Analysis of Customer Purchase Behaviour — A Market Basket Analysis

Enrolment Nos. —14103033, 14103056, 14103271

Name of Students —Nishant Srivastava, Stuti, Kanika Gupta

Name of Supervisor —Ankita Verma

May-2018

Submitted in partial fulfilment of the Degree of
Bachelor of Technology

in

Computer Science Engineering

DEPARTMENT OF COMPUTER SCIENCE ENGINEERING &
INFORMATION TECHNOLOGY

JAYPEE INSTITUTE OF INFORMATION TECHNOLOGY, NOIDA

TABLE OF CONTENTS

Publications from the project

- Research Paper titled “Improved Market Basket Analysis with Utility Mining” presented at the International Conference on Internet of Things and Connected Technologies 2018, and currently in process for publication in Elsevier Journal.

(II)

DECLARATION

We thus pronounce that this project is our own work and that, to the best of our insight and conviction, it contains no material already distributed or composed by someone else nor material which has been acknowledged for the honor of some other degree or recognition of the college or other establishment of higher learning, aside from where due affirmation has been made in the content.

Place: Signature:

Date:

Name: Nishant Srivastava
Stuti
Kanika Gupta

Enrolment No:14103033
14103056
14103271

(III)

CERTIFICATE

This is to certify that the work titled **Analysis of Customer Purchase Behaviour — A Market Basket Analysis** submitted by **Nishant Srivastava, Stuti, Kanika Gupta** in partial fulfilment for the award of degree of B. Tech. (CSE) of Jaypee Institute of Information Technology, Noida has been carried out under my supervision. This work has not been submitted partially or wholly to any other University or Institute for the award of this or any other degree or diploma.

Signature of Supervisor
Name of Supervisor Ankita Verma
Designation
Date

(IV)

ACKNOWLEDGEMENT

We would like to acknowledge the following people for their support and assistance with this project. Firstly, we would like to thank Jaypee Institute of Information Technology, for providing us with this opportunity to pursue this project. We are highly indebted to Mrs. Ankita Verma for her guidance and constant supervision as well as for providing necessary information regarding the project & also for her support in completing the research project. We offer our thanks towards the Faculty of Jaypee Institute of Information Technology for their kind co-task and support which helped us in finishing of this undertaking.

Our thanks and appreciation likewise go to our associates in building up the undertaking and individuals who have enthusiastically bailed us out with their capacities.

Nishant Srivastava (14103033)

Stuti (14103056)

Kanika Gupta (14103271)

(V)

SUMMARY

Our focus since the very start of the project has been on developing an algorithm which overcomes all the shortcomings of association rule mining and mines item sets not on the basis of only frequency, but also incorporates the actual utility of the item, the term utility being subject to relative understanding (ex. Price for Shopkeeper, Discount for Customers Etc.). Hence we read numerous literature pertaining to the above and created an ensemble algorithm known as the Two-phase algorithm which mines high utility item sets by calculating all item sets' utility and selectively pruning the low utility ones by using the downward closure property, which reduces the runtime of the algorithm.

The next phase of our major project involved creating a recommender system which utilised the item sets mined by our algorithm and also those mined from A-priori algorithm and made recommendations to the users based on it. Running on Wamp, the recommender system works in the form of an E-commerce website hosted on a localhost, with all the product details and mined rules existing on a database on a MySQL server. The website is coded in PHP, and has features such as User registration, Login, Add to Cart, Comments, User reviews and provides recommendation at the time when product is saved in the cart and also at the time of checkout thus attempting to create aa personalized experience for every user.

Hence the E-commerce website acts as a very useful tool on which we can test and improve our recommendation system and further our work on Market basket analysis and customer feedback.

(VI)

LIST OF FIGURES

LIST OF TABLES

(VIII)

LIST OF SYMBOLS & ACRONYMS

I. HUIM: High Utility Item Set Mining

II. ARM: Association Rule Mining

III. CRM: Customer Relationship Management

IV. MBA: Market Basket Analysis

V. JVM: Java Virtual Machine

CHAPTER 1

INTRODUCTION

1.1 General Introduction

Data Mining is a tool for retrieving novel and useful information contained in huge data stores. Conventional approaches to data mining techniques have majorly targeted the discovery of correlations among items that occur often in transactional databases. This method referred to as frequent item set mining believes that recurring item sets must be more significant to the user. Alternatively, we attempt to simulate an algorithm for a recent development called Utility Mining, which studies the usefulness or

utility of item sets, in addition to their frequency. we collected data and mined for frequent itemsets using two algorithms, namely Association rule mining, and Two phase algorithm which used High Utility Itemset Mining. With the proliferation of online shopping and E-commerce to the masses has greatly impacted the buying habits of people and a recommendation system has a huge contribution to its popularity. Recommendation algorithms provide an effective form of targeted marketing by creating a personalized shopping experience for each customer. Through the rules retrieved we in our project try to implement a recommendation system on an self-developed ecommerce website with the aim of comparing the accuracy of the prevalent A-priori algorithm and our proposed algorithm of Utility Mining with the aim of analysing customer behaviour.

Data Mining & CRM

Use of information digging methods for Customer Relationship Management has turned out to be broadly perceived as a critical business approach. Numerous associations have gathered an abundance of information about their present clients, potential clients, and providers. Information mining utilizes measurable, numerical, manmade brainpower and machine-learning methods to separate and recognize helpful data and in this way pick up information from extensive databases empowering associations to take business choices that encourage Customer Identification, Customer Attraction, Customer Retention and Customer Development.

Figure 1 - Understanding CRM using Data Mining

Market Basket Analysis

Market basket analysis (otherwise called association rule mining) is a technique for finding client obtaining designs by extracting rules or co-events from stores' value-based databases. Finding, for instance, that general store clients are probably going to buy dairy products like milk , bread, and cheddar together can help administrators in planning store format, sites, item blend and packaging, and other promoting techniques.

tid	*X*
100	{beer, chips, wine}
200	{beer, chips}
300	{pizza, wine}
400	{chips, pizza}

Table 1 – An example transaction database D

To date, the A-priori algorithm (Agrawal et al., 1993) is quite a prevalent method for mining the association rules from transaction logs, which fulfil the minimum support and confidence levels as per the users' requirement. Support assesses how frequently the transactional logs comprise both A and B, while confidence assesses the rule's correctness. Confidence is defined as the ratio of the number of transactions containing both A and B to the number of transactions containing A only. In a-priori, given two mutually exclusive subsets of products, A and B, an association rule A→ B points to a pattern that if a customer buys A, then he or she buys B as well.

Itemset	Cover	Support	Frequency
{}	{100, 200, 300, 400}	4	100%
{beer}	{100,200}	2	50%
{chips}	{100,200,400}	3	75%
{pizza}	{300,400}	2	50%
{wine}	{100,300}	2	50%
{beer, chips}	{100,200}	2	50%
{beer, wine}	{100}	1	25%
{chips, pizza}	{400}	1	25%
{chips, wine}	{100}	1	25%
{pizza, wine}	{300}	1	25%
{beer, chips, wine}	{100}	1	25%

Table 2 – Itemsets and their support in D

Rule	Support	Frequency	Confidence
{beer} ⇒ {chips}	2	50%	100%
{beer} ⇒ {wine}	1	25%	50%
{chips} ⇒ {beer}	2	50%	66%
{pizza} ⇒ {chips}	1	25%	50%
{pizza} ⇒ {wine}	1	25%	50%
{wine} ⇒ {beer}	1	25%	50%
{wine} ⇒ {chips}	1	25%	50%
{wine} ⇒ {pizza}	1	25%	50%
{beer, chips} ⇒ {wine}	1	25%	50%
{beer, wine} ⇒ {chips}	1	25%	100%
{chips, wine} ⇒ {beer}	1	25%	100%
{beer} ⇒ {chips, wine}	1	25%	50%
{wine} ⇒ {beer, chips}	1	25%	50%

Table 3 – Association rules and their support and confidence in D

However, the practical utility of frequent item set mining is confined by the importance of the identified item sets. While, a-priori algorithm has concentrated on frequent item sets, in reality, rare ones are of greater interest. Hence, during the mining process one should not merely concentrate on identifying frequent or rare item sets but also those item sets which have both high frequency and high utility in the database. The term utility implies the significance or usefulness of the presence of item sets in transactional logs, and is expressed as a quantity in terms of profit or sales, etc.

Utility Mining

An emerging area is that of Utility Mining which not only considers the frequency of the item sets but also considers the utility associated with the item sets by taking metrics like profit and sales into consideration. The term utility in utility based mining refers to the quantitative representation of user preference i.e, measurement of importance of that item set in the user's perspective is the utility value of an item set.

Transacion ID	Quantity of Item sold in Transaction		
	Item A	Item B	Item C
T1	2	0	1
T2	4	0	2
T3	4	1	0
T4	0	1	1
T5	5	1	2
T6	10	1	5
T7	4	0	2
T8	1	0	0
T9	3	0	0
T10	5	0	0

Table 4 – Transaction Database

Item Name	Unit Profit (in INR)
Item A	5
Item B	100
Item C	40

Table 5 – Unit Profit associated with Items

Itemset	Support(%)	Profit(INR)
A	**90**	190
B	**40**	400
C	**60**	**520**
AB	30	395
AC	**50**	**605**
BC	30	**620**
ABC	20	**555**

Table 6 – Support and Profit for all item set

For example, on the off chance that a business expert engaged with some retail inquire about necessities to discover which thing sets in the stores gain greatest deals income for the stores he or she will characterize utility of any things as the money related benefit that the store acquires by offering every unit of that thing set. This plans to beat the impediments of existing regular things mining procedures.

Recommendation Engine

Recommendation Engines can be defined as information filtering system aiming to analyse and estimate the 'rating' that user would give to an item. There are times when people don't know what

they want until they see it. Recommender Systems show the user/customer a whole new range of possibilities and products, which they might not know they would want to buy. Some basic recommendation techniques identified are memory based, knowledge based, rule-based, and so on. With recommendation engines the surfing over various sites becomes more personalised and viable for users.

Market Basket Analysis and Recommendation Engine

A typical list of recommendations on the basis of the user's interest in a product is produced through a variety of methods and that is the aim of a recommender system. Using association rules to generate recommendations is one such method. The production of a set of association rules is a typical analysis goal when applying market basket analysis which is in the form : *IF {pasta, wine, garlic} THEN pasta-sauce.* To generate recommendations for new basket data and/or new transactions is the deployment of the rule engine in a productive environment. Measures like support, lift and confidence, define how trustworthy each rule is. So with threshold values for these measures the probability that the next time one goes to the supermarket and buys pasta and wine, some pasta sauce will be recommended.

1.2 Problem Statement

Our objective is to propose an ensemble algorithm which can tackle the issues of considering the purchase of an item as a binary variable and not its frequency in traditional ARM method and the shortcomings of HUIM (High Utility Item set Mining) method, wherein the Downward Closure property cannot be applied, in order to find a pruned and useful set of high utility items. The high utility items obtained through the proposed algorithm will be then deployed in an e-commerce website to accurately predict the next product bought by an online customer using a simple recommender system and compare its result with the prevalent method i.e. A-priori algorithm of association rule mining.

1.3 Empirical Study

With the large amount of data that we have today from a variety of fields like retail markets, banking sector, medical field and so on, it is important that we extract useful information to make sense out of data.

A-priori is the most accepted technique to filter frequent item sets. It works in multiple passes while probing the data-store many a times. Subsequently, various effective ways have been asserted to

derive critical rules from the data bases (Agrawal et al., 1993). Han et al (2000) proposed a novel way of extracting association rules without generating candidate sets by introducing a data structure, FP-tree and an FP- growth method. In practical world applications, an item may be valued owing to its importance or utility. Bhattacharya et al. (2012) devised a high utility mining algorithm depending on the profit gained from the item set which should satisfy a minimum threshold thus having a minimum support. Reddy et al. (2012) proposed an improved UP-growth high utility mining algorithm. A new method for utility frequent item set mining was offered by Jabbar et al. (2016), which mines novel high frequency products by assigning weights to items' quantity, significance, utility and user defined support. The traditional market basket analysis lacks in discovering crucial buying patterns in a multi-store setting, owing to the tacit assumption that items under consideration are available and visible every time, everywhere. An A-priori- like algorithm for automatically mining association rules in a multi-store environment was proposed by Chen et al (2005). Liu et al (1999) proposed mining of association rules with numerous minimum supports to approach the rare item problem. Agrawal et al (1997) proposed three cohesive procedures for mining association rules coupled with item constraints. Apart from the many data mining algorithms, a number of nature inspired algorithms have also been proposed to further research in Market Basket Analysis.

While there has been considerable work done on mining rules more efficiently (e.g., Ng. et al. 1998, Bayardo and Agrawal 1999, Zaki (2000), research into the use of rules to make effective recommendations is scarce. Zaïane (2002) proposed a method that finds all eligible rules (rules whose antecedents are subsets of the basket and whose consequents are not), and recommends the consequent of the eligible rule with the highest confidence. The notion of combining rules has been explored in a few studies in the past. Given a customer's basket, Lin et al. (2002) calculate the score for each item as the sum of the products of the supports and confidences of all eligible rules with this item as the consequent. The item with highest score is recommended to the customer. During the last twenty years, the amount of attention given to recommendation frameworks is developing at an expanding pace. This development makes new business openings and testing research issues in programming advancement, information mining, plan of better calculations, showcasing, administration and related issues. Client and business requests are presently setting piece of the examination motivation in proposal frameworks writing. As of late, new research groups (e.g. organize examination) from various fields have begun to include in the exploration of proposal frameworks keeping in mind the end goal to comprehend the monetary conduct of online purchasers and its suggestions to business process and rivalry. Software engineering writing and related fields are regularly enhanced by bibliographic surveys on the progressions of suggestion frameworks.

1.4 Approach to problem in terms of technology /platform to be used

Our project is divided mainly in two halves. For the first part, we study market basket analysis and the concept of association rule mining in order to find associations between products from their buying trends. The first part mainly was an implementation of the prevalent method i.e. A-priori and the algorithm proposed by us, a miscellany of HUIM and A-priori. This implementation helped us in deriving various associations present in the dataset. To implement the same, we use JAVA, a concurrent, class-based, object-oriented, and specifically designed to have as few implementation dependencies as possible. The platform used was Netbeans which helped us in building various classes functions like conversion of csv to input format, development of algorithms and so on.

The second part required us to develop a test website to check the recommendations from the algorithms implemented and compare their accuracy accordingly. For this the platform that we used was PHP-Hypertext Preprocessor with a database management system WAMP.

1.5 Support for Novelty/ significance of problem

Conventional approaches to data mining techniques have majorly targeted the discovery of correlations among items that occur often in transactional databases. This method referred to as frequent item set mining believes that recurring item sets must be more significant to the user. After analyzing and weighing all the possible solutions in order to accomplish the utility maximization achieved by the High Utility Item set mining method we implement a Bi-phase algorithm for the quicker discovery of high utility item sets. With our proposed method we attempt to simulate an algorithm for a recent development called Utility Mining, which studies the usefulness or utility of item sets, in addition to their frequency. This High Utility Item Set Mining facilitates the recognition of item sets having utility value greater than a lower limit specified by the seller. This algorithm negates the issue of absence of downward closure property of A-priori which cannot be implemented in HUIM, by approaching it differently. Though high *confidence* is attained, yet the limitation of our approach lies in low *support*.

The ongoing practicality of the whole research and our contribution has various practical purposes. Some of them are:

1) Based on the insights, like the utility associated and frequent occurrence of the items in various transactions, from such rule discovery one can organize the store to increase revenues.
2) Various recommendations help the customers widen their horizon and help them in venturing various possibilities unknown earlier.
3) The recommendations also help in expansion of the associations and discovering the new trends in the public.

1.6 Tabular comparison of other existing approaches/ solution to the problem framed

The problem statement discussed above can be divided into two parts as follows:

a) Retrieval of Association Rules i.e. Association Rule Mining

b) Building a recommendation system

Existing Solution Approaches	Description of the method	Limitation of the method
A-priori algorithm	Frequent item set mining and association rule learning over transactional databases by identifying frequent individual items and extending them to supersets till those sets appear as much as required threshold.	• Considers the appearance of an item in a transaction even if the customers may purchase more than one of the same item, and the unit cost may vary among items. • All items are viewed of as having same importance.
High Utility Item set Mining	An extension of frequent item set mining but purchase quantities are taken into account as well as the unit profit of each item.	Downward Closure Property of A-priori cannot be directly applied to discover high utility item sets only making the process time consuming.

Table 7 – Association Rule Mining methods comparison

Existing Solution Approaches	Description of the method	Limitation of the method
Collaborative Filtering	Based on the idea that people who agreed in their evaluation of certain items in the past are likely to agree again in the future filters information to provide recommendations.	• Sufficient number of users required • Matrix sparsity is an issue • Cannot recommend an item that has not been previously rated. • Cannot recommend items to someone with unique tastes.

Content Based Filtering	Content-based filtering, also known as cognitive filtering, recommends items based on a collation between the matter of the items and a user profile. The content of each item is represented as a set of descriptors or terms.	• Not every word has similar importance • Longer words have higher chance of overlap with the profile.

Table 8 – Recommendation System comparison

CHAPTER 2
Literature Survey

2.1 Summary of the Papers Studied

Paper 1 - High Utility Item Set Mining

The proposed approach helps identify item set patterns which have maximum value revenue-wise to a sales person rather than just depending on the frequency of occurrence. The proposed algorithm depends on the profit gained from the item set which should satisfy a minimum threshold thus having a minimum support. Each item is assigned a utility value (Unit Profit) and each transaction shows the number of each item sold. All possible item sets are then obtained and analyzed on the basis of support and profit (Number of unit items*respective profit values). Those having both minimum support and a threshold profit value are considered high utility item sets.

Paper 2 - Knowledge discovery of hidden consumer purchase behavior: A market basket analysis

The analysis done in this paper aims at discovering the structure of associations among different products' sales in order to plan strategically for marketing decisions. It uses the principles of market basket analysis and analyses the datasets obtained from Kuwait's various departmental stores.

Paper 3 - Market basket analysis in a multiple store environment

The traditional market basket analysis neglects to find essential buying designs in a multi-store condition, as a result of an understood suspicion that items under thought are on rack all the time over all stores.. Temporal rules are developed to overcome the weakness of static association rules but these results are often biased. Chen et al proposes an Apriori- like algorithm for consequently removing affiliation administers in a multi-store condition. The standards created additionally contain data on store (area) and time where the guidelines hold.

Paper 4 - Application of data mining techniques in customer relationship management: A literature review and classification

A scholastic writing survey of the use of information mining strategies to CRM. It gives a scholastic database of writing between the time of 2000– 2006 covering 24 diaries and proposes a classification

plan to characterize the articles. Discoveries of this paper demonstrate that the exploration region of client maintenance got most research consideration. Of these, most are identified with coordinated one-to-one marketing and loyality programs separately. Then again, classification and association models are the two regularly utilized models for information mining in CRM.

Paper 5 - A novel algorithm for utility-frequent item set mining in market basket analysis

In this journal the author has proposed a novel approach for utility frequent item set mining. The technique mines novel frequent item sets by offering significance to items amount, importance, weightage, utility and client characterized support. It focuses around intriguing quality, centrality and utility of a thing. This approach can be utilized to give important proposal to a venture to enhance business utility.

Paper 6 - Mining Association Rules with Multiple Minimum Supports

The key component of association rule mining, Minimum Support, is utilized to prune the inquiry space and farthest point the quantity of rules created. Notwithstanding, utilizing just a solitary least support certainly expect that all things in the information are of the same having comparable frequencies in the database. In numerous applications, a few items come up as often as possible in the information, while others seldom show up. Setting least support too high, those rules that include rare items won't be found and setting the support low to discover both successive and rare items may cause combinatorial blast in light of the fact that those incessant things will be related with each other in all conceivable ways. This quandary is known as the rare item problem. The strategy enables clients to indicate different multiple supports attempts to solve situations like above.

Paper 7 - A Case-Based Recommendation Approach for Market Basket Data

An investigation of recommendation approach for circumstances in which a client's experience relies upon the arrangement of items chosen together more than on every item's stand-alone properties. Case-based thinking (CBR) is especially fitting on the grounds that the arrangements of items chosen together can be satisfactorily displayed and contrasted along with the execution of a case based reasoner profits by the presence of a lot of information. The paper utilizes case-based thinking (CBR) to recognize and suggest the items that appear to be more appropriate for finishing a client's purchasing knowledge provided that he or she has officially chosen a few items.

Paper 8 - Product Recommendation System

In this paper, based on the research on some existing models and algorithms, to predict the rating for a product that a customer has never reviewed, Item Similarity, Bipartite Projection and Spanning Tree were implemented, based on the data of all other users and their ratings in the system on some existing datasets. The results indicated Spinning Tree has the best result, for old users, and Item Similarity best with mean squared error (MSE), for new users. Bipartite Projection has the best result and in terms of computational performance, Bipartite Projection is the swiftest algorithm.

Paper 9 - A Recommendation Engine by Using Association Rules

This investigation deals with a recommendation engine which was produced to customize an internet business site. Here, the technique is association rule mining and the personalization approach is collaborative filtering. The product was created by the programming language C# and association rules were created by A-priori algorithm. The recommendation engine had been tried by existing information before it was conveyed to an e-commerce web site.

Paper 10 - Effective Personalization Based on Association Rule Discovery from Web Usage Data

In this paper the authors present a versatile framework for recommender frameworks utilizing association rule mining from clickstream data. In particular, they show an information structure for putting away the found incessant item sets which is particularly appropriate for recommender frameworks. The recommendation calculation uses this information structure to create recommendations effectively progressively, without the need to produce all association rules from visit item sets. Moreover, through nitty gritty experimental assessment it demonstrates that the framework can beat a portion of the deficiencies of recommender frameworks in view of association rules, by utilizing multiple support levels for various kinds of site hits and differing estimated client histories.

2.2 Integrated summary of the literature studied

Though the field of frequent item set mining has been a heavily researched one ever since its inception, traditional association rule mining is still the most cited and all-prevailing in the field of market basket analysis. Though some research has been done in the field of utility item set mining, there is a lack of a benchmark algorithm which can be taken as a base to further research in this avenue. The claims of efficient algorithms in terms of computational complexities stand on thin ground as it varies largely with the size of the dataset and the specifications of the machine used to run the transactions. A large proportion of the research done takes association rule mining as a standard algorithm and provides methods to optimize its output in terms of the number of interesting

association rules generated. The use of neural networks in market basket analysis is an emerging area and not a lot of work has been done to incorporate them. One of the most consequential utility of the aforementioned is in recommender systems where various algorithms like Item Similarity, Bipartite Projection and Spanning Tree are used to predict user ratings or certain algorithms use association rule mining itself in order to personalise a website to each user's preference through the clickstream data. Lastly, even after a decade of research market basket analysis continues to be dominated by algorithms focusing more on statistical aspects rather than the semantic value of the rules generated, limiting their usefulness in the business world.

Work	Research question	Methodology	Contribution	Limitation	Scope
Chau et al, 2009	Data-mining techniques in CRM.	Search using descriptor, ‘‘customer relationship management” and ‘‘data mining”, to further eliminate ones related to data mining in CRM.	First of its kind work on data mining techniques to CRM.	Overlooked articles without a keyword index.	Facilitates future research in data mining for CRM.
Mostafa, 2015	Market Basket Analysis to obtain purchase patterns in Kuwait.	Mines useful rules by association rules mining (ARM) technique and patterns of products' closeness on the basis of rules.	To design price promotion strategies.	Products purchased within the same time but not together at the same time are not handled.	Provides a rich picture of Kuwaiti consumer behaviour helping retailers.
Raorane, Kulkarni, & Jitkar, 2012	Analysis of data to identify consumer behaviour	Market Basket Analysis performed on retail transactional data at store to find rules with support and confidence.	Useful for organizations in retailing business for their decision making process.	In many practical situations rare ones are of higher interest than frequent item sets.	Provides valuable recommendation to the enterprise to enhance business utility.
Bhattacharya & Dubey, 2012	Citing the limitations of frequent item set mining and proposing a utility based mining approach.	Each item sold compared with minimum support and a threshold profit value	Identifies item set patterns which have maximum value revenue-wise other than its frequency.	Item sets may possess support lower than the minimum threshold, hence nullifying the advantages.	Facilitates the development of profit maximizing marketing strategies.
Chen et al, 2005	Alteration of A-priori to cater to a multiple store environment.	A-priori like methodology wherein each transaction is appended with a timestamp and a store identifier.	Useful for obtaining product, inventory, and delivery strategies for the store chain.	Fails to explore the generation of association rules iteratively, online, in a distributed environment, or in parallel.	May be expanded by considering temporal, spatial and location constraints.
Dai H. et al, 2001	Effective and scalable ways for Web personalization to tackle the lack of explicit user ratings	Matches the current user's activity against the discovered patterns. .	Scalable framework presented	ARM recommendations-primitive nature	Can be extended to real time market situations
Srikant, Vu, & Agrawal, 1997	Generating association rules with item constraints	Generating rules using a Boolean expression.	Extracting rules according to user constraints.	Multiple taxonomy can lead to conflicting rules.	Facilitates knowledge accumulation and creation
Murat Efe Aras, Ozgur Cakir, 2012	Offer products, users will find interesting, and get higher converts via recommendation engine by using ARM.	C# programming language and works on .NET framework	Recommendation system when used in week 3 increases the basket ration when compared to week 1 when it wasn't used	Random recommendations don't make difference	Can be extended to real time market situations

Table 9 – Summary of Related Work

CHAPTER 3

Analysis, Design and Modelling

3.1 Overall Description of the Project

Analysing the market and its trends and associations between products through the various purchases and expenditures done by the customers in order to give them the most personalised experience possible during their next turn through recommendation based o association rules gathered.
The overall description of the project can be done in the following sub-sections:

3.1.1 Project Perspective

Taking the base of prevalent and popular methods namely A-priori algorithm and HUIM our project aims at combining the best of both methods in order to come up with an ensemble algorithm which not only considers the frequency of the items present but also its utility in terms of the profit associated with it or the revenue it helps in generating. Our project further aims at using these rules, retrieved from both A-priori and the proposed algorithm, in a rule-based recommendation system which according to a user's basket and the rules generated will recommend items according to various aspects like support, confidence and so on.
*perspective diagram

3.1.1.1 Hardware Specification

1. CPU: 2.4GHz
2. Computer Processor: Intel i5
3. Computer Memory: 8 GB
4. Network: Required

3.1.1.2 Software Specification

1. Operating system: Linux/Windows/MacOS
2. Tool Specification:

Tool	Version Number
PHP	5.6.35
MySQL	5.7.21
Apache	2.4.33

MariaDB	10.2.14
JVM	25.111
R	3.4.0
RStudio	1.0.143
WAMP	3.1.3
Netbeans	8.2

Table 10 – Version Specification

3.1.2 Project Functions

Various functionalities that can/should be performed in order to make the best use of the whole project are as follows:

Association Rule Mining

i. Conversion of the transactional database (CSV format) to input data format
ii. Application of the algorithms to retrieve rules by setting minimum support, minimum confidence or minimum utility as required.

Recommendation Engine

i. Viewing the product details
ii. Adding various products to cart
iii. Rating and adding comments for a product

3.1.3 Assumptions

Our project assumes the following:

- No security or validation is required for using the e-commerce service.
- The consumer has a localhost server, and connection to network.

3.2 Functional Requirements

- A system with JVM installed
- A system with WAMP/XAMPP installed
- A dataset of transactions of a supermarket.
- Set of association rules from algorithms implemented in a database along with the products available in the supermarket.

3.3 Non-Functional Requirements

- Speed
- Stability

- Security
- Efficiency

3.4 Logical Database requirements

The following classification is a logical classification of those data entities as well as their attributes.

Data	Attributes	Use
Cart	• ID • Product	This data is used to identify the products in the cart
Category	• ID • title	Identifies the different categories of products.
Comments	• id • item_id • comments • commentValue • user_id • com_date	This data is used to store user comments on different products..
Fpgrowth_association_rules	• ID • ItemIDGroup • RecommendItemCode • SUP • CONF	This data stores the various rules obtained from ARM.
Order	• Id • Name • Contact • Address • Email • Item • Amount • Status • dateOrdered • dateDelivered • itemid	This data stores the information of products which are checked out by the users
Products	• ID • imgURL • Product • Descriptions • Price • Category	This data stores the product information.

Twophase	• ID • ItemIDGroup • RecommendItemCOde • SUP • UTIL	This data stores the rules generated from our Twopphase algorithm.
User	• Id • Username • password	This data stores the user information.

Table 11 – Database Description

3.5 Design Diagrams

3.5.1Use Case Diagrams

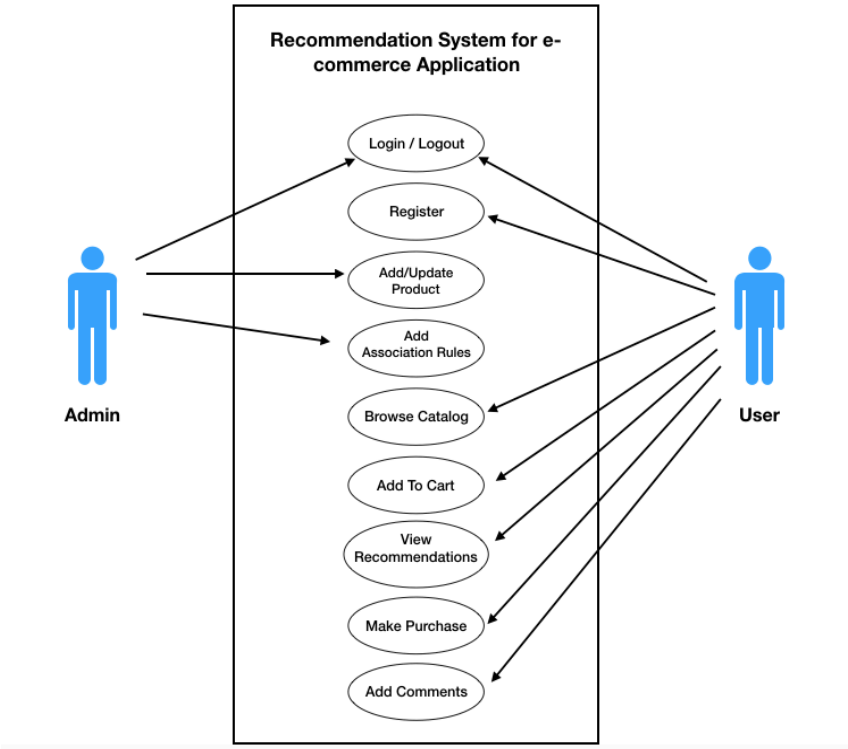

Figure 2 – Use Case Diagram of Recommendation Engine

3.5.2Control Flow diagrams

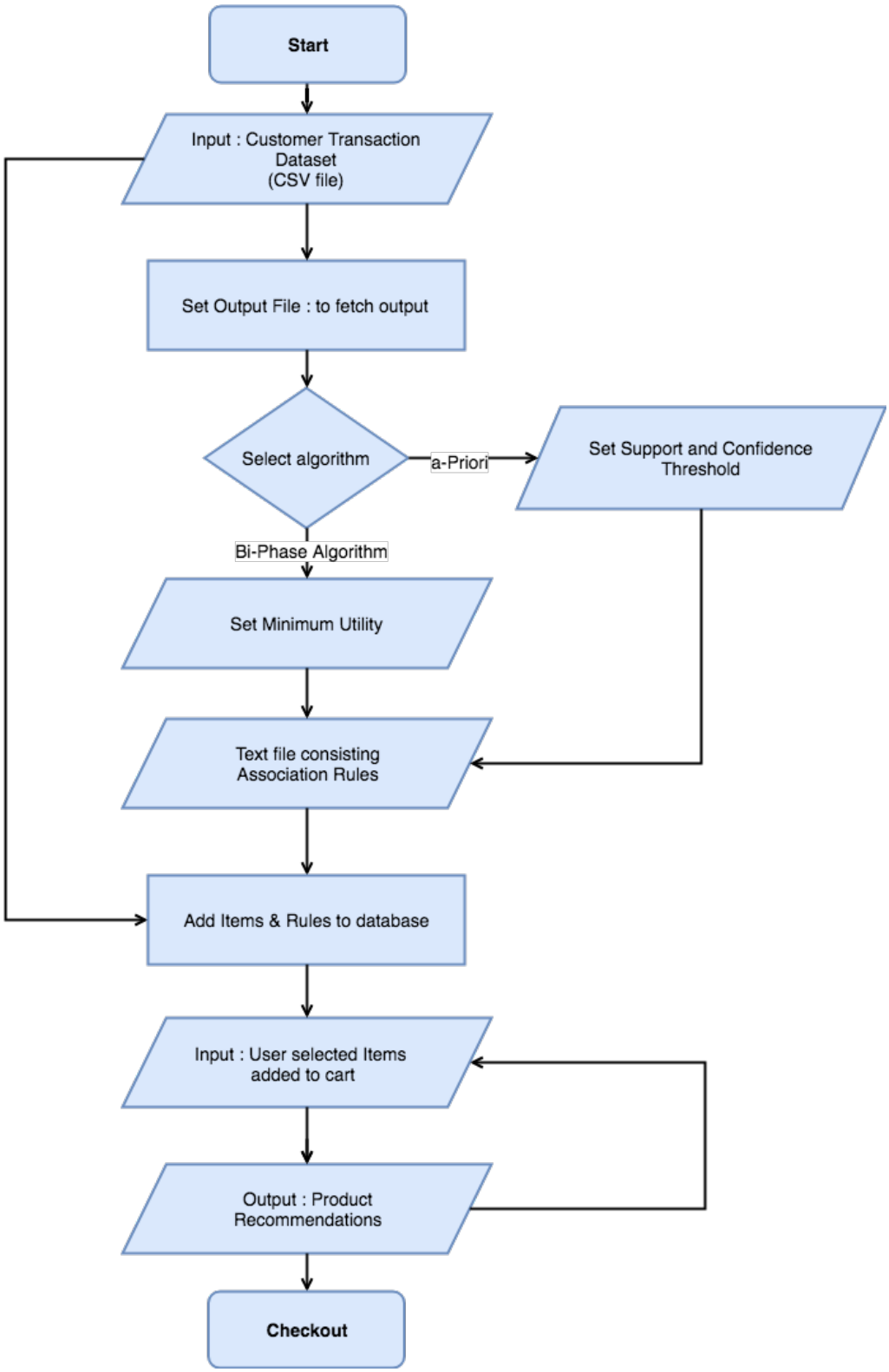

Figure 3 – Control Flow Diagram

3.5.3Activity Diagrams

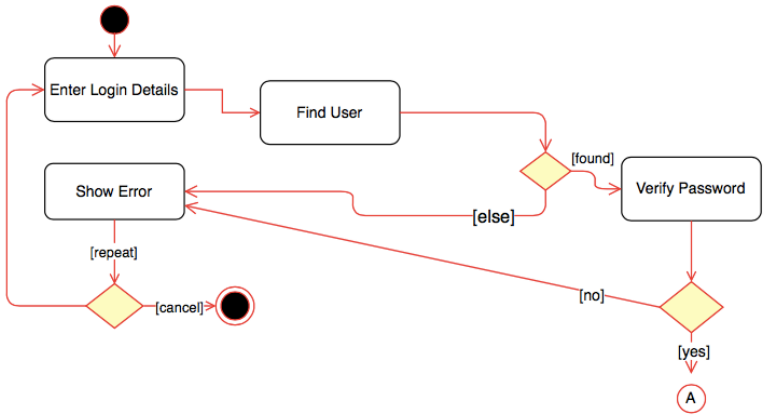

Figure 4 (a) – Activity Diagram of Recommendation Engine

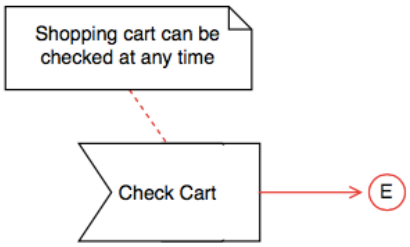

Figure 4 (b) – Activity Diagram of Recommendation Engine

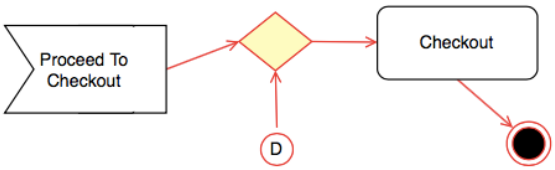

Figure 4 (c) – Activity Diagram of Recommendation Engine

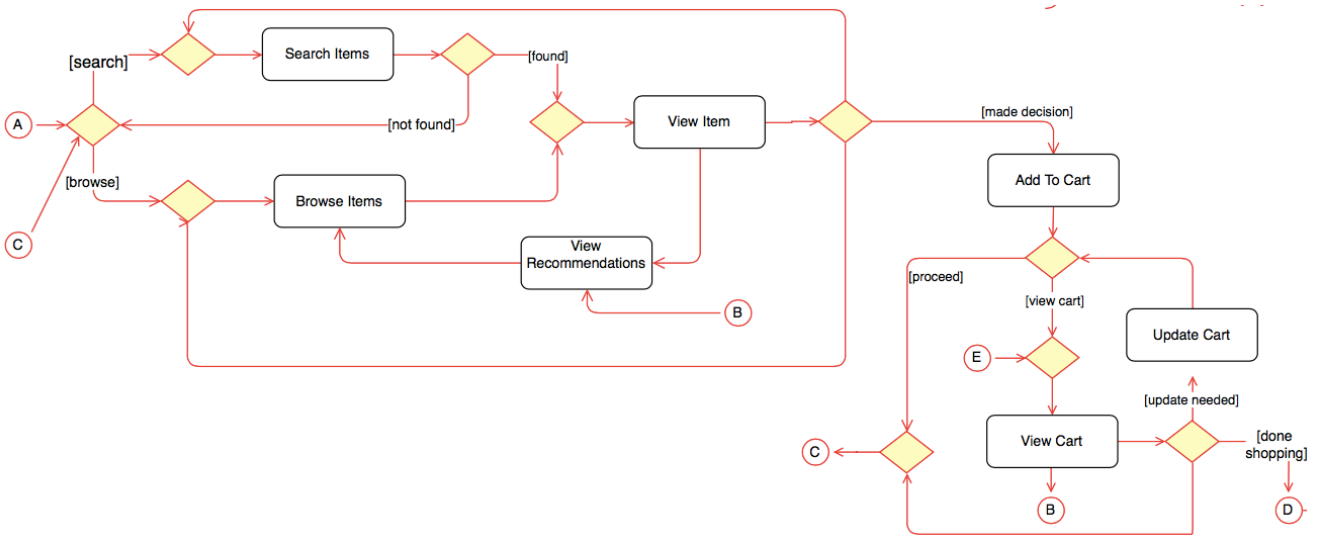

Figure 4 (d) – Activity Diagram of Recommendation Engine

Chapter 4
Implementation Details and Issues

4.1 Implementation Details and Issues

4.1.1 Implementation Issues

Our project is developed to mine for association rules in a data set. For every complex data set, there exists some data co-relation that can be mined to find connections between certain aspects (Data columns). In this project the data set collected has over 52 lakh entries which are processed by different algorithms to provide us with concrete rules of association that can be used for practical purposes in Data Science. The input (which usually is in csv format) gets converted into a base txt file, with simple input that can be parsed easily and in less time for quicker processing and results. The algorithms in the program are run on this file to produce results of data co- relation, with high confidence and probability of accuracy. This result can be used for further analysis on data mining based marketing. In our project we used these derived rules to recommend items to our user on the basis of their searches and items in their cart. These recommendations were sorted according to their respective support, confidence and utilities. These recommendations from both the algorithms implemented, i.e. A-priori and the proposed algorithm, are used is to compare their precision, recall, f-score along with the usability of results that are generated. More study can also yield results as to which logic is more suited for which purpose, time complexity, and type of data that is to be analysed.

Following were the issues faced :

1. High complexity of algorithms with huge data sets to be processed caused large running time causing the time for rule retrieval only go up as the number of transactions went higher, that needed to be minimised to improve the efficacy of the program.
2. The large data contained noise as well which had to be filtered out to make sure the results generated had rules with maximum accuracy and confidence.
3. Algorithm had to be designed in such a way that the result contained all the rules that could prove to be of value. A very high threshold value in filtering could let some valuable results get lost and a very low value would cause rules that are unnecessary also creep into the final result file.
4. Due to the dependency on the database for the rules directory the whole process of input was elongated more than expected.

5. Finding the right way to recommend i.e. choosing between support, utility and confidence.

4.1.2 Algorithms

1. Utility Mining Algorithm

TID: Transaction ID
F: frequency of each item in a transaction
C: cost of each item
E: user specified threshold
$\bar{E}$: utility threshold
HU: High Utility Item Set
HTWU: High Transaction Weighted Utilisation Item Set

Input: Transactional database, Utility (cost) of each item
Output: Association Rules

1. ∀TID
   calculate transactional utility using F X C
2. ∀X ∈ itemset
   calculate transaction weighted utilisation (twu)
3. If twu(X)>E
   X ⊆ HTWU
4. If E = $\bar{E}$ → HU ⊆ HTWU

2. Recommendation Engine Algorithm

Q: Search Query
X: Product
AR: Association Rules

Input: Search Query (Q)
Output: Recommendation using Association Rules

1. If Q=true
   Result=∀P ∈ Product Data
2. ∀Result
   If X ∈ Result is selected
   match(X, A->B) such that R1=A
   ∀X∈ Association Rules such that X is the antecedent
   retrieve AR(X)
3. ∀AR(X)
   If support(AR(X))=max OR utility(AR(X))=max
   recommend the consequents of A->B where A=X i.e. B

4.2 Risk Analysis and Mitigation

RISK ID	Classification	Description of Risk	Risk Area	Probability	Impact	RE(P*I)
1	Development Environment-Development Process	Huge Size of Data	Data Pre-processing or data cleaning	0.5	H	0.2*5=1
2	Development Environment-Development Process	Missing Values in data	Data Pre-processing or data cleaning	0.1	L	0.1*1=0.1
3	Development Environment-Development Process	Incorrect/ missed database entry	Project Development	0.1	M	0.1*3=0.3
4	Program Constraints-Resources	Hardware Limitation (usage by Program is more than the actual resources)	Output fetching	0.4	M	0.4*3=1.2

Table 12 – Risk Analysis

Chapter 5

Testing

5.1 Testing Plan

Type of Test	Will Test be Performed	Comments/Explanations	Software Development
Requirements testing	Yes	Performed to check the various requirements from aspects like input, and development and testing environment	PHP, Java, MySQL, Excel
Unit	Yes	Performed on individual modules of implementation	PHP, Java, MySQL
Integration	Yes	Integration testing performed after Unit Testing after merging different modules. After merging testing is done	PHP, Java, MySQL
Performance	Yes	Performance testing checks the responsiveness of our implementations under different work load	PHP, Java, MySQL
Load	Yes	To check the efficiency and optimal nature of the developed project	PHP, Java, MySQL

Table 13 – Test Classification

Activity	Start Date	Completion Date	Hours	Comments
Obtain pre-processed input data	10/29/2017	10/31/2017	Up to 2 hours	Raw data when put for cleaning took time due to its heavy size
ARM fetching	11/27/2017	12/2/2017	Up to 10-15 hours maximum	Was time consuming due to the various possibilities of minimum thresholds
Smooth functioning of website developed	4/30/2018	4/30/2018	2-3 hours	Checking various features of the website

Recommendation Generation for accuracy comparison	5/5/2018	5/7/2018	Up to 5 hours	Database directory of rule updation extended the period

Table 14 - Test Schedule

Test Environment

1. PhpMyAdmin: A software tool written in PHP which handles the administration of Mysql over the web. We needed this software in order to run our website on a local server.
2. MySQL: Is an open source relational database management system(RDBMS) based on structured Query language (SQL). While testing, MySQL was used to host the data.
3. WAMP : WAMP (Windows, Apache, MySQL, and PHP) was used as a testing environment as it seamlessly integrates the above technologies.

Software Requirements

1. Operating system: Linux/Windows/MacOS
2. Language: PHP-Hypertext Preprocessor, R, Java, MySQL
3. Tools: RStudio, WAMP, Netbeans 8.2

Hardware Requirements

1. CPU: 2.4GHz
2. Computer Processor: Intel i5
3. Computer Memory: 8 GB
4. Network: Required

5.2 Component decomposition and type of testing required

S. No.	Components that require testing	Type of Testing Required
1	Obtain pre-processed input data	Requirements
2	ARM fetching	Unit
3	Smooth functioning of website developed	Unit

	Recommendation Generation for accuracy comparison	Performance/Load

Table 15 – Component Decomposition and identification of tests required

5.3 Limitations of the Solution

Our solution mainly faces the following limitations -

1. *Sparsity:* It is hard to find users that have rated the same items and hence the user/ratings matrix is sparse, and if there are many items to be recommended, and many users.
2. *Cold Start:* A healthy number of users should be present already in the system.
3. *Popularity Bias*: Can only recommend items which are widely popular and cannot cater to unique tastes.
4. *First Rater*: An item which hasn't been rated yet cannot be recommended.
5. *Infrastructural constraint*: Our algorithm requires more resources than currently available with us and hence cannot process a large dataset of items.

Chapter 6

Findings & Conclusion

6.1 Findings

6.1.2 Evaluation of interestingness measures

We have used three measures – Support, Confidence and Lift to rank the mined association rules on the basis of their importance. Support is calculated as $P(A \cap B)/Total\ transactions$ and often expressed as a value ranging from 0 to 1. Confidence means the probability that a set of items/products is selected given that another set has already been selected. This is also often expressed as a value ranging from 0 to 1. Lift is calculated as $(A \cap B)/P(A)*P(B)$ where A and B represent set of items/products. A lift greater than 1 signifies that the antecedents and consequents are positively correlated.

```
> inspect(sort(mbaUM$model,by="confidence")[1:5])
    lhs                rhs     support     confidence lift      count
[1] {23286}         => {23287} 0.03116883 1.0000000  25.666667 12
[2] {22698,22699}   => {22697} 0.02597403 1.0000000  19.250000 10
[3] {20726,20727}   => {20725} 0.02597403 1.0000000  10.694444 10
[4] {22698}         => {22697} 0.02857143 0.9166667  17.645833 11
[5] {23209,85099B} => {23203} 0.02857143 0.9166667   8.607724 11
> inspect(sort(mbaUM$model,by="lift")[1:5])
    lhs                rhs     support     confidence lift     count
[1] {23286}         => {23287} 0.03116883 1.0000000  25.66667 12
[2] {23287}         => {23286} 0.03116883 0.8000000  25.66667 12
[3] {21932}         => {21933} 0.02597403 0.8333333  24.67949 10
[4] {21933}         => {21932} 0.02597403 0.7692308  24.67949 10
[5] {22697,22699} => {22698} 0.02597403 0.7142857  22.91667 10
```

Figure 5 – LIFT and Confidence Measure of Rules

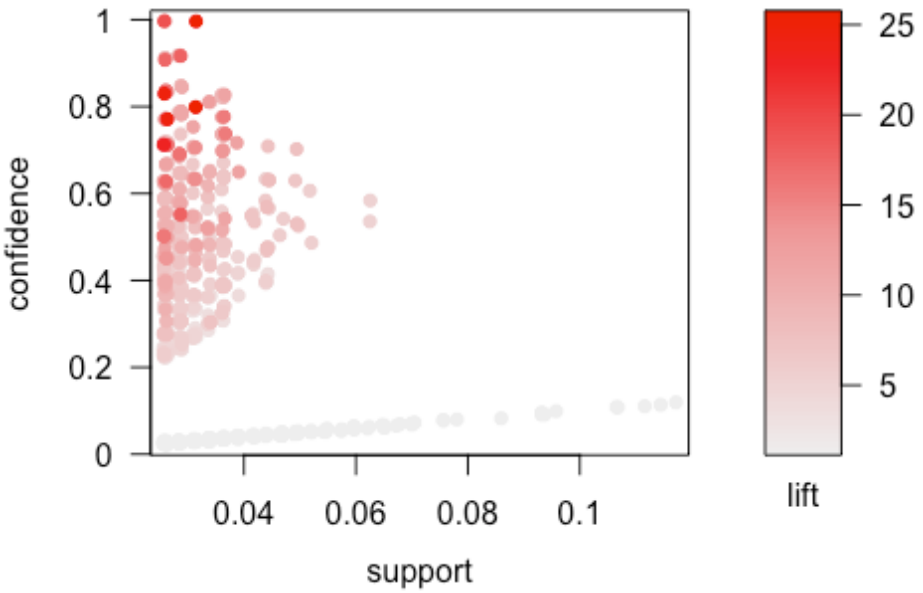

Figure 6 – Scatter plot between support, lift and confidence

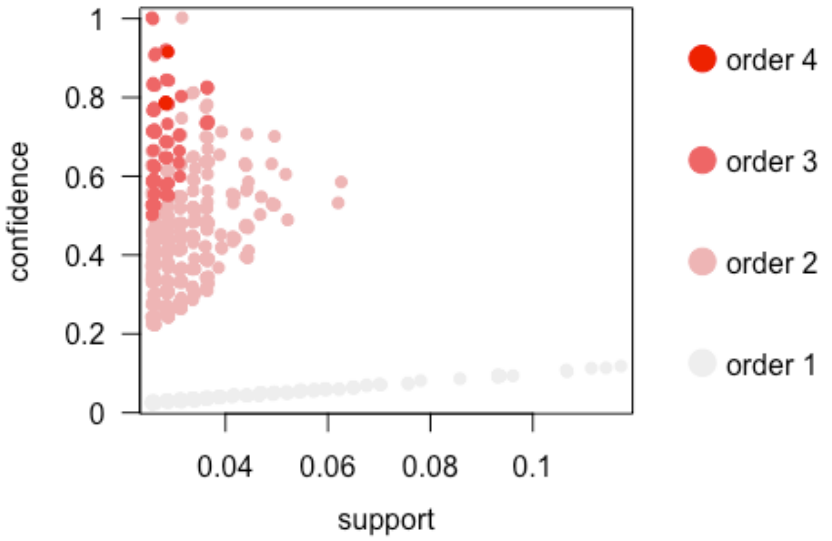

Figure 7 – Plot between support and confidence

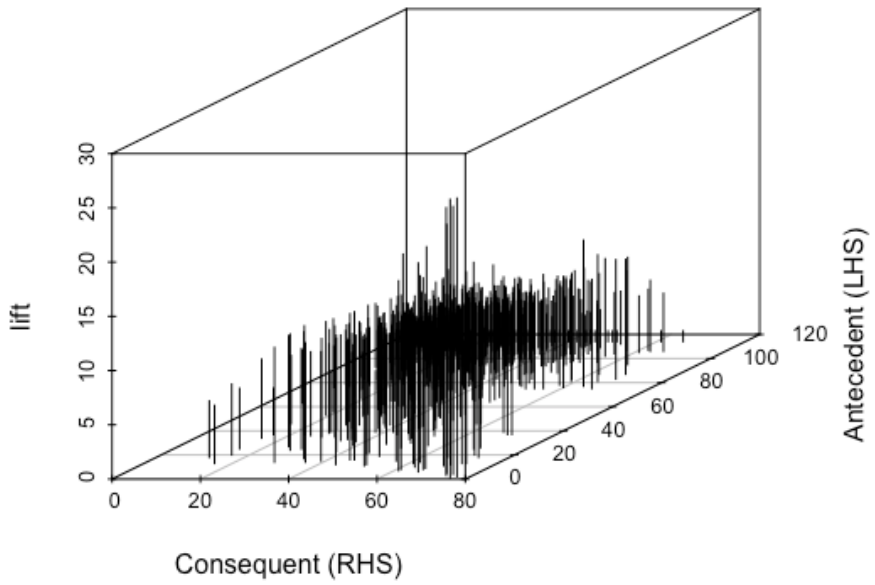

Figure 8 – Matrix plot relating the antecedents and consequents

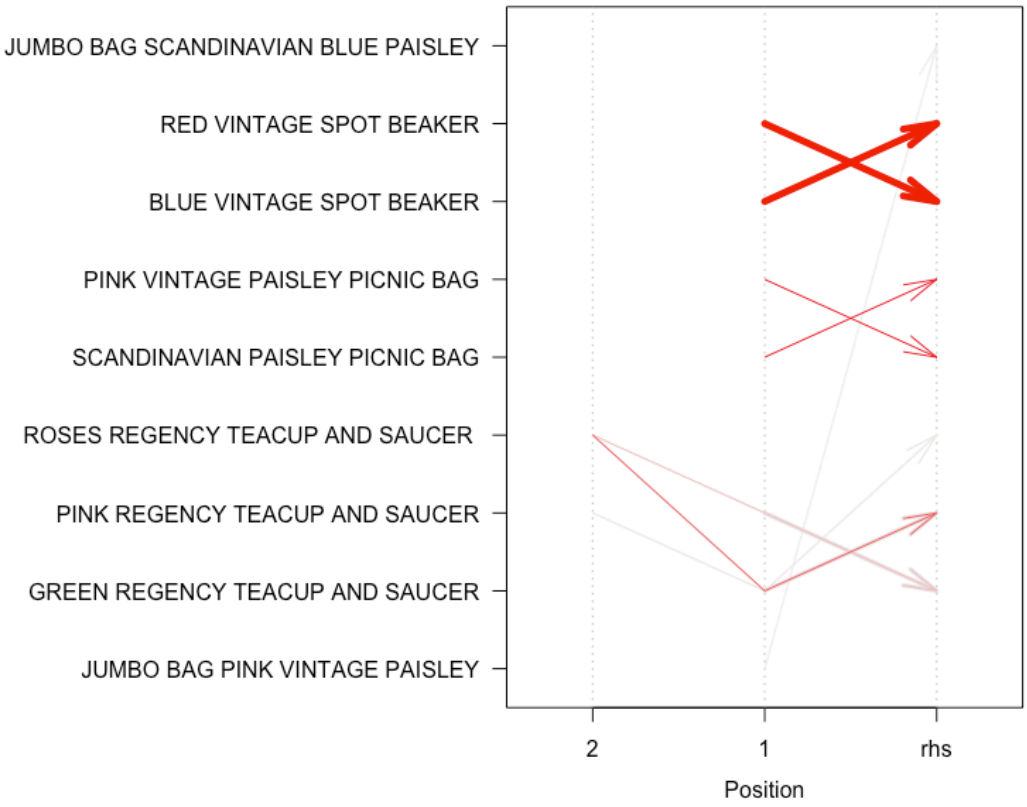

Figure 9 – Parallel Coordinates plot for 10 rules

```
> head(quality(rules))
     support confidence lift count
1 0.03896104 0.03896104    1    15
2 0.02597403 0.02597403    1    10
3 0.04935065 0.04935065    1    19
4 0.02857143 0.02857143    1    11
5 0.02857143 0.02857143    1    11
6 0.02597403 0.02597403    1    10
```

Figure 10 – Top 5 rules

6.1.2 Accuracy Comparison of the Algorithms

In order to compare the efficiency of A-priori and the proposed ensemble algorithm we performed multiple test cases on the test set of 10 each wherein we checked for the precision, recall and the F-score of all the test cases on both the algorithms. We calculated precision, recall and F-score using the following formulas:

$$\text{Precision} = \frac{\text{All Desired Retrived}}{\text{All Retrieved}} \qquad (1)$$

$$\text{Recall} = \frac{\text{All Desired Retrived}}{\text{All Desired}} \quad (2)$$

$$\text{F} - \text{Score} = \frac{2\text{X}(\text{Precision X Recall})}{\text{Precision}+\text{Recall}} \quad (3)$$

Test Case ID	Precision	Recall	F-score
1	45%	78%	0.57
2	60%	82%	0.69
3	52%	74%	0.61
4	58%	76%	0.66

Table 16 – Performance Measures of A-priori algorithm

Test Case ID	Precision	Recall	F-score
1	61%	100%	0.76
2	52%	95%	0.67
3	70%	85%	0.77
4	61%	98%	0.75

Table 17 – Performance Measures of Utility based Association rules

The Precision-Recall Curve shows the trade-off between precision and recall for different test cases. A high area under the curve signifies a greater recall and precision i.e, a low false positive rate and low false negative rate.

Figures 9 and 10 show the PR curves for the algorithm based on association rule based recommender system and the algorithm based on utility rules based recommender system.

Another way to graphically analyse the performance of a recommender system is the Receiver Operating Characteristic (ROC) curve. It is a plot between true positive rate (TPR) and false positive rate (FPR). The true positive rate is also called Recall or Sensitivity and FPR is also called as (1-specificity). The closer the graph is to the top left border, the more accurate the test. A greater AUC is a measure of accuracy.

Figures 11 and 12 show the ROC curves for the algorithm based on association rule based recommender system and the algorithm based on utility rules based recommender system

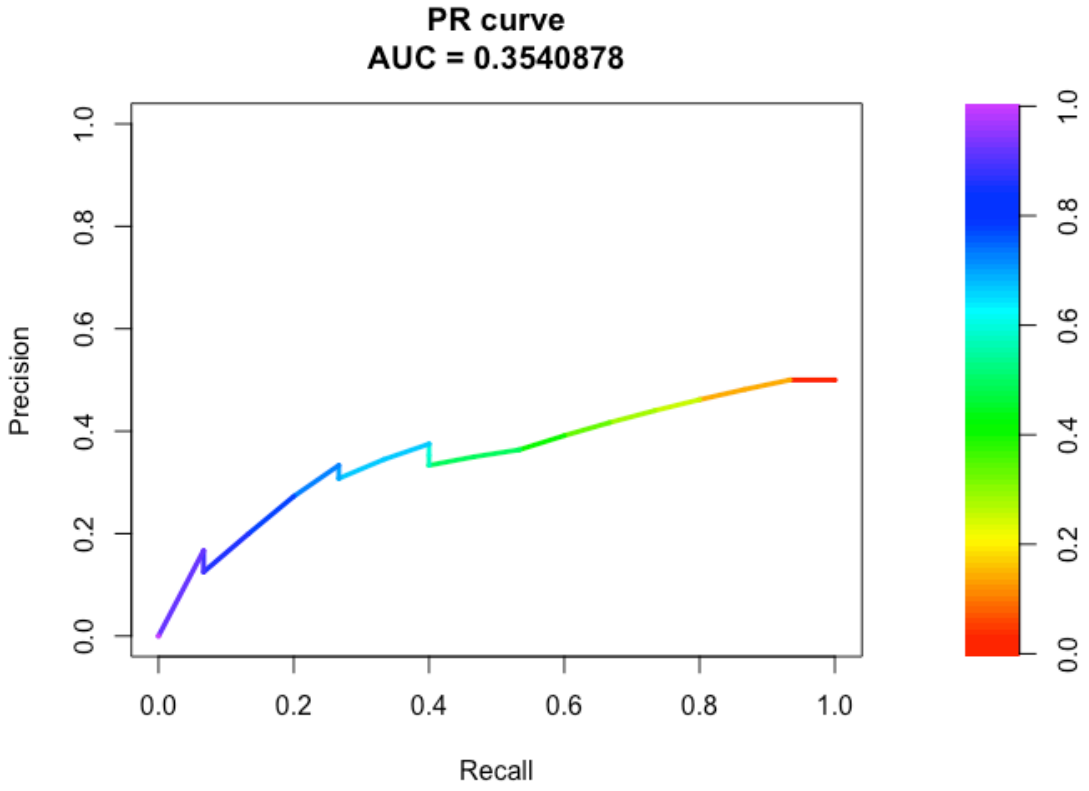

Figure 11 – PR curve for A-priori algorithm

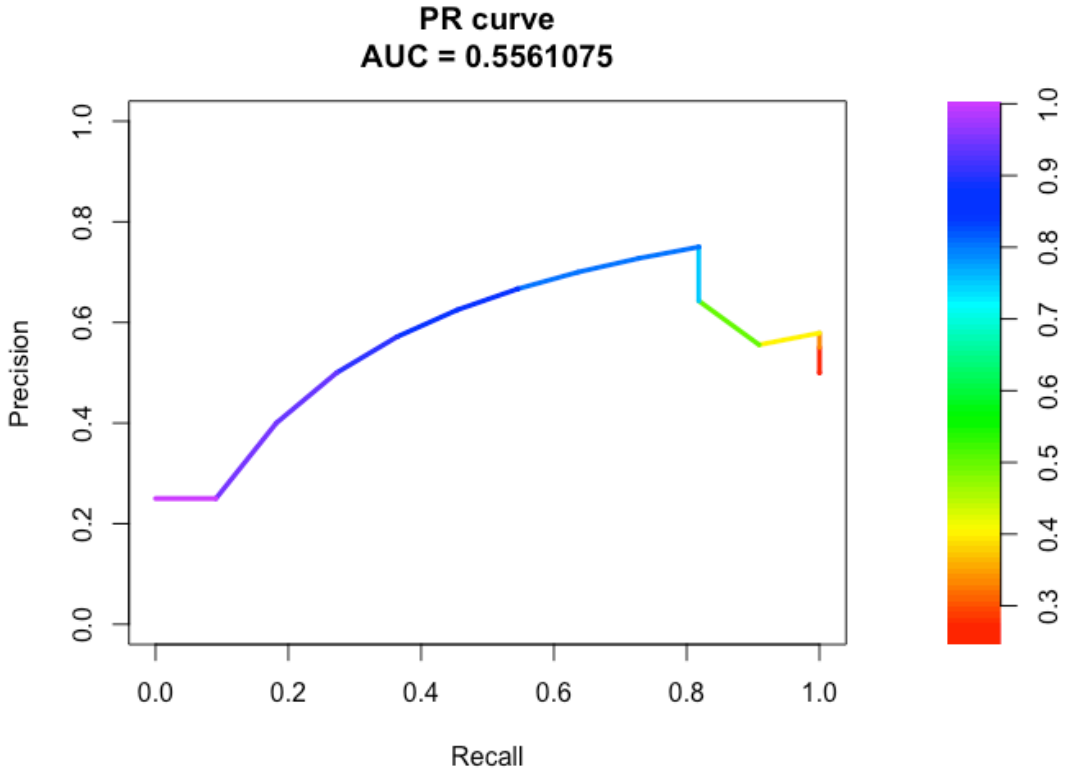

Figure 12 – PR curve for Utility Based algorithm

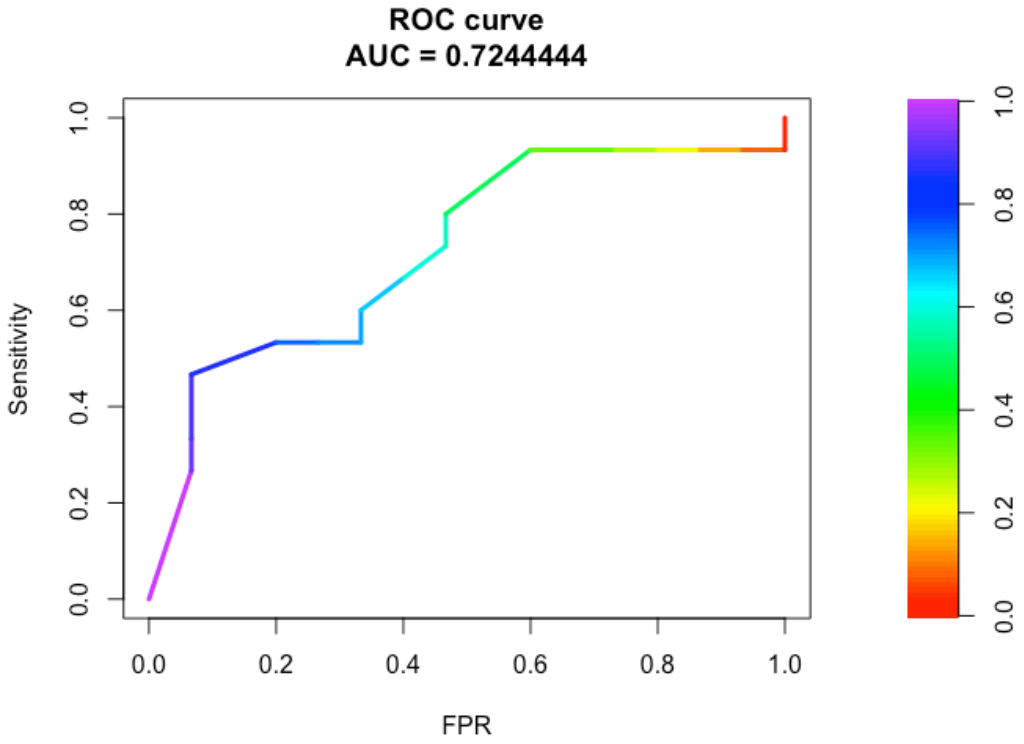

Figure 13 – ROC curve for A-priori algorithm

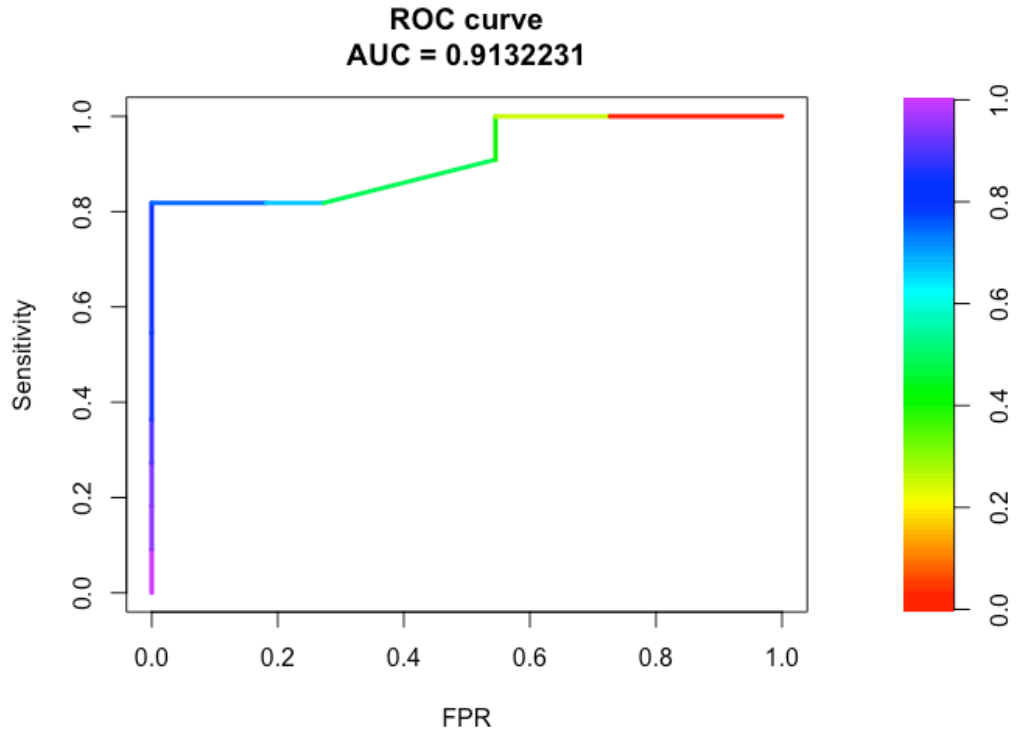

Figure 14 – ROC curve for Utility based algorithm

6.2 Conclusion

We have proposed a Bi-Phase algorithm that recognizes high utility item sets more effectively. The transaction-weighted utilization mining confines the search space and encompasses high utility item sets. Only one additional database prune is required to eliminate the overvalued item sets. This overcomes the limitations presented by the conventional frequent item set mining by by taking both frequency and utility in account as shown in the eq.s (1) and (2).

$$U(X1) = N1 \, X \, C1 \quad (1)$$

$$TU(X) = U(X1) + U(X2) + \cdots U(Xn) \quad (2)$$

where X1, X2, X3..∈ item set X having frequency N1, N2, N3.. and utility (cost) C1, C2, C3.. respectively and U(A1) is the utility of the item and TU(A) is the transactional utility of the item set A.

In the recommendation engine, we implemented all the rules we retrieved in order to check the various comparison measures between the prevalent traditional method i.e. the A-priori algorithm and our proposed method, the Utility based algorithm. We also incorporated a wholesome E-commerce website experience by providing various facilities such as User Login, like, comment and review faculties along with providing recommendations while the items are in cart and also when they've been checked out.

In the following table, we have included three measures upon which we've compared the two algorithm – namely Precision, Recall, F-score and ROC curve.

- Precision is the ratio of retrieved items that are relevant to the query and all the retrieved items. It was found that for the algorithm proposed by us, the measure of precision was **7 percentage points** more than the traditional algorithm.
- Recall is the ratio of relevant items retrieved and all items that are relevant to us. Similarly, it was found that the utility based algorithm exceeded the A-priori algorithm by the measure of **16 percentage points**.
- F-score is the single measure which comprehensively coalesces the precision and recall score, thus giving us a single measure for the same i.e. the mean of precision and recall. Hence as stated above, it was found that the F-score for traditional algorithm is less than the Utility based algorithm by **11 percentage points**.

- The higher AUC for PR curve and ROC curve for utility based rules signifies a greater accuracy of the recommender system as opposed to association rule based system.

Comparison Measures	A-priori Algorithm	Utility Based Algorithm
Precision	54%	**61%***
Recall	78%	**94%***
F-Score	0.63	**0.74***

Table 18 - Comparison Measures of Utility based Association rules and A-priori algorithm
***Bold values represent better performance**

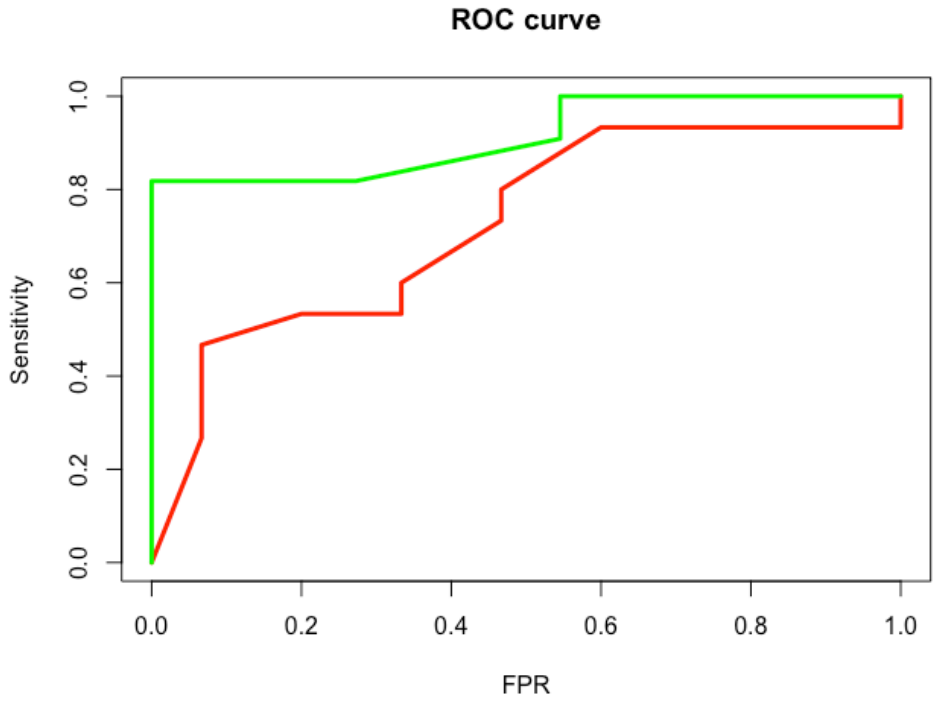

Figure 15 – Overlapping ROC curve
***Red - A-priori, Green – Utility based algorithm**

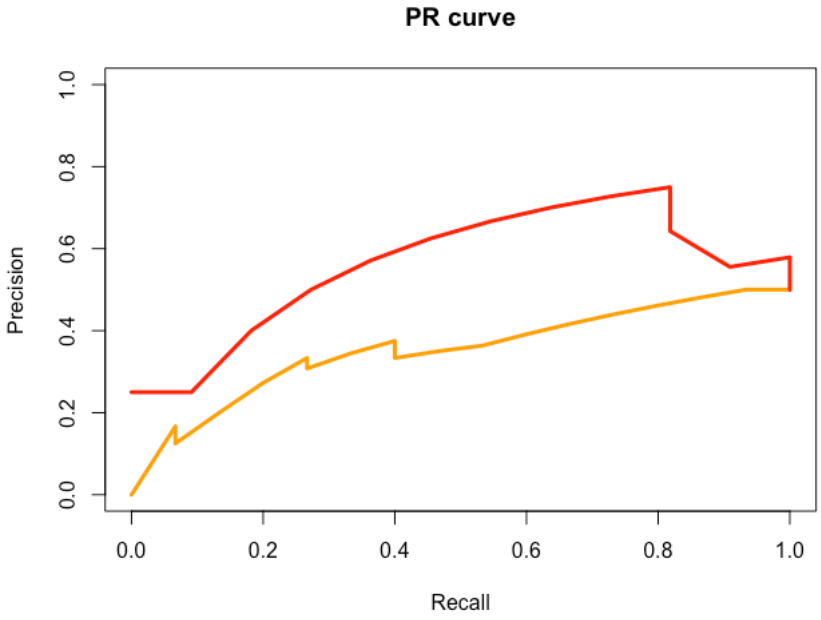

Figure 16 – Overlapping PR curve
***Orange - A-priori, Red – Utility based algorithm**

However, the algorithm had to be designed in such a way that the result contained all the rules that could prove to be of value. A very high threshold value in filtering could let some valuable results get lost and a very low value would cause rules that are unnecessary also creep into the final result file. Also high complexity of algorithms with huge data sets to be processed caused large running time, which was to be minimised to enhance the efficacy of the program considering the data contained noise which had to be filtered out to make sure the results generated had rules with maximum accuracy and confidence.

6.3 Future Work

This was initially introduced by the means of A-priori algorithm, the most known algorithm for association rule mining from a transactions given, satisfying the minimum support and confidence levels specified by users. A-priori however overestimates the results hence, efforts were made to retrieve useful rules from the data bases and various efficient methods have been proposed. Utility Mining then emerged as a whole new area of research which takes both frequency and utility by taking metrics like profit and sales into consideration. Ongoing research part of this project is to compare the algorithms, their implementation time and the accuracy as well as the usability of results that are generated. More study can also yield results as to which logic is more suited for which purpose, time complexity, and type of data that is to be analyzed. An algorithm generating rules with

high utility and simultaneously integrating the traditional concepts of minimum support and confidence can open new avenues in Market Basket Analysis and its applications. With the increasing customer base and the competitive market environment from a data so useful which can help only in expanding the horizon of not only for service providers but for service receivers. It is rightly said many times people don't know what they want until they see it and that is where we see the future of Recommendations Engine.

References

1. Agrawal, R., & Srikant, R. (1994, September). Fast algorithms for mining association rules. In *Proc. 20th int. conf. very large data bases, VLDB* (Vol. 1215, pp. 487-499).
2. Agrawal, R., & Srikant, R. (1995, March). Mining sequential patterns. In *Data Engineering, 1995. Proceedings of the Eleventh International Conference on* (pp. 3-14). IEEE.
3. Agrawal, R., Imieliński, T., & Swami, A. (1993, June). Mining association rules between sets of items in large databases. In *Acm sigmod record* (Vol. 22, No. 2, pp. 207-216). ACM.
4. Bhattacharya, S., & Dubey, D. (2012). High Utility Itemset Mining. International Journal of Emerging Technology and Advanced Engineering, 2(8), 476-481.
5. Cakir, O., & Aras, M. E. (2012). A recommendation engine by using association rules. *Procedia-Social and Behavioral Sciences*, *62*, 452-456.
6. Chen, Y. L., Tang, K., Shen, R. J., & Hu, Y. H. (2005). Market basket analysis in a multiple store environment. *Decision support systems*, *40*(2), 339-354.
7. Han, J., Pei, J., & Yin, Y. (2000, May). Mining frequent patterns without candidate generation. In *ACM sigmod record(Vol. 29, No. 2, pp. 1-12). ACM.*
8. Hipp, J., Güntzer, U., & Nakhaeizadeh, G. (2000). Algorithms for association rule mining—a general survey and comparison. *ACM sigkdd explorations newsletter, 2(1), 58-64.*
9. Hu, J., & Zhang, B. (2012). Product Recommendation System. *CS224W Project Report*.
10. J. McAuley, C. Targett, J. Shi, A. van den Hengel. Image-based recommendations on styles and substitutes. SIGIR, 2015
11. Jabbar, M. A., Deekshatulu, B. L., & Chandra, P. (2016). A Novel Algorithm for Utility-Frequent Itemset Mining in Market Basket Analysis. In *Innovations in Bio-Inspired Computing and Applications* (pp. 337-345). Springer, Cham.
12. Lee, D., Park, S. H., & Moon, S. (2013). Utility-based association rule mining: A marketing solution for cross-selling. *Expert Systems with applications*, *40*(7), 2715-2725.
13. Liao, C. W., Perng, Y. H., & Chiang, T. L. (2009). Discovery of unapparent association rules based on extracted probability. *Decision Support Systems*, *47*(4), 354-363.
14. Liu, Bing, Wynne Hsu, and Yiming Ma. "Mining association rules with multiple minimum supports." *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*. ACM, 1999.
15. Mobasher, B., Dai, H., Luo, T., & Nakagawa, M. (2001, November). Effective personalization based on association rule discovery from web usage data. In *Proceedings of the 3rd international workshop on Web information and data management* (pp. 9-15). ACM.

16. Mostafa, M. M. (2015). Knowledge discovery of hidden consumer purchase behaviour: a market basket analysis. *International Journal of Data Analysis Techniques and Strategies*, *7*(4), 384-405.
17. Ngai, E. W., Xiu, L., & Chau, D. C. (2009). Application of data mining techniques in customer relationship management: A literature review and classification. *Expert systems with applications*, *36*(2), 2592-2602.
18. R. He, J. McAuley. Modeling the visual evolution of fashion trends with one-class collaborative filtering. WWW, 2016
19. Raorane, A. A., Kulkarni, R. V., & Jitkar, B. D. (2012). Association rule–extracting knowledge using market basket analysis. *Research Journal of Recent Sciences.*
20. Reddy, B. Adinarayana, O. Srinivasa Rao, and M. H. M. Prasad. "An Improved UP-Growth High Utility Itemset Mining." *arXiv preprint arXiv:1212.0317* (2012).
21. Srikant, R., Vu, Q., & Agrawal, R. (1997, August). Mining association rules with item constraints. In *KDD (Vol. 97, pp. 67-73).*