

**UNIVERSITY OF NAIROBI
SCHOOL OF COMPUTING
AND
INFORMATICS**

^Mining Retail Outlet Transaction Data ^

BY

**Odek Ochieng' Felix
"" P56/7437/2006**

Supervisor

Dr. Peter W. Wagacha

August 2011

Submitted in partial fulfillment of the requirements of the Master of
Science in Information Systems


Declaration

This research is my original work and has not been submitted for any degree in any university.

Signed^ 

Date **O&CMoil**

This project has been submitted for examination with my approval as the university supervisor.

Signed^ 

Date & 3-*i \

Supervisors Name: Dr. Peter Waiganjo

Abstract

Retailing is increasingly becoming a high performance sector in the Kenya economy and retailers are fast seeking a competitive edge through technology. We describe the exploitation of Data Mining techniques and in particular association rule mining to analyze various baskets of a popular retail shop in Kisumu. The aim of the basket analysis was to allow retailers to quickly and easily look at the size, content and value of their customers' products to understand patterns, affinities and associations with a view to identifying cross-sell opportunities, improve shop floor layout and organization, encourage impulse buying driving promotions and advertisement based on the database intelligence and identify new business opportunities. We used CRISP-DM (Cross-Industry Standard Process for Data Mining) methodology for data mining and predictive analytics. CRISP DM was adopted because of its ability to iteratively move back and forth in all the six stages of data mining namely business understanding, data understanding, data preparation, modeling, evaluation and deployment. The dataset used for this case study contained data for both loyal and non loyal customers. We cleaned the data and converted it to binary format for analysis. The data was divided into two partitions in equal portions of two months periods. The results observed were that regularities in both partitions were fairly consistent such as the rules and itemset generated. Best transacted items revolved around basic Fast Moving Consumer Goods. With the regularities observed a floor plan and a stimulus response model were proposed for the retail shop with a view to improving impulse buying therefore improving sales.

Key words: Data mining, Association rule mining and market basket analysis.

Dedication

To my dearest parents, who have raised and educated me with their unconditional love & care.

Acknowledgments

My greatest gratitude is to my project advisor, Dr. Peter Waiganjo, was providing me with professional directions and solid academic training patiently. He was always inspiring and encouraging throughout my graduate study. His intellectual guidance and attitude towards life have not only led me to achieve a higher goal in academic but also inspired me to have greater expectations of myself.

Other thanks go to the entire panel of School of Computing and Informatics' insightful criticism and comments. I would also like to thank the following people very sincerely for their unique support, individually and collectively; Omondi Lwande, Orembe Onge'ye, Robert Odek and Pascal Odhiambo. Their efforts and time were sincerely appreciated. Also, some credits go to my class-mates for their help, encouragement and friendship.

Finally, I am grateful to everyone who has assisted me during my studies. Thank you all.

Table of Contents

Mining Retail Outlet Transaction Data.....	i
Declaration.....	jj
Abstract.....	jjj
Dedication.....	iv
Acknowledgments.....	v
List of Figures.....	viii
List of Tables.....	viii
List of Abbreviations.....	ix
CHAPTER ONE: INTRODUCTION.....	1
1.0 Overview.....	I
1.1 Motivation.....	1
1.2 Problem Statement.....	2
1.3 Justification for the study.....	2
1.4 Objective.....	3
1.5 Research questions.....	3
1.6 Scope.....	3
1.7 Structure of the report.....	3
CHAPTER TWO: LITERATURE REVIEW.....	4
2.0 Data Mining.....	4
2.1.0 Data Mining Methods.....	4
2.1.1 Classical Techniques: Statistics, Neighborhoods and Clustering.....	4
2.1.2 Next Generation Techniques: Trees, Networks and Rules.....	7
2.2.0 Association Rule Mining.....	13
2.2.1 Itemset Generation Phase.....	14
2.2.3 The Apriori Principle.....	16
2.2.4 Frequent Pattern Growth (FP-Growth) Algorithm.....	20
2.2.5 The ECLAT Algorithm.....	23
CHAPTER THREE: METHODOLOGY.....	24
3.0 The KDD Process.....	24
3.1.0 Data mining frameworks.....	25
3.1.1.SEMMA.....	25
3.1.2 CRISP DM.....	27
CHAPTER FOUR: THE DATA DESCRIPTION.....	31
4.0. The Retail shop.....	31
4.1 TheDataset.....	32
4.2.0 Dataset Processing.....	32
4.2.1 The data Attributes.....	33

4.2.2 Data segmentation	34
4.2.3. Data formatting.....	36
4.3. Basket analysis Platform.....	37
CHAPTER FIVE: RESULTS AND FINDINGS.....	38
5.0. Results and Findings.....	38
5.1.0 Generated Rules.....	38
5.1.2 Itemset Generated.....	41
5.1.3 Item transaction performance.....	43
5.2. Findings.....	45
5.2.1 The most transacted items that meet the minimum threshold.....	45
5.2.2 The most important regularities in the dataset.....	45
5.2.3 Deployment of Generated regularities to business operations.....	45
5.2.4 Models generated from the regularities.....	47
5.3 Conclusions and Recommendations.....	49
5.4 Recommendations and future work.....	49
REFERENCES.....	50

List of Figures

Figure 2.0:	Statistical Classification	6
Figure 2.1:	K-Nearest Neighbor.....	7
Figure 2.2:	Simple clustering of loan dataset	8
Figure 2.3:	Decision Tree.....	9
Figure 2.4:	Neural Network.....	12
Figure 2.5:	Itemset Lattice.....	17
Figure 2.6:	Apriori Principle.....	19
Figure 2.7:	Frequent Pattern Growth Tree.....	24
Figure 3.0:	KDD Process.....	28
Figure 3.1:	CRISP DM 1.0 Processes.....	31
Figure 5.0:	Itemset Performance Graph first partition.....	48
Figure 5.1:	Itemset Performance Graph Second partition.....	48
Figure 5.2:	Initial Shop Layout	50
Figure 5.3:	Proposed Shop Layout	51
Figure 5.4:	Stimulus Response Model.....	53

List of Tables

Table: 2.0:	Pseudo code for frequent item generation.....	19
Table 2.1:	Transaction database.....	21
Table 4.0:	Sample of Card master Transaction Table.....	33
Table 4.1:	Sample of Item transaction Table.....	35
Table 4.2:	Shows Clean data	35
Table 4.3:	Sample of data converted into binary format.....	39
Table 5.0:	Top 21 rules for the first partition of data.....	38
Table 5.1:	Top 30 rules for the second partition of data.....	39
Table 5.2:	Top 27 itemset for the first partition.....	41
Table 5.3 :	Top 30 itemsets for the second partition.....	42

List of Abbreviations

BOGOF		Buy One Get One Free
CAR! ,		Classification and Regression Trees
CHALL)		Chi- Squared Automatic Interaction Detection
CRISP-DM	-	Cross Industry Standard Process for Data Mining
DM	-	Data Mining
FP	-	Frequent Pattern Growth
KDD	-	Knowledge Discovery in Database
KNN	-	K-Nearest Neighbor
SEMMA	-	Sample Explore Modify Model Assess
WEKA	-	Waikato Environment for Knowledge Analysis
TID	-	Transaction Identification
FMCG	-	Fast moving Consumer Goods

CHAPTER ONE: INTRODUCTION

1.0 Overview

The exploitation of data mining techniques in retail industry is progressively being recognized as a sure means of achieving success and retail shops that apply them will never go wrong. In this paper, we present a means of exploiting data mining techniques in a retail shop with a view to improving layout, encouraging impulse buying, promoting cross selling and revamping internal operations.

Data mining, or the efficient discovery of interesting patterns from large collections of data, has been recognized as an important area of database research. The most commonly sought patterns are association rules which are a class of important regularities in data. Association Rules (Agrawal, Imielinski, & Swami, 1993; Agrawal & Srikant, 1994) in data mining is a technique that analyzes the correlations and patterns between sets of items. The key strength of association rule mining is that it can efficiently discover the complete set of associations that exist in data. These associations provide a complete picture of the underlying regularities in the domain. Different techniques, and algorithms have been proposed for solving this problem (Tan, Steinbach & Kumar, 2004; Gregory Piatetsky-Shapiro, et al, 1996). These algorithms are fully explained in chapter 2.

1.1 Motivation

There is an increasing focus on data mining, which has been defined as the application of data analysis and discovery algorithms to large databases with the goal of discovering (predictive) models (Gregory Piatetsky-Shapiro, et al, 1996). Business Week (Berry 1994) estimated that over half of all retailers are using or planning to use database marketing, and those who do use it have good results.

Retail stores are massively collecting large volumes of data in their daily business operations and the retail shops in themselves stock hundreds of thousands of items. One leading retail shop in Kisumu had more than 250,000 unique items and posted an average of 7,000 transactions per day. This volume of data was a sea of sitting knowledge that can yield strategic business

intelligence when extracted. Transactions at these stores were routinely captured at the point of sale. Furthermore major retail shops and many more were using Customer loyalty cards largely to retain their clients. These cards equally provided a means of understanding the customers' bio information and buying characteristics. Indeed they provided a means of understanding the value of the customer to the shop and how to reach them.

With data mining analysis, retailers can drive more profitable advertisement and promotions (database driven), attract more customers into the stores, increase the size and value of basket purchases, improve loyalty card promotions with longitudinal analysis, test and learn by using a market place as a laboratory, empower planners and vendors to make smarter decisions, march inventory to needs by customizing layouts, assortments and pricing to the local demographics.

1.2 Problem Statement

That most retail shops are sitting on enormous amount of information on their databases is evident with the sprawling of retail shops and the massive queues of customers transacting within these shops. The introduction of customer loyalty cards, the acceptance of usage of visa and credit cards are additional tools that capture customer transaction behavior and demographics. Most managers plan their product placement and replenishment, promote their product and organize advertisement based largely on their experience rather than on the basis of database driven intelligence.

1.3 Justification for the study

Retailers always assume risk every time they make decisions around buying, replenishment, advertising, promotions and assortment planning. These decisions need not be based on experience or instincts. A 1 % lift in sales or 0.01% improvement in margin can tip the balance between success, survival or failure. Every retailer's top-line sales and success require constant fine tuning of controls available to the retailer. Most retailers still suffer from the old age retail problem of stocking too much of the wrong item and not enough of the right one. The right product moves and the wrong one sits until it is marked down. Customer's life is further complicated when he or she cannot get enough quantities of a popular product. Data mining therefore will leverage retailers on smarter decision making process.

1.4 Objective

The main goal is to exploit data mining techniques to perform basket analysis with a view to using the knowledge mined to improve on sales and assortment planning.

The specific objectives are:

- i. To examine algorithms for association rule mining
- ii. To extract interesting and useful patterns from a retail shop database.
- iii. To develop models based on generated patterns.

1.5 Research questions

This study was guided by the following questions:

- i. What are the most frequently transacted items in the database within the set metrics?
- ii. What are the most interesting regularities in the database?
- iii. How can such regularities aid marketing strategies?
- iv. What model(s) can be generated from such regularities?

1.6 Scope

The study was based on transactional data collected from a leading retail shop in Kisumu Town. The choice of this shop was informed by virtue of its location, at the centre of the city. Being that other branches are equally dispersed within and outside the city, we believed that the finding from the study will be a true reflection of what was applicable in other branches across the city.

1.7 Structure of the report

The report was done in five chapters. Chapter one introduced data mining and the objective of mining retail outlet transaction data. Chapter Two part one reviewed data mining and part two reviewed association rule mining and the algorithms that implement it. Chapter Three described the mining methodologies and or process, and the choice of CRISP_DM employed in this work. In Chapter Four, data was described and the rules and item sets were generated and models developed. Chapter Five contained the analysis and discussions of the finding, conclusion and recommendation.

CHAPTER TWO: LITERATURE REVIEW

2.0 Data Mining

Data mining is a step in Knowledge Discovery (KDD) process aimed at discovering patterns and relationships in preprocessed & transformed data, (Marc M. Van Hulle. 2004). According to Ho Tu Bao (2002) data mining is a step in the knowledge discovery process consisting of particular data mining algorithms that, under some acceptable computational efficiency limitations, finds patterns or models in data.

Data mining involves fitting models to, or determining patterns from, observed data. The fitted models play the role of inferred knowledge: Whether the models reflect useful or interesting knowledge is part of the overall, interactive KDD process where subjective human judgment is typically required.

Data mining techniques allow inferring recommendation rules or building recommendation models from large datasets (Schafer, J. B., 2006). In commercial applications, Machine Learning algorithms are used to analyze the demographics and past buying history of customers, and find patterns to predict future buying behavior. The two primary goals of data mining in practice tend to be prediction and description. Prediction involves using some variables or fields in the database to predict unknown or future values of other variables of interest. Description focuses on finding human interpretable patterns describing the data. The relative importance of prediction and description for particular data mining applications can vary considerably.

2.1.0 Data Mining Methods

Data mining methods can largely be classified into two (Kurt Thearling et al, 1995):

2.1.1 Classical Techniques: Statistics, Neighborhoods and Clustering

These techniques have been used for decades on existing business problems;

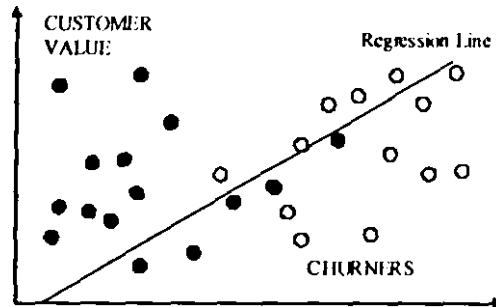
i. Statistics

By strict definition "statistics" or statistical techniques are not data mining. They were being used long before the term data mining was coined to apply to business applications. However,

statistical techniques are driven by the data and are used to discover patterns and build predictive models.

There are a variety of different types of regression in statistics but the basic idea is that a model is created that maps values from predictors in such a way that the lowest error occurs in making a prediction.

FIGURE 2.0: Statistical classification



ii. Nearest Neighbor

Nearest neighbor is a prediction technique that is quite similar to clustering - its essence is that in order to predict what a prediction value is in one record look for records with similar predictor values in the historical database and use the prediction value from the record that it "nearest" to the unclassified record.

Income

- Pays Bills
- Defaults

FIGURE 2.1 The nearest neighbors are shown graphically for three unclassified records: A, B, and C.

K-nearest neighbor algorithm (KNN) is part of supervised learning that has been used in many applications in the field of data mining, statistical pattern recognition and many others. An object is classified by a majority vote of its neighbors. K is always a positive integer. The neighbors are taken from a set of objects for which the correct classification is known.

The most used distance is the Euclidean distance, though other distance measures such as the Manhattan and Chebyshev distances could in principle be used instead.

The algorithm on how to compute the K-nearest neighbors is as follows:

- i.
 - o Determine the parameter K = number of nearest neighbors beforehand. This value is all up to you.
 - o Calculate the distance between the query-instance and all the training samples. You can use any distance algorithm,
 - o Sort the distances for all the training samples and determine the nearest neighbor based on the K-th minimum distance,
 - o Since this is supervised learning, get all the Categories of your training data for the sorted value which fall under K.

- o Use the majority of nearest neighbors as the prediction value,

iii. **Clustering**

Clustering is a common descriptive task where one seeks to identify a finite set of categories or clusters to describe the data. The categories may be mutually exclusive and exhaustive, or consist of a richer representation such as hierarchical or overlapping categories. Examples include discovering homogeneous sub-populations for consumers in marketing databases and identification of sub-categories of spectra from infrared sky measurements.

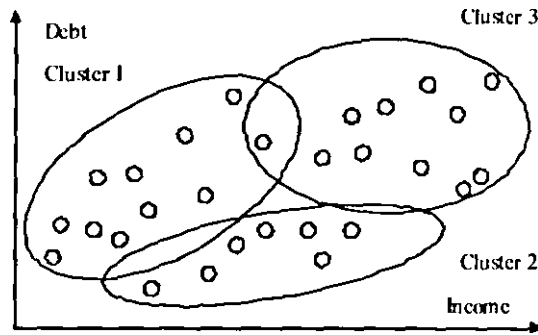


FIGURE 2.2 A simple clustering of the loan data set into three cluster

2.1.2 Next Generation Techniques: Trees, Networks and Rules

i. **Trees**

A decision tree is a predictive model that, as its name implies, can be viewed as a tree. Specifically each branch of the tree is a classification question and the leaves of the tree are partitions of the dataset with their classification. For instance if we were going to classify customers who are loyal in a retail shop, the decision tree might look like Figure 2.3.

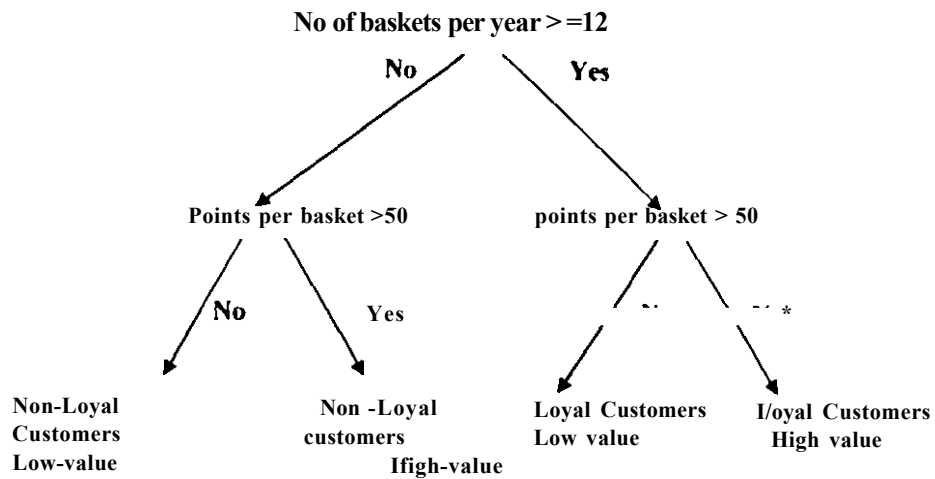


FIGURE 23 A decision tree is a predictive model that makes a prediction on the basis of a series of decision.

The first component is the top decision node, or root node, which specifies a test to be carried out. The root node in this case is "No of baskets per year ≥ 12 ." The results of this test cause the tree to split into branches, each representing one of the possible answers. In this case, the test "No of baskets per year ≥ 12 ." can be answered either "yes" or "no," and so we get two branches. Depending on the algorithm, each node may have two or more branches. For example, CART generates trees with only two branches at each node. Such a tree is called a binary tree. When more than two branches are allowed it is called a multiway tree. Each branch will lead either to another decision node or to the bottom of the tree, called a leaf node.

By navigating the decision tree you can assign a value or class to a case by deciding which branch to take, starting at the root node and moving to each subsequent node until a leaf node is reached. Each node uses the data from the case to choose the appropriate branch.

With this sample tree a marketing officer can determine whether the customer a loyal customer, high value customer or non loyal customer. A customer who purchases goods worth 50 points means spending a minimum of 5000/= (five thousand Kenya Shillings in one visit). If such a customer makes more than one visit per month to the shop then he can be classified as "High value, Loyal Customer".

Decision trees models are commonly used in data mining to examine the data and induce the tree and its rules that will be used to make predictions. A number of different algorithms may be used for building decision trees including CHAID (Chi-squared Automatic Interaction Detection), CART (Classification And Regression Trees), Quest, and C5.0.

Decision trees are grown through an iterative splitting of data into discrete groups, where the goal is to maximize the "distance" between groups at each split. One of the distinctions between decision tree methods is how they measure this distance.

Decision trees which are used to predict categorical variables are called classification trees because they place instances in categories or classes. Decision trees used to predict continuous variables are called regression trees.

Decision trees make few passes through the data (no more than one pass for each level of the tree) and they work well with many predictor variables. As a consequence, models can be built very quickly, making them suitable for large data sets.

Trees left to grow without bound take longer to build and become unintelligible, but more importantly they overfit the data. Tree size can be controlled via stopping rules that limit growth. One common stopping rule is simply to limit the maximum depth to which a tree may grow. Another stopping rule is to establish a lower limit on the number of records in a node and not do splits below this limit. An alternative to stopping rules is to prune the tree. The tree is allowed to grow to its full size and using built-in heuristics or user intervention, the tree is pruned back to the smallest size that does not compromise accuracy. For example, a branch or subtree that the user feels is inconsequential because it has very few cases might be removed. CART prunes trees by cross validating them to see if the improvement in accuracy justifies the extra nodes.

A common criticism of decision trees is that they choose a split using a "greedy" algorithm in which the decision on which variable to split does not take into account any effect the split might have on future splits. In other words, the split decision is made at the node "in the moment" and it is never addition, all splits are made sequentially, so each split is dependent on its predecessor. Thus all future splits are dependent on the first split, which means the final solution could be

very different if a different first split is made. The benefit of looking ahead to make the best splits based on two or more levels at one time is unclear. Such attempts to look ahead are in the research stage, but are very computationally intensive and presently unavailable in commercial implementations.

Furthermore, algorithms used for splitting are generally univariate; that is, they consider only one predictor variable at a time. And while this approach is one of the reasons the model builds quickly — it limits the number of possible splitting rules to test — it also makes relationships between predictor variables harder to detect,

ii. Networks

Neural networks are very powerful predictive modeling techniques but some of the power comes at the expense of ease of use and ease of deployment. Neural nets may be used in classification problems (where the output is a categorical variable) or for regressions (where the output variable is continuous).

A neural network as shown in Figure 2.4 starts with an input layer, where each node corresponds to a predictor variable. These input nodes are connected to a number of nodes in a hidden layer. Each input node is connected to every node in the hidden layer. The nodes in the hidden layer may be connected to nodes in another hidden layer, or to an output layer. The output layer consists of one or more response variables.

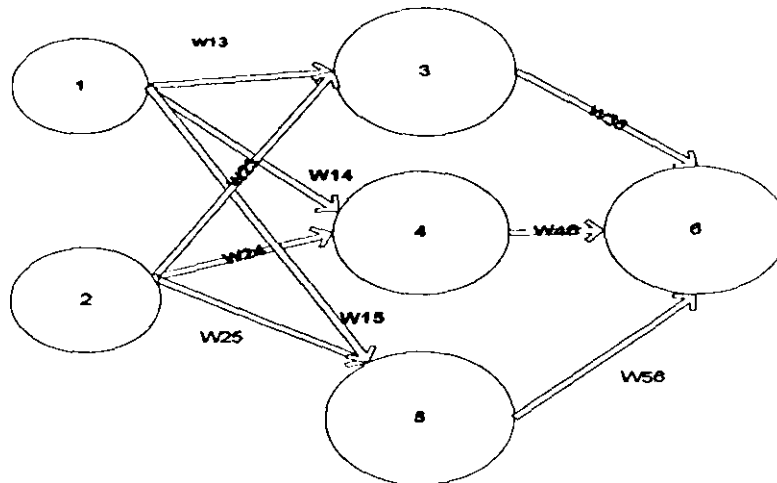


FIGURE 2.4: Neural network

After the input layer, each node takes in a set of inputs, multiplies them by a connection weight W_{xy} (e.g., the weight from node 1 to 3 is W_{13} — (Figure 5), adds them together, applies a function (called the activation or squashing function) to them, and passes the output to the node(s) in the next layer. For example, the value passed from node 4 to node 6 is:

Activation function applied to $([W_{1A} * \text{value of node 1}] + [W_{2A} * \text{value of node 2}])$

Each node may be viewed as a predictor variable (nodes 1 and 2 in this example) or as a combination of predictor variables (nodes 3 through 6). Node 6 is a non-linear combination of the values of nodes 1 and 2, because of the activation function on the summed values at the hidden nodes. In fact, if there is a linear activation function but no hidden layer, neural nets are equivalent to a linear regression; and with certain non-linear activation functions, neural nets are equivalent to logistic regression.

The connection weights (W 's) are the unknown parameters which are estimated by a training method. Originally, the most common training method was **back propagation**; newer methods include conjugate gradient, quasi-Newton, Levenberg-Marquardt, and genetic algorithms. Each training method has a set of parameters that control various aspects of training such as avoiding local optima or adjusting the speed of convergence.

The architecture (or topology) of a neural network is the number of nodes and hidden layers, and how they are connected. In designing a neural network, either the user or the software must choose the number of hidden nodes and hidden layers, the activation function, and limits on the weights. While there are some general guidelines, you may have to experiment with these parameters. One of the most common types of neural network is the feed-forward back propagation network.

Back propagation training is simply a version of gradient descent, a type of algorithm that tries to reduce a target value (error, in the case of neural nets) at each step. The algorithm proceeds as follows.

Feed forward: The value of the output node is calculated based on the input node values and a set of initial weights. The values from the input nodes are combined in the hidden layers, and the values of those nodes are combined to calculate the output value.

Back propagation: The error in the output is computed by finding the difference between the calculated output and the desired output (i.e., the actual values found in the training set). Next, the error from the output is assigned to the hidden layer nodes proportionally to their weights. This permits an error to be computed for every output node and hidden node in the network. Finally, the error at each of the hidden and output nodes is used by the algorithm to adjust the weight coming into that node to reduce the error.

This process is repeated for each row in the training set. Each pass through all rows in the training set is called an epoch. The training set will be used repeatedly, until the error no longer decreases. At that point the neural net is considered to be trained to find the pattern in the test set. Because so many parameters may exist in the hidden layers, a neural net with enough hidden nodes will always eventually fit the training set if left to run long enough. But how well it will do on other data?

To avoid an overfitted neural network which will only work well on the training data, you must know when to stop training. Some implementations will evaluate the neural net against the test data periodically during training. As long as the error rate on the test set is decreasing, training will continue. If the error rate on the test data goes up, even though the error rate on the training data is still decreasing, then the neural net may be overfitting the data.

iii. Rule induction

Rule induction on a data base can be a massive undertaking where all possible patterns are systematically pulled out of the data and then an accuracy and significance are added to them that tell the user how strong the pattern is and how likely it is to occur again. In general these rules are relatively simple such as for a market basket database of items scanned in a consumer market basket you might find interesting correlations in your database such as:

If Kodak-battery is purchased then a towel is purchased 90% of the time and this pattern occurs in 3% of all shopping baskets.

If bar-soap is purchased then dettol is purchased 60% of the time and these two items are bought together in 6% of the shopping baskets.

The rules that are pulled from the database are extracted and ordered to be presented to the user based on the percentage of times that they are correct and how often they apply.

This project will be using the rule induction logic as developed in association rule mining in Basket Analyzer Tool

2.2.0 Association Rule Mining

Association Rules (Agrawal, Imielinski, & Swami, 1993; Agrawal & Srikant, 1994) in data mining is a technique that analyzes the correlations and patterns between sets of items. Let I be the domain of literals called *items*. A record called a *transaction* contains a set of items $\{A, B, C, \dots\}$. The input to the association rule mining algorithm is a set of transactions, T . We call any set of items $\{A, B, C, \dots\}$ collectively an *itemset*.

An *association rule* is a relation of the form $A \Rightarrow B$ in T , where A, B are itemsets, and $A \cap B = \emptyset$. A is the antecedent of the rule and B is the consequent of the rule.

An itemset has a measure of statistical significance associated with it called *support*. Support for an itemset X in T ($support(X)$) is the number of transactions in T containing X . For a rule, $A \Rightarrow B$, the associated support is $support(A \cup B)$. A *support fraction* is the ratio of the support value to the total number of transactions.

The strength of a rule is given by another measure called *confidence*. The confidence of $A \Rightarrow B$ is the ratio $support(A \cup B) / support(A)$.

The problem of association rule mining is to generate all rules that have support and confidence greater than some user-specified thresholds. Itemsets that have support greater than the user-specified support are called *large* itemsets. For a large itemset S , if $A \subset S$ and $\text{support}(S) / \text{support}(A) \geq \text{confidence threshold}$, then $A \Rightarrow S$ is an association rule.

The problem of association rule mining is, thus, broken down into two tasks (Marc M. Van Hulle, 2004; Agrawal & Srikant, 1994; Feng Zhang, Hui-You Chang, 2005):

(i) The task of determining all large itemsets. This stage is split into two parts

a) *candidate-generation phase* - a set of itemsets called *candidate itemsets* are chosen, this set is chosen such that it contains all potential *large itemsets*.

b) *large-itemset generation phase* - the support for the candidates are counted, those with support greater than or equal to the user-specified minimum support qualify to become *large*.

(ii) The task of determining the rules with enough confidence, from the large itemsets.

2.2.1 Itemset Generation Phase

A lattice structure can be used to enumerate all possible itemset as shown in figure 2.5. The figure shows an itemset lattice for $I = \{a, b, c, d, e\}$. In general a dataset that contains a k itemset can potentially generate upto $2^k - 1$ frequent itemset excluding the null set.

Because k can be very large in practical situation, the search space of items that need to be generated is exponentially large.

FIGURE 2.5: itemset lattice

There are several ways to reduce the complexity of frequent itemset generation:

- i) **Reduce the number of candidate itemset (M).** The Apriori Principle is an effective way to eliminate some of the candidate itemset without counting their support value.
- ii) **Reduce the number of comparison.** Instead of matching each candidate itemset against every transaction, reduce the number of comparison either by storing the candidate itemsets or by compressing the data set.

2.2.2 The Association Rule Mining Algorithms

Several algorithms have been developed for Rule mining but principally there are three key algorithms generally used in association rule mining namely Apriori, Frequent pattern Growth (FP) and ECLAT algorithms.

2.2.3 The Apriori Principle.

The Apriori algorithm (Agrawal, Imielinski, & Swami, 1993; Agrawal & Srikant, 1994) provides an efficient method for generating association rules. It obtains its efficiency by using potentially smaller candidate sets when counting the support for itemsets to identify large itemsets. It uses the fact that for a given support all the subsets of a large itemset need to be large itemsets themselves.

Suppose $\{c, d, e\}$ is a frequent itemset, clearly any transaction that contains $\{c, d, e\}$ must also contain its subsets $\{c, d\}$, $\{c, e\}$, $\{d, e\}$, $\{c\}$, $\{d\}$ and $\{e\}$. Consequently if $\{c, d, e\}$ is frequent then all subsets of $\{c, d, e\}$ must also be frequent.

Conversely if an itemset such $\{a, b\}$ is infrequent then all of its supersets must also be infrequent too. The entire sub graph containing the superset $\{a, b\}$ can be pruned immediately once $\{a, b\}$ is found to be infrequent.

This strategy of pruning the exponential search space based on the support measure is known as **support based pruning**. Such a pruning strategy is made possible by a key property of support measure, namely, that the support of an itemset never exceeds the support for its subsets. This is known as the **anti-monotone** property of support measure (Agrawal, Imielinski, & Swami, 1993; Agrawal & Srikant, 1994).

Any measure that possesses an anti-monotone property can be incorporated directly into the mining algorithm to effectively prune the exponential search space of candidate itemset.

$abcde^{\wedge}$

FIGURE 2.6: Apriori principle: If $\{c, d, e\}$ is frequent then all its subsets are frequent.

a) Frequent itemset generation in the Apriori algorithm

- The algorithm initially makes a single pass over the dataset to determine the support of each item. Upon completion of this step, the set of all frequent 1-itemsets **F1** will be known as shown in Steps **I** and **2** of **Table 2.0**.
- Next the algorithm will iteratively generate new candidate k itemset using the frequent $(k-1)$ itemset found in the previous iteration. (Step5). Candidate generation is implemented using a function called Apriori-gen.
- To count the support of the candidate, the algorithm needs to make an additional pass over the dataset. (Steps **6-10**). The subset function is used to determine all the candidate itemset in C_k that are contained in each transaction t .
- After counting their support, the algorithm eliminates all candidate itemset whose support counts are less than *minsup*. (Step **12**).

- The algorithm terminates when there are no new frequent item set generated. I.e. $F_k = \emptyset$ (Step 13)

The pseudo code for the frequent itemset generation part of Apriori algorithm is shown below. Let Q denote the set of candidate k -itemsets and F^* denote the set of frequent k -itemset:

- The algorithm initially makes a single pass over the dataset to determine the support of each item. Upon completion of this step, the set of all frequent 1-itemsets F_1 will be known (steps 1 and 2).
- Next, the algorithm will iteratively generate new candidate k -itemset using the frequent $(k-1)$ itemsets found in the previous iteration (step 5). Candidate generation is implemented using a function called *apriori-gen*.
- To count the support of the candidate the algorithm needs to make an additional pass over the dataset (steps 6-10). The subset function is used to determine all the candidate itemsets in C^* that are contained in each transaction t .
- After counting their support, the algorithm eliminates all candidate itemsets whose support count are less than *minsup* (step 12).
- The algorithm terminates when there are no new frequent itemset generated, i.e. $F_k = \emptyset$

<i>Steps</i>	<i>Pseudo-Code</i>
1	$k = 1.$
2	$f_k = \{c \mid \text{ACT}(c) \geq \text{NX} \text{ minsup}\}$ (Find all frequent itemset)
3	<i>Repeat</i>
4	$k=k+1$
5	$C_k = \text{Apriori-gen}(F_{k-1})$ {Generate candidate itemset}
6	For each transaction $t \in D$ do.
7	$a = \text{Subset}(C_k, t)$
8	For each candidate $c \in C_k$ do.
9	$\text{count}(a) = \text{count}(a) + 1$ {increment support count}
10	end for
11	end for
12	$F_k = \{c \mid c \in C_k \wedge \text{ACT}(c) \geq \text{NX} \text{ min sup}\}$ /Extract k frequent itemset}
13	Until $F_k = \langle f \rangle$
14	Result $\cup F_k$

Table: 2.0: Pseudo code for frequent item generation part of Apriori algorithm

The frequent itemset generation part of a priori algorithm has two important characteristics:

First, it's a **level wise** algorithm: i.e. it traverses itemset lattice one level at a time, from frequent 1-itemset to the maximum size of frequent itemsets.

Second it employs a **generate-and-test** strategy for finding frequent itemsets. At each iteration new candidate itemsets are generated from the frequent itemsets found in the previous iteration. The support of each candidate is then counted and tested against the **minsup** threshold. The total number of iteration needed by the algorithm is $k^{\wedge} + 1$ where k^{\wedge} is the maximum size of the frequent itemset.

The Apriori-gen function shown in step5 of the algorithm above algorithm generates candidate item set by performing the following two operations:

- I. Candidate generation: This operation generates new candidate k-itemset based on the frequent (k-1) itemset found in the previous iteration.
- II. Candidate pruning: This operation eliminates some of the candidate k-itemset using support based pruning strategy.

To illustrate the candidate pruning operation, consider a candidate k-itemset $\{i_1, i_2, \dots, i_k\}$. The algorithm must determine whether all of its proper subsets, $X - \{i_y\}$ ($\forall y = 1, 2, 3, \dots, X$) are frequent. If one of them is infrequent then X is immediately pruned. This approach can effectively reduce the number of candidate itemset considered during support pruning. The complexity of this operation is $O(k)$ for each candidate k-itemset. If m of the k subsets were used to generate a candidate then only the remaining k-m itemsets need be considered during candidate pruning.

Examples of such applications that utilize the Apriori principle for rule generation include

b) Rule generation Phase

Each frequent K-itemset, Y, can produce upto $2^k - 2$ association rules, ignoring rules that have empty antecedents or consequents ($\emptyset \rightarrow Y$ or $\emptyset \rightarrow Y$). Association rule can be extracted by partitioning the itemset Y into non-empty subsets X and $Y - X$ such that $X \wedge Y - X$ satisfies the confidence threshold.

Note that all such rules must have met the minimum threshold because they are generated from a frequent itemset.

2.2.4 Frequent Pattern Growth (FP-Growth) Algorithm

The FP-growth algorithm is currently one of the fastest approaches to frequent item set mining. It is based on a prefix tree representation of the given database of transactions (called an FP-tree), which can save considerable amounts of memory for storing the transactions (Bonchi and B. Goethals,2004). The basic idea of the FP-growth algorithm can be described as a recursive elimination scheme.

i. Preprocessing

Similar to several other algorithms for frequent item set mining, like, for example, Apriori or Eclat, FP-growth preprocesses the transaction database as follows; in an initial scan the frequencies of the items (support of single element item sets) are determined. All infrequent items—that is, all items that appear in fewer transactions than a user-specified minimum number—are discarded from the transactions, since, obviously, they can never be part of a frequent item set. In addition, the items in each transaction are sorted, so that they are in descending order w.r.t. their frequency in the database. Although the algorithm does not depend on this specific order, experiments showed that it leads to much shorter execution times than a random order. An ascending order leads to a particularly slow operation in my experiments, performing even worse than a random order F. Bonchi and B. Goethals(2004).

a d f	d	8	da
a c d e	b	7	dcac
bd	c	5	db
bed	a	4	dbc
be	e	3	be
abd	f	2	dba
bde	g	1	dbe
b c e g			bee
c d f			dc
abd			dba

Table 2.1: Transaction database (left), item frequencies (middle), and reduced transaction database with items in transactions sorted descendingly w.r.t their frequency (right).

Figure 2.7: FP-tree for the (reduced) transaction database shown in Table 1.

ii. Building the Initial FP-tree

After all individually infrequent items have been deleted from the transaction database, it is turned into an FP-tree. An FP-tree is basically a prefix tree for the transactions. That is, each path represents a set of transactions that share the same prefix, each node corresponds to one item. In addition, all nodes referring to the same item are linked together in a list, so that all transactions containing a specific item can easily be found and counted by traversing this list. The list can be accessed through a head element, which also states the total number of occurrences of the item in the database. As an example, Figure 1 shows the FP-tree for the (reduced) database shown in Table I on the right. The head elements of the item lists are shown to the left of the vertical grey bar, the prefix tree to the right of it.

iii. Projecting an FP -tree

The core operation of the FP-growth algorithm is to compute an FP-tree of a projected database, that is, a database of the transactions containing a specific item, with this item removed. This projected database is processed recursively, remembering that the frequent item sets found in the recursion share the removed item as a prefix.

2.2.5 The ECLAT Algorithm

Eclat was the first FI-miner using a vertical encoding of the database combined with a depth-first traversal of the search space (organized in a prefix-tree). Eclat is a plain FI-miner traversing the IT-tree in a depth-first manner in a pre-order way, from left-to-right [Mohammed J. Zaki ,1999; Yui Liu et al. 2003].

The Eclat-based algorithms have the advantage of fast support computing through tid-list intersection. By independent task parallelism, they gain very good speedups on distributed memory multiprocessors. The main drawback of these algorithms is that they need to generate and redistribute the vertical TID-lists of which the total size is comparable to that of the original database. Also, for a long frequent itemset, the major common parts of the TID-lists are repeatedly intersected for all its subsets. To alleviate this situation, difiset optimization [Mohamed j. Zaki and Karam Gouda, 2003] has been proposed to track only the changes in TID-lists instead of keeping the entire TID-lists through iterations so that it can significantly reduce the amount of data to be computed.

CHAPTER THREE: METHODOLOGY

3.0 The KDD Process

Knowledge discovery in databases (KDD) is the non-trivial process of identifying valid, novel, potentially useful, and ultimately understandable patterns in data. KDD process, as presented in (Fayyad et al, 1996), is the process of using DM methods to extract what is deemed knowledge according to the specification of measures and thresholds, using a database along with any required preprocessing, sub sampling, and transformation of the database. There are considered five stages, presented in figure 8:

Selection - this stage consists on creating a target data set, or focusing on a subset of variables or data samples, on which discovery is to be performed. Sometimes the combination of data from ubiquitous sources can be useful, but possible matters of compatibility have to be observed;

Pre-processing - this stage consists on the target data cleaning and pre processing in order to obtain consistent data. The less noise contained in data the higher is the efficiency of data mining. Elements of the pre-processing span the cleaning of wrong data, the treatment of missing values and the creation of new attributes;

Transformation - this stage consists on the transformation of the data using dimensionality reduction or transformation methods, the reduction can be made via lossless aggregation or a loss full selection of only the most important elements. A representative selection can be used to draw conclusions to the entire data;

Data Mining - this stage consists on the searching for patterns of interest in a particular representational form, depending on the DM objective (usually, prediction);

Interpretation/Evaluation - this stage consists on the interpretation and evaluation of the mined patterns.

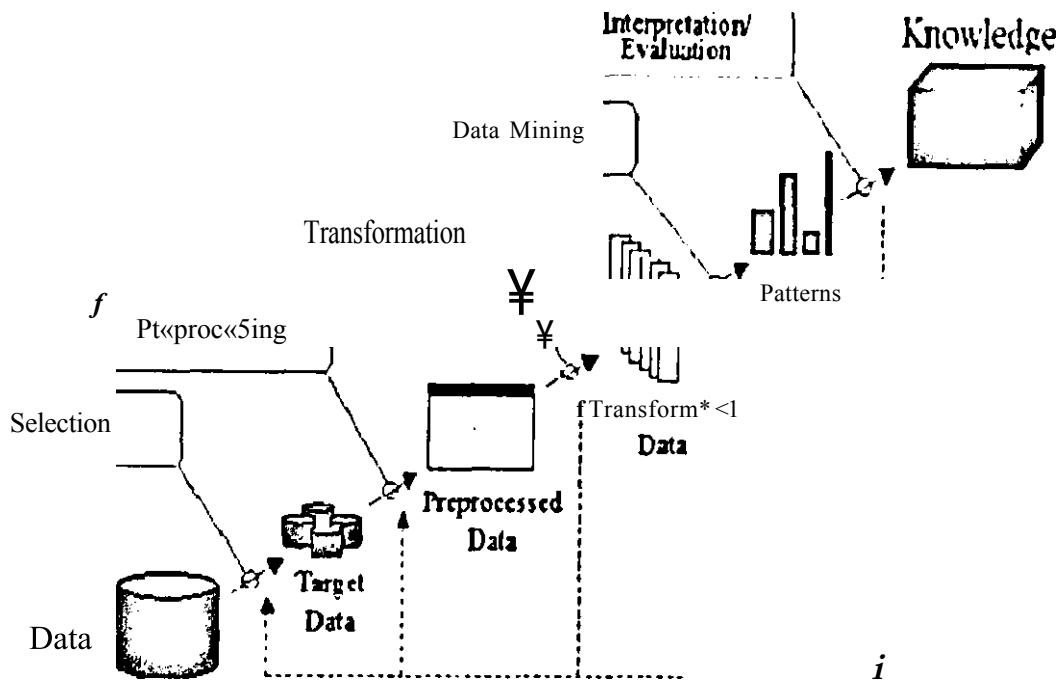


Figure 3.0: KDD Process

3.1.0 Data mining frameworks

The academics efforts are centered in the attempt to formulate a general framework for DM (Dzeroski, 2006). The bulk of these efforts are centered in the definition of a language for DM that can be accepted as a standard, in the same way that SQL was accepted as a standard for relational databases (Han et al, 1996) (Meo et al, 1998) (Imielinski et al, 1999) (Sarawagi, 2000) (Botta et al, 2004).

The efforts in the industrial field concern mainly the definition of processes/methodologies that can guide the implementation of DM applications. The widely used processes/methodologies include:

3.1.1. SEMMA

The acronym SEMMA - sample, explore, modify, model, assess - refers to the core process of conducting data mining. Beginning with a statistically representative sample of data, SEMMA makes it easy to apply exploratory statistical and visualization techniques select and transform

the most significant predictive variables, model the variables to predict outcomes, and confirm a model's accuracy. SEMMA is focused on the model development aspects of data mining:

Sample data by extracting a portion of a large data set big enough to contain the significant information, yet small enough to manipulate quickly. For optimal cost and performance, SAS Institute advocates a sampling strategy, which applies a reliable, statistically representative sample of large full detail data sources. Mining a representative sample instead of the whole volume reduces the processing time required to get crucial business information.

Explore data by searching for unanticipated trends and anomalies in order to gain understanding and ideas. Exploration helps refine the discovery process. If visual exploration doesn't reveal clear trends, you can explore the data through statistical techniques including factor analysis, correspondence analysis, and clustering. For example, in data mining for a direct mail campaign, clustering might reveal groups of customers with distinct ordering patterns. Knowing these patterns creates opportunities for personalized mailings or promotions.

Modify your data by creating, selecting, and transforming the variables to focus the model selection process. Based on your discoveries in the exploration phase, one may need to manipulate your data to include information such as the grouping of customers and significant subgroups, or to introduce new variables. One may also need to look for outliers and reduce the number of variables, to narrow them down to the most significant ones. One may also need to modify data when the "mined" data change. Because data mining is a dynamic, iterative process, you can update data mining methods or models when new information is available.

Model data by allowing the software to search automatically for a combination of data that reliably predicts a desired outcome. Modeling techniques in data mining include neural networks, tree-based models, logistic models, and other statistical models -- such as time series analysis, memory-based reasoning, and principal components. Each type of model has particular strengths, and is appropriate within specific data mining situations depending on the data. For example, neural networks are very good at fitting highly complex nonlinear relationships.

Assess data by evaluating the usefulness and reliability of the findings from the data mining process and estimate how well it performs. A common means of assessing a model is to apply it to a portion of data set aside during the sampling stage. If the model is valid, it should work for this reserved sample as well as for the sample used to construct the model. Similarly, you can test the model against known data. For example, if you know which customers in a file had high retention rates and your model predicts retention, you can check to see whether the model selects these customers accurately. In addition, practical applications of the model, such as partial mailings in a direct mail campaign, help prove its validity.

It thus confers a structure for his conception, creation and evolution, helping to present solutions to business problems as well as to find the DM business goals. (www.sas.com/offices/europe/uk/technologies/analytics/datamining/miner/semma.html, Santos & Azevedo, 2005,)

3.1.2 CRISP DM

CRISP-DM (the Cross-Industry Standard Process for Data Mining) is the industry standard methodology for data mining and predictive analytics (www.crisp-dm.org accessed on 21st February 2011). The life cycle of a mining project consists of six phases as shown in figure 3.1. The sequence of the phases is not rigid as moving back and forth between different phases is always required. It depends on the outcome of each phase, which phase or which particular task of a phase has to be performed next. The arrows indicate the most important and frequent dependencies between phases.

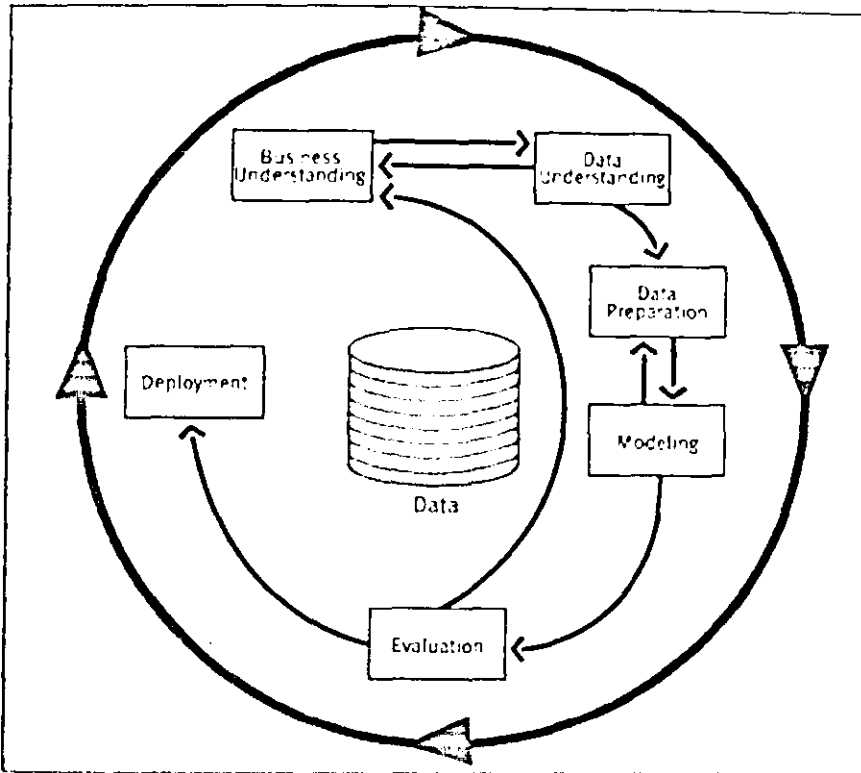


Figure 3.1 CRISP-DM 1.0 Processes (Image Courtesy of James P. Greichen)

Business understanding

This initial phase focuses on understanding the project objectives and requirements from a business perspective, then converting this knowledge into a data mining problem definition and a preliminary plan designed to achieve the objectives.

Data understanding

The data understanding phase starts with an initial data collection and proceeds with activities in order to get familiar with the data, to identify data quality problems, to discover insights into the data or to detect interesting subsets to form hypotheses for hidden information.

Data preparation

The data preparation phase covers all activities to construct the Final dataset (data that will be fed into the modeling tool(s) from the initial raw data. Data preparation tasks are likely to be

performed multiple times and not in any prescribed order. Tasks include table, record and attribute selection as well as transformation and cleaning of data for modeling tools.

Modeling

In this phase, various modeling techniques are selected and applied and their parameters are calibrated to optimal values. Typically, there are several techniques for the same data mining problem type. Some techniques have specific requirements on the form of data. Therefore, stepping back to the data preparation phase is often necessary.

Evaluation stage

At this stage in the project a model (or models) is built that appear to have high quality from a data analysis perspective. Before proceeding to final deployment of the model, it is important to more thoroughly evaluate the model and review the steps executed to construct the model to be certain it properly achieves the business objectives. A key objective is to determine if there is some important business issue that has not been sufficiently considered. It compares results with the evaluation criteria defined at the start of the project.

A good way of defining the total outputs of a data mining project is to use the equation:

$$\mathbf{Results = Models + Findings}$$

In this equation, it can be seen that the total Output of data Mining project is not just the models (although they are, of course, important) but also findings which is defined as anything (apart from the model) which will guide in meeting the business objective, or important in leading to new questions, line of approach or side effects e.g. data quality problems uncovered by the data mining exercise.

It is important to note that although the models are directly related to business questions the finding need not be related to any questions or business objective.

At the end of this phase, a decision on the use of the data mining results should be reached.

Deployment stage

Creation of the model is generally not the end of the project. Even if the purpose of the model is to increase knowledge of the data, the knowledge gained will need to be organized and presented

in a way that the customer can use it. It often involves applying "live" models within an organization's decision making processes, for example in real-time personalization of Web pages or repeated scoring of marketing databases. However, depending on the requirements, the deployment phase can be as simple as generating a report or as complex as implementing a repeatable data mining process across the enterprise. In many cases it is the customer, not the data analyst, who carries out the deployment steps. However, even if the analyst will not carry out the deployment effort it is important for the customer to understand up front what actions need to be carried out in order to actually make use of the created models.

In this work we adopted the CRISP- DM methodology because of its ability to iteratively move back and forth guaranteeing a mark of reliability on the deliverables.

CHAPTER FOUR: THE DATA DESCRIPTION

4.0. The Retail shop

The dataset used in this study was collected from one of the leading retail shops in Kisumu City of Nyanza province in Kenya. The goal of the chain store was to create a chain of superstores in strategic locations delivering quality, value, service, variety and lifestyle, with convenient opening hours giving everyone the opportunity to shop.

The store aimed at providing the largest variety and highest quality of local and international brands at reasonable and uniform prices in the region, with unmatched service provided by warm, friendly and always helpful staff, in a modern ambience for a pleasant shopping experience, providing ample secure parking and exciting customer rewards program for shoppers with loyalty card continuously enhancing lifestyles and delivering value.

The retail shop had three branches in Kisumu City alone and ten (10) other branches country wide with an expansion plan of five other branches countrywide. It sold a variety of products and in its database, the products span to over 200,000 different products each with its own unique code. The retail shop captured customer transactions at the point of sale. Each branch had more than five points of sales and each point of sale sent all the transaction to a central repository of the branch.

The shop introduced customer loyalty cards in 2006 to increase customer loyalty and to provide other avenues of marketing strategy in which each individual customer was reached for promotional and other advertisement activities.

These cards in addition to capturing the transactional history of the customers also contain the demographic information as contained in the national Identification card. These information include; Name, Sex, ID number, date of Birth and profession.

For purposes of this study, the demographic information for the loyalty card holders was not availed to us on grounds of protecting customer confidentiality.

The customer base of the shop was very large and each point of sale recorded an average of one thousand two hundred transactions per day.

The shop opened at 8:30 AM and closed at 8:30 PM and at its point of sale each transaction was attached with its timestamp. This expanded drastically the domains and dimensions for mining including sequential rule mining for different products or for particular cluster of customers,

4.1 The Dataset

The dataset used for this case study contained data for both loyal and non loyal customers; time of transactions and transactions performed, and item properties. The entire dataset covered the period of 12/02/2007- 11/12/2007.

There were 1,794,664 purchase transactions by both loyal and non-loyal customers. These transactions involved 220,222 unique itemset. It was important to note at this stage that the dataset given was only a small portion of the entire dataset picked from one point of sale. We removed single itemset transactions performed by both loyal and non-loyal customers, which reduced the number of purchase transactions to 976,660 by potentially 18,724 customers. We then proceeded to grouping together items with similar code prefix and or usage. We observed that certain brands such as Supa loaf and Sunblest had completely different codes but both were bread. This reduced the number of transactions to 440,370 with 502 unique items. We referred to this dataset as clean data shown in **table 4.2**. The average price of a purchased item was Shillings 37.81.

4.2.0 Dataset Processing

Before we applied the Data Mining techniques described in Section 3, we processed the clean dataset to remove incomplete data for the demographic, item, and purchase transaction attributes.

The original database had a number of tables some which had no data. Of interest from these tables were **BILLJTXN** table which captured all the transactions at the point of sale. The details of **BILLJTXN** table are described in **section 4.2.1**. Equally important was the **CardTransMaster** table shown in **table 4.0**. The table was linked directly to **BILLJTXN** master which helped to identify the particular products associated with respective card numbers.

This association was particularly useful in tracking individual customer's buying behavior and product affinity.

BRNC ID	CardNo	Car <1 type	mbr	traaDatct ate	tr.ty p	ret lot	bin_>ard	ROM	balpoi *u	l >r N<mr
18	209030865400001065	N	000002	11/23/2007 9 12:05 AM	A	877	9101823112007091024000102	8	570	PAD
18	108079812400001189	N	000003	11/23/2007 9 13 32 AM	A	606	910182311200709125000003	6	371	PAH
18	211034058400001060	N	000005	11/23/2007 9 21:27 AM	A	1023	9101823112007091832000005	10	764	PAD
18	110076297200001148	N	000015	11/23/2007 10 12 41 AM	A	250	9101823112007101228000015	2	3	PAD
18	104060215600001060	N	000022	11/23/2007 10 22:45 AM	A	1382	91018231120071021091000022	13	402	PAD
18	106053609200001164	N	000025	11/23/2007 10 32 45 AM	A	1130	9101823112007103204000025	11	923	PAD
18	202079355100001062	N	000026	11/23/2007 10 37:07 AM	A	163	9101823112007103652000026	1	124	PAJ
18	210066111400001184	N	000037	11/23/2007 11 21:29 AM	A	2878	9101823112007111711000037	21	1604	PAD
18	208064510200001184	N	000043	11/23/2007 11 30 19 AM	A	225	9101823112007112955000043	2	46	PAD
18	102068652100001182	N	000045	11/23/2007 11 35:09 AM	A	603	9101823112007113249000045	6	61	PAD
18	108054582100001181	N	000046	11/23/2007 11 40:22 AM	A	120	9101823112007113551000046	1	200	PAD
18	208074869400001186	N	000047	11/23/2007 11 43:22 AM	A	139	9101823112007114259000047	1	64	PAD
18	104052502000001062	N	000050	11/23/2007 12 38:42 PM	A	14322 5	9101823112007115548000050	1404	3141	PAD
18	209031045900001065	N	000054	11/23/2007 12 11:56 PM	A	68	9101823112007121140000054	0	1343	PAD

Table 4.0: Sample of Card master Transaction Table

4.2.1 The data Attributes

Each record in the data set contained information about the transaction ID (BILL_SCD), date of purchase (BILL_RUN-DATE), Product __code, Product_Description, Product-Price, Product quantity, Product amount, VAT, Product-cost, Card_Number and Points_accrued .

Attributes whose variables were constant such as VAT-EXMPT, TXN TYPE, PRDCT DSCNT were eliminated because constant variables were obvious conclusions thus needed not be predicted.

The most important item attributes were BILL_SCD, PRDCT_CODE, PRDCT_LNG DSCPTN, PRDCT_PRCE and PRDCT_COST. BILL_SCND helped to identify Transaction ID which was the identifier of a basket. PRDCT_CODE attribute helped in the sorting of item categories. Similar items tended to share a code range, for example ferries bread had product code range between **100032710** to **100032723** as shown in **Table 4.1** . PRDCT_LNG_pSCRPTN gave full description of the item and both PRDCTPRCE and PRDCT_COST helped in the generation of profitability matrix.

4.2.2 Data segmentation

The original database contained more than 210,222 different unique items. Such a large number of items could not be fitted in the allowable column range both in access and excel sheets which was fixed at 255. The Cleaned data resulted in 502 unique items which we partitioned into two based on the time frame. The first partition contained dataset between February 12th to June 22nd and the second partition contained data between 23rd of June to December 11th 2007.

BILLSCNCD	BIIX KUN DATE	PRDCT CODE ~	PKIX'I I\G ijM K PIN	PRDCT PR C	PRIMT QN IT	PRIXT AMN T
9101823112007090716000001	11/23/2007 9:07:16 AM	818950022	DAILY STANDARD NEWS PAPER	35	1	35
9101823112007091024000002	11/23/2007 9:10:24 AM	788320008	KXH AM. PURPOSE FIJOUR 2KG	117	3	351
9101823112007091024000002	11/23/2007 9:10:30 AM	707690011	KHNSALI IKG	17	1	17
9101823112007091024000002	11/23/2007 9:10:33 AM	747160091	MUMIAS BROWN SUGAR IKG	75	1	75
9101823112007091024000002	11/23/2007 9:10:36 AM	707640180	ROYCO MCHU/I MIX 200G JAR	63	1	63
9101823112007091024000002	11/23/2007 9:11:03 AM	778230071	M(X)NG WHOLH IKG	82		82
9101823112007091024000002	11/23/2007 9:11:23 AM	717730021	FRKSM FRI COOKING OH. LLT	119		119
9101823112007091024000002	11/23/2007 9:11:34 AM	788310003	HODARI MAI/L FU)UR 2KG	40	3	120
9101823112007091024000002	11/23/2007 9:11:46 AM	777460085	CAD DAIRY MILK FRUIT & NUT 40G	50	1	50
9101823112007091250000003	11/23/2007 9:12:50 AM	747160057	MUMIAS WHITE SUGAR IKG	75	4	300
9101823112007091250000003	11/23/2007 9:13:03 AM	778210115	MWLA-(IOLD PISNORI RICH IKG	79	2	158
9101823112007091250000003	11/23/2007 9:13:13 AM	747170063	FAMARI IFA IJOOSE 250CJ	85	1	85
9101823112007091250000003	11/23/2007 9:13:16 AM	707640180	ROYCO MCHUZI MIX 200G JAR	63	1	63

Table 4.1: Sample of Item transaction Table

BILL TXN	BILL_RUN_DATE	PRDCT CODE	PRDCT LNG DSCRPTN
9071812022007161323000002	2/12/2007 4:13:23 PM	9019050041	AIRTIME
9071812022007161323000002	2/12/2007 4:15:54 PM	9019050041	AIRTIME
9071815022007082811000009	2/14/2007 8:28:11 AM	7037100098	WATER
9071815022007110211000010	2/14/2007 11:02:11 AM	8018900952	SUZUKI
9071814022007185936000004	2/14/2007 6:59:36 PM	7017040059	JUICK
9071814022007185936000004	2/14/2007 6:59:38 PM	7158040060	BEEF
9071814022007190022000005	2/14/2007 7:00:24 PM	7047160057	SUGAR
9071814022007190046000006	2/14/2007 7:00:46 PM	7047160057	SUGAR
9071814022007190113000007	2/14/2007 7:01:13 PM	7107640122	SPICES
9071814022007190113000007	2/14/2007 7:01:16 PM	7047160057	SUGAR
90718170220070830300000%	2/16/2007 8:30:30 AM	7117730029	COOKING-FAT
90718170220070830300000%	2/16/2007 8:30:48 AM	7107690011	SALT
9071817022007083030000096	2/16/2007 8:30:59 AM	7047160057	SUGAR
9071817022007083258000097	2/16/2007 8:32:58 AM	7047160057	SUGAR
9071817022007083258000097	2/16/2007 8:33:00 AM	7087510369	CRISPS

Table 4.2: Shows Sample of Clean data

4.2J. Data formatting

CRISP-DM-1.0 explains formatting transformation to primarily syntactic modification made to the data that do not change its meaning but is required by modeling tool. WEKA can preprocess data in ARFF, CSV, C4.5 and binary formats. Basket analyzer uses Apriori algorithm developed by WEKA and it imports data in binary format. Table 4.3 shows data converted into binary format.

				<i>t</i>	mint juice rise			
0 0	0 0	0 0	0 0	0	0	0	0	0
0 1	0 0	0 0	0 0	0	0	0	0	0
1 0	1 1	1 0	0 0	0	0	0	0	0
0 0	0 0	0 1	0 0	0	0	0	0	0
1 0	0 1	0 0	0 1	1	0	0	0	0
1 0	1 1	0 0	0 1	0	0	0	0	0
0 0	0 0	0 0	0 0	0	1	1	1	0
1 0	0 0	0 0	0 0	0	0	1	0	1
1 0	0 1	0 0	0 0	0	0	0	0	0
1 0	0 0	0 0	0 0	0	0	0	1	0
1 0	0 1	0 0	0 0	0	0	0	0	0
0 0	0 0	0 0	0 0	0	0	0	0	0
0 0	0 0	0 0	0 0	0	0	0	0	0
0 0	0 0	0 0	0 0	0	0	0	1	0
0 0	0 0	0 0	0 0	0	0	0	0	0
0 0	0 0	0 0	0 0	0	0	0	0	0
1 0	0 1	0 0	0 0	0	0	0	0	0
0 0	0 0	0 0	0 0	0	0	0	0	0
0 0	0 1	0 0	0 0	0	0	0	0	1
0 0	0 0	0 0	0 0	0	0	0	0	0
1 0	0 0	0 0	0 0	0	0	0	0	0
1 0	0 1	0 0	0 0	0	0	0	0	0
0 0	0 0	0 0	0 0	0	0	0	0	0
1 0	0 0	0 0	0 0	0	0	0	0	0
0 0	0 0	0 0	0 0	0	0	0	0	0
0 0	0 0	0 0	0 0	0	0	0	0	0

Table 4.3: Sample of data converted into binary format

4 B a s k e t a n a l y s i s P l a t f o r m

For this project basket analyzer software that implemented Apriori algorithm was used. Apriori algorithm was a more a stable, complete and easy to use approach to rule generation as described in section 2. Furthermore basket analyzer could generate several levels of itemsets from the data set alongside rules. Itemsets were equally very important for driving promotions for cross sells and non moving products.

CHAPTER FIVE: RESULTS AND FINDINGS

5.0. Results and Findings

Transactional data was mined in two partitions over six months periods, to explore the midterm characteristics of transaction dynamics in terms of unique items, itemsets and general rules behavior over time.

5.1.0 Generated Rules

Initially when the statistical metrics were placed at 75% confidence and at 60% support, no rules were generated. But when the metrics were relaxed to 20% confidence and 10 % support, 621 rules were generated for the first partition of data part of which is shown in Table 5.0 and 923 rules generated for the second partition part of which is shown in Table 5.1

IF		THEN	Confidencii C/.)	Lin	Supportn (%)	No. of Transactions
salt,magarine	= >	sugar	100	2.25	8.33	6
salt,rice	==>	lotion	100	4.235	8.33	6
salt,rice	= - >	detergent	100	5.143	8.33	6
water,lotion	= >	soap	100	2.769	9.72	7
lotion,wheat-fluor	= >	soap	100	2.769	8.33	6
detergent,milk	==>	soap	100	2.769	9.72	7
detergent, wheat-fluor	= - >	soap	100	2.769	9.72	7
milk,ugali	- - >	soap	100	2.769	8.33	6
soap,ugali	= >	milk	100	2.483	8.33	6
mug,rice	- >	soap	100	2.769	9.72	7
lotion,cookin-fat	==>	detergent	100	5.143	8.33	6
detergent/ice	= >	lotion	100	4.235	9.72	7
lotion,rice		detergent	100	5.143	9.72	7
lotion,wheat-fluor	- - >	detergent	100	5.143	8.33	6
lotion,cookin-fat	- - >	rice	100	5.538	8.33	6
lotion,wheat-fluor	- >	rice	100	5.538	8.33	6
salt,detergent,wheat-floor	==>	soap	100	2.769	8.33	6
salt, soap, wheat- fluor	= >	detergent	100	5.143	8.33	6
salt,soap,detergent	- - >	wheat-floor	100	4.8	8.33	6
salt,dctergent,rice	- =>	lotion	100	4.235	8.33	6

Table 5.0. Top 21 rules for the first partition of data

It		Til UN	('Mndtin* (%)	Uft	Supports (%)	Transaction* or
CKNG_FAT,SOAP	= =>	SUGAR	88.82	3.09	10.44	151
SALT	= ->	SUGAR	81.25	2.826	8.98	130
SUGARJP	- ->	SOAP	77.22	3.639	8.43	122
CKNGFAT, SUGAR	- ->	SOAP	76.65	3.613	10.44	151
SOAP,TP		SUGAR	75.31	2.62	8.43	122
BEVERAGE		SUGAR	74.59	2.594	9.33	135
CKNGFAT		SUGAR	73.51	2.557	13.61	197
DETERGENT	- ->	SOAP	72.97	3.439	9.33	135
SOAP	= ->	SUGAR	72.64	2.527	15.41	223
MARGARINE	= ->	SUGAR	71.94	2.502	9.74	141
DETERGENT	= >	SUGAR	69.19	2.407	8.85	128
TP	= =>	SOAP	68.64	3.235	11.2	162
SOAP,SUGAR	- ->	CKNGFAT	67.71	3.656	10.44	151
TOOTHPASTE		SOAP	67.34	3.174	9.26	134
TP	= ->	SUGAR	66.95	2.329	10.92	158
MARGARINE	= >	SOAP	65.31	3.078	8.85	128
DETERGENT		TP	63.78	3.911	8.15	118
CKNGFAT	= >	SOAP	63.43	2.99	11.75	170
TOOTHPASTE	= >	SUGAR	62.81	2.185	8.64	125
TOOTHPASTE		TP	59.8	3.666	8.22	119
CKNGFAT	= >	SOAP,SUGAR	56.34	3.656	10.44	151
LOTION	- ->	SOAP	55.56	2.619	10.02	145
SOAP	- ->	CKNGFAT	55.37	2.99	11.75	170
SOAP,SUGAR	= ->	TP	54.71	3.354	8.43	122
SUGAR		SOAP	53.61	2.527	15.41	223
SOAP	- ->	TP	52.77	3.235	11.2	162
TP	- ->	SOAP,SUGAR	51.69	3.354	8.43	122
LOTION	= >	SUGAR	51.34	1.786	9.26	134
TP	= >	TOOTHPASTE	50.42	3.666	8.22	119

Table 5.1: Top 30 rules for the second partition of data

When the scores were reduced to 20% confidence and 10% support and arranged in descending order, the results of Table 5.0 and Table 5.1 were realized. Sample of the generated rules were:

Soap, lotion, detergent, rice •^wheat-flour

If itemset soap, lotion, detergent and rice is purchased then wheat flour is purchased 100% of the time and this happens in 8.33% of all shopping baskets.

Lotion, wheat-flour detergent

If itemset lotion, wheat-flour is purchased then detergent is purchased 100% of the time and this happens in 8.33% of all shopping baskets.

Water, lotion soap

jj-

itemset water, lotion is purchased then soap is purchased 100% of the time and this happens in 8.33% of all shopping baskets.

Cookingfat, soap sugar

If itemset cookingfat, soap is purchased then sugar is purchased 88.82% of the time and this happens in 10.44% of all shopping baskets.

Toothpaste Toilet-paper

If itemset tooth paste is purchased then toilet paper is purchased 59.8% of the time and this happens in 8.22% of all shopping baskets.

Water, lotion soap

If

itemset water, lotion is purchased then soap is purchased 100% of the time and this happens in 8.33% of all shopping baskets.

Mug, rice soap

If

itemset mug rice, lotion is purchased then soap is purchased 100% of the time and this happens in 8.33% of all shopping baskets.

From tables 5.0 and 5.1 it was observed that most rules generated were fairly consistent and at high scores both in the first and second partitions. Such rules include *Cookingfat, soap sugar; Mug, rice soap; Lotion, wheat-flour detergent*; operating at a minimum of 8.33% support. Again the tables 5.0 and 5.1 reveal that most of the rules generated moved around basic domestic items such as milk, diapers, lotion, soap, salt, sugar, rice, detergent, mug, cooking-fat, margarine, soda, toothbrush, cooking-flour, baking-flour, spaghetti. This meant that customers most frequently bought basic items. It must be noted that the data was collected in 2007; which year was an election year and was prone to a lot of political uncertainty. Part of the explanation for this was that most people were interested in stocking their food store

for fears of uncertainty; the other possible reason was that majority of the customers were lower working class whose core demand were to meet their basic needs.

It can therefore be concluded that the most interesting rules with high percentage scores revolved around the basic commodities.

5.1.2 Itemset Generated

At the same metric levels of 25% confidence and 10% support, 207 itemsets were generated for the first partition and 156 itemsets were generated for the second partition part of which partitions are observed in Table 5.2 and Table 5.3.

itemset	Size	Support (%)	No. of Transactions
sugar		43.06	31
milk		41.67	30
mug		38.89	28
bread		36.11	26
soap		36.11	26
cookin-fat		26.39	19
water		25	18
cake		23.61	17
lotion		23.61	17
juice		22.22	16
chew-gum		22.22	16
salt		19.44	14
beverage		19.44	14
wheat-floor		19.44	14
soap,milk		19.44	14
detergent		18.06	13
crisps		18.06	13
rice		18.06	13
bread,mug		18.06	13
spice		16.67	12
magarine		16.67	12
tissue-paper		16.67	12
toothpaste		16.67	12
beer		16.67	12
sugar,soap	2	16.67	12
bread,cake	2	16.67	12
soap,lotion	2	16.67	12

Table 5.2 ; Top 27 itemset for the first partition

Itemset	Size	Support (%)	No. of Transactions
SUGAR	1	28.75	416
MILK	1	21.42	310
SOAP	1	21.22	307
MUG	1	19.7	285
BREAD	1	18.52	268
CKNGFAT	1	18.52	268
JUICE	1	18.11	262
LOTION	1	18.04	261
CHWNG_GUM	1	17.69	256
TP	1	16.31	236
SOAP,SUGAR		15.41	223
TOOTHPASTE	.1 ,	13.75	199
CKNG_FAT,SUGAR		13.61	197
MARGARINE	1	13.55	196
DETERGENT	1	12.79	185
BEVERAGE	1	12.51	181
SODA	1	12.02	174
CKNG_FAT,SOAP		11.75	170
SOAP,TP		11.2	162
SALT	1	11.06	160
SUGAR,TP		10.92	158
SPICES	1	10.85	157
BKNG_FLR	1	10.71	155
UGALI	1	10.64	154
CKNG_FAT,SOAP,SUGAR		10.44	151
RICE	1	10.16	147
CAKE	1	10.09	146
LOTION,SOAP	2	10.02	145
MARGARINE,SUGAR	2	9.74	141
BEVERAGE,SUGAR	2	9.33	135
DETERGENT,SOAP	2	9.33	135
BREAD,MILK	2	9.26	134
LOTION3UGAR	2	9.26	134
SOAP,TOOTH PASTE	2	9.26	134
BREAD,SUGAR	2	9.12	132
SALT,SUGAR	2	8.98	130
DETERGENT,SUGAR	2	8.85	128

Table 5.3 : Top 30 itemsets for the second partition

The itemset generation was done in multiple levels. The level one itemset include sugar, mug, soap, water, milk, beverage, lotion margarine. The second level is a two itemset lattice such as sugar-lotion; sugar-bread; Cooking_fat-rice; rice-wheat_floun bread-mug; bread - milk; soap-Toilet_paper; soap-lotion; detergent-rice, detergent-milk, sugar-toothpaste among others. The three itemset lattice include salt-detergent-wheat_flour; salt-soap-wheat-flour; soap-detergent-rice; The details are shown in Tables 5.2 and 5.3. It can clearly be seen that the majority of the itemsets had support metric above 10%. This is a comparatively high support level.

In can therefore be concluded that the itemsets generated equally revolves around basic commodities. Sugar-milk, soap-bread, cooking fat-wheat flour are among the highly sort after itemsets in the shop at a minimum of 10% of all the shopping involving any item of the total transactions acted.

5.13 Item transaction performance

No. of Transactions

¹ No. of Transactions

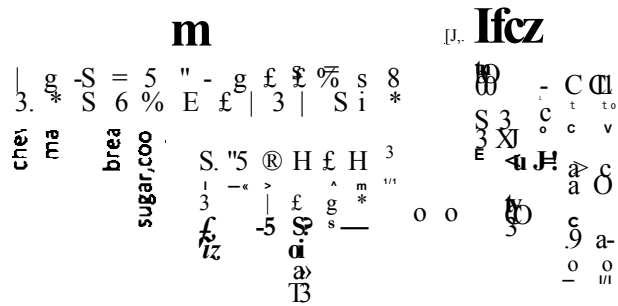


Figure 5.0: item performance graph top 19 item (set) (first partition)

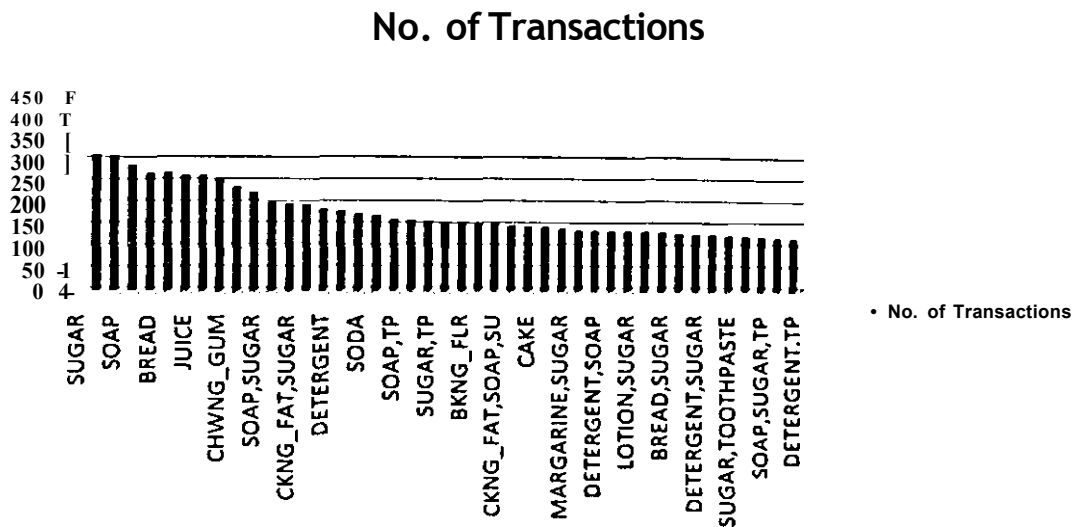


Figure 5.1: Item performance graph top 15 items (second Partition)

According to figures 5.0 and 5.2 the general item performance could be observed at a preset metrics of 20% confidence at 10% support. It was observed that the highly sort items both in the first and second part of the year were the following;

Sugar, Milk, Mug, Bread, Soap, Cooking Fat, Juice, Lotion, Toilet paper, Beverage, Wheat flour.

All these items transacted at above 15% support which is a high threshold with reference to total amount of transacted data.

The same tables revealed that *beef, bags, bed sheets, ink, ruler; tape, after-shaves* were among the least transacted itemset.

A large number of items performed averagely. This include water, cake, braids, pads, polish, scourers, veges, margarines, sodas, salt, etc.

5.2. Findings

5.2.1 The most transacted items that meet the minimum threshold

As explained in section 5.1 the most transacted items were; Sugar, Milk, Mug, Bread, Soap, Cooking Fat, Juice, Lotion, Toilet paper, Beverage, Wheat flour, soap, include water, cake, braids, pads, polish, scourers, veges, margarines, sodas, salt, etc.

5.2.2 The most important regularities in the dataset

Several rules were generated with high degree of confidence when the support metric was lowered to 10% and confidence metric set 30%. Indeed 602 rules were generated in the first partition and 912 rules generated for the second partition. The most important regularities were observed at above 50% confidence and above 8 % support. The regularities are shown in table 5.0 and table 5.1. Indeed table 5.0 has top rules with a 100 % confidence and most interesting rules in table 5.0 have 60% confidence above. This was almost a sure indication that the items affected were definitely to be traded with high degree of certainty and stocking and replenishing of the items was sure business move. Again it must be stated that rules generated are quite interesting.

5.2.3 Deployment of Generated regularities to business operations

The generated regularities are particularly useful in aiding business decisions processes. Such decisions involving item placement, promotions and advertisement, and buying can be supported by such regularities.

On item assortment and shop layout, the itemset generated in tables 5.2 and 5.3 can be used to aid in item assortment and shop layout. Items such as cooking fat and sugar, such lotions and soaps, such as toothpaste and toiletries can be assorted and placed in a manner that they are adjacent to each other for ease of reachability by the customers.

The first floor of the shop contained the general Fast Moving Consumer Goods (FMCG). The general shop layout plan is shown in figure 5.3

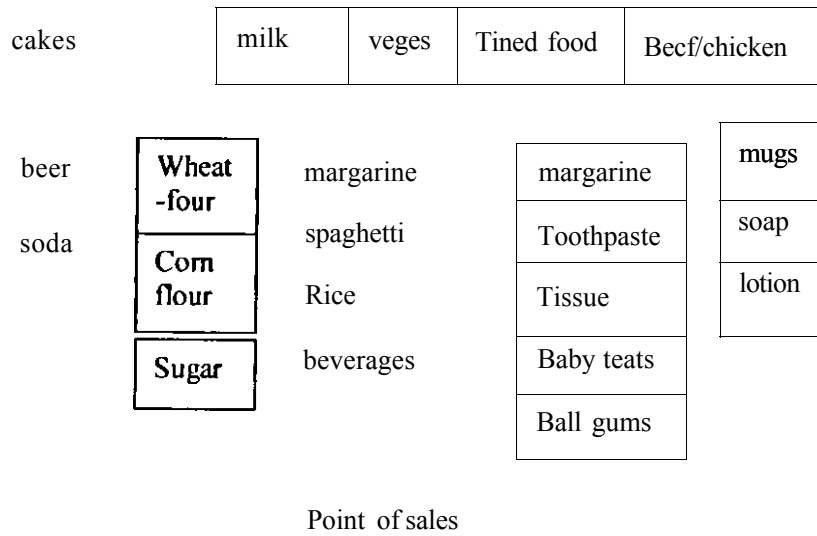


Figure 5.3: The Initial shop layout

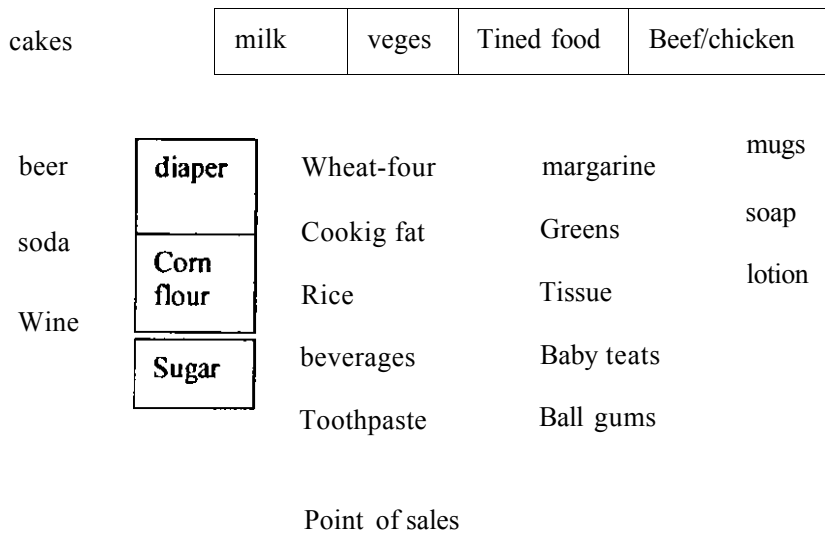


Figure 5.4: The proposed shop lay out

Figure 5.4 shows shop lay out with slight modifications such that goods that seldom sell are placed in such a strategic way as to be in between goods that were always likely to sell together.

In such a case whenever a customer was to buy for instance soap, such customers were highly likely to buy cooking fat, and as a such items like greens that rarely sold could be placed between soap and cooking fat with the hope of improving their identification hence could provoke impulse buying.

On promotions and advertisement, Promotional offers are designed in retail stores to promote the sale of goods. Discount and incentive oriented promotions are used as part of a broader promotional strategy which may include special events and events building promotions.

"Buy One, Get One Free" is a common type of sales promotions. This marketing technique is universally known in the marketing industry by the acronym BOGOF, and it is regarded as one of the most effective forms of special offers for goods (Pradip Kumar Bala, 2009). Originally, "buy one get one free" was a random, end-of-season or stock clearance method used by shops who were left with a large quantity of stock that they were looking to sell quickly. More recently it has become a popular, planned and considered marketing method. "Buy Two Get One Free" equally is a promotional strategy that is adapted to the prospects of making losses.

Among the least moving products explained in section 5.0., BOGOF strategy can be employed to influence their disposal. People seldom buy items like salt, tinned beef, bed sheets, ink etc. These products could be promoted along with the itemset generated in table tables 4.5 and 4.6.

5.2.4 Models generated from the regularities

Why do people buy what they buy? How do people go about making decisions and choices in the market place and how can sales promotions influence these decisions and choices? Pradip Kumar Bala (2009).

Linda Teunter (2002). When a consumer goes shopping, he or she implicitly, or explicitly, has to make four key decisions for each product category. Whether to buy in the category and, if so, where (which store), which brand, and what quantity. All four decisions may be influenced by consumer characteristics (e.g., income, family size, purchase frequency) and by the marketing environment e.g., the prices and promotion activities of the various brands and stores. Marketing mix variables can affect these four decisions to differing degrees. For example, price might have

a substantial influence on a consumer's brand choice decision but might not affect category purchase or timing decisions.

Consumers' actions or their reactions to marketing mix stimuli include increased awareness of, interest in, and desire for a product, in addition to actual purchase of the product.

Based on the above questions and arguments and based on the regularities in the data, stimulus-response model shown in Figure 5.0 was adopted. Stimuli are assumed to operate through or upon unknown consumer processes, which remain unmodeled intervening processes (Bagozzi 1986).

Many forces not under the direct control of firms also influence consumer behavior. These are labeled environmental factors and include economic conditions, social determinants, and cultural influences. Marketers have little or no control over these, but they do try to anticipate and forecast their effects.

STIMULUS BLACK BOX CONSUMER RESPONSE

Marketing Mix

variables

Product
Place
Promotion
Price

Environmental

factors

Economic
Conditions
Social forces
Cultural input

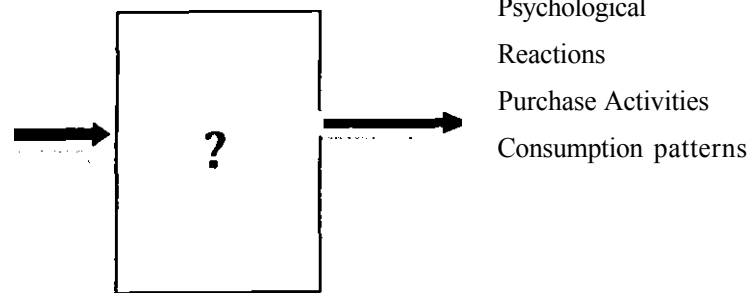


Figure 5.0: Stimulus-response model

By use of a stimulus-response approach, marketers can discover the reactions of consumers to different advertising appeals, package designs, and prices, to name a few stimuli. The stimulus-response model is an appealing model. First of all, it is simple, which makes it easy to understand and communicate to others. Second, it is a highly useful managerial tool and it has been found to work well in the past.

Such strategies as "Buy Two Get One Free", Product placement can easily be tested under this model.

5.3 Conclusions and Recommendations

The customer purchase patterns approach using associations rule mining technique, is an effective way of extracting the rules from the raw data and inferring the buying pattern among them. Indeed it is a sure means of supporting business decisions governing retail shop operations. A significant opportunity for improving sales was identified based on the itemset and rules generated. Certain items were transacted more frequently in both partitions of data set. This could easily inform the problem of stocking enough of the correct items and less of wrong items. In terms of prediction the rules and itemsets obtained show that different items and itemset have a high level of confidence and support matrix which then guarantee the certainty of their sales. The package like buy two packets of sugar get a packet of salt free can make a big business sense than promoting items which customers are not likely to buy.

From the association rules with sufficient coverage we can predict product the customer tends to buy along with the purchase of particular products,

5.4 Recommendations and future work

The work done so far leaves ample amount of space for future improvements and comparison. Promotions and layout based on these findings need to be studied to identify to what extent this impacts on the business performance of the shop.

The loyal customer card details need to be included in future mining work in order to understand fully the demographic characteristics and its relationship with the promotions and advertisement and the value of the baskets. And finally, the change in the floor layout needs to be investigated to show the extent to which it affects sales.

REFERENCES

- Agrawal, R., & Srikant, R. (1994). Fast algorithms for mining association rules.
- Agrawal, R., Imielinski, T., & Swami, A. (1993). Mining Association Between Sets of
- Ansari, A., Essegai, S. & Kohli, R. (2000). Internet Recommendation Systems. *Journal of Marketing Research*, 37(3), 363-375.
- Ardissono, L. & Goy, A. (2000). Tailoring the Interaction with Users in Web Stores.
- Bagozzi, R.P. (1986), *Principles of Marketing Management*, Chicago: Science Research Associates Inc.
- Botta, Marco, et al, 2004. Query Languages Supporting Descriptive Rule Mining: A Comparative Study. *Database Support for Data Mining Applications*. LNAI 2682, pp 24-51
- Christian Borgelt "An Implementation of FP- Growth Algorithm". In *First International Proceedings of OSDM on Open Source Data Mining*. PP. 1 -5, 2005
- Dzeroski, S., 2006. Towards a General Framework for Data Mining.. In Dzeroski, S and Struyf, J (Eds.), *Knowledge Discovery in Inductive Databases*. LNCS 47474. Springer-Verlag.
- F. Bonchi and B. Goethals. FP-Bonsai: the Art of Growing and Pruning Small FP-trees. *Proc. 8th Pacific-Asia Conference on Knowledge Discovery and Data Mining (PAKDD'04, Sydney, Australia)*, 155-160. Springer-Verlag, Heidelberg, Germany 2004
- Feng Zhang, Hui-You Chang, (2005). On a Hybrid Rule Based Recommender System. In *Proceedings of the 2005 The Fifth International Conference on Computer and Information Technology (CIT'05)*.
- Gregory Piatetsky-Shapiro, et al, (1996). From Data mining to knowledge discovery in databases. In *Proceedings of the International Conference on Very Large Data Bases*, 407- 419.
- International Conference on Management of Data*, 207-216.
- Items in Massive Database. In *International Proceedings of the ACM-SIGMOD*
- J. E. Tang, D. Y. Shee, T.I. Tang, "A Conceptual Model for Interactive Buyer-Supplier Relationship in Electronic Commerce," *International Journal of Information Management*, vol. 21, no. 1, pp. 49-68, Feb. 2001.

- J. E. Tang, D. Y. Shee, T.I. Tang, "A Conceptual Model for Interactive Buyer-Supplier Relationship in Electronic Commerce," *International Journal of Information Management*, vol. 21, no. 1, pp. 49-68, Feb. 2001.
- Larry Gordon, (2008): *Leading practices in market basket analysis*. Los Altos CA 94022.
- Linda H. Teunter (2002). *Analysis of Sales Promotion Effects on Household Purchase Behavior*. Rottadam.
- Mohammed J. Zaki and Karam Gouda. Fast vertical mining using diffsets. In Proc. of the ACM SIGKDD Int'l Conf. on Knowledge Discovery and Data Mining, pages 326-335, August 2003.
- Mohammed J. Zaki. Parallel and distributed association mining: A survey. *IEEE Concurrency*, 7(4): 14-25, 1999.
- N. Belkin and B. Croft, "Information Filtering and Information Retrieval," *Comm. ACM*, vol. 35, no. 12, pp. 29-37, 1992.
- Pradip Kumar Bala (2009). *A Data Mining Model for investigating the Impact of Promotion in Retailing*. In proceedings 2009 IEEE International Advance Computing Conference (IACC 2009) Patiala, India, 6-7 March 2009.
- R. Baeza-Yates and B. Ribeiro-Neto, *Modern Information Retrieval*. Addison-Wesley, 1999.
- Schafer, J. B. (2006). *The Application of Data-Mining to Recommender Systems*. In: *Encyclopaedia of Data Warehousing and Mining*, Wang, J. (Ed.). Information Science Publishing.
- Two Crows cooperation. (2005). *Introduction to Data Mining and Knowledge Discovery* Third Edition.
- U. Shardanand and P. Maes, "Social Information Filtering: Algorithms for Automating 'Word of Mouth'," *Proc. Conf. Human Factors in Computing Systems*, 1995. *User Modeling and User-Adapted Interaction* 10(4), 251-303.
- Ying Liu, Wei-keng Liao, and Alok Choudhary (2006). *Parallel Data Mining Algorithms for Association Rules and Clustering*. CRC Press LLC.
- Ying Liu, Wei-keng Liao, and Alok Choudhary. Design and evaluation of a parallel HOP clustering algorithm for cosmological simulation. In Proc. of the 17th Int'l Parallel and Distributed Processing Symposium, April 2003.

<http://www.sas.com/offices/europe/uk/tecml>