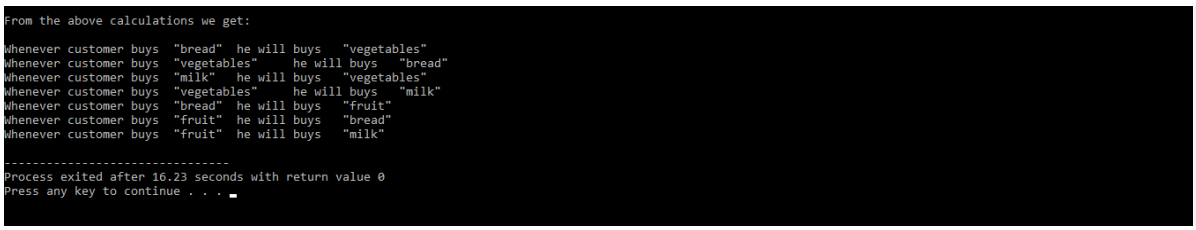
7.2 Pseudocode

```
INPUT: S, support where S= dataset, min_support = real
OUTPUT: Set of Frequent Itemsets
Require: S≠ Ø  0 <= min_support <= 1
  procedure GETFREQUENTITEMSETS
  freqSets [] <— null
  for all Itemsets i in S do
  if support >= min_support then
  freqSets[] <- i
  end if
  end for
  end procedure

INPUT: S where S = dataset
OUTPUT: Set of Candidate Itemsets
Require: S ≠ Ø
  procedure GENERATECANDIDATES
  i <-- 2
 num <— NumAttributes( S )
 candidates[ ] <— null
 while i < num do
 candidates[ ] 4— all sets of size i, support
end while
end procedure
```

7.3 Output Screen

7.4 Result Analysis



8. Conclusion and Future Scope:

By using the approach and implementing the concept of Market Basket Analysis we were able to find the support and confidence of various items and were able to obtain the desired combination of products that were closely associated to each other. This technique can also be used by several other Store Managers and retailers to generate the item combinations and decide which item set to be kept together and which set shouldn't be kept. The future scope of this project can be that this approach can be accepted and used by many Store Managers in order to increase their sales and also reduce customer efforts and provide more and more customer satisfaction.

Schedule: (PERT Chart)

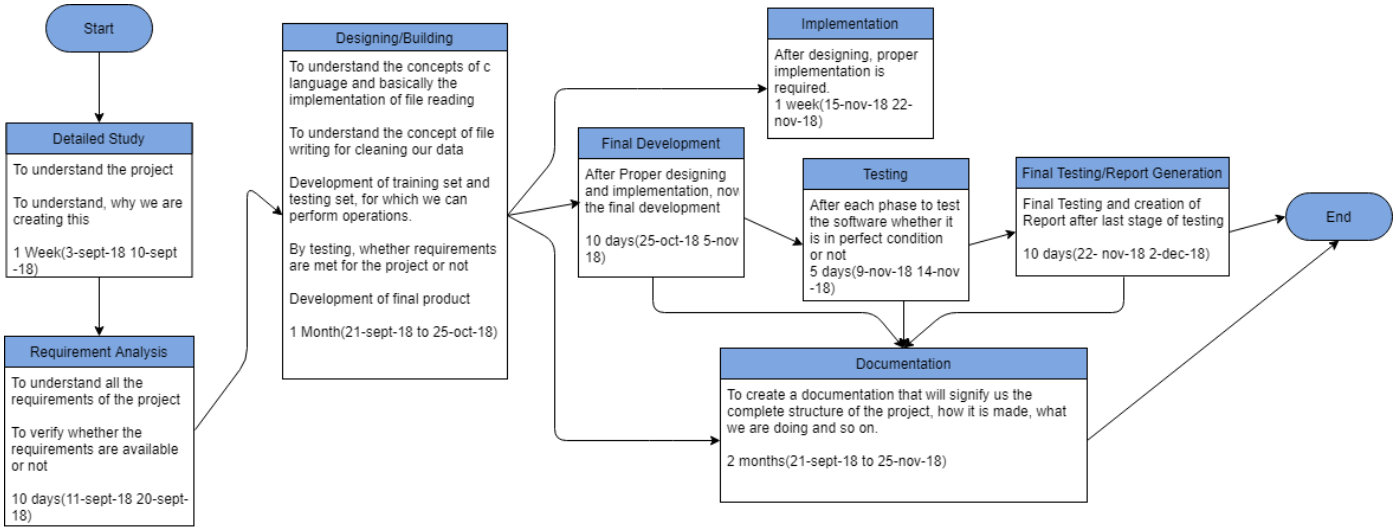


Fig 1.2 Pert Chart

References:

Documented Reference:

[1] https://www.sciencedirect.com/science/article/pii/S1877050916305208

[3] https://raw.githubusercontent.com/CSIT.../1813_Sanjeev_MarketBasketAnalysis.pdf

Book Reference:

[2] Data Mining-concept and techniques by Jiawei Han, Jian Pei, Micheline Kamber

[4] Data Mining Techniques by Arun k Pujari

Appendix Project Code

```
#include<stdio.h>
#include<stdlib.h>
float pair(int data[][6],int a, int b)
{
 int flaga=0,flagb=0,counter,i,j;
 float supp;
 counter=0;
 for(i=0;i<100;i++)
 {
     for(j=0;j<5;j++)
     {
      if(a==data[i][j])
   {
    flaga = 1;
   }

   if(b==data[i][j])
   {
    flagb = 1;
   }
  }

      if(flaga == 1 && flagb==1)
  {
    counter++;
    flaga=0;
    flagb=0;
  }

  else
  {
   flaga=0;
   flagb=0;
  }

 }
 supp = (float)counter/100;
 return supp;
}
void display(int va)
{
 if(va==1)
 printf("cleaner");
 else if(va==2)
 printf("salt");
 else if(va==3)
 printf("curd");
 else if(va==4)
 printf("beverages");
 else if(va==5)
 printf("snack");
 else if(va==6)
 printf("meat");
```

```
else if(va==7)
printf("milk");
else if(va==8)
printf("dessert");
else if(va==9)
printf("water");
else if(va==10)
printf("bags");
else if(va==11)
printf("vegetables");
else if(va==12)
printf("flour");
else if(va==13)
printf("beef");
else if(va==14)
printf("potato");
else if(va==15)
printf("yogurt");
else if(va==16)
printf("rice");
else if(va==17)
printf("chicken");
else if(va==18)
printf("sauce");
else if(va==19)
printf("bread");
else if(va==20)
printf("newspapers");
else if(va==21)
printf("berries");
else if(va==22)
printf("cream");
else if(va==23)
printf("spices");
else if(va==24)
printf("cereals");
else if(va==25)
printf("butter");
else if(va==26)
printf("onions");
else if(va==27)
printf("margarine");
else if(va==28)
printf("fruit");
else if(va==29)
printf("gum");
else if(va==30)
printf("cake");
else if(va==31)
printf("waffles");
else if(va==32)
printf("pork");
else if(va==33)
printf("cheese");
```

```
 else if(va==34)
 printf("wine");
 else if(va==35)
 printf("detergent");
 else if(va==36)
 printf("chocolate");
 else if(va==37)
 printf("eggs");
 else if(va==38)
 printf("pastry");
 else if(va==39)
 printf("coffee");
 else if(va==40)
 printf("oil");
 else if(va==41)
 printf("sugar");
 else if(va==42)
 printf("juice");
 else if(va==43)
 printf("soda");
 else if(va==44)
 printf("beer");
 else if(va==45)
 printf("soup");
 else if(va==46)
 printf("fish");
 else
 printf("grapes");
}
int main()
{
int i,j,k;
char ch;
char nx;
int data[100][6];
FILE *fp;
fp=fopen("mba_data2.txt","r");
if(fp==NULL)
{
    printf("File not found");
    exit(0);
    return(1);
}
while(!feof(fp))
{
    ch=fgetc(fp);
    if(ch!=',' && ch!='\n' && ch!=' ')
    {
        nx=fgetc(fp);
        if(nx!=',' && nx!='\n' && nx!=' ')
        {
            data[i][j]= (nx-'0')+((ch-'0')*10);
        }
        else
```

```
        {
            data[i][j]=ch-'0';
        }
        j++;
    }
    if(ch=='\n')
    {
        i++;
        j=0;
    }
}
fclose(fp);

const char unique[47][20]={"cleaner", "salt", "curd", "beverages", "snack", "meat",
"milk", "dessert", "water", "bags", "vegetables", "flour", "beef", "potato", "yogurt",
"rice", "chicken", "sauce", "bread", "newspapers", "berries", "cream", "spices",
"cereals", "butter", "onions", "margarine", "fruit", "gum", "cake", "waffles", "pork",
"cheese", "wine", "detergent", "chocolate", "eggs", "pastry", "coffee", "oil", "sugar",
"juice", "soda", "beer", "soup", "fish", "grapes"};

int values[47]=
{1,2,3,4,5,6,7,8,9,10,11,12,13,14,15,16,17,18,19,20,21,22,23,24,25,26,27,28,29,30,3
1,32,33,34,35,36,37,38,39,40,41,42,43,44,45,46,47};

int b[47];
float c[47];
for(k=0;k<47;k++)
{
 int count=0;
 for(i=0;i<100;i++)
 {
  for(j=0;j<6;j++)
  {
   if(values[k]==data[i][j])
   {
    count++;
    break;
   }
  }
 }
 b[k]=count;
 c[k]=(float)b[k]/100;
}
printf("Enter minimum support");
float min_support;
scanf("%f",&min_support);
if(min_support>1||min_support<0)
{
 printf("Enter correct support");
 exit(0);
}
printf("\n");
float temp1;
int temp2;
```

```
printf("\n Occurence of each item\n");
printf("\nItem ID\t occurence\n");
for(i=0;i<47;i++)
{
 printf("%d\t%d\n",values[i],b[i]);
}
printf("\n\n\n");
printf("\n Support of each item\n");
printf("\nItem ID\t support\n");
for(i=0;i<47;i++)
{
 printf("%d\t%.4f\n",values[i],c[i]);
}
for(i=0;i<47;i++)
{
 for(j=0;j<46;j++)
 {
  if(c[j]<c[j+1])
  {
   temp1=c[j];
   c[j]=c[j+1];
   c[j+1]=temp1;
   temp2=values[j];
   values[j]=values[j+1];
   values[j+1]=temp2;
  }

 }
}
int x=0;
for(i=0;i<47;i++)
{
 if(min_support<=c[i])
 {
  x=i+1;
 }
 else
 {
  break;
 }
}
printf("\n\n\n");
printf("\n Support of items having more than min_support\n");
printf("\nItem ID\t support\n");
for(i=0;i<x;i++)
{
 printf("%d\t%.4f\n",values[i],c[i]);
}
int doublets[30][2];
float dub_sup[30];
int d_count=0,l,m;
for(l=0;l<x-1;l++)
{
 for(m=l+1;m<x-1;m++)
```

```
 {
  doublets[d_count][0]=values[l];
  doublets[d_count][1]=values[m];
  dub_sup[d_count]=pair(data,values[l],values[m]);
  d_count++;
 }
}
printf("\n Support of each pair\n");
printf("\nItem Pair\t   support\n");
for(i=0;i<d_count;i++)
{
 printf("%d\t%d\t%.4f\n",doublets[i][0],doublets[i][1],dub_sup[i]);
}
float swap1;
int swap2,swap3;
for(l=0;l<d_count;l++)
{
 for(m=0;m<d_count-1;m++)
 {
  if(dub_sup[m]<dub_sup[m+1])
  {
   swap1=dub_sup[m];
   dub_sup[m]=dub_sup[m+1];
   dub_sup[m+1]=swap1;
   swap2=doublets[m][0];
   doublets[m][0]=doublets[m+1][0];
   doublets[m+1][0]=swap2;
   swap3=doublets[m][1];
   doublets[m][1]=doublets[m+1][1];
   doublets[m+1][1]=swap3;
  }
 }
}

int d_min=0;;
for(i=0;i<d_count;i++)
{
 if(min_support<=dub_sup[i])
 {
  d_min=i+1;
 }
 else
 {
  break;
 }
}

printf("\n Support of pairs having more than min_support\n");
printf("\nItem Pair\t support\n");
for(i=0;i<d_min;i++)
{
 printf("%d\t%d\t%.4f\n",doublets[i][0],doublets[i][1],dub_sup[i]);
}
```

```
//Confidence
printf("\nConfidence of \n");
int condi[10][2];
float confs[10][2];
for(i=0;i<d_min;i++)
{
 printf("%d->%d\t\t%d->%d\n",doublets[i][0],doublets[i][1],doublets[i][1],doublets[i][0]);
 for(j=0;j<x;j++)
 {
  if(doublets[i][0]==values[j])
  {
   condi[i][0]=doublets[i][0];
   condi[i][1]=doublets[i][1];
   confs[i][0]=dub_sup[i]/c[j];
   printf("%.4f\t\t",confs[i][0]);
  }
 }
 for(j=0;j<x;j++)
 {
  if(doublets[i][1]==values[j])
  {
   confs[i][1]=dub_sup[i]/c[j];
   printf("%.4f\n\n",confs[i][1]);
  }
 }
}
float conf;
printf("Enter minimum confidence");
scanf("%f",&conf);
if(conf>1||conf<0)
{
 printf("Enter correct confidence\n");
 exit(0);
}
float min_conf[10];
int min_con[10][2];
int ix=0;
printf("\nItems having more than min confidence\n");
for(i=0;i<d_min;i++)
{
 if(confs[i][0]>conf)
 {
  printf("%d->%d\t\t",doublets[i][0],doublets[i][1]);
  printf("%.4f\n",confs[i][0]);
  min_con[ix][0]=doublets[i][0];
  min_con[ix][1]=doublets[i][1];
  min_conf[ix]=confs[i][0];
  ix++;
 }
 if(confs[i][1]>conf)
 {
  printf("%d->%d\t\t",doublets[i][1],doublets[i][0]);
  printf("%.4f\n",confs[i][1]);
  min_con[ix][0]=doublets[i][1];
```

```
  min_con[ix][1]=doublets[i][0];
  min_conf[ix]=confs[i][1];
  ix++;
 }
}
printf("\nFrom the above calculations we get:\n\n");
for(i=0;i<ix;i++)
{
 printf("Whenever customer buys\t\"");
 display(min_con[i][0]);
 printf("\"\t he will buys\t\"");
 display(min_con[i][1]);
 printf("\"\n");
}
return 0;
}
//380 loc
```