



# Masters Project Final Report

December 2016

<b>Project Title</b>	<b>Product Recommendation for E-Commerce Application via Web Server Logs</b>		
<b>Student Name</b>	<b>L. G. Kavindu Bandara Thundeniya</b>		
<b>Registration No. &amp; Index No.</b>	<b>2013/MIT/83 13550831</b>		
<b>Supervisor's Name</b>	<b>H. A. Caldera</b>		
<b>Please Circle the appropriate</b>	<b>Master's Program</b>		<b>Type</b>
	<b><u>MIT</u></b>	<b>MCS</b>	<b>Research      <u>Implementation</u></b>

<b>For Office Use ONLY</b>

# **Product Recommendation for E-Commerce Application via Web server Logs**

**L.G. Kavindu Bandara Thundeniy  
2016**



# **Product Recommendation for E-Commerce Application via Web server Logs**

**A dissertation submitted for the Degree of Master  
of Information Technology**

**L.G.Kavindu Bandara Thundeniya  
University of Colombo School of Computing  
2016**



# Declaration

The thesis is my original work and has not been submitted previously for a degree at this or any other university/institute.

To the best of my knowledge it does not contain any material published or written by another person, except as acknowledged in the text.

Student Name: **L. G. Kavindu Bandara Thundeniya**

Registration Number: **2013/MIT/083**

Index Number: **13550831**

---

Signature:

Date:

This is to certify that this thesis is based on the work of Mr. L. G. Kavindu Bandara Thundeniya under my supervision. The thesis has been prepared according to the format stipulated and is of acceptable standard.

Certified by:

Supervisor Name: **H. A. Caldera**

---

Signature:

Date:

# Abstract

E-commerce applications use product recommendation on their application for increase sales. Application owners use their own business experience and business related assumptions for identify recommendations. These recommendations not accurate most of time and it leads application consumers with multiple choices for a specific product which leads to product overload. These incorrect recommendations badly affected for archived expected sales targets and give bad user experience. Therefor it is important to identify most accurate recommendations and present them on store front.

This developed system focused on web usage mining. System extract product association rules generated from machine learning tool and integrate those findings with E-Commerce application without user involvement. System use weka-3-6-12 as machine learning tool.

Scope of research includes access log data acquisition, data pre-processing, data cleaning, product based association rule generation and integrate identified recommendations with E-Commerce application.

Selected E-commerce application for integrated with developed system build on Magento e-commerce platform (Enterprise version 12) and hosted on Ubuntu server with Apache 2.0, PHP 5.5 and MySQL.

System generated recommendations evaluate with already used recommendations using data analytic tools with dependent sample t-Test. Test results prove system generated recommendations are more accurate and relevant compare with already used recommendations.

# Acknowledgement

My sincerest gratitude goes to my project supervisor Dr. H.A Caldera, Senior Lecturers at University of Colombo School of Computing, for the invaluable advices, guidance, comments and helpful discussions which enabled me to enthusiastically plan and carry out this project successfully. I am also indebted and grateful for helping me to complete this dissertation successfully.

My sincere thanks is also extended to Mr. Remco Dekker CTO of ISM E-Company, for his valuable guidance and encouragement throughout this project and valuable time to have progress meetings regarding the project.

Finally I wish to express my gratitude to my parents and my friends for their help, encouragement, blessings and guidance.

# Table of Contents

<b>Declaration .....</b>	<b>ii</b>
<b>Abstract.....</b>	<b>iii</b>
<b>Acknowledgement.....</b>	<b>iv</b>
<b>Table of Contents .....</b>	<b>v</b>
<b>List of Figures.....</b>	<b>viii</b>
<b>List of Tables .....</b>	<b>ix</b>
<b>List of Acronyms .....</b>	<b>x</b>
<b>CHAPTER 1 - INTRODUCTION.....</b>	<b>1</b>
1.1 Overview.....	1
1.2 Problem domain and the Motivation .....	2
1.2.1 Why Products Recommendation System Required .....	2
1.2.1 How to solve the problem.....	2
1.3 Objective of the Project.....	3
1.4 Project Scope.....	3
1.5 Benefits from the system.....	3
1.6 Structure of the system.....	5
<b>CHAPTER 2 -LITERATURE REVIEW .....</b>	<b>7</b>
2.1 Literature Review .....	7
2.2 Recommended Similar Systems.....	8
2.2.1 Similar Systems .....	8
2.2.2 Comparison of Recommendation Similar Systems .....	10
2.3 Data Mining Tools .....	12
2.3.1 Weka .....	12
2.3.2 KNIM.....	12
2.3.3 Orange.....	13
2.3.4 Comparison of Data Mining Tools .....	13
2.4 Data Mining Algorithms .....	14
2.4.1 Apriori Algorithm .....	14
2.4.2 Eclat Algorithm.....	15
2.4.3 FP-Growth Algorithm.....	15
2.5 Recommendation Techniques .....	15
<b>CHAPTER 3 –ANALYSIS AND DESIGN.....</b>	<b>18</b>

3.1	Proposed System .....	18
3.1.1	Data Acquisition .....	19
3.1.2	Data Pre-Processing .....	20
3.1.3	Data Cleaning .....	21
3.1.4	Pattern Analysis - Association Rules .....	22
3.2	Selected Technologies .....	23
3.3	User Interface Design .....	23
3.3.1	Back End User Interface .....	23
3.3.2	Front End User Interface .....	24
3.4	Database Design .....	26
<b>CHAPTER 4 –IMPLIMENTATION.....</b>		<b>27</b>
4.1	Introduction .....	27
4.2	Implemented Environment.....	27
4.3	Development Tools .....	27
4.3.1	PHPStrom 6.0 .....	27
4.3.2	DbVisualizer 7.0 .....	28
4.4	Major Code Structure .....	29
4.5	Reuse Existing Code .....	30
<b>CHAPTER 5 - EVALUATION .....</b>		<b>34</b>
5.1	Introduction .....	34
5.2	Test Plan.....	34
5.3	Evaluation .....	35
5.4	Project Assessment.....	37
<b>CHAPTER 6 - CONCLUTION .....</b>		<b>38</b>
6.1	Major Achievement .....	38
6.2	Support from MIT .....	38
6.3	Lesson Learnt .....	39
6.4	Future works.....	39
<b>REFERANCES .....</b>		<b>40</b>
<b>APPENDIX A – MAGENTO INSTALLATION .....</b>		<b>43</b>
<b>APPENDIX B – WEKA USER GUIDE.....</b>		<b>54</b>
<b>APPENDIX C – ROW DATA FILES .....</b>		<b>56</b>
<b>APPENDIX D – ARFF DATA FILES .....</b>		<b>58</b>
<b>APPENDIX E – ASSOCIATION RULES.....</b>		<b>60</b>



<b>APPENDIX F – CODE LISTING .....</b>	<b>64</b>
--	-----------

# List of Figures

Figure 1. 1: Recommendation widget.....	4
Figure 2. 1: Item to Item collaborative filtering approach.....	9
Figure 2. 2: Frequently Bought Together .....	9
Figure 2. 3: Feedback Ratings .....	10
Figure 2. 4: Apriori Algorithm .....	15
Figure 3. 1: Proposed System .....	18
Figure 3. 2: Describe Clickstream .....	19
Figure 3. 3: Data Pre-Processing Steps.....	20
Figure 3. 4: WEKA Data Format (ARFF) .....	21
Figure 3. 5: Back End User Interface .....	24
Figure 3. 6: Front End Use Interface .....	25
Figure 3. 7: Database Design .....	26
Figure 4. 1: Unprocessed Data File.....	28
Figure 4. 2: WEKA Data Format (ARFF) .....	29
Figure 4. 3: Code Structure.....	30
Figure 4. 4: Database Connection Code .....	31
Figure 4. 5: Data Pre-Processing Code.....	32
Figure 4. 6: Automate Shellsript .....	33
Figure 5. 1: Variable View .....	35
Figure 5. 2: Data View.....	36
Figure 5. 3: Generated Analyses Report.....	37

## List of Tables

Table 2. 1: Comparison of similar Recommendation Systems.....	12
Table 2. 2: Comparison of Data Mining Tools .....	14

## List of Acronyms

<b>WEKA</b>	- Waikato Environment for Knowledge Analysis
<b>PHP</b>	- PHP: Hypertext Preprocessor
<b>CRM</b>	- Customer Relationship Management
<b>CBF</b>	- Content-Based Filtering
<b>CF</b>	- Collaborative Filtering
<b>UID</b>	- Unique Identifier
<b>OOAD</b>	- Object Oriented Analysis and Design
<b>UML</b>	- Unified Modeling Language
<b>MVC</b>	- Model View Controller

# CHAPTER 1 - INTRODUCTION

## 1.1 Overview

E-commerce has obtained huge popularity in recent years. It has changed the traditional way of doing businesses. This rapid growth of e-commerce has generated new challenges to both customers and companies. The customer is provided with multiple opportunities for a specific product which leads to product overload. Therefore it has led to confused customer where customer is not able to choose effectively from the offered products [1]. As a result, the need for new marketing strategies such as one-to one marketing and customer relationship management (CRM) has been stressed both from researches [2]. One effective solution to address this issue is to make use of recommendation system that provides each customer with a list of product recommendation that customer would be interested in.

Recommendation system can be classified into: Content based system and Collaborative Filtering system. Content based system examines the properties of products recommended. Collaborative Filtering system makes use of product consumer interaction data and ignoring other facts to provide recommendation [3]. Collaborative filtering focuses on identifying customers whose interests are similar to those of a given customer and recommends relevant items of a given customer [2]. Despite its popularity and widespread use it suffers from two major limitations [4], [5]. The first is related to scarcity. The number of ratings already obtained is very less in comparison to the total number of ratings that need to be predicted since collaborative filtering requires explicit non-binary user ratings for like products. Therefore, collaborative filtering based recommendations systems are unable to accurately compute the related products and identify the right item to recommend. The second issue is related to scalability. As number of customers and products increases in an E-commerce site the computation time to locate related products grows linearly resulting in poor scalability [1], [4]. Studies in [6] show that web usage mining can be used to overcome the issues associated with collaborative systems.

## 1.2 Problem domain and the Motivation

### 1.2.1 Why Products Recommendation System Required

Most of E-commerce applications use product recommendation on their application for increase sales. Related products and Cross-sell products are most common and widely used two types of product recommendations. Related products appear in the product info page, but they are products that are meant to be purchased in addition to the one that the customer is viewing. Cross-sell items can appear both in the product page and in the shopping cart but they are a bit like an impulse buy – similar to items at the cash registers in grocery stores.

For the moment to identify related and Cross-sell products application owners use their business experience and some business related assumptions. This is not accurate most of the time. Those incorrect recommendations badly affected for archived expected sales targets. And also those incorrect recommendations give bad user experience, this will not help site user to find required information and relevant item quickly and easily. Because of that, incorrect recommendations badly affect for e-commerce applications and it is one of the biggest problem application owners has to come across.

### 1.2.1 How to solve the problem

Web usage mining is an application of data mining techniques to discover interesting and useful patterns from web data. User's click stream data can act as a very rich source of information to provide effective recommendation. Click stream data is defined as customer's path through a website. It provides information about customers shopping pattern and behaviour, like details about the products viewed by the customer, the products they buy, they items they add to their shopping cart etc. This information is captured in web log files. Analyses of this usage data helps identify customers' preferences and interests. Furthermore, this data can be used to discover interesting relationships, correlation and rules. In our proposed system we try to provide the customer with better quality recommendations. A good quality recommendation has a significant impact on customers' future shopping behaviour. Poor quality of recommendations can lead to two types of distinct errors: false negatives, items that are not recommended even

though the customers like them and false Positive, items that are recommended even though the customer does not like them. In an E-commerce domain, the most important errors to avoid is false positives, as it can result in irritated, unsatisfied customers thus reducing their probability to revisit the site once again. Therefore it is highly important to provide the customer with the type of product he or she is interested in.

### 1.3 Objective of the Project

Instead of business related guessing and assumption, we provide a solution based on access.log analyse. E-commerce application access log store following information on each user request. The IP address of visitor, The date and time of the visit, The file requested, The request status (success (200), failed(404)), Number of bytes that were transferred, Where visitor came from and the keywords used to find site, Browser and operating system of the visitor.

Analysing above information, we can identify users buying patterns (what are the other product already on cart in same user sessions), site navigation (what are the other pages user view on same session before place order) etc... Base on that, we can provide more accurate Related Products, Cross-sell products that provide better user experience. Users will be able to find required information and relevant products easily and quickly. That helps to archived sales target of e-commerce applications.

### 1.4 Project Scope

Scope of the project consist with access log data acquisition, data pre-processing, data cleaning, product based association rule generation and integrate identified recommendations with E-Commerce application.

### 1.5 Benefits from the system

Developed system gives following benefits. Mainly after integrated developed system with exist e-commerce application give better user experience. Show users personalized and more relevant products while there on the site. Also give users discover more of what they like. Few other benefits list below.

## Products Recommendation for E-commerce Application via Web Server Logs

- Increased Traffic/Page Views. By giving users more relevant suggestion, web site receives more traffic and higher business.
- Increased Sales.
- Improve customer retention. The more users interest on site, the more they buy, the higher they are likely to return.

This diagram shows (Figure 1.1) a visitors' shopping journey and what recommendation widgets will help you capitalize on their intent.

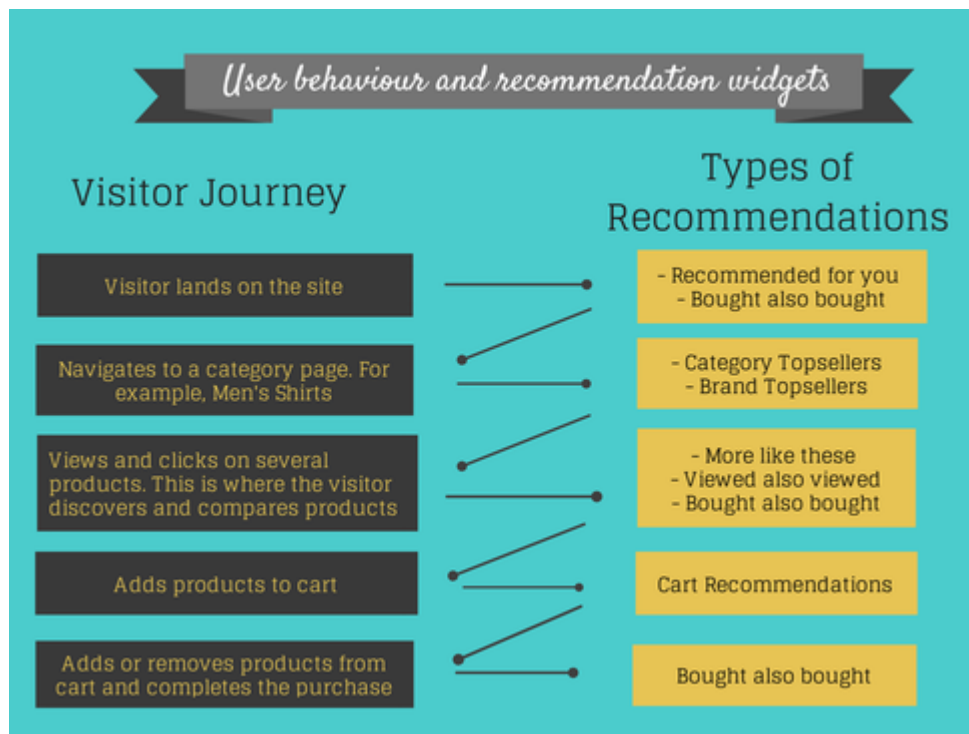


Figure 1. 1: Recommendation widget



## 1.6 Structure of the system

### **Chapter 1 – Introduction**

The first chapter will simply describe overview of the system and background of the current system and it includes the introduction of the project, and the introduction to the problem, the scope, objectives of the project finally structure of the thesis.

### **Chapter 2 – Literature Review**

This chapter will examine the literature that is relevant to understanding the development of, and interpreting the results of this convergent study.

### **Chapter 3 – Analysis and Design**

This chapter describes the background of the system. It describes similar systems that available and alternative technologies available relevant to the system. It will include essential background information with references and benefits and weaknesses of those alternatives. The design diagrams and methodical approach to the design of the system is being described and alternative solutions and their justification for not using in the system.

### **Chapter 4 – Implementation**

This chapter covers the implementation techniques used in the project. The Hardware and software requirements also being described get together with tools and techniques.

### **Chapter 5 – Evaluation**

The definition of evaluation is often problematic and it can be argued that evaluation does not need a definition. This chapter provides evidence about testing. A proper test plan is explained in order to verify and validate test cases.

### **Chapter 6 – Conclusion**

In this chapter critical evaluations of the system and suggestions provided for future use. The lessons learnt during the project and the success criteria also being included in this chapter.

### **References**

List the references which have been used.

### **Appendix A-Magento Installation**

Server configuration and magento installation contain in this appendix A

#### **Appendix B-WEKA User Guide**

Brief introduction about WEKA machine leaning tool in this appendix.

#### **Appendix C-Row Data Files**

Information about access log and sample file includes this appendix C.

#### **Appendix D-ARFF Data Files**

Describe ARFF Data File with sample od system generated data files.

#### **Appendix E-Association Rules**

Associated rules extracted using Apriori algorithm with WEKA machine learning tool.

#### **Appendix F-Association Rules**

Major codes used on system are listed in this Appendix F.

## CHAPTER 2 -LITERATURE REVIEW

This chapter provides an overview of previous research on knowledge sharing and intranets. It introduces the framework for the case study that comprises the main focus of the research described in this dissertation.

### 2.1 Literature Review

The entire process of web usage mining is broadly divided into two important tasks: data preparation and pattern discovery. [2] Web servers hold the data required for web usage mining. Estimates in [8], [9] show that 80% of data mining time goes in pre-processing the web log data. The pre-processing task can follow either of the two techniques: In the first technique web logs are mapped into corresponding data formats, and then appropriate mining algorithms are adapted to further analyse it.[10] The second technique makes use of special pre-processing process to convert the log data to fit specific mining algorithms. The data preparation tasks construct a server session file where each session is a sequence of requests of different types made by single user during a single visit to a site. [2] A set of various pre-processing tasks are followed for web log data. A detailed description of data preparation methods for mining web browsing patterns is given in [11]. Different methods to discover usage patterns namely Apriori [12], Naïve Bayesian [13], and Agglomerative clustering [14] are discussed. The pattern discovery tasks involve the discovery of association rules, sequential patterns, and user classifications [2]. Usage pattern extracted from web data can be applied to a wide range of applications [2].

This research provides product recommendation based on web log data, sales data and customer related data [2]. Last few years, a wide variety of recommendation techniques have been known and developed. Most of the current recommendation systems recommend products that have a high probability of being purchased [16]. They employ content-based filtering (CBF) [17], collaborative filtering (CF) [18] [19], and other data mining techniques, for example, decision tree [20], association rule [21], and semantic approach [21].

There are two methods in CF as user based collaborative filtering and item based collaborative filtering [21]. User based CF assumes that a good way to find a certain user's interesting item is to find other users who have a similar interest. So, at first, it tries to find the user's neighbours based on user similarities and then combine the neighbour users' rating scores, which have previously been expressed, by similarity weighted averaging. And item based CF fundamentally has the same scheme with user based CF. It looks into a set of items; the target user has already rated and computes how similar they are to the target item under recommendation. After that, it also combines his previous preferences based on these item similarities [21].

Authors in [7] discuss a Navigation Pattern that constructs a tree to store web access information using NP-Miner Algorithm. Based on this information real time recommendations are provided to online users. The research proves that this algorithm efficiently performs online dynamic recommendation in a stable manner. In [1] [2] a personalized recommendation system for an Internet shopping mall is described. This system makes use of web usage data, association rules, and product taxonomy and decision tree induction to provide better quality recommendation. This research tries to provide effective recommendation to all visitors of an E-commerce site regardless of them being registered or unregistered. This helps us to retain existing customers and attract one time visitors of the site.

## 2.2 Recommended Similar Systems

In this section presents E-commerce businesses that apply one or more alternations of recommender system technology in their web sites.

### 2.2.1 Similar Systems

#### **Amazon.Com**

Amazon use massive use of an item to item collaborative filtering approach(See Figure 2.1). For each item Amazon builds a neighborhood of related items when someone buy/look at an item , Amazon then recommends items from that item's neighborhood.



Figure 2. 1: Item to Item collaborative filtering approach

User to user collaborative filtering approach is varying in item to item collaborative filtering approach. There are two separate recommendation lists in information page for each book in their catalog. The first recommends books frequently purchased by customers who purchased the selected book. The second recommends authors whose books are frequently purchased (See Figure 2.2) by customers who purchased works by the author of the selected book.

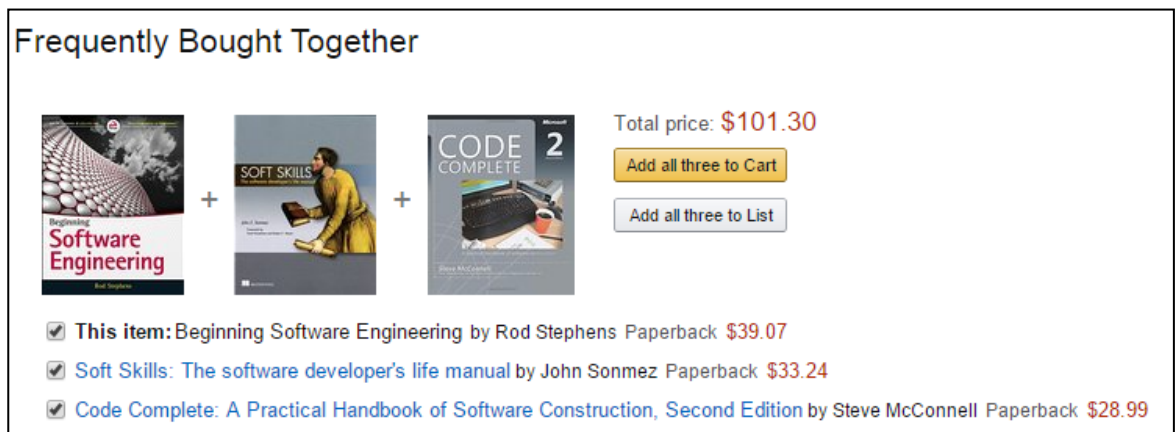


Figure 2. 2: Frequently Bought Together

## eBay

eBay Feedback profile feature allows both buyers and sellers to contribute to feedback profiles of other customers with whom they have done business. Feedback is an important part of the eBay community. When you understand what the numbers and stars mean, find it easier to evaluate a member's reputation. The number of positive, negative, and neutral Feedback ratings a member has received over time is part of the Feedback score (See Figure 2.3).

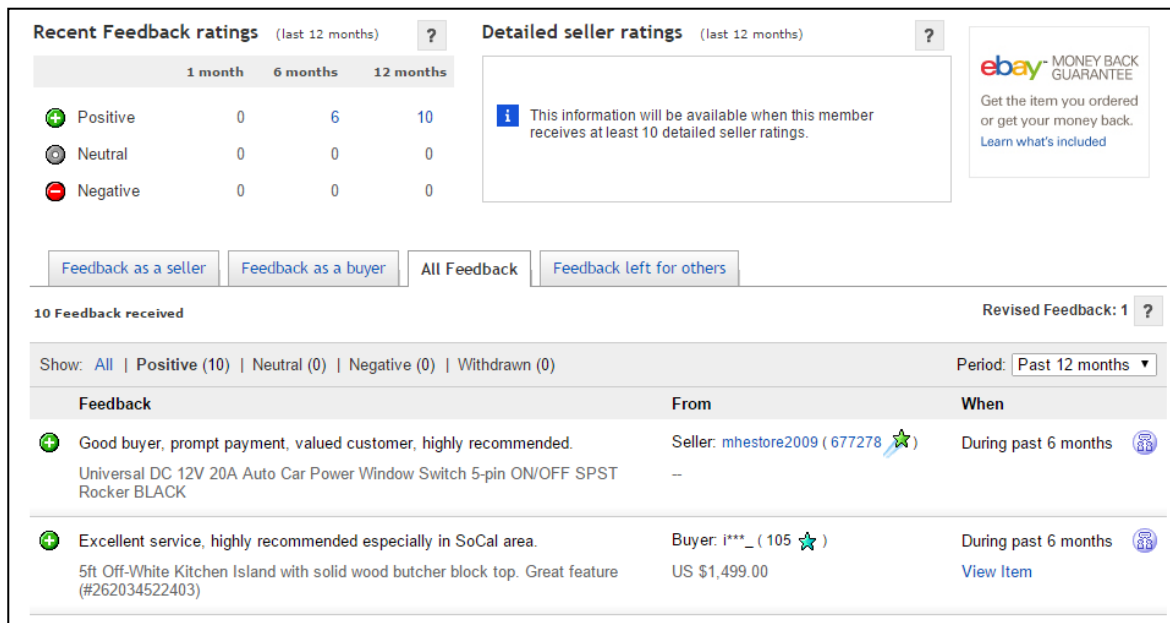


Figure 2. 3: Feedback Ratings

### 2.2.2 Comparison of Recommendation Similar Systems

The following, Table 2.1 is the Comparison of recommendation similar systems.

System	System Feature	Recommendation Interface	Recommendation Technology	Finding Recommendations
Amazon.com	Customers who Bought	Similar Item	Item to Item Correlation Purchase data	Organic Navigation
	Eyes	Email	Attribute Based	Keywords/

# Products Recommendation for E-commerce Application via Web Server Logs

				freeform
	Amazon.com Delivers	Email	Attribute Based	Selection options
	Book Matcher	Top N List	People to People Correlation Likert	Request List
	Customer Comments	Average Rating Text Comments	Aggregated Rating Likert Text	Organic Navigation
CDNOW	Album Advisor	Similar Item Top N List	Item to Item Correlation Purchase data	Organic Navigation Keyword/freeform
	My CDMOW	Top N List	People to People Correlation Likert	Organic Navigation Request List
eBay	Feedback Profile	Average Rating Text Comments	Aggregated Rating Likert Text	Organic Navigation
Levis	Style Finder	Top N List	People to People Correlation Likert	Request List
Moviefinder.com	Match Maker	Similar Item	Item to Item Correlation Editor's choice	Navigate to an item
	We Predict	Top N List Ordered Search Results Average Rating	People to People Correlation Aggregated Rating Likert	Keywords/freeform Selection options Organic Navigation

Reel.com	Movie Matches	Similar Item	Item to Item Correlation Editor's choice	Organic Navigation
	Movie Map	Browsing	Attributed Based Editor's choice	Keywords/freeform

Table 2. 1: Comparison of similar Recommendation Systems

## 2.3 Data Mining Tools

Here in this section the open source data mining tools are mentioned

### 2.3.1 Weka

Weka is a collection of machine learning algorithms for data mining tasks. WEKA non-Java version was mainly developed for analyzing data from the agricultural domain. In Java-based version, the tool is practical and used in many different applications including visualization and algorithms for data analysis and predictive modeling. Weka contains tools for data pre-processing, classification, regression, clustering, association rules, and visualization. It is also well-suited for developing new machine learning schemes.

The key features of Weka,

- It provides many different algorithms for data mining and machine learning
- It is open source and freely available
- It is platform-independent
- It is easily useable by people who are not data mining specialists
- It provides flexible facilities for scripting experiments
- It has kept up-to-date, with new algorithms.

### 2.3.2 KNIM

Konstanz Information Miner (KNIM) is an open source data analytics, reporting and integration platform. KNIME integrates various components for machine learning and



data mining through its modular data pipelining concept. A graphical user interface allows assembly of nodes for data pre-processing, for modelling and data analysis and visualization.

### 2.3.3 Orange

Orange is a component-based data mining and machine learning software suite and featuring a visual programming front-end for explorative data analysis and visualization, and Python bindings and libraries for scripting. It includes a set of components for data pre-processing, feature scoring and filtering, modelling, model evaluation, and exploration techniques. It is implemented in C++ and Python.

### 2.3.4 Comparison of Data Mining Tools

The following, Table 2.2 is the Comparison of Data Mining Tools.

Procedure	KNIME	RapidMiner	Weka	Orange
Partitioning of dataset to training and testing sets	Limited	Limited	Limited	Limited
Descriptor Scaling	Have Facility	Have Facility	Does not have the facility to save parameters for scaling to apply to future datasets	Does not have the facility of scaling
Descriptor Selection	Wrapper methods	Have Facility	Have the facility but not the part of knowledge flow	Wrapper methods
Parameter optimization of machine	No automatic facility	Have Facility	Does not have automatic facility	Does not have automatic facility

learning methods				
Model validation using cross validation and/or independent validation set	Only limited error measurement methods	Have Facility	Have the facility but is not capable of saving the model so have to rebuild model for every future data set	Have the facility but is not capable of saving the model so have to rebuild model for every future data set

Table 2. 2: Comparison of Data Mining Tools

## 2.4 Data Mining Algorithms

Data mining is method of extracting the useful information and knowledge from very large amount of data. Apriori, is basic algorithm for finding frequent item sets. But it takes more time for finding the frequent item sets, It needs to scan the database again and again which is time consuming process. Eclat algorithms generate frequent items only once. Frequent item sets are those items which are frequently occur in the database. There are number of algorithms for finding frequent item sets.

### 2.4.1 Apriori Algorithm

Apriori is an algorithm (See Figure 2.3) for frequent item set mining and association rule learning over transactional databases. It proceeds by identifying the frequent individual items in the database and extending them to larger and larger item sets as long as those item sets appear sufficiently often in the database. The frequent item sets determined by Apriori can be used to determine association rules which highlight general trends in the database: this has applications in domains such as market basket analysis [G].

```

 $F_1 = \{\text{frequent 1-itemsets}\};$ 
for ( $k = 2; F_{k-1} \neq \emptyset; k++$ ) do begin
   $C_k = \text{apriori-gen}(F_{k-1});$  //New candidates
  foreach transaction  $t \in \mathcal{D}$  do begin
     $C_t = \text{subset}(C_k, t);$  //Candidates contained in  $t$ 
    foreach candidate  $c \in C_t$  do
       $c.\text{count}++;$ 
    end
     $F_k = \{c \in C_k \mid c.\text{count} \geq \text{minsup}\};$ 
  end
 $\text{Answer} = \cup_k F_k;$ 

```

Figure 2. 4: Apriori Algorithm

### 2.4.2 Eclat Algorithm

Eclat algorithm is very simple algorithm to find the frequent item sets. This algorithm uses vertical database. It cannot use horizontal database. Éclat algorithm scans the database only once [H].

### 2.4.3 FP-Growth Algorithm

FP-Growth Algorithm allows frequent item set discovery without candidate item set generation. There are two steps:

- **Step1:** Build a compact data structure called FP-Tree (Build using 2 phases over the data set)
- **Step2:** Extracts frequent item sets directly from the FP-Tree (Traversal through the FP-Tree)

## 2.5 Recommendation Techniques

Effective recommendation is provided even in the case, where the user clears the cache memory on his/her browser. Also it tries to provide better recommendations if different users access the same system and browser by providing a combination of recommendation based on most recent session and timestamp.

Following assumptions for the recommendation:

**1) Product based recommendation technique:**

This technique is more suitable to provide recommendation for unregistered users.

Following assumptions for the recommendation:

- i. For a Website, a session  $S$  is a collection of sequence of Web pages  $\{url1, url2, \dots, urln\}$ .
- ii. A session identifier is associated with each session  $\{s\_id1, s\_id2, \dots, s\_idn\}$ .
- iii. Each url consists of important information such as ip address, time stamp, product identifier url1  $\{ip\_addr, ts, p\_id\}$  which will be considered for analysis.
- iv. Every product  $P$  is associated with product identifier  $\{p\_id1, p\_id2, \dots, p\_idn\}$ , a manufacturer identifier  $\{m\_id1, m\_id2, m\_id3, \dots, m\_idn\}$  and a category Identifier  $\{c\_id1, c\_id2, \dots, c\_idn\}$ .

In this approach we fetch last three sessions based on most recent timestamp. For each session we extract products in descending order and place last two products in the recommendation list. If the recommendation list, has less than ten products we fetch related product based on category and manufacturer details. If recommendation list has any redundant product than we filter it out and related product is added to the recommended set.

This approach helps us to reduce false positive errors that normally occur in traditional recommendation technique.

**2) User based recommendation technique**

This technique provides recommendation to the registered users of the web portal.

Following assumptions are made for recommendation.

- i. We have  $m$  registered users and  $n$  transactions in processed log file.
- ii. Let each user be associated with unique identifier (UID).
- iii. We assume that we have a minimum of three sessions for each user to provide effective recommendation since we are using mining operation.

A list of ten products is shown as recommendation. Other terminology used for user based technique is similar to product based recommendation technique. For each unique

## Products Recommendation for E-commerce Application via Web Server Logs

user that exists in log file we fetch sessions in descending order based on most recent timestamp. In each session we retrieve all visited products. If a specific product is already ordered then it is discarded from the recommendation list and related product is added in recommendation set. If a specific product is added to cart or wish list then it is shown as top recommendation. If recommendation list has a count of less than ten then we check for the next recent session and repeat the above procedure both the technique provides recommended products to the end user. Based on this recommendation list pattern analysis is done.

## CHAPTER 3 – ANALYSIS AND DESIGN

This chapter identifies specific requirements of the project in detail. Functional and non-functional requirements of the system are being identified. The design diagrams and methodical approach to the design of the system is being described and alternative solutions and their justification for not using in the system.

### 3.1 Proposed System

The proposed system lots of users visiting the E-Commerce Application and entire clickstream data will be collected and maintained in a log file. The log file will then be processed to remove irrelevant data. Different techniques are then applied on cleaned log file to provide effective recommendation. The proposed system is shown in Figure 3.1.

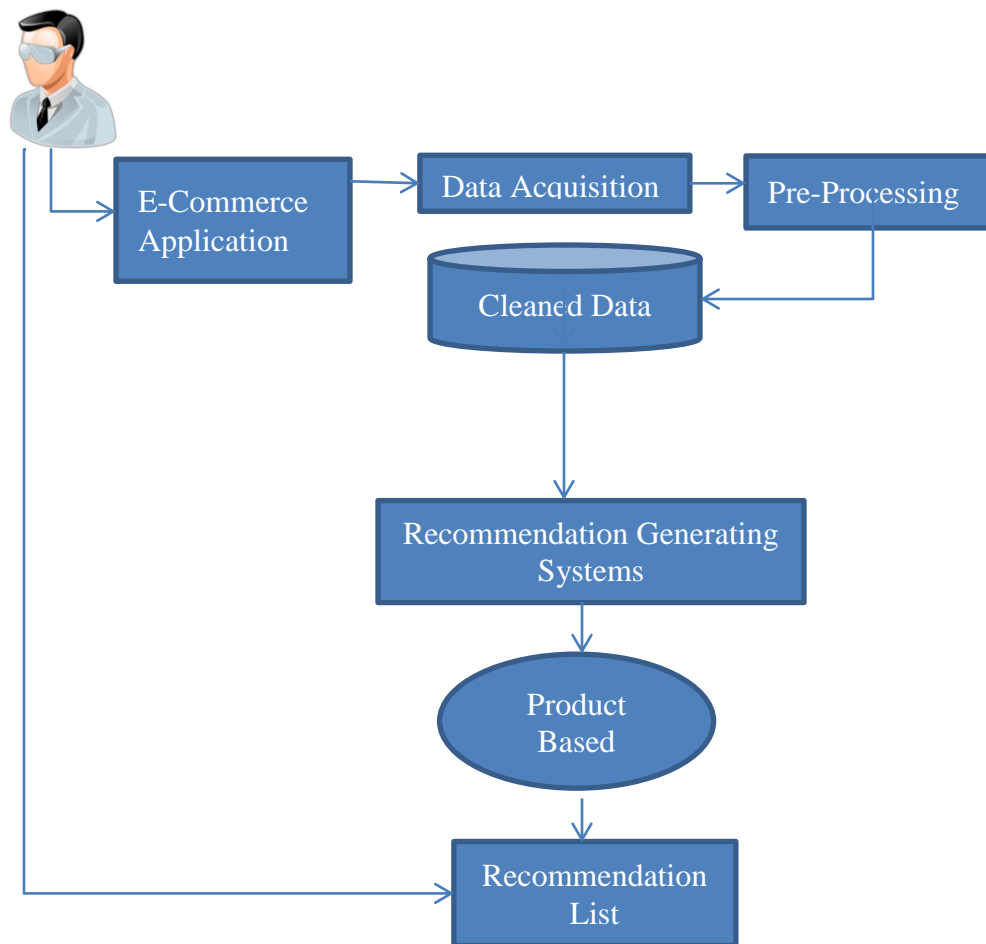


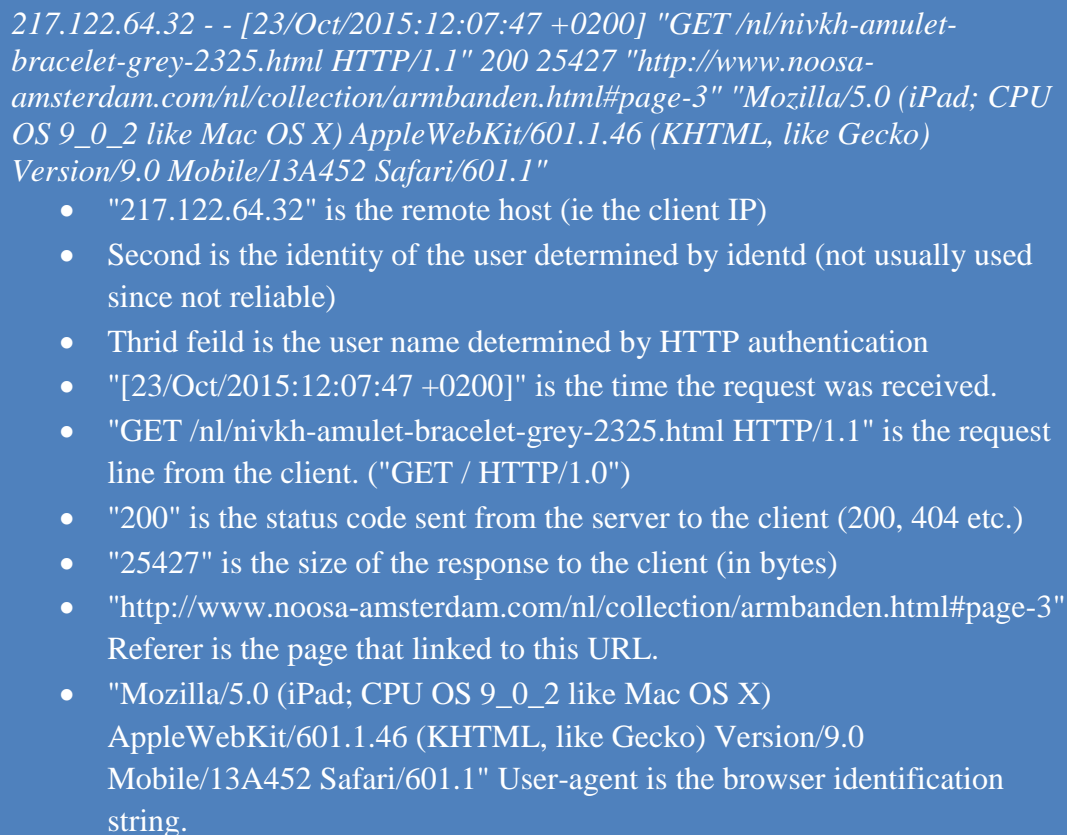
Figure 3. 1: Proposed System

The entire system has following phrases:

- Data Acquisition
- Data Pre-Processing
- Data Cleaning
- Pattern Analysis
- Recommendation Generation

### 3.1.1 Data Acquisition

In this phase the entire clickstream data of all the customers, which consists of all the web pages visited is collected and maintained in a log file which contain common log file format. Example clickstream described in Figure 3.2.



*217.122.64.32 - - [23/Oct/2015:12:07:47 +0200] "GET /nl/nivkh-amulet-bracelet-grey-2325.html HTTP/1.1" 200 25427 "http://www.noosa-amsterdam.com/nl/collection/armbanden.html#page-3" "Mozilla/5.0 (iPad; CPU OS 9\_0\_2 like Mac OS X) AppleWebKit/601.1.46 (KHTML, like Gecko) Version/9.0 Mobile/13A452 Safari/601.1"*

- "217.122.64.32" is the remote host (ie the client IP)
- Second is the identity of the user determined by identd (not usually used since not reliable)
- Thrid feild is the user name determined by HTTP authentication
- "[23/Oct/2015:12:07:47 +0200]" is the time the request was received.
- "GET /nl/nivkh-amulet-bracelet-grey-2325.html HTTP/1.1" is the request line from the client. ("GET / HTTP/1.0")
- "200" is the status code sent from the server to the client (200, 404 etc.)
- "25427" is the size of the response to the client (in bytes)
- "http://www.noosa-amsterdam.com/nl/collection/armbanden.html#page-3" Referer is the page that linked to this URL.
- "Mozilla/5.0 (iPad; CPU OS 9\_0\_2 like Mac OS X) AppleWebKit/601.1.46 (KHTML, like Gecko) Version/9.0 Mobile/13A452 Safari/601.1" User-agent is the browser identification string.

Figure 3. 2: Describe Clickstream

### 3.1.2 Data Pre-Processing

Good and exceptional quality of data should be provided as an input in effective data analysis. Irrelevant and Inconsistent data is consists in collected web log data and need to be cleaned for effective analysing. Following steps are followed for data pre-processing as shown in Figure 3.3.



Figure 3. 3: Data Pre-Processing Steps

**Field Split:** Split individual fields by making use of separator character such as space.

**Data Cleaning:** Elimination of useless data.

**User Identification:** It is important to determine between different users for analysing different user access behaviour patterns. A different user ID will be assigned to different IP address. In case of same IP address referrer information and browser details will be used to distinguish among different web users.

**Session Identification:** A session is defined as an ordered sequence of web pages visited by a user. Maximum session time consider to be 24 hours.

**Data Formatting:**

Data will be formatted to The Weka data format (ARFF):

A dataset start with a declaration of its name:

➤ @relation supermarket

Followed by a list of all the attributes in the dataset

➤ @attribute nominal\_attribute {first\_value}

For each product create nominal attribute with product id

After the attribute declarations, the actual data is introduced by a

➤ @data



Data Tag which is followed by a list of all the instances. The instances are listed in comma separated format, with a question mark representing a missing value.

Example is shown in Figure 3.4

```
Example:
@relation supermarket
@attribute '152' { t}
@attribute '154' { t}
.
.
.
@attribute '1697' { t}
@attribute '1698' { t}
@data
?,?,?,?,?,?,?,?,?,?,?,?,?,t,?,?,?,?,?,?,?,?,?,?,?,?,?,t,?,?,?,?,t,?,?
t,t,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,?,t,?,?,?,?,?,?,?,?,?,?
?,?,?,?,?,?,?,?,?,t,?,?,?,?,?,?,?,?,?,t,?,?,?,?,?,?,?,?,?,?,?,?,?,?
```

Figure 3. 4: WEKA Data Format (ARFF)

### 3.1.3 Data Cleaning

Data cleaning is a process of filtering out irrelevant and outliers' data [1].The following requests has been excluded from analysis:

- Filename suffixes such as gif, jpeg, GIF, JPEG, and JPG are removed
- Automatically generated by Web client browsers, requests generated by Web bots. (e.g. Web crawlers – Googlebot, bingbot, NewRelicPinger, nagios plugins, monitoring plugins, msnbot, crawl, slurp, spider, dotbot)
- Requests connected with administrative tasks.

### 3.1.4 Pattern Analysis - Association Rules

Integrated formatted data with WEKA Machine learning tool can be generated associated rules.

- **Data mining perspective**
  - ✓ Market basket analysis: looking for associations between items in the User Session.
  - ✓ Rule form: Body  $\Rightarrow$  Head [support, confidence]
  - ✓ Example: buys(x, “diapers”)  $\Rightarrow$  buys(x, “beers”) [0.5%, 60%]
- **Machine learning approach:** treat every possible combination of attribute values as a separate class, learn rules using the rest of attributes as input and then evaluate them for support and confidence. Problem: computationally intractable (too many classes and consequently, too many rules).
- **Best rules found:**
  - 1755=t 1758=t 1846=t 14  $\Rightarrow$  1839=t 13    conf:(0.93)
  - 1758=t 1839=t 20  $\Rightarrow$  1846=t 18    conf:(0.9)
  - 1755=t 1758=t 1839=t 15  $\Rightarrow$  1846=t 13    conf:(0.87)
- ✓ Tuples are transactions, attribute-value pairs are items.
- ✓ Association rule: {1755,1758,1846,...}  $\Rightarrow$  {1839,...}, where 1755,1758,1758,... are items.
- ✓ Confidence (accuracy) of  $A \Rightarrow B$  :  $P(B|A) = (\# \text{ of transactions containing both } A \text{ and } B) / (\# \text{ of transactions containing } A)$ .
- ✓ Support (coverage) of  $A \Rightarrow B$  :  $P(A,B) = (\# \text{ of transactions containing both } A \text{ and } B) / (\text{total } \# \text{ of transactions})$
- ✓ We looking for rules that exceed pre-defined support (minimum support) and have high confidence.

## 3.2 Selected Technologies

After analysis the business problem and discussed with supervisor it was decide to use WEKA as machine learning tool and use PHP for develop recommendation system.

WEKA is platform independent portable software. It is freely available under GNU (General Public License) and there is very large collection of different data mining algorithms supported for WEKA.

## 3.3 User Interface Design

### 3.3.1 Back End User Interface

Web shop Administrator can use following back end interface for change related product information shown in Figure 3.5.

# Products Recommendation for E-commerce Application via Web Server Logs

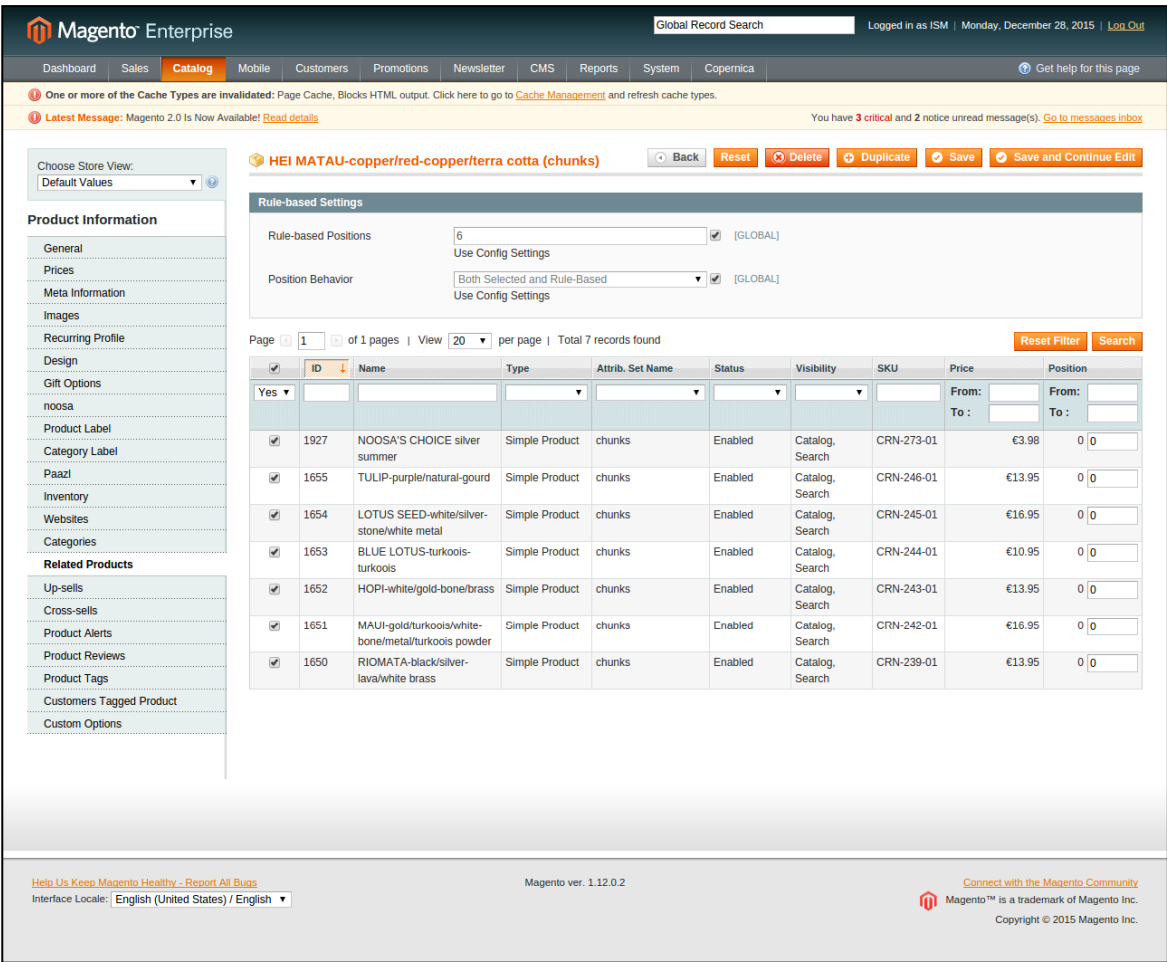


Figure 3. 5: Back End User Interface

## 3.3.2 Front End User Interface

Related products are display on bottom section of main product detail page shown in Figure 3.6.

## Products Recommendation for E-commerce Application via Web Server Logs

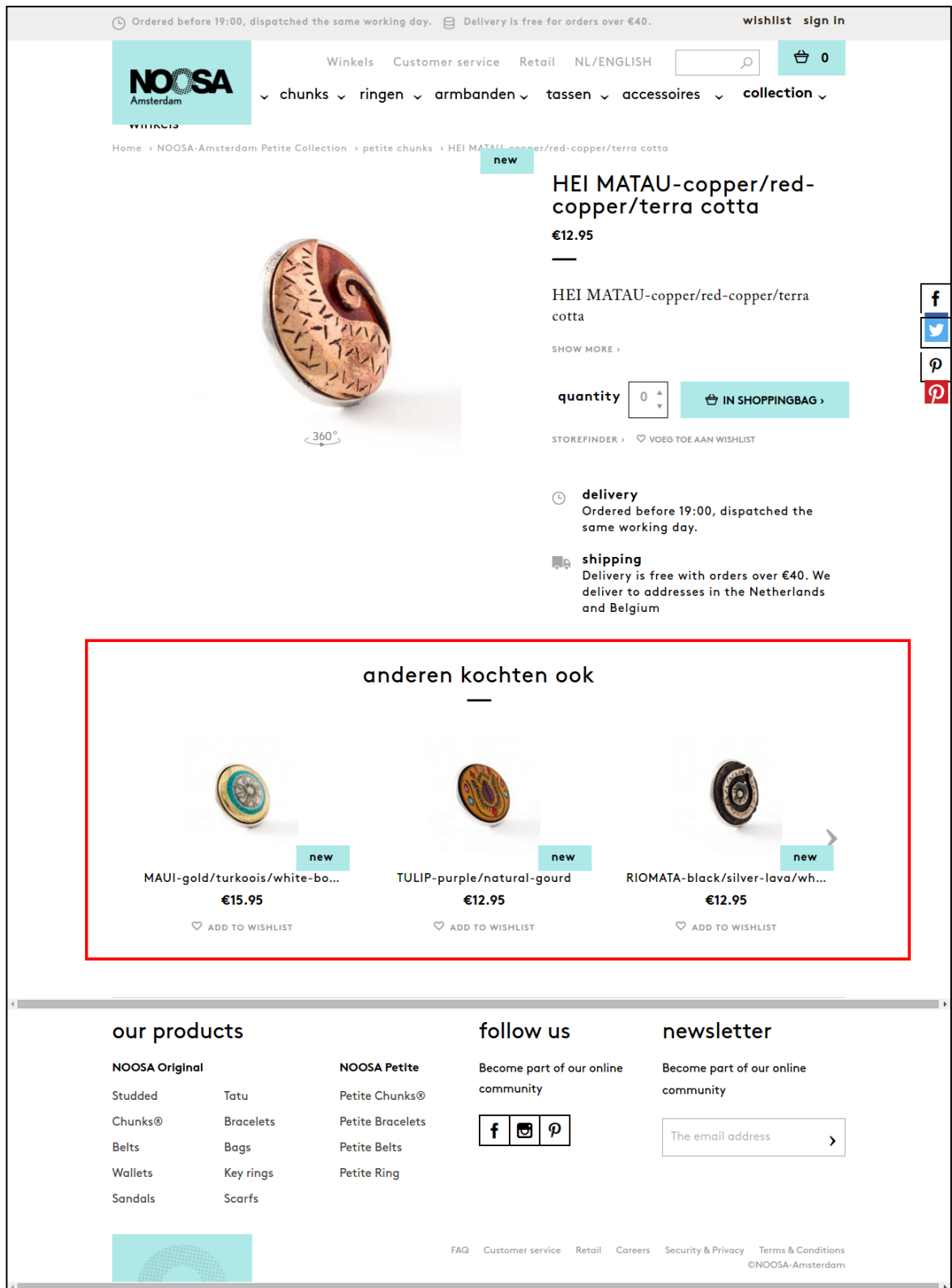


Figure 3. 6: Front End Use Interface

### 3.4 Database Design

- ✓ catalog\_product\_link\_type table content available products types.  
Ex: Related products, Cross-sell Products and Up-sells products
- ✓ catalog\_product\_link table store main products and related products association.
- ✓ catalog\_product\_entity table maintain basic products information.

Above details shown in Figure 3.7.

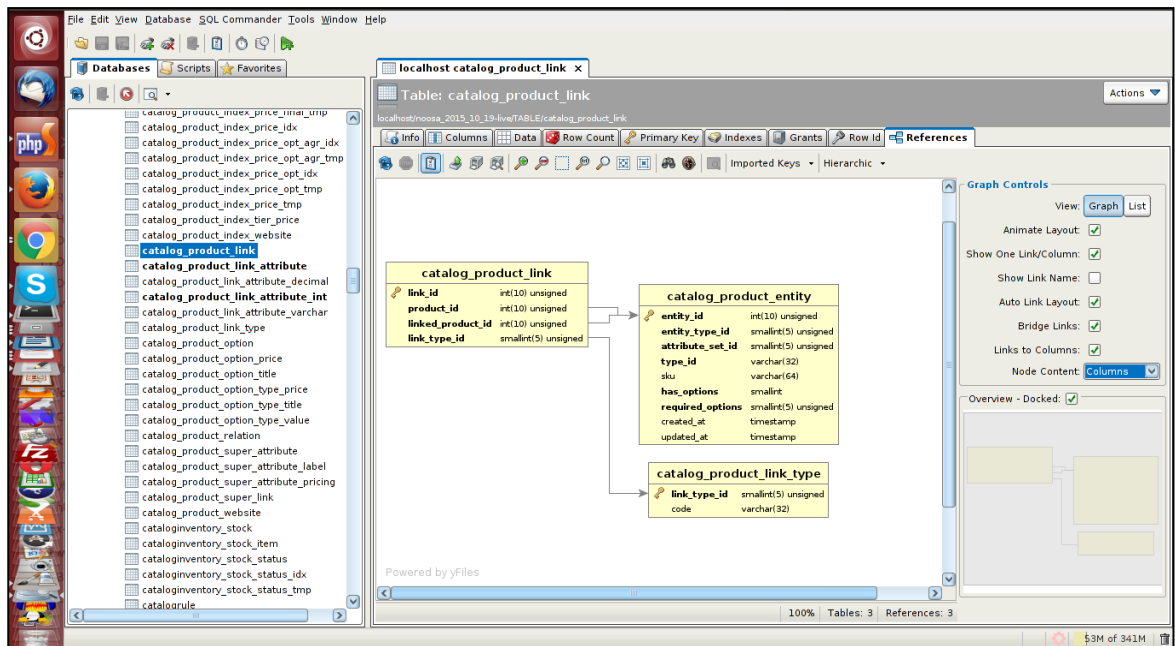


Figure 3. 7: Database Design

## CHAPTER 4 –IMPLEMENTATION

This chapter covers the implementation techniques used in the project. The Hardware and software requirements also being described get together with tools and techniques.

### 4.1 Introduction

The System Implementation comprise of Installing and configuring of the Magento E-Commerce platform, and WEKA machine learning tool. Here most of discussion was related to the actual Implementation of the system at ISM-APAC acceptance servers.

### 4.2 Implemented Environment

The development machine should have at least following hardware configuration and software installed to implement the system.

The proposed system is implemented using Ubuntu 14.0 server, PHP 5.5, MySQL 5.6 and PHPStorm6.0 IDE. The E-commerce application developed using Magento Enterprise version 12.0 which is shopping cart system based on model view controller (MVC) framework. It is a very rich tool that has an intuitive admin tool and control over the entire store. All experiments are performed on a computer system with a CPU clock rate of 2 GHz and 4 GB of main memory.

WEKA machine learning tool run on same server with java version "1.7.0\_79" and OpenJDK Runtime Environment (IcedTea 2.5.5) (7u79-2.5.5-0ubuntu0.14.04.2)

### 4.3 Development Tools

#### 4.3.1 PHPStorm 6.0

PHPStorm is a commercial, cross-platform IDE for PHP [1] built on JetBrains' IntelliJ IDEA platform. PhpStorm provides an editor for PHP, HTML and JavaScript with on-the-fly code analysis, error prevention and automated refactoring for PHP and JavaScript code. PhpStorm's code completion supports PHP 5.3, 5.4, 5.5 & 5.6[2] (modern and legacy projects), including generators, the finally keyword, list in for each, namespaces,

closures, traits and short array syntax. It includes a full-fledged SQL editor with editable query results.

## 4.3.2 DbVisualizer 7.0

DbVisualizer is the universal database tool for developers, DBAs and analysts. It is the perfect solution since the same tool can be used on all major operating systems accessing a wide range of databases.

Figure 3.6 Shows front end of the system that offers different electronic products divided as per different categories. The proposed approach has many modules. The first and the most important step are data acquisition and pre-processing the clickstream data. Figure 4.1 Shows unprocessed log data. Upon application of different pre-processing steps a cleaned log file is obtained. Figure 4.2 shows the formatted ARFF Data file support for WEKA Tool. Relevant data formatted files available in Appendix D.

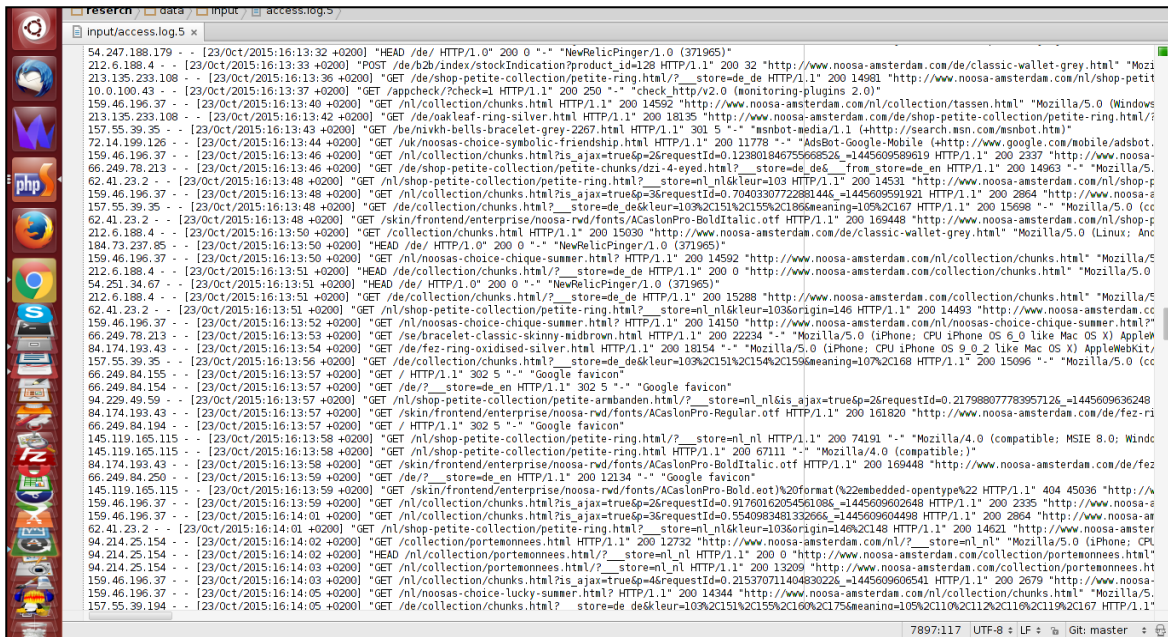


Figure 4. 1: Unprocessed Data File



## 4.4 Major Code Structure

Developed integration system mainly contains Data Processing and Data Integration Module. (Shown in Figure 4.3)

- 29

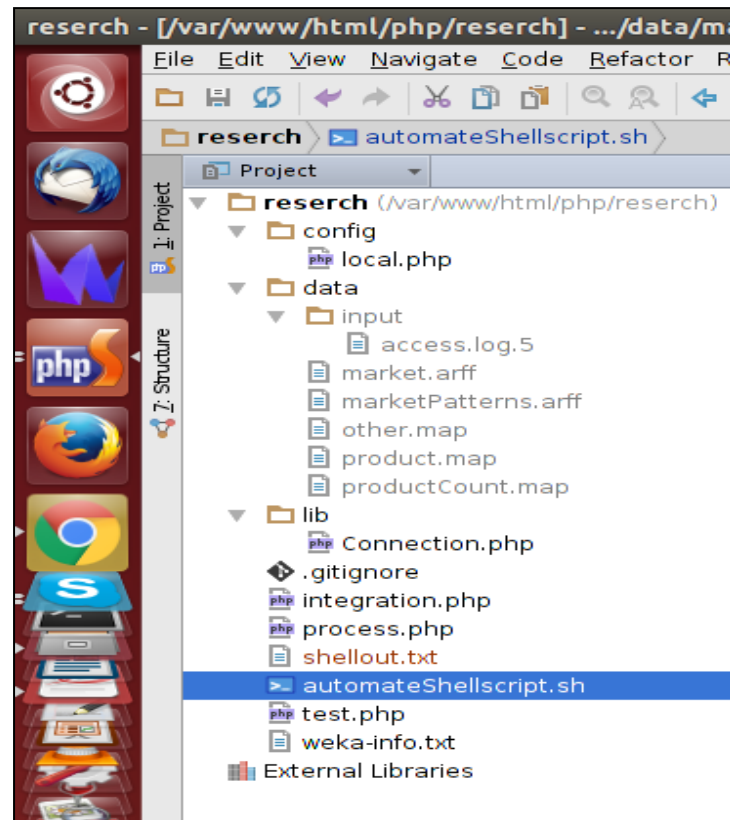


Figure 4. 3: Code Structure

## 4.5 Reuse Existing Code

### ***Database Connection Code***

Database Connection code is shown in Figure 4.4.

```
/**
 * Database connection
 */
class Connection{

    private static $instance = NULL;

    /**
     * Get DB instance
     */
    public static function getInstance() {
        if (!isset(self::$instance)) {
            try{
                $pdo_options[PDO::ATTR_ERRMODE] = PDO::ERRMODE_EXCEPTION;
                self::$instance = new PDO('mysql:host='.DB_HOST.';dbname='.DB_NAME,
                DB_USER, DB_PASS, $pdo_options);
            } catch (PDOException $e) {
                die('Database connection failed: ' . $e->getMessage());
            }
        }
        return self::$instance;
    }
}
```

```

    } catch (Exception $e){
        print($e);
    }
}
return self::$instance;
}
}

```

Figure 4. 4: Database Connection Code

### ***Data Pre- Processing Code***

Data cleaning is a process of filtering out irrelevant and outliers' data. For an example Filename suffixes such as gif, jpeg, GIF, JPEG, and JPG are removed and Automatically generated by Web client browsers, requests generated by Web bots are removed (e.g. Web crawlers – Googlebot, bingbot, NewRelicPinger, nagios plugins, monitoring plugins, msnbot, crawl, slurp, spider, dotbot).

Data Pre-Processing Code is shown in Figure 4.5.

```

if (preg_match( '#\b(Googlebot|bingbot|NewRelicPinger|nagios-plugins|monitoring-
plugins|msnbot|crawl|slurp|spider|dotbot)\b#',
    $data[9] ) == false )
{
    if (strpos($data[9], 'Gecko/') == false) {

        if (preg_match('#\b(v2_soap|.otf)\b#', $data[5]) == false) {

            if (preg_match('#\b(302|301)\b#', $data[6]) == false) {

                if (strpos($data[5], '.html') == true) {

                    if ($productUrl = $this->isProductLog($data[5])) {

                        if (isset($mapProductId[$productUrl])) {
                            $data[5] = $mapProductId[$productUrl];
                        } elseif(isset($mapNotProductUrl[$productUrl])) {
                            continue;
                        } else{
                            $post = null;
                            $queryInput = "select product_id from core_url_rewrite where
request_path like '% " . $productUrl . "%' and options IS NULL limit 1";

                            $queryPrep = $dbConnection->query($queryInput);

```



## Products Recommendation for E-commerce Application via Web Server Logs

```
export CLASSPATH=/home/ism-apac/kavindu/mit/sem3/MIT3104-DataM/weka-3-6-12/weka.jar:.
#echo $CLASSPATH

php=`which php`
${php} process.php

java weka.associations.Apriori -N 30 -T 0 -C 0.7 -D 0.05 -U 1.0 -M 0.02 -S -1.0 -c -1 -I -t
/var/www/html/php/reserch/data/market.arff >
/var/www/html/php/reserch/data/marketPatterns.arff

${php} integration.php
```

Figure 4. 6: Automate Shellscrip

## CHAPTER 5 - EVALUATION

This chapter provides evidence about testing. A proper test plan is explained in order to verify and validate test cases.

### 5.1 Introduction

The system testing was carried out according to an organized manner where each module was tested individually in the initial stages. The complete system was then test to validate weather satisfy the expected. All the test cases were set up at the Production servers. After testing the initial prototype of each component, the prototype was evaluated and sometimes modified until satisfies the requirements.

### 5.2 Test Plan

It was decide test each prototype of the system, soon after developing expected components. The project schedule was prepared by allocating time slots to these testing phases. Also it was decide to get the client's involvement in the testing phase.

Unit testing was carried out to ensure that each unit operates properly. First they were tested individually without other system components. Then the Sub-system was tested using a collection of components and interactions between these modules were closely examined. Therefore this stage was allocated with more time period comparing to the others.

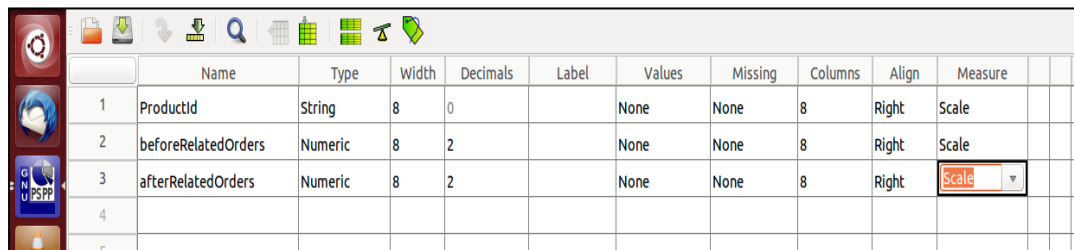
These test phases validated "weather we are building the right product" and also helped verify weather "we are building the product right". Sub-system was integrated and entire system was tested at next stage to ensure that system is working correctly with all the sub systems. This was the milestone where it validates the system's functional and non-functional requirements. As each functional and non-functional requirement was tested

individually at this phase. An Alpha testing phase was planted in the testing process to verify the operational standard. This helped to reveal the weak points in the system.

### 5.3 Evaluation

System generated product recommendations evaluate with already used product recommendations using PSPP dependent sample t-Test. GNU PSPP is a program for statistical analysis of sampled data. It is a Free replacement for the proprietary program SPSS.

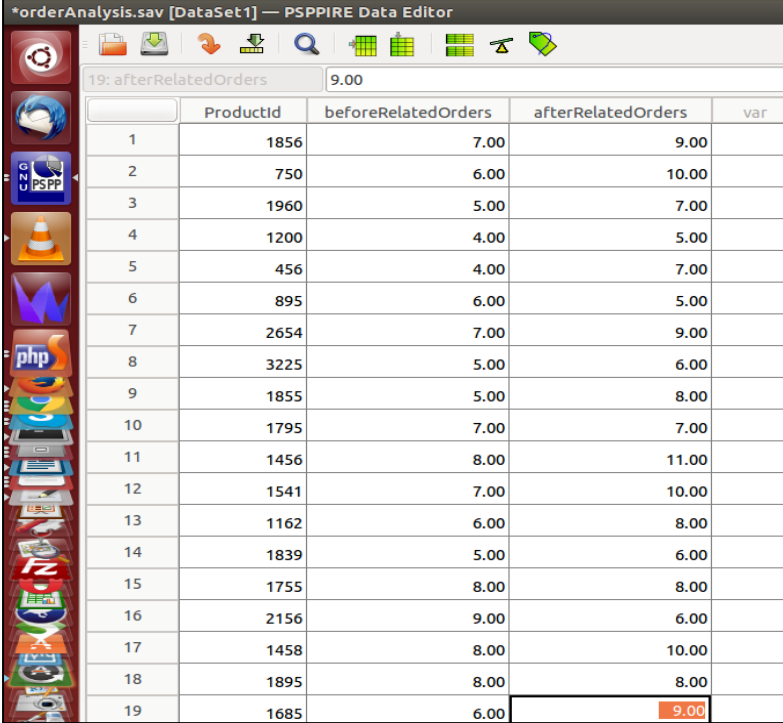
For implement dependent sample t-Test use variables as productid, total order existing recommended product contain and total order system generated recommended products contains. Variable view shown in Figure 5.1



	Name	Type	Width	Decimals	Label	Values	Missing	Columns	Align	Measure
1	Productid	String	8	0		None	None	8	Right	Scale
2	beforeRelatedOrders	Numeric	8	2		None	None	8	Right	Scale
3	afterRelatedOrders	Numeric	8	2		None	None	8	Right	Scale
4										

Figure 5. 1: Variable View

Main products and related products contain order information queried from order table on magento E-commerce application. Data view of PSPP Shown Figure 5.2



	ProductId	beforeRelatedOrders	afterRelatedOrders	var
1	1856	7.00	9.00	
2	750	6.00	10.00	
3	1960	5.00	7.00	
4	1200	4.00	5.00	
5	456	4.00	7.00	
6	895	6.00	5.00	
7	2654	7.00	9.00	
8	3225	5.00	6.00	
9	1855	5.00	8.00	
10	1795	7.00	7.00	
11	1456	8.00	11.00	
12	1541	7.00	10.00	
13	1162	6.00	8.00	
14	1839	5.00	6.00	
15	1755	8.00	8.00	
16	2156	9.00	6.00	
17	1458	8.00	10.00	
18	1895	8.00	8.00	
19	1685	6.00	9.00	

Figure 5. 2: Data View

For analysis generated data used Paired-Sample t-Test. Use variable one as after related orders and variable two as beforerelatedorders with 95% confidence intervals. Generated analytical results showing on Figure 5.3.

Paired sample statistic table interpret mean, standard deviation and standard error mean. Mean of afterrelatedorders is 7.84. Mean for beforerelatedorders is 6.37. When comparing before and after related order means system generated recommendations has more orders. For paired sample test table represent t-score, degree of freedom and significant-value equal to 3.75 with 9 degree of freedom. It is give significant level as 0.006, because of 0.006 less than 0.05 we can rejects the nullhypothesies. It shows system generated recommendations make a difference of total number of orders considering generated information we can prove system generated recommendations are more accurate than previously used recommendation.



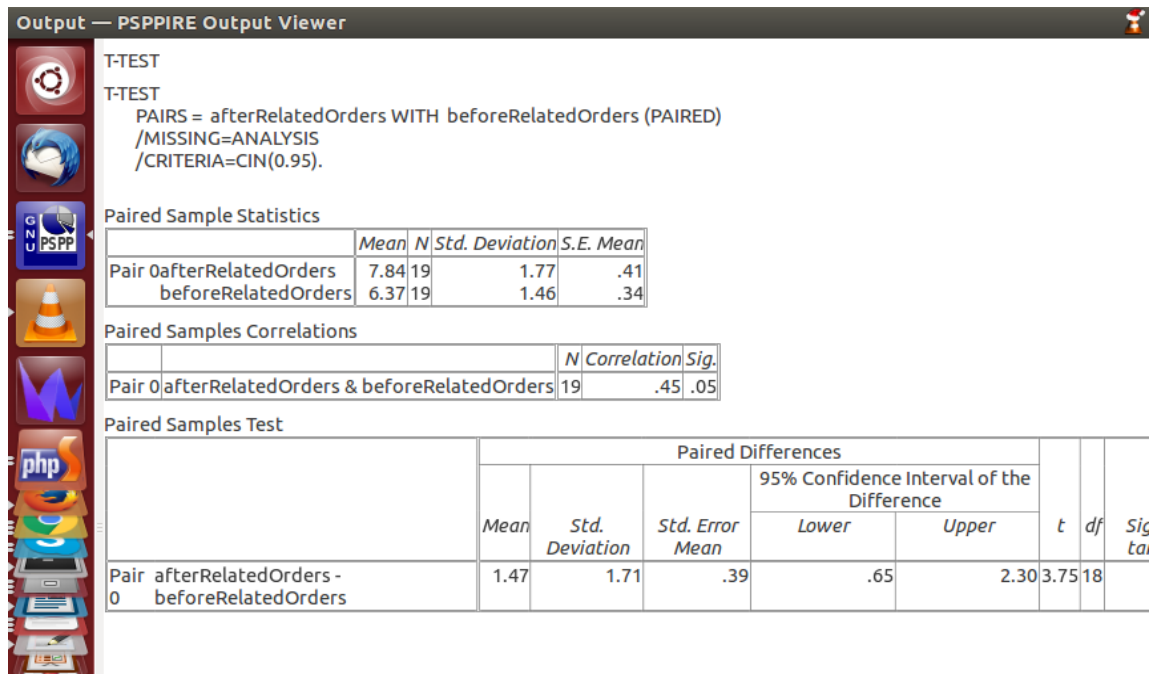


Figure 5. 3: Generated Analyses Report

## 5.4 Project Assessment

The objective at the start of the project was to provide a system to identify products recommendation and automatically integrate them to E-Commerce application. The major objectives were successfully satisfied at the end of the Project. While successfully archiving all the functional and non-functional requirements of the project, it was able to meet the time constrains as well. The project works carried out referring the project time schedule prepared earlier and helped to archive objectives in timely manner.

Overall the project was able to achieve all the goals and objectives that were set up at the beginning. Current system was in good working condition. It can handle all product recommendation for E-Commerce application.

## CHAPTER 6 - CONCLUSION

### 6.1 Major Achievement

Throughout the course of this research lot of useful, valuable and interesting goals were achieved. Especially all the initial project goals were achieved. Few of other major achievements can be listed as access log processing module developed using PHP and integration module develop for integrate generated recommendations with Magento E-Commerce application. With minimum adoption, integration and processing modules can be used with any existing system.

Created automated shell script is another major achievement. Shell script scheduled to execute on configured time intervals. It will automatically process latest access log and integrated formatted data with WEKA machine learning tool. Then shell script will process associations rule generated from Machine learning tool and integrate identified recommendations with Magento E-Commerce application.

Another major achievement is evaluating system generated recommendation with GNU PSPP statistical analysis tool. It proved system generated recommendations are more accurate and relevant that previously used recommendations.

### 6.2 Support from MIT

The project was carried out as partial fulfilment of the requirement of the MIT program. The Knowledge obtained about various aspects of Information Technology during the MIT program provides a great assistance in the research of the project.

There was important configuration had to be done to develop product recommendation, such as develop data pre-processing module and develop integration module. The knowledge obtains from learning object oriented programming; programme designing, rapid application development and data mining help to do this task.

Web Application Development knowledge learnt in the 2nd and 3rd semester helped a great deal in working with PHP web server. System and Network administration knowledge learnt in 2nd semester help to develop Ubuntu base server side. The lessons learnt in Project Management were used thought out the whole project. Also the opportunity provided by the UCSC to complete a project prior to completions of the Master Degree gave the opportunity to engage in an actual System Development project and it helped to learn about professional issues in real world environment.

### 6.3 Lesson Learnt

During the course of this project lot of new concepts and lessons was learnt.

- The main lesson learnt was software project management. All the phases in typical software project, beginning from project proposal to final system implementation were comprehensively covered
- Also during the project learnt how clients and getting their views and ideas in project planning interact.
- Project includes integration WEKA Tool into E- Commerce application. And that give the hand on experience in working with practical system integration.
- Gained comprehensive understanding about the Linux platform and many other related open source technologies.
- Gained knowledge about data mining concepts.
- Learned how the system was designed and managed within the given constraints like time schedule, human, technical and monetary resources etc.

### 6.4 Future works

The system was fully operational and all its basic functionalities have been configured successfully. Incorporating more control such as real time personalise recommendation generation will be considering as next phase after this project.

## REFERENCES

- [1] Y. Cho, and J. Kim, "Application of Web usage mining and product taxonomy to collaborative recommendations in e-commerce", *Expert systems with Applications*, vol. 26, no. 2, pp. 233-246, February 2004.
- [2] Y. Cho, J. Kim, and S. Kim, "A personalized recommender system based on web usage mining and decision tree induction", *Expert Systems with Applications*, vol. 23, no. 3, pp. 329-342, October 2002.
- [3] Z. Huang, D. Zeng, and H. Chen, "A comparative study of recommendation algorithms in e-commerce applications", *IEEE Intelligent Systems*, vol. 22, pp. 68-78, 2007.
- [4] B. Sarwar, G. Karypis, J. Konstan, and J. Riedl, "Analysis of recommendation algorithms for e-commerce", *Proceedings of the 2nd ACM conference on Electronic commerce*. ACM, pp.158-167, 2000.
- [5] M. Claypool, A. Gokhale, T. Miranda, P. Murnikov, D. Netes, and M. Sartin, "Combining content-based and collaborative filters in an online newspaper", *Proceedings of ACM SIGIR workshop on recommender systems*, vol. 60, 1999.
- [6] B Mobasher, R Cooley, J Srivastava, "Automatic personalization based on Web usage mining", *Communications of the ACM*, vol.43 no. 8 pp. 142-151, 2000. Add extra literature review start
- [7] YM Huang, YH Kuo, JN Chen, and YL Jeng, "NP-miner: A real-time recommendation algorithm by using web usage mining", *Knowledge-Based Systems*, vol.19, no.4, pp. 272-286, 2006.

- [8] C.R. Varnagar, N.N. Madhak, T. M. Kodinariya, and J. N. Rathod, "Web usage mining: A review on process, methods and techniques", Information Communication and Embedded Systems (ICICES), International Conference on. IEEE, pp. 40-46, 2013.
- [9] P. Nithya, and P. Sumathi, "Novel pre-processing technique for web log mining by removing global noise and web robots." In Computing and Communication Systems (NCCCS) IEEE, pp. 1-5,2012.
- [10] J. Borges, and M. Levene, "Data mining of user navigation patterns", Web usage analysis and user profiling, Springer Berlin Heidelberg ,pp.92-112, 2000.
- [11] R. Cooley, B. Mobasher, J. Srivastava, "Data preparation for mining world wide web browsing patterns", Knowledge and information systems, vol.1, pp. 5-32, 1999.
- [12] W.Bin and L. Zhijing, "Web Mining Research", Fifth International Conference on Computational Intelligence and Multimedia Applications, pp. 84 – 89, 2003.
- [13] M. Khosravi and M.J. Tarokh , "Dynamic Mining of Users Interest Navigation Patterns Using Naive Bayesian Method", Intelligent Computer Communication and Processing (ICCP), IEEE, (pp. 119-122, 2010.
- [14] B. Devi, Y. Devi, B. Rani and R. Rao, "Design and Implementation of Web Usage Mining Intelligent System in the Field of e-commerce." International Conference on Communication Technology and System Design Procedia Engineering, vol. 30 , pp.20-27, 2012.
- [15] R. Agrawal & R. Srikant, 1994. Fast Algorithms for Mining Association Rules in Large Databases.Proceedings of the 20th International Conference on Very Large Databases, pages 487-499.

- [16] A.V.Bodaptati, "Recommendation systems with purchase data, Journal of Marketing Research", 45 (1), pp.77-93,2008.
- [17] A.Zenebe, A.F.Norcio, "Representation, similarity measures and aggregation methods using fuzzy sets for content-based recommender systems", Fuzzy Sets and Systems, 160 (1), pp. 76-94, 2009.
- [18] J.L. Herlocker, J.A. Konstan, J. Loren, G. Terveen, T.Riedl, "Collaborative filtering recommender systems",ACM Transactions on Information Systems, 22 (1), pp. 5-53, 2004.
- [19] H.W. Ye, "A Personalized collaborative filtering recommendation using association rules mining and selforganizing map", Journal of Software, 6(4), pp.732-739,2011.
- [20] Y.M. Zhang, S.Y. Jiang, "A Splitting criteria based on similarity in decision tree learning", Journal of Software, 7(8), pp.1775-1782, 2012
- [21] T.P. Liang, Y.F. Yang, D.N. Chen, Y.C. Ku, "A semanticexpansion approach to personalized knowledge recommendation", Decision Support Systems, 45 (3), pp.401-412, 2008.
- [22] Zh. Xiong, P. Chen, Y. Zhuang, "Improvement of ECLAT algorithm for association rules based on hash boolean matrix," Application Research of Computers, vol.4, pp.1323-1325, 2010.
- [23] P. Nithya, and P. Sumathi, "Novel pre-processing technique for web log mining by removing global noise and web robots." In Computing and Communication Systems (NCCCS) IEEE, pp. 1-5,2012.

# APPENDIX A – MAGENTO

## INSTALLATION

Magento is the most popular content management system for e-commerce websites. It is used by small businesses and large enterprise companies alike, and can be enhanced by thousands of extensions and themes. Magento uses the MySQL database system, the PHP programming language, and parts of the Zend Framework.

In here describe how to install Magento with an Apache web server on Ubuntu 14.04, including some common performance and security configurations along the way.

### Step 1 — Configure Apache and PHP

Before we download and install Magento, Apache and PHP need to be configured to properly handle Magento's traffic and computing tasks, and some additional modules will need to be installed and activated.

#### Apache Virtual Host

First, we need to configure a virtual host file so that Apache knows how to manage our Magento site correctly. We can use our text editor to create a new virtual host file in `/etc/apache2/sites-available/`. In this example, the virtual host is called `magento`, but you can name it something else if you prefer.

```
sudo nano /etc/apache2/sites-available/magento.conf
```

Magento's virtual host needs are fairly simple. Begin with a `<VirtualHost>` tag that listens for all traffic from port 80, the standard HTTP port. Then add a line telling Apache where to find your site's files with the `DocumentRoot` directive:

```
<VirtualHost *:80>
```

```
    DocumentRoot /var/www/html
```

```
</VirtualHost>
```

We need to add some additional parameters for Magento to work properly. Inside of a `<Directory>` tag pointing to our document root, we'll enter the options for Apache's directory indexing, symlink support, and multilingual support. We'll also add a line that allows `.htaccess` files to override Apache settings, which provides more fine-grained control of individual folders.

```
...
```

```
<Directory /var/www/html/>
    Options Indexes FollowSymLinks MultiViews
    AllowOverride All
</Directory>
```

```
...
```

With all of these components in place, your virtual host file will look like this:

```
<VirtualHost *:80>
    DocumentRoot /var/www/html
    <Directory /var/www/html/>
        Options Indexes FollowSymLinks MultiViews
        AllowOverride All
    </Directory>
</VirtualHost>
```

When you're finished writing up the server block, save and close the file. To enable the new site with Apache, use the `a2ensite` command:

```
sudo a2ensite magento.conf
```

We also want to disable the default virtual host that came with Apache, as it will conflict with our new virtual host. To disable a site with Apache, use the `a2dissite` command:

```
sudo a2dissite 000-default.conf
```

PHP Settings



Next, we need to change how much memory Apache grants to PHP processes. Magento uses PHP for nearly everything it does, and needs a decent amount of memory for complex operations like indexing products and categories. By default, PHP allocates a maximum of 128MB of memory to each script running on Apache. We should bump that limit up to a reasonable amount to ensure that none of Magento's scripts run out of memory, which would cause the script to crash.

Open Apache's PHP configuration file with your text editor and root privileges:

```
sudo nano /etc/php5/apache2/php.ini
```

Find the following line, which declares the memory limit per PHP script:

```
memory_limit = 128M
```

Change that line so that the limit is raised to 512MB. Your store's memory needs may be higher depending on the number of products in your catalog and the number of visitors you receive daily. Some larger stores need to set their memory limit to 2GB or more, but 512MB should be adequate for now.

```
memory_limit = 512M
```

Note: Be sure to use "M" at the end of the memory number.

When you are finished making this change, save and close the file. The next time that you restart Apache, the memory limit change will take effect.

Magento needs a couple of PHP modules in addition to the ones that come with PHP. We can get these directly from Ubuntu's default repositories after we update our local package index:

```
sudo apt-get update
```

```
sudo apt-get install libcurl3 php5-curl php5-gd php5-mcrypt
```

These extensions will allow Magento to properly handle HTTP requests, image thumbnails, and data encryption. Now that we have all of the packages that we need, we can enable URL rewriting support for Apache and encryption support for PHP:

```
sudo a2enmod rewrite
```

```
sudo php5enmod mcrypt
```

Once all of these configuration and extension changes have been made, it's time to restart the Apache server instance so that the changes are applied:

```
sudo service apache2 restart
```

### **Step 2 — Create a MySQL Database and User**

Magento uses a MySQL database to manage site data, like product and order information. We have MySQL installed and configured, but we need to make a database and a user for Magento to work with.

Begin by logging into the MySQL root account:

```
mysql -u root -p
```

You will be prompted for MySQL's root account password, which you set when you installed MySQL. Once the correct password has been submitted, you will be given a MySQL command prompt.

First, we'll create a database that Magento can write data to. In this example, the database will be called magento, but you can name it whatever you prefer.

```
CREATE DATABASE magento;
```

Note: Every MySQL statement must end in a semi-colon (;), so check to make sure that you included that if you are running into any issues.

Next, we are going to create a new MySQL user account that will be used exclusively to operate on the new database. Creating one-function databases and accounts is a good idea, as it allows for better control of permissions and other security needs.

I am going to call the new account `magento_user` and will assign it a password of `password`. You should definitely use a different username and password, as these examples are not very secure.

```
CREATE USER magento_user@localhost IDENTIFIED BY 'password';
```

At this point, you have a database and a user account that are each specifically made for Magento. However, the user has no access rights to the database. We need to link the two components together by granting our user access privileges to the database:

```
GRANT ALL PRIVILEGES ON magento.* TO magento_user@localhost IDENTIFIED BY 'password';
```

Now that the user has access to the database, we need to flush the privileges so that MySQL knows about the recent privilege changes that we've made. Once that is done, we can exit out of the MySQL command prompt.

```
FLUSH PRIVILEGES;
```

```
exit
```

You should now be back to your regular SSH command prompt.

### **Step 3 — Download and Set Up Magento Files**

We are now ready to download and install Magento. To see what the latest stable version of the Magento Community Edition is, head over to the community download page. In this example, the current release number was 1.9.0.1, but you should substitute that number for the latest release available to you. It is always recommended to use the latest version of Magento, as new releases often include important security updates in addition to new and improved features.

Use `wget` to download the Magento file archive to your home directory:

```
cd ~
```

```
wget http://www.magentocommerce.com/downloads/assets/1.9.0.1/magento-1.9.0.1.tar.gz
```

We can extract the archived files to rebuild the Magento directory with `tar`:

```
tar xzvf magento-1.9.0.1.tar.gz
```

You will now have a directory called `magento` in your home directory. We'll need to move the unpacked files to Apache's document root, where it can be served to visitors of our website. We will use `rsync` to transfer our Magento files there, since `rsync` will include important hidden files like `.htaccess`. Once the transfer is complete, we can clean up our home directory by deleting the `magento` folder and archive there.

```
sudo rsync -avP ~/magento/ /var/www/html/
```

```
rm -rf ~/magento*
```

`rsync` will safely copy all of the contents from the directory that you unpacked to the document root at `/var/www/html/`. Now we need to assign ownership of the files and folders to Apache's user and group:

```
sudo chown -R www-data:www-data /var/www/html/
```

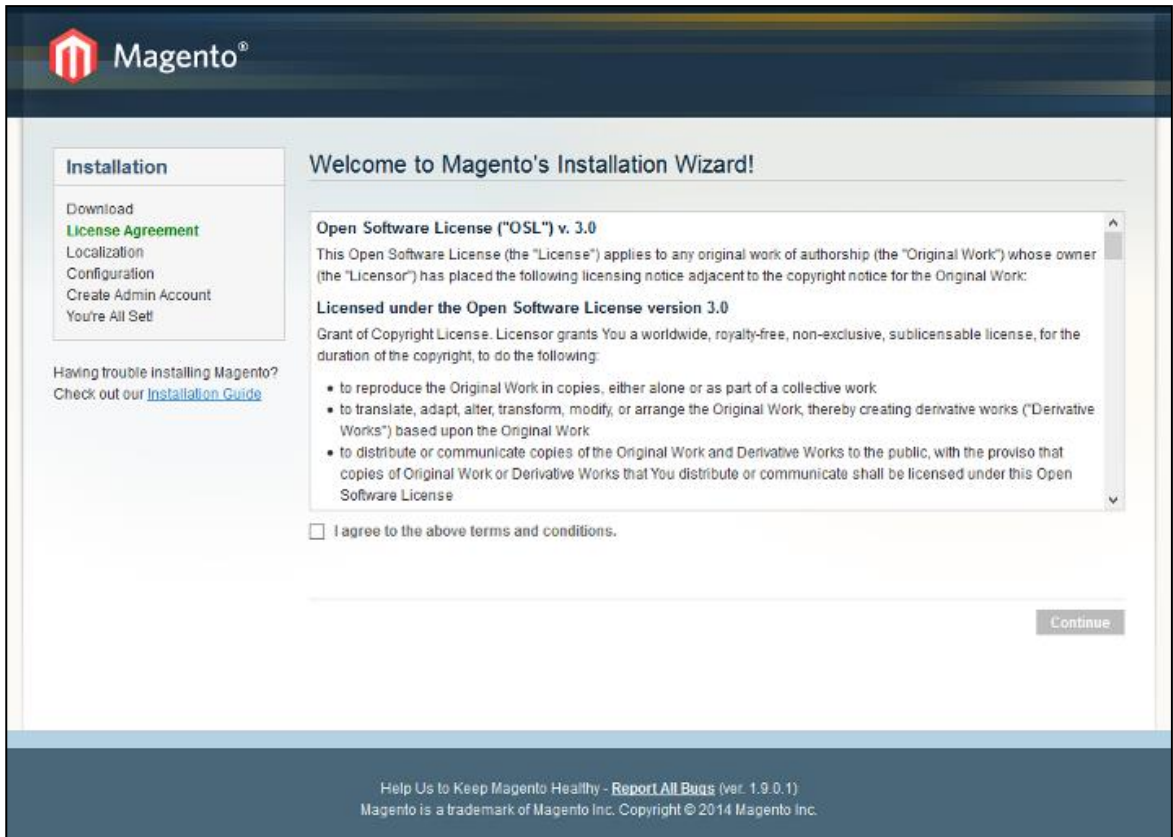
This change will allow Apache to create and modify Magento files. We are now ready to complete the installation process via Magento's browser-based configuration interface.

### **Step 4 — Completing Installation Through the Web Interface**

To access the web interface with your browser, navigate to your server's domain name or public IP address:

```
http://server_domain_name_or_IP/
```

If the previous steps have been followed correctly, you will be presented with Magento's installation wizard. The first page will display the license agreement, which you will need to agree to before you can hit Continue.

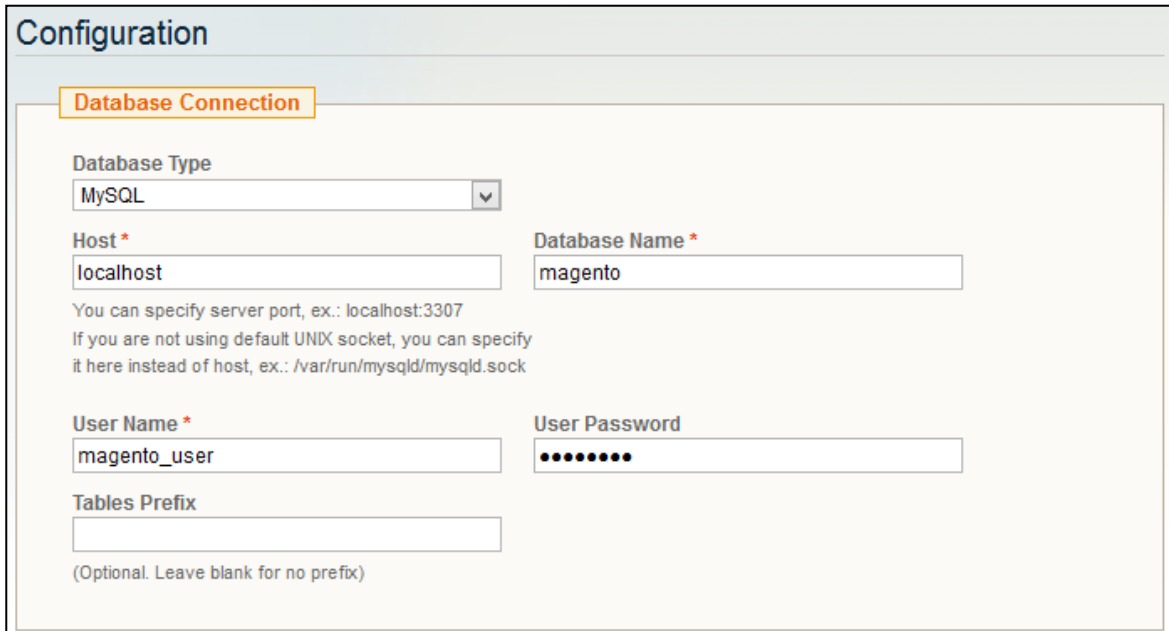


The next page is where you can change your locale settings, like language, time zone, and currency. Change these to the settings most appropriate to where your products are being sold, then hit Continue.

The screenshot shows the "Localization" page of the Magento 1.9.0.1 installation wizard. The page has a title "Localization" and a sub-section "Locale settings". Below this, there are three required fields: "Locale \*" with a dropdown menu showing "English (United States) / English (United S..."; "Time Zone \*" with a dropdown menu showing "US Eastern Standard Time (Etc/GMT+5)"; and "Default Currency \*" with a dropdown menu showing "US Dollar". At the bottom right of the page, there is a red asterisk icon followed by the text "\* Required Fields" and an orange "Continue" button.

The Configuration page is broken down into three sections. The first section is where you can set up how Magento connects to its MySQL database. Make sure that Database Type

is set to "MySQL" and Host is set to "localhost". You'll then need to fill in Database Name, User Name, and User Password with the settings that you decided on when creating the MySQL database and user account earlier.



The screenshot shows the 'Configuration' page with the 'Database Connection' tab selected. The form contains the following fields and values:

- Database Type:** A dropdown menu with 'MySQL' selected.
- Host \*:** A text input field containing 'localhost'.
- Database Name \*:** A text input field containing 'magento'.
- User Name \*:** A text input field containing 'magento\_user'.
- User Password:** A password input field with 10 dots.
- Tables Prefix:** An empty text input field.

Below the 'Host \*' field, there is explanatory text: 'You can specify server port, ex.: localhost:3307' and 'If you are not using default UNIX socket, you can specify it here instead of host, ex.: /var/run/mysqld/mysqld.sock'. Below the 'Tables Prefix' field, there is a note: '(Optional. Leave blank for no prefix)'.

The next section on the Configuration page is where you can configure your store's URL, along with a couple of other Apache-controlled functions. Make sure that Base URL matches your server's domain name; if you don't have a domain name set up yet, you can use your server's public IP address for now. It's a good idea to change the Admin Path to something less obvious than "admin" to make it more difficult for someone to find your admin panel. You should also consider checking Use Web Server (Apache) Rewrites to make your site's URLs more friendly to users and search engines. The Apache module needed to support this functionality, `mod_rewrite`, has already been enabled and is ready for use.

**Web access options**

**Base URL \***

**Admin Path \***

Additional path added after Base URL to access your Administrative Panel (e.g. admin, backend, control etc.).

☒ **Enable Charts**  
Enable this option if you want the charts to be displayed on Dashboard.

☐ **Skip Base URL Validation Before the Next Step**  
Check this box only if it is not possible to automatically validate the Base URL.

☒ **Use Web Server (Apache) Rewrites**  
You could enable this option to use web server rewrites functionality for improved search engines optimization.  
Please make sure that `mod_rewrite` is enabled in Apache configuration.

☐ **Use Secure URLs (SSL)**  
Enable this option only if you have SSL available.

The last part of the Configuration page is for selecting the method of session data storage. Magento ships with two methods of saving user session data. The File System method stores sessions in files on the server and is the simplest method to start with. The Database method stores sessions as entries in the MySQL database and is ideal for Magento installations that span across multiple servers. For now, we can stick with the File System method, since it will generally perform better out of the box.

**Session Storage Options**

**Save Session Data In**

\* Required Fields

**Continue**

After you select your configuration options and hit Continue, you'll move on to the admin account creation page. This is where you will create the administrative account that is in charge of maintaining the Magento store. Make sure that the username and password are both secure and difficult to guess. The Encryption Key field should be left blank unless you are migrating data over from an existing Magento installation. If you leave the field

blank, Magento will generate a new encryption key when you click Continue and will display it for you on the next page. Make sure that you save that encryption key somewhere safe in case you need it for migration purposes later.

### Create Admin Account

**Personal Information**

First Name \*  
Example

Last Name \*  
User

Email \*  
webmaster@example.com

**Login Information**

Username \*  
magento\_admin

Password \*  
.....

Confirm Password \*  
.....

**Encryption Key**

Magento uses this key to encrypt passwords, credit cards and more. If this field is left empty the system will create an encryption key for you and will display it on the next page.

\* Required Fields

Continue

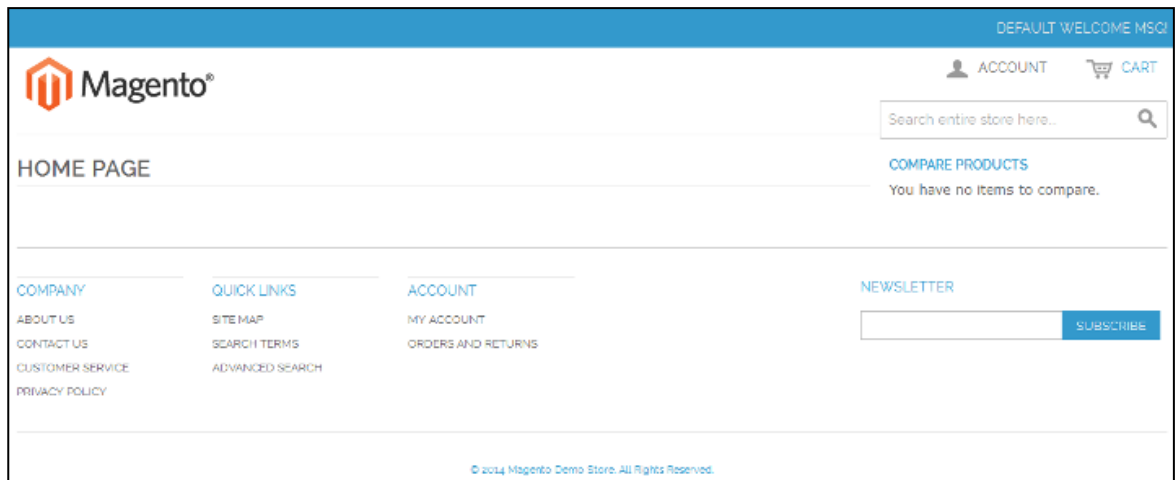
Once you have completed the web-based configuration, your Magento installation will be ready to use. Check out your new storefront by visiting your server's domain name or public IP address in your web browser:

`http://server_domain_name_or_IP/`

If everything was configured correctly, you should be presented with a storefront that looks something like this:



## Products Recommendation for E-commerce Application via Web Server Logs



## APPENDIX B – WEKA USER GUIDE

### Step 1: Install Weka

Go to the Weka website, <http://www.cs.waikato.ac.nz/ml/weka/>, and download the software. On the left hand side, click on the link that says download. Select the appropriate link corresponding to the version of the software based on your operating system and whether or not you already have Java VM running on your machine (if you don't know what Java VM is, then you probably don't).

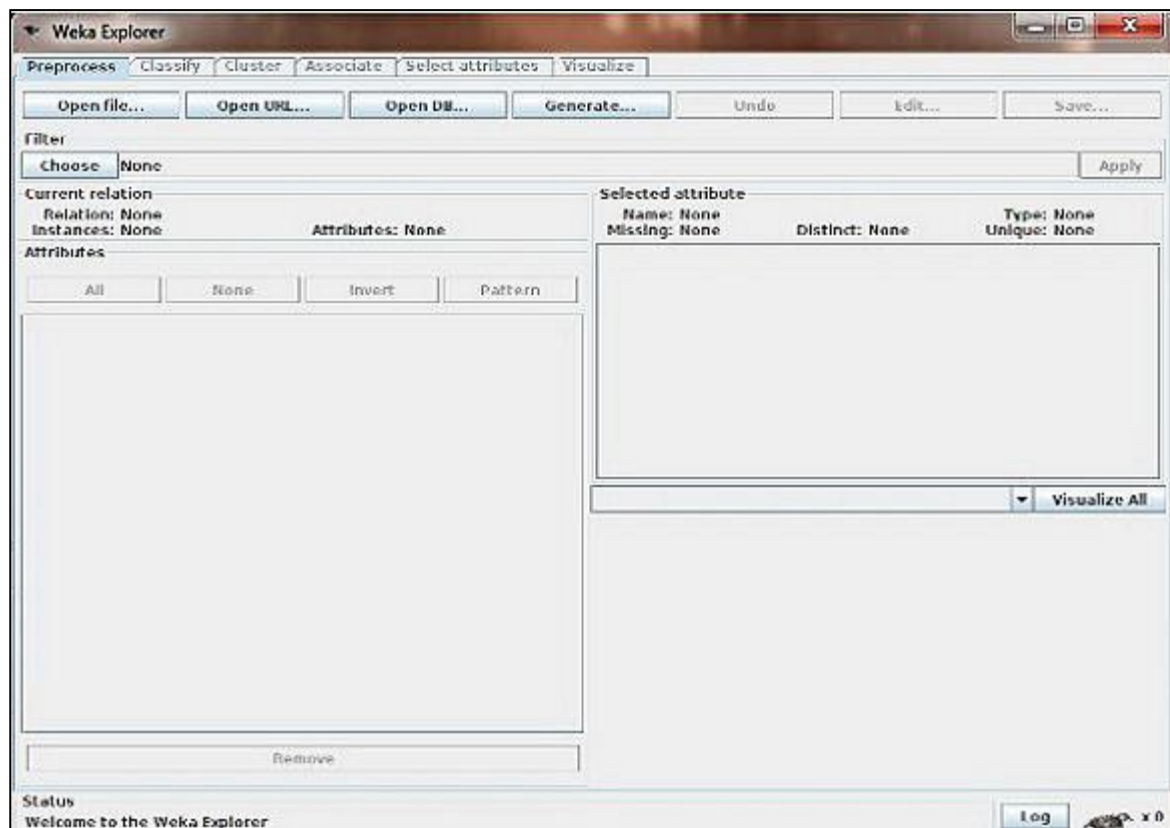
The link will forward you to a site where you can download the software from a mirror site. Save the self-extracting executable to disk and then double click on it to install Weka. Answer yes or next to the questions during the installation. Click yes to accept the Java agreement if necessary. After you install the program Weka should appear on your start menu under Programs (if you are using Windows).

### Step 2: Running Weka

From the start menu select Programs, then Weka, then Weka 3\*.

You will see the Weka GUI Chooser. Select Explorer. The Weka Explorer will then launch.





## APPENDIX C – ROW DATA FILES

This Section shows the Row Data Files.

Web application access log use Format called CLF (Common Log Format). Example of clickstream in details shown in below.

```
127.0.0.1 User identifier frank [10/Oct/2000:13:55:36 -0700] "GET /apache_pb.gif
HTTP/1.0" 200 2326
```

A "-" in a field indicates missing data.

127.0.0.1 is the IP address of the client (remote host) which made the request to the server.

User identifier is the RFC 1413 identity of the client.

Frank is the userid of the person requesting the document.

[10/Oct/2000:13:55:36 -0700] is the date, time, and time zone when the server finished processing the request, by default in strftime format %d/%b/%Y:%H:%M:%S %z.

"GET /apache\_pb.gif HTTP/1.0" is the request line from the client. The method GET, /apache\_pb.gif the resource requested, and HTTP/1.0 the HTTP protocol.

200 is the HTTP status code returned to the client. 2xx is a successful response, 3xx a redirection, 4xx a client error, and 5xx a server error.

2326 is the size of the object returned to the client, measured in bytes.

```
157.55.39.35 - - [23/Oct/2015:08:11:53 +0200] "GET
/nl/collection/chunks.html?kleur=81%2C154%2C156%2C159%2C160%
2C186 HTTP/1.1" 200 16014 "-" "Mozilla/5.0 (compatible;
bingbot/2.0; +http://www.bing.com/bingbot.htm) "
```

```
157.55.39.29 - - [23/Oct/2015:08:11:58 +0200] "GET /se/grey-
nivkh-small-flap-wallet-amulet-2321.html HTTP/1.1" 301 5 "-"
"Mozilla/5.0 (compatible; bingbot/2.0;
+http://www.bing.com/bingbot.htm) "
```

```
80.245.147.81 - - [23/Oct/2015:08:11:59 +0200] "GET
/de/collection/chunks.html?is_ajax=true&p=2&requestId=0.3918
```

## Products Recommendation for E-commerce Application via Web Server Logs

```

929030187428&_=1445580718875      HTTP/1.1"      200      2345
"http://www.noosa-amsterdam.com/de/collection/chunks.html"
"Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/43.0.2357.134 Safari/537.36"
80.245.147.81 - - [23/Oct/2015:08:12:04 +0200] "GET
/de/collection/armbänder.html HTTP/1.1" 200 15243
"http://www.noosa-amsterdam.com/de/collection/taschen.html"
"Mozilla/5.0 (Windows NT 6.1; WOW64) AppleWebKit/537.36
(KHTML, like Gecko) Chrome/43.0.2357.134 Safari/537.36"
54.251.34.67 - - [23/Oct/2015:08:12:07 +0200] "HEAD /de/
HTTP/1.0" 200 0 "-" "NewRelicPinger/1.0 (371965)"
66.249.79.132 - - [23/Oct/2015:08:12:08 +0200] "GET
/be/bracelet-petite-multi-light-grey.html HTTP/1.1" 404
11708 "-" "Mozilla/5.0 (iPhone; CPU iPhone OS 6_0 like Mac
OS X) AppleWebKit/536.26 (KHTML, like Gecko) Version/6.0
Mobile/10A5376e Safari/8536.25 (compatible; Googlebot/2.1;
+http://www.google.com/bot.html)"
66.249.79.132 - - [23/Oct/2015:08:12:09 +0200] "GET
/at/bracelet-navajo-white.html HTTP/1.1" 404 12298 "-"
"Mozilla/5.0 (compatible; Googlebot/2.1;
+http://www.google.com/bot.html)"
80.245.147.81 - - [23/Oct/2015:08:12:14 +0200] "GET
/de/wrap-bracelet-classic-skinny-light-grey.html HTTP/1.1"
200 18388 "http://www.noosa-
amsterdam.com/de/collection/armbänder.html" "Mozilla/5.0
(Windows NT 6.1; WOW64) AppleWebKit/537.36 (KHTML, like
Gecko) Chrome/43.0.2357.134 Safari/537.36"
50.31.164.139 - - [23/Oct/2015:08:12:19 +0200] "HEAD /de/
HTTP/1.0" 200 0 "-" "NewRelicPinger/1.0 (371965)"
80.245.147.81 - - [23/Oct/2015:08:12:21 +0200] "GET
/de/collection/chunks.html?is_ajax=true&p=3&requestId=0.6469
204709865153&_=1445580741344 HTTP/1.1" 200 2874

```

## APPENDIX D – ARFF DATA FILES

This Section shows the ARFF Data Formatted Files.

A dataset has to start with a declaration of its name:

@relation name

Followed by a list of all the attributes in the dataset (including the class attribute). These declarations have the form

@attribute attribute\_name specification

If an attribute is nominal, specification contains a list of the possible attribute values in curly brackets:

@attribute nominal\_attribute {first\_value, second\_value, third\_value}

If an attribute is numeric, specification is replaced by the keyword numeric: (Integer values are treated as real numbers in WEKA.)

@data

Data tag, which is followed by a list of all the instances. The instances are listed in comma-separated format, with a question mark representing a missing value.

```
@relation supermarket
@attribute '147' { t}
@attribute '151' { t}
@attribute '152' { t}
@attribute '154' { t}
@attribute '162' { t}
@attribute '2229' { t}
@attribute '2239' { t}
@attribute '2251' { t}
@attribute '2252' { t}
@attribute '2258' { t}
@attribute '2259' { t}
@attribute '2262' { t}
```

## Products Recommendation for E-commerce Application via Web Server Logs

```
@attribute '2287' { t}
@attribute '2288' { t}
.
.
@attribute '2334' { t}
@attribute '2335' { t}
@data
?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, t, t, ?, ?, ?, ?, ?, ?, t, ?, ?, ?, ?, ?, t,
t, ?, ?, ?, ?, ?, t, t, ?, ?, ?, ?, ?, ?, t, ?, t, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?,
?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?,
t, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, t, ?, ?, ?, t,
?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, t, t, ?, ?, ?, ?, ?, ?, ?, ?,
?, ?, ?, ?, ?, ?, ?, ?, t, t, ?, ?, ?, t, t, ?, ?, ?, ?, t, ?, ?, ?, t, t, ?, ?, ?, ?,
?, ?, ?, ?, t, ?, t, ?, ?, ?, ?, ?, ?, t, t, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?,
t, ?, ?, ?, ?, ?, ?, t, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, t, t, ?, t,
?, t, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, t, t, ?, ?, ?, t, ?, ?, t
?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?,
?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?,
?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?,
?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?,
?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?,
?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?,
?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?,
?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?
?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?,
?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, t, ?, ?, ?, ?,
?, ?, ?, ?, t, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?,
?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?,
?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, t, ?, ?, ?, ?,
?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?
?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?, ?
```

## APPENDIX E – ASSOCIATION RULES

Association rules extracted using APRIORI Algorithm with WEKA machine learning tool.

Apriori

=====

Minimum support: 0.02 (11 instances)

Minimum metric <confidence>: 0.7

Number of cycles performed: 20

Generated sets of large itemsets:

Size of set of large itemsets L(1): 20

Large Itemsets L(1):

193=t 11

1026=t 12

1515=t 13

1568=t 86

1613=t 13

1704=t 25

1755=t 109

1756=t 17

1757=t 60

1758=t 34

1759=t 21

1760=t 74

1839=t 49

1846=t 44

1853=t 98

1865=t 104



## Products Recommendation for E-commerce Application via Web Server Logs

2107=t 20

2108=t 18

2115=t 11

2289=t 14

Size of set of large itemsets  $L(2)$ : 29

Large Itemsets  $L(2)$ :

1568=t 1755=t 11

1568=t 1760=t 12

1568=t 1839=t 11

1568=t 1853=t 11

1568=t 1865=t 41

1755=t 1757=t 16

1755=t 1758=t 19

1755=t 1760=t 27

1755=t 1839=t 22

1755=t 1846=t 20

1755=t 1853=t 23

1755=t 1865=t 25

1757=t 1760=t 12

1757=t 1853=t 16

1757=t 1865=t 13

1758=t 1760=t 11

1758=t 1839=t 20

1758=t 1846=t 21

1758=t 1853=t 12

1758=t 1865=t 11

1760=t 1846=t 16

1760=t 1853=t 17

1760=t 1865=t 21

1839=t 1846=t 29

## Products Recommendation for E-commerce Application via Web Server Logs

1839=t 1853=t 16

1839=t 1865=t 18

1846=t 1853=t 16

1846=t 1865=t 16

1853=t 1865=t 17

Size of set of large itemsets  $L(3)$ : 7

Large Itemsets  $L(3)$ :

1755=t 1758=t 1839=t 15

1755=t 1758=t 1846=t 14

1755=t 1760=t 1853=t 11

1755=t 1839=t 1846=t 17

1758=t 1839=t 1846=t 18

1839=t 1846=t 1853=t 12

1839=t 1846=t 1865=t 11

Size of set of large itemsets  $L(4)$ : 1

Large Itemsets  $L(4)$ :

1755=t 1758=t 1839=t 1846=t 13

Best rules found:

1. 1755=t 1758=t 1846=t 14 ==> 1839=t 13      conf:(0.93)
2. 1758=t 1839=t 20 ==> 1846=t 18      conf:(0.9)
3. 1755=t 1758=t 1839=t 15 ==> 1846=t 13      conf:(0.87)
4. 1758=t 1846=t 21 ==> 1839=t 18      conf:(0.86)
5. 1755=t 1846=t 20 ==> 1839=t 17      conf:(0.85)
6. 1755=t 1758=t 19 ==> 1839=t 15      conf:(0.79)
7. 1755=t 1839=t 22 ==> 1846=t 17      conf:(0.77)
8. 1755=t 1839=t 1846=t 17 ==> 1758=t 13      conf:(0.76)

## Products Recommendation for E-commerce Application via Web Server Logs

```
9. 1758=t 1839=t 20 ==> 1755=t 15      conf:(0.75)
10. 1846=t 1853=t 16 ==> 1839=t 12      conf:(0.75)
11. 1839=t 1853=t 16 ==> 1846=t 12      conf:(0.75)
12. 1755=t 1758=t 19 ==> 1846=t 14      conf:(0.74)
13. 1758=t 1839=t 1846=t 18 ==> 1755=t 13      conf:(0.72)
14. 1755=t 1846=t 20 ==> 1758=t 14      conf:(0.7)
```

=== Evaluation ===

Elapsed time: 0.262s

## APPENDIX F – CODE LISTING

Data Integration module

```
<?php

/**
 * User: Kavindu Bandara
 * Date: 09/02/15
 * Time: 10:04 AM
 */

require_once 'config/local.php';
require_once 'lib/Connection.php';

/**
 * Class process
 */
class integration
{
    public function integrate(){

        $matchedRules = [];
        $initial = false;
        $handle = fopen(MINE_OUT_PUT, "r");
        if ($handle) {
            $is_need = false;
            while (($line = fgets($handle)) !== false) {

                if (strpos($line, MINE_OUT_RULE) !== false) {
                    $is_need = true;
                }
            }
        }
    }
}
```

## Products Recommendation for E-commerce Application via Web Server Logs

```
if (strpos($line, MINE_OUT_RULE_END) !== false) {
    $is_need = false;
}
// process the line read.

if($is_need){
    if($initial){
        $matchedRules[] = $line;
    }
    $initial = true;
}
}

fclose($handle);
} else {
    // error opening the file.
}
/**
 * 20. 1667=t 1668=t 3 ==> 1672=t 3   conf:(1)
    21. 1668=t 3 ==> 1667=t 1672=t 3   conf:(1)
 */
if(count($matchedRules) > 1){

    $leftMatch = [];

    foreach($matchedRules as $matchedRule){
        $matchedRule = explode(' ==> ', $matchedRule);

        $leftRule = explode('=t', $matchedRule[0]);
        $il = 0;
```

## Products Recommendation for E-commerce Application via Web Server Logs

```
array_pop($leftRule);
foreach($leftRule as $leftSide){

    if($il == 0){
        $leftSide = trim(end(explode(' ', $leftSide)));
    }
    $leftSide = trim($leftSide);

    $rightRule = explode('=t', $matchedRule[1]);
    $ir = 0;
    array_pop($rightRule);
    foreach($rightRule as $rightSide){

        $rightSide = trim($rightSide);

        if(isset($leftMatch[$leftSide][$rightSide])){
            $leftMatch[$leftSide][$rightSide] = $leftMatch[$leftSide][$rightSide] +
1;

        } else {
            $leftMatch[$leftSide][$rightSide] = 1;
        }
        $ir++;
    }
    $il++;
}

print_r($leftMatch);
$dbConnection = Connection::getInstance();
```

## Products Recommendation for E-commerce Application via Web Server Logs

```
foreach($leftMatch as $parent => $associations){
    foreach($associations as $association => $valueRelate){

        $queryInput = "SELECT link_id FROM catalog_product_link WHERE
product_id = $parent AND linked_product_id = $association AND link_type_id = 1";
        $stmt = $dbConnection->prepare($queryInput);
        $stmt->execute();

        $editRow=$stmt->fetch(PDO::FETCH_ASSOC);
        $num = $stmt->rowCount();
        if($num > 0){
            $queryUpdate = "UPDATE catalog_product_link_attribute_int SET
`value` = $valueRelate WHERE link_id = " . $editRow['link_id'] . " ";
            $stmt = $dbConnection->prepare($queryUpdate);
            $stmt->execute();
        }else{
            $queryInsert = "INSERT INTO catalog_product_link(product_id,
linked_product_id, link_type_id ) VALUES ( $parent , $association , 1)";
            $stmt = $dbConnection->prepare($queryInsert);
            $stmt->execute();

            $lastId = $dbConnection->lastInsertId();

            $queryInsertLink = "INSERT INTO
catalog_product_link_attribute_int(product_link_attribute_id, link_id, `value` ) VALUES
( 1 , $lastId , $valueRelate)";
            $stmt = $dbConnection->prepare($queryInsertLink);
            $stmt->execute();
        }
    }
}
```

```
$newClass = new integration();  
echo $newClass->integrate();
```

## Data Processing Module

```
<?php
/**
 * User: Kavindu Bandara
 * Date: 09/18/15
 * Time: 11:54 AM
 */

require_once 'config/local.php';
require_once 'lib/Connection.php';

/**
 * Class process
 */

class process
```



```
{  
    /**  
    * @param $string  
    * @return bool|mixed|string  
    */  
    function isProductLog($string)  
    {  
        $start = 'GET '  
        $end = '.html';  
        $endLength = 5;  
        $string = ' ' . $string;  
        $ini = strpos($string, $start);  
        if ($ini == 0) {  
            return "  
        }  
        $ini += strlen($start);  
        $len = strpos($string, $end, $ini) - $ini;  
        $stringRequired = substr($string, $ini, $len + $endLength);  
        $result = explode('/', $stringRequired);  
        if (count($result) >= 4) {  
            return false;  
        } elseif (count($result) < 4) {  
            return end($result);  
        }  
    }  
}  
  
public function mineAccessLogs()  
{  
  
    /**  
    * Define variables
```

## Products Recommendation for E-commerce Application via Web Server Logs

```
*/  
  
$dbConnection = Connection::getInstance();  
$mapProductId = array();  
$mapNotProductId = array();  
$userSession = array();  
$browserSession = array();  
$allProducts = array();  
  
$pattern = DATA_PATH_LOG;  
$fileList = glob($pattern);  
  
/**  
 * Get Already Map product from csv  
 */  
$file = fopen(MAP_PRODUCT_ID, 'r');  
while (($line = fgetcsv($file)) !== FALSE) {  
    $mapProductId[$line[0]] = $line[1];  
}  
fclose($file);  
  
/**  
 * Get Already Map product from csv  
 */  
$file = fopen(MAP_OTHER_ID, 'r');  
while (($line = fgetcsv($file)) !== FALSE) {  
    $mapNotProductId[$line[0]] = $line[1];  
}  
fclose($file);  
  
$row = 1;
```

## Products Recommendation for E-commerce Application via Web Server Logs

```
foreach($fileList as $file){
    if (($handle = fopen($file, "r")) !== false) {
        $row++;
        while (($data = fgetcsv($handle, 1000, ' ')) !== false) {

            if(!isset($data[9]) && $data[9] != ""){
                continue;
            }

            if (preg_match('#\b(Googlebot|bingbot|NewRelicPinger|nagios-
plugins|monitoring-plugins|msnbot|crawl|slurp|spider|dotbot)\b#',
                $data[9] ) == false )
            {
                if (strpos($data[9], 'Gecko/') == false) {

                    if (preg_match('#\b(v2_soap|.otf)\b#', $data[5]) == false) {

                        if (preg_match('#\b(302|301)\b#', $data[6]) == false) {

                            if (strpos($data[5], '.html') == true) {

                                if ($productUrl = $this->isProductLog($data[5])) {

                                    if (isset($mapProductId[$productUrl])) {
                                        $data[5] = $mapProductId[$productUrl];
                                    } elseif(isset($mapNotProductUrl[$productUrl])) {
                                        continue;
                                    } else{
                                        $post = null;
                                        $queryInput = "select product_id from core_url_rewrite where
request_path like '%" . $productUrl . "%' and options IS NULL limit 1";
```

## Products Recommendation for E-commerce Application via Web Server Logs

```
//          $query = $dbConnection->prepare($queryInput);
//          $query->execute();
//          $post = $query->fetch();

$queryPrep = $dbConnection->query($queryInput);
$queryPrep->setFetchMode(PDO::FETCH_ASSOC);
$post = $queryPrep->fetch();

if(isset($post['product_id']) && $post['product_id'] != null &&
$post['product_id'] != ""){
    $mapProductUrlId[$productUrl] = $post['product_id'];
    $data[5] = $post['product_id'];
}else{
    $mapNotProductUrl[$productUrl] = 'No Match found';
    continue;
}
}

if (!in_array($data[9], $browserSession)) {
    $browserSession[] = $data[9];
}
$key = array_search($data[9], $browserSession);
$data[9] = $key;

//$allProducts[$data[5]] = '?';
if(isset($allProducts[$data[5]]) && $allProducts[$data[5]] > 0){
    $allProducts[$data[5]] = $allProducts[$data[5]]+1;
}else{
    $allProducts[$data[5]] = 1;
}
```

```
        $userSession[$data[0]][$key][] = $data[5];
    }
}
}
}
}
}
}
}
}
fclose($handle);
}
}

//Save MAP Data
file_put_contents(MAP_PRODUCT_ID, "");
foreach ($mapProductUrlId as $key=>$value) {
    file_put_contents(MAP_PRODUCT_ID, $key.", ".$value . "\n", FILE_APPEND );
}

//Save MAP NON
file_put_contents(MAP_OTHER_ID, "");
foreach ($mapNotProductUrl as $key=>$value) {
    file_put_contents(MAP_OTHER_ID, $key.", ".$value . "\n", FILE_APPEND );
}

//array_filter($allProducts);
unset($allProducts[""]);
ksort($allProducts);
```

## Products Recommendation for E-commerce Application via Web Server Logs

```
file_put_contents(MAP_PRODUCT_COUNT, "");
$processAllProduct = array();
foreach($allProducts as $productIds=> $numCount){

    file_put_contents(MAP_PRODUCT_COUNT, $productIds.", ".$numCount . "\n",
FILE_APPEND );

    if($numCount > 2){
        $processAllProduct[$productIds] = '?';
    }

}

/**
 * fput csv on file new
 */

file_put_contents(DATA_OUT_PUT, "");

file_put_contents(DATA_OUT_PUT,          "@relation          supermarket"."\\n",
FILE_APPEND);
foreach($processAllProduct as $key => $productId){
    file_put_contents(DATA_OUT_PUT,  "@attribute  '$key'  {   t}" .  "\\n",
FILE_APPEND );
}
file_put_contents(DATA_OUT_PUT, "@data"."\\n", FILE_APPEND);

foreach($userSession as $user){

    foreach($user as $session){

        $isHaveValue = 0;
```

## Products Recommendation for E-commerce Application via Web Server Logs

```
$updatedRow = $processAllProduct;
foreach($session as $associted){
    if(isset($updatedRow[$associted])){
        $updatedRow[$associted] = 't';
        $isHaveValue++;
    }
}
if($isHaveValue == false){
    $debog = "";
}
if($isHaveValue > 1){
    file_put_contents(DATA_OUT_PUT, implode(' ' , $updatedRow) . "\n",
FILE_APPEND );
}

}

}

}

public function testFile(){
    file_put_contents(DATA_OUT_PUT, "@relation supermarket", FILE_APPEND);
}

}

$newClass = new process();
echo $newClass->mineAccessLogs();
```