The Pennsylvania State University
The Graduate School
Capital College

Applying Data Mining to Demand Forecasting and Product Allocations

A Master's Paper in
Computer Science
By
Bhavin Parikh

Submitted in Partial Fulfillment
of the Requirements
for the degree of

Master of Science

November 2003

Acknowledgment

I would like to take this opportunity to thank my project advisor, Dr. Q. Ding. Her knowledge and experience helped me to finish this project successfully. Her guidance and encouragement throughout this project are deeply appreciated. I am grateful to the committee members: Dr. T. Bui, Dr. P. Naumov, and Dr. L. Null for reviewing this paper.

I would like to gratefully acknowledge Spectra Marketing for their kind permission to use their data and information for this project. My hearty thanks to Dr. C. Getchell, SVP of Research Group, for his guidance and comments. I would like to thank Dr. C. Getchell, Richard Mahon and Roger Meimban (members of Research Group) for providing data and information for this project. I would like to thank Leo Tubay, Director of Development Group, for helping me to setup SAS Exceed Software. I also would like to thank Chandu Patel, Oracle DBA in Development Group, for helping me during Oracle 9i installation and to setup sqlnet.ora and tnsnames.ora in SAS UNIX Servers. I would like to extend my hearty thanks to Dennis Klos, SVP of Development Group, for providing me hardware and software resources as well as encouragement for this project.

Finally I would like to thank Hemali, my wife, for providing me constant encouragement during the project.

Abstract

This paper presents data mining based solution for demand forecasting and product allocations applications. Accurate demand forecasting remains difficult and challenging in today's competitive and dynamic business environment, but even a little improvement in demand prediction may result in significant saving for retailers and manufactures. This project aims to improve the accuracy of demand forecasting by implementing multi-relational data mining process on store, product and shopper's data sets. This paper proposes two data mining models, Pure Classification model and Hybrid Clustering Classification model. Pure Classification model uses k-Nearest Neighbor Classification technique, and Hybrid Clustering Classification first uses k-Mean Mode Clustering to define clusters and then k-Nearest Neighbor classification to find k most similar objects. Hybrid Clustering Classification model introduces a new concept of combining existing data mining techniques on the multi-relation data sets. Experimental results show that Hybrid Clustering Classification is promising in practical situations. The data mining application is implemented in Java 2 version 1.4.1. The user interface is created with the help of Java Swing classes. Oracle 9i Release 9.2.0.1.0 is used as the application database system. The application is designed to be extendable at every level of the system.

Table of Contents

List of Figures

List of Tables

1 Introduction

Demand forecasting and product allocations are key business functions for retailers and manufacturers. Demand forecasting helps retailers to identify under-achieving stores where the potential sales of a product appear to exceed the actual sales of the product. Product allocations assist manufacturers to allocate products to stores and accounts. The majority of retailers still conduct demand forecasting and planning using outdated, often homegrown systems that lack forecasting algorithms and analytical tools [1]. Product demand in a store can depend upon various store attributes, such as size related factors and information about different departments, and shopper attributers such as income, age, education etc., and the product attributes such as brand name. Some other factors such as competition between stores and population density can also affect actual sales of a product in a store. Extensive information related to stores, products and shopper relations are being collected for retailer and manufacturer industries. Without using data mining techniques and models, not only is it difficult to design system with better demand forecasts but it is also not efficient to handle large data set with many relations and attributes.

The paper introduces a new sophisticated demand forecasting and product allocation application based on various data mining techniques. The business objective is to predict potential sales of products considering various store, product and shopper's demographic attributes. This objective drives the entire data mining process. Figure 1.1 shows the complete data mining process used to achieve the business objective. The next step involves identifying the required external data sources. The Loading and Integration process loads data from the external sources and combines them in one database instance. Then the data preparation steps are performed to ensure the quality of the selected data sets. The proposed data mining models, Pure Classification (PC) and Hybrid Clustering Classification (HCC), implement different mining techniques on the processed data. Both models support multi-relation data mining with efficient search and indexing techniques, combining with existing data mining techniques. The Hybrid Clustering Classification model introduces a new concept of combining existing data mining techniques on the multi-relation data set. Finally, the Evaluation step analyzes test experiments of each

model, and then compares both models in terms of accuracy and performance. Experimental results show that the Hybrid Clustering Classification model provides better accuracy with significant efficiency improvement over the Pure Classification model.

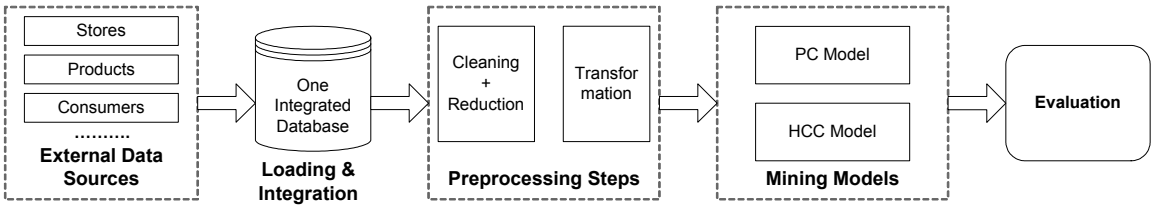

Figure 1: Data Mining Process Overview

The concept of combining existing data mining techniques to produce new hybrid methods is relatively new. Dan Steinberg and Scott Cardell [3] introduced two new hybrid models, CART-Logit model and CART-Neural Net model, by combining decision tree methods with neural network techniques and logistic regression. Their test experiments on real world examples and artificial data sets show that the hybrid model is always better than either component alone. Jin Jia and Keiichi Abe [4] proposed a new algorithm for generating a tree classifier by using instance-based learning and clustering method. The new algorithm utilizes a clustering method as preprocessing and a k-Nearest Neighbor classification as a complementary classification applied to each cluster [4].

The paper is organized as follows. Section 2, Data Set Overview and Data Preparation, describes data set and preprocessing steps to prepare the data for data mining models. Sections 3 and 4 introduce Pure Classification and Hybrid Clustering Classification models with evaluation, respectively. Section 5 reviews the application from user and programmer's perspectives. The paper ends with some conclusions and future enhancements, in Section 6.

2 Data Set Overview and Data Preparation

Data Set Overview provides an understanding of each key relation and its attributes. Data Preparation discusses all preprocessing steps to prepare the data for mining models. This section also studies the evaluation strategies used to evaluate PC and HCC models.

2.1 Data Set Overview

The application data set contains the following key information: the stores and their facts, products and their attributes, and shoppers and their demographics attributes. Additionally, it also contains actual sales data for products and stores carrying the products. Actual weekly sales data for each product and store carrying the product are stored in STORE_PRODUCT_SALES table. Spectra Marketing provided all necessary data set for this project. Figure 2 shows all key tables along with the number of records in each table. Appendix A shows the complete database schema of this application. Database constraints are defined to ensure efficiency during reading data from multiple tables.

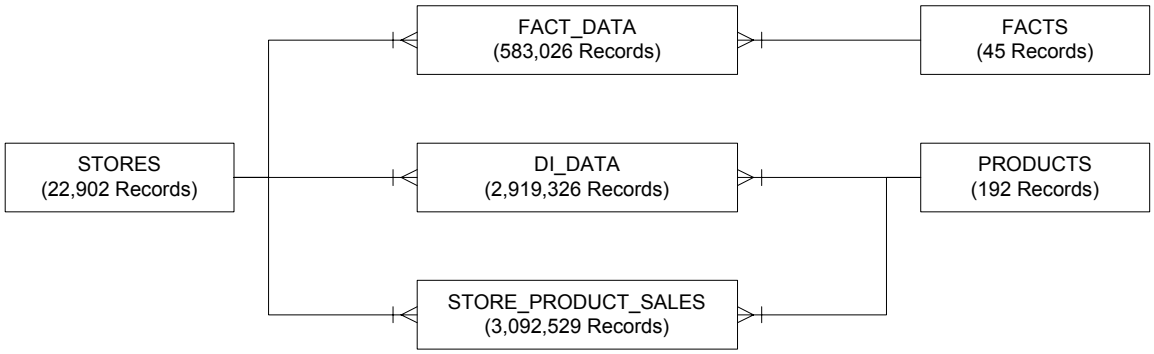

Figure 2: Application Core Schema Diagram

Store data set: The three tables, STORES, FACTS, and FACT_DATA, in Figure 2 represent the store data set. The store data set contains 22902 distinct stores that are representative samples of all stores in the USA. Each store has 55 different facts including the location information for each store. The location attributes basically represent longitude, latitude, and address information. All the facts but location attributes

are divided into three data types. Many attributes are listed in the domain analysis section.

- Numerical attributes: Store size related facts are examples of numerical attributes. Selling area in square foot, the number of full-time employees, and the number of front-end checkouts are the numerical attributes. There are a total of 11 numerical store attributes in the store data set.
- Categorical attributes: The set contains 8 categorical attributes. The store's format and account information are important categorical attributes.
- Boolean attributes: The store dataset contains information about the presence of departments and in-store services such as the presence of a bakery and the presence of prepared food as Boolean attributes. There are a total of 26 Boolean attributes in the store dataset.

Product data set: The product data set contains information about 192 different products of various product categories. The PRODUCTS table represents all 192 unique products. Product brand and product category are the two categorical attributes in the product data set.

Shopper's demography data set: The shopper demographical attributes, such as age and the presence of children, represent the customers who shop for the products in stores. Unlike the stores and the products data set, the shopper's demography data set is related both to stores and to products. The data set contains 9 different demographical attributes, and all of those are numerical attributes. The DI_DATA table represents the shopper's demography information for each product and the stores carrying the product.

2.2 Data Preparation

An important portion of any data mining project is to prepare the data for mining models. The Data Preparation process starts after identifying the business objective. Data Loading chooses and loads the right data set from external data sources, and Data Integration combines them into one database instance. Domain Analysis helps to understand details about the problem and the data sets, and an understanding of domains

helps to clean data and reduce dimensions. Data Cleaning and Reduction briefly touches some basic methods to work with inconsistent and incomplete data. Finally Data Transformation prepares the data for the classifier model.

2.2.1 Data Loading and Integration

We identified three key data sources - stores, products, and shopper's demography of the year 2002 - for this project. All these data files were available in various formats and from different locations. The products data set, shopper's demography, sales data, and most of the stores data were available in different SAS files, and the remaining store's information was available in Text/Excel files. After identifying the correct data set, the data were extracted from the external data sources according to their formats and loaded into the Oracle 9i instance. SQL*Loader utility was used to read SAS data files and Text/Excel files as shown in Figure 3.

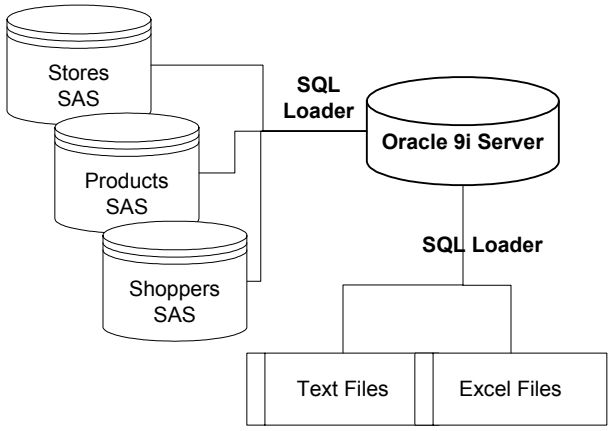

Figure 3: Integration from External Data Sources

2.2.2 Domain Analysis

Domain analysis plays a very important role in any data mining project. The main purpose of this step in the project was to understand details about the problem and the data set. The domain analysis started immediately after the business problem identification, and continued throughout this project. The first step was to work with a domain expert group to identify the relevant data for this project. We studied the external store, product, and demography data sources to understand each attribute. Basic statistical analysis was performed to compute range, mean, standard deviation, and population

distribution for each numerical attribute, and population distribution for categorical and Boolean attributes. Figure 4 shows an example of a population distribution diagram that represents a store's population for each density value. Density represents the population intensity near a store, where low value represents sparse area and high value represents dense area. Tables 1 and 2 list store numerical attributes and categorical/boolean attributes respectively. Appendix B provides a list of all attributes used in this application.

Store Numerical Fact	Pop Count	Distinct Values	Min	Max	Mean	SD
Population density	22,902	224	0	223	112	65
Total households	22,902	8,706	223	41,535	6,418	4,010
Stores competition	22,902	1,569	10	2,840	892	505
All comodity value	22,902	66	6,000	1,825,000	556,515	450,438
# of checkouts	21,967	42	1	73	23	16
# of end-aisle for contract	8,430	71	1	77	36	21
Full time employees	21,967	299	3	865	192	169
Selling area in sq. ft.	21,967	170	1,000	196,000	85,771	49,722
Average weekly volume	21,967	63	6,000	1,250,000	494,667	403,873
Grocery selling area	8,599	82	3,000	165,000	47,512	33,006
Grocery weekly volume	8,599	41	34,000	1,000,000	461,805	294,972

Table 1: List of Store Numerical Attributes

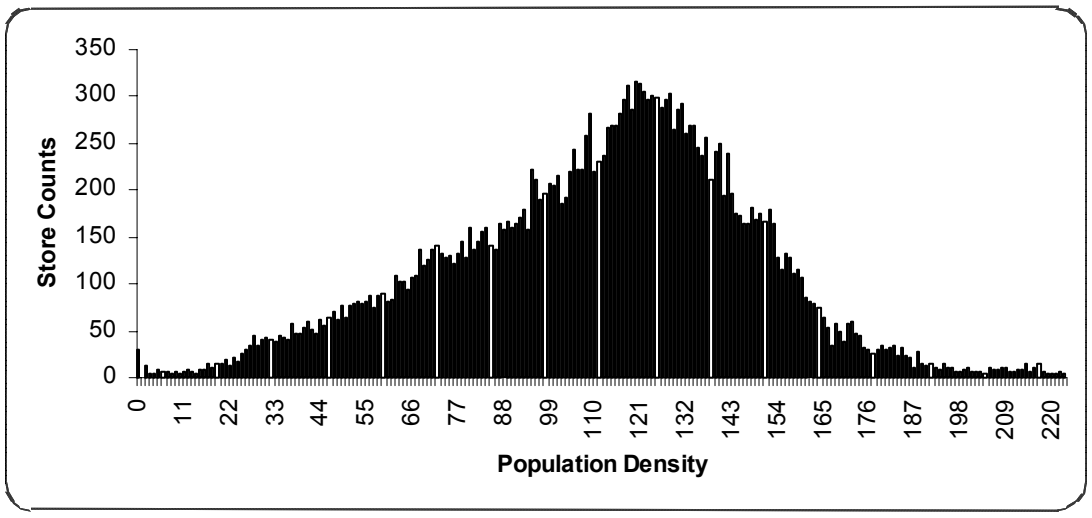

Figure 4: Store Populations over Density

Store Categorical/Boolean Fact	Pop Count	Distinct Values
Store format	22,902	8
Account of store	22,902	129
Merchandising type	14,694	6
Firm size of store	22,902	10
Type of store	22,902	7
Presence of an in-store bakery	9,086	
Presence of a full service bank	9,016	
Presence of beer and/or wine sections	21,960	
Presence of a bulk food section	8,952	
Presence of coffee grinders	8,963	
Presence of an in-store service-deli	9,084	
Presence of a fast food franchise	9,001	
Presence of an in-store seafood counter	9,008	
Presence of an in-store floral center	9,014	
Carries a full line of general merchandise	22,902	
Presence of gasoline	1,557	
Suepermarket emphasizing gourmet products and brands	8,809	
supermarket appealing to hispanic customers	8,772	
Limited assortment (<1500 items)	22,902	
Store sells lottery tickets	8,954	
Presence of an in-store restaurant	9,006	
Presence of an in-store salad bar	8,952	
Presence of a seasonal aisle	8,963	
Presence of an in-store cosmetic counter	9,004	
Presence of a specialty cheese center	8,986	
Presence of an in-store service meat/butcher	9,002	
High volume cigeratte sales	228	
Presence of an in-store video rental department	8,992	
Store sells videos	8,955	
Indicating store remodeled since last survey	8,564	
Presence of prepared food section	8,962	

Table 2: List of Store Categorical and Boolean Attributes

The domain analysis process continued even after the data preparation step. Understanding the importance of different attributes helped us to implement attribute weighting in data mining models. Some attributes may be given more importance by assigning larger weights to them.

2.2.3 Data Cleaning and Data Reduction

This step is necessary to ensure the quality of the selected data, and to improve the accuracy of the mining models. The external data sources are quite clean, but still they

are not consistent enough to apply directly to the data mining models. One of the earlier issues was that each original data set was in a different format and isolated from each other, so they did not have the same structure, naming conventions, or consistency among them. The following efforts were taken to remove the incomplete and inconsistent data.

1. The first issue was to handle inconsistent data types. Some attributes in one external data source were VARCHAR2 or STRING, and the same attributes in other external data sources were NUMBER. So we converted data types of the attributes to the most suitable ones.
2. Some important attributes and/or actual values (class label) were missing in store data sets. For example, a couple of stores were removed because they were missing many facts. Some store and product records were removed because there were missing important relationships.
3. Some records were found to be duplicates, and thus they were deleted from the database.
4. We identified many outliers in one external data source due to the mistakes in generating the data set from other information, and all of them were corrected by regenerating them from original data sources.
5. Some attributes that did not have a large enough population or had just one value for all records were deleted from the system.

In many cases, we ignored missing and incomplete records. But sometimes it is more useful for a data mining model to replace the missing values with the most suitable values. The next section discusses the data transformation, which is the last step in data preprocessing methods.

2.2.4 Data Transformation

The purpose of this step is to transform the selected data into appropriate forms to produce the analytical data model. Jiawei Han and Micheline Kamber [2] suggest various transformation methods based on selection of data mining models and input data sets [2]. We applied the log transformation first, and then normalized all the numerical attributes

in the range of [0, 1] using min max normalizations in our experiments. Both techniques are described briefly as following:

Min Max Normalization

This method transforms the data in the same range of values such as [Min, Max]. Otherwise, the attributes with larger ranges will be treated as more important. Normalizing the all-numerical attributes will also help speed up the learning process.

Min Max Normalization maps a value v of attribute A to v' in the range $[new\_\max_A, new\_\min_A]$ by computing [2]:

$$v' = \frac{v - \min_A}{\max_A - \min_A}(new\_\max_A - new\_\min_A) + new\_\min_A$$

Log Transformation

Log transformation helps to correct problems with skewed data and unequal variations. Many attributes of the data set had no upper bound and the largest values in the different attributes were many times larger than the smallest values of the same attributes. We use the natural logarithm (base e) for the log transformation method. The following log function transforms v to v'.

$$v' = \log_e(v)$$

2.3 Evaluation Strategy

It is necessary to decide what evaluation strategies are required to analyze the PC and HCC models. Sometimes it is difficult to measure numeric prediction as the errors are not just present or absent but they come in different sizes [8]. This section describes two types of evaluation measures that are used in this application to measure the accuracy of the data mining models. The selected evaluation measures are simple and not very computation-intensive.

Error Measures: The application uses Root Relative Squared Error and Relative Absolute Error measures. Root Relative Squared Error (RRSE) is the most suitable error

measure for this application. This error measure emphasizes the relative error differences rather than absolute error values. For example, 10% error is equally important whether it is an error of 50 in a prediction of 500 or an error of 0.5 in a prediction of 5. Relative Absolute Error simply adds up individual errors without taking into account their signs. Both error measures are defined as follows [8]:

$$\text{Root Relative Squared Error} = \sqrt{\frac{(p_1 - a_1)^2 + \cdots + (p_n - a_n)^2}{(a_1 - \bar{a})^2 + \cdots + (a_n - \bar{a})^2}}, \textit{where} \quad \bar{a} = \frac{1}{n}\sum_{i=1}^{n} a_i$$

$$\text{Relative Absolute Error} = \frac{|p_1 - a_1| + \cdots + |p_n - a_n|}{|a_1 - \bar{a}| + \cdots + |a_n - \bar{a}|},$$

where $a_1, \cdots, a_n$ are actual values and $p_1, \cdots, p_n$ are predicted values.

Prediction Accuracy: This evaluation method counts the percentage of test samples for which the actual and modeled results are within n percentage. The following ranges are used in the evaluation of the mining models.

- Prediction within 5 percentage difference between actual values and predicted values
- Prediction within 6 percentage to 10 percentage difference between actual values and predicted values
- Prediction within 11 percentage to 25 percentage difference between actual values and predicted values
- Prediction within 26 percentage to 50 percentage difference between actual values and predicted values
- Prediction with more than 50 percentage difference between actual values and predicted values

The Data Preparation section covers all the steps before designing data mining models. Without sufficient and efficient data preprocessing techniques, data mining models can never find interesting results or patterns. After the data preparation step, the preprocessed data is now ready to be used by the analytical data mining models. We will introduce the Pure Classification (PC) model in the next section.

3 Pure Classification (PC) Model

The Pure Classification (PC) model uses the k-Nearest Neighbor classification technique with weighted distance measure and attribute weighting. The k-Nearest Neighbor classification method is an instance-based learning method that reads training samples by joining among multiple relations and stores them in memory, and then classifies test samples directly from the training samples instead of learning any rules. k-Nearest Neighbor classifiers are among the simplest and yet most efficient classification rules and are widely used in practice [6]. Many research studies have shown that the predictive accuracy of k-NN algorithms is comparable to that of decision trees, rule learning systems, and neural network learning algorithms on many practical tasks. In addition, it has been shown that the probability of error of Nearest Neighbor classification is bounded above by twice the optimal Bayes probability of error [5]. The k-NN classifier provides good predictive accuracy with large sample data at the cost of high computation and storage requirements. This section discusses distance measures that are used in both data mining models to find the distance between objects. The k-NN classifier workflow and its evaluation are described in detail later in this section.

3.1 Distance Measures

There are many different ways a distance function can be defined, and it is hard to find the most suitable one especially when there are attributes other than numerical ones. The PC model (and the HCC model too) does not depend upon any specific distance function. In other words, the application can work with any distance function such as the Euclidean or Manhattan distance function. Euclidean distance is the most commonly used distance function for the k Nearest Neighbor classifier and k Mean Clustering classifier, and thus we decided to use the modified version of Euclidean distance as a default distance measure. The weighted Euclidean distance function $d(i, j)$ between instances i and j is defined as follows:

$$d(i,j) = \sqrt{\sum_{p=1}^{n} d_{ij}^{(p)}} \ ,$$

where $d(i, j)$ is computed based on the attribute data type p:

If p is a numerical attribute: $d_{ij}^{(p)} = w^{(p)}(d_i^{(p)} - d_j^{(p)})^2$.

If p is a boolean or categorical attribute: $d_{ij}^{(p)} = w^{(p)}$ if $d_i^{(p)} = d_j^{(p)}$; otherwise $d_{ij}^{(p)} = 0$.

3.2 Pure Classification Algorithm

This section describes the k-Nearest Neighbor technique with the weighted distance measure. Building the classifier, and classify unknown samples from the classifier are the two important steps in the following algorithm.

Algorithm: k-Nearest Neighbor Model

Inputs: integer k - number of nearest neighbors, training set, testing set, attributes with weights, distance function

Output: predicted values of the test samples, evaluation measures

Method:

1. create a separate instance for each training record with its actual value;
2. for each test sample s
    a. find k nearest neighbors from the training set by computing distance between each instance of the training set and the test sample s;
    b. compute the average value of the actual values of k nearest neighbors, and assign as predicted value for the test sample s;
3. compute evaluation measures from a set of predicted values and actual values of the test samples;

The k-NN classifier is the simplest classification prediction algorithm, but high computational costs and storage requirements make the classifier unpractical in production environments with multiple relational data sets. Many variations of the k-NN classifier are available to reduce memory and time computations [5].

3.3 Test Experiments and its Evaluation

The sample data set is divided randomly into a training data set and a testing data set. The training data set is used to build a classifier model, and the testing data set is used to evaluate the classifier model. Currently 70% of the total sample data set is selected randomly for the training, and the remaining 30% is selected for the testing. The sampling technique is loosely coupled with the application, and so it can be changed without any affect on the application and models. To evaluate the k-NN classifier model, 500 random test samples were selected from the testing data set for all the test experiments of this model.

The k-Nearest Neighbor classifier is tested with different sizes of k, the integer representing the number of nearest neighbors. Figure 5 shows how the different choices for k affected the predictive accuracy. Table 2 provides the results of each test in detail. The Root Relative Square Error (RRSE) is found to be between 34.36% to 42.39 for different k. The error rate is highest with the value of k = 5, and lowest with the value of k = 11. After looking at the nearest neighbors of some samples, our educated guess is that the actual values of the first 3 were quite close to the actual value of the unknown samples, but in some cases the actual value of the 4^{th} or 5^{th} nearest neighbor is drastically different than those of the first 3 or 4 nearest neighbors. The effect of the outliers decreases with increasing numbers of nearest neighbors. One outlier effect among 8 to 10 nearest neighbors is very minimal. So the Nearest Neighbor with either 3 or more than 5 provides better results than the Nearest Neighbor with 5. Our suggestion is to try weighted distance average rather than standard average calculation for predicting potential value for unknown samples.

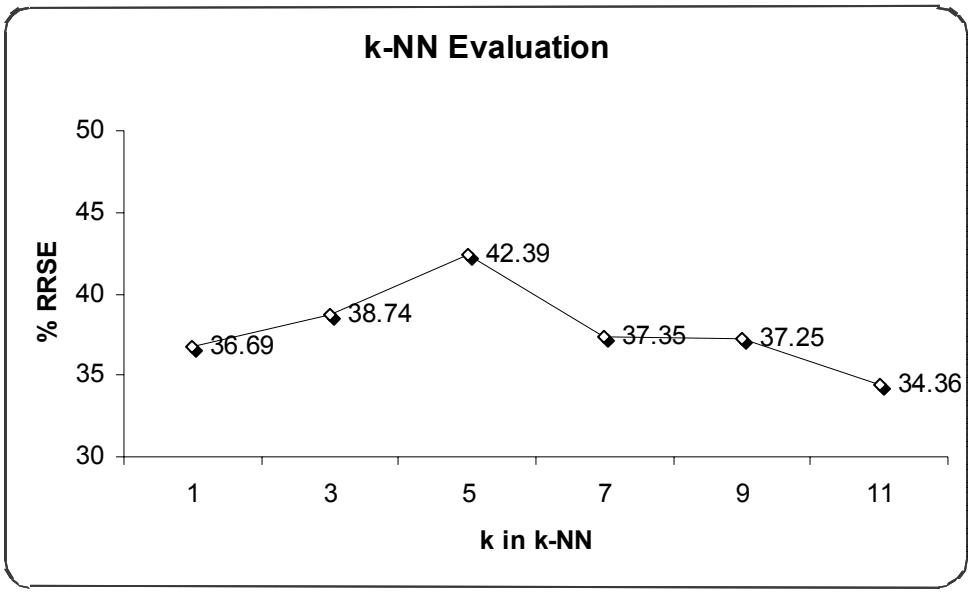

Figure 5: k-NN Evaluation for Different k

k-NN Evaluation Matrix No. of Test Samples = 500 No. of Attributes = 20	k = 1	k = 3	k = 5	k = 7	k = 9	k=11
Root Relative Square Error	36.69%	38.74%	42.39%	37.35%	37.25%	34.36%
Relative Absolute Error	0.32224	0.28034	0.27934	0.28119	0.27037	0.25716
Predicted within 5% diff.	10%	14%	13%	11%	14%	14%
Predicted within 6 to 10% diff.	9%	8%	9%	10%	7%	8%
Predicted within 11 to 25% diff.	26%	26%	27%	28%	27%	29%
Predicted within 26 to 50% diff.	28%	28%	26%	28%	26%	23%
Predicted with > 50% diff.	28%	23%	26%	24%	25%	26%

Table 3: k-NN Evaluation Matrix for Different k

The k-Nearest Neighbor predictor becomes a true optimal predictor with a very large training set and with a higher k. However, the model memory and time performance starts to degrade as the value of k increases. In the next section, we propose a more effective solution by combining the k-Nearest Neighbor technique with the Clustering algorithm.

4 Hybrid Clustering Classification (HCC) Model

Expensive time-computations and large memory requirements combined with a training set with more than 30 attributes makes the k-Nearest Neighbor difficult to use in practical situations. Many studies have shown that the k-Nearest Neighbor classification method with effective search optimization cannot handle more than 30 attributes [5]. The Clustering technique seems to be a good solution particularly for this application where more than half of the attributes of the data set are related to stores only. The purpose is to use the Clustering technique to divide the store training set into small groups based on their similarity, and then apply the classification technique to the small groups rather than the complete store training data set. The proposed Hybrid Clustering Classification (HCC) model works effectively on the multiple relational data sets by combining the k-Mean Mode Clustering technique and the k-Nearest Neighbor classifier. The k-Mean Mode Clustering is a variation of k-Means algorithm to cluster data with mixed numeric and categorical values. The HCC model workflow describes how both techniques are used together on store, product, and shopper relations. This section includes the test experiment results of the HCC model and a comparison of performance between the HCC model and the PC model.

4.1 Hybrid Clustering Classification Algorithm

This section outlines the algorithm of the hybrid model to understand how the clustering and classification techniques work together. The HCC model needs three types of inputs: the first is specific to the clustering technique, the second is specific to the classification technique, and the third are common inputs applicable to both techniques. The same distance measures, discussed in Section 3.1, are applied to this model.

The following algorithm first uses the k-Mean Mode Clustering to divide the training data set into k clusters as described in step 2, and then k-Nearest Neighbor Classification to classify the unknown test samples from the predefined k clusters as mentioned in step 4. The k-Mean Mode Clustering uses the store attributes to define clusters, and the k-Nearest Neighbor Classification uses the remaining attributes besides the store attributes.

Algorithm: k-Mean Mode Clustering with k-Nearest Neighbor Classification Model

Inputs:

Common Inputs: training set, testing set, distance function

Clustering Inputs: integer p representing the number of clusters, clustering attributes with weights

Classification Inputs: integer q representing the number of nearest neighbors, integer n representing the number of included clusters, classification attributes with weights

Output: predicted values of the test samples, evaluation measures

Method:

1. arbitrarily choose p instances as the initial cluster centers;
2. repeat
    a. assign each instance to the cluster which has the closest mean and mode;
    b. calculate new mean and mode for each cluster, i.e., compute the mean and mode values of the instances based on the attribute type for each cluster;

   until there is no change in any p clusters;
3. define the final set of p clusters with centroid points;
4. for each test sample s
    a. find n closest clusters by computing the distance between the centroid of each cluster and the test sample s;
    b. store all instances of n included clusters, the closest clusters of Step 4.a, as classifier set;
    c. find q nearest neighbors from the classifier set by computing the distance between each instance of the classifier set and the test sample s;
    d. compute the average value of the actual values of q nearest neighbors, and assign as predicted value for the test sample s;
5. compute error rate and prediction accuracy measures from the set of predicted values and the actual values of the test samples;

Some details are ignored, such as loading training/test data into memory, to avoid complexity in the above algorithm. Resulting cluster definitions can also be saved in the database for later use. Basically the hybrid model divides the difficult classification task into two simpler tasks.

4.2 Test Experiments and Their Evaluation

The HCC model is evaluated with the same 500 test samples for all test experiments as it is used for the PC model. There are three key input parameters for the HCC model: the number of clusters, the number of included clusters and the number of nearest neighbors.

Test Case 1: Experiment with the number of clusters

The overall clustering quality is measured by computing within-cluster variance (WC) and between-cluster variance (BC) [7]. Within-cluster variance is the sum of squares of distances from each point to the center of the cluster to which it belongs, while between-cluster variance is the sum of squares of the distance between cluster centers [7]. Figure 6 shows the between-cluster/within-cluster variance ratio vs. the number of clusters. The experiments show that the between-cluster to within-cluster ratio increases as the number of clusters increases. The ratio factor indicates how compact the clusters are and how far they are from each other.

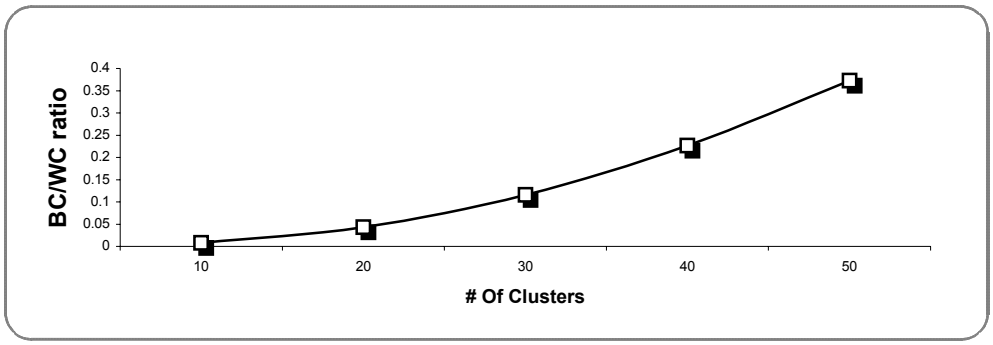

Figure 6: Experiments with the Number of Clusters

Test Case 2: Experiment with the number of included clusters

The number of included clusters 1, 2, and 3 are selected in this experiment. The HCC model predicts the test samples from the instances of included clusters rather than from all instances. The HCC model is tested for number of clusters from 10 to 50 for each included clusters 1, 2 and 3. For the number of included clusters 1, the model uses instances of only 1 nearest cluster from unknown test sample. Similarly, for the number of included clusters 2 and 3, the model uses instances of 2 and 3 nearest clusters, respectively. The first series, second series, and third series in the figure 7 show the results of included clusters 1, included clusters 2, and included clusters 3, respectively. It has observed that the error rate percentage with included clusters 2 or 3 is more stable than the error rate percentage with included clusters 1.

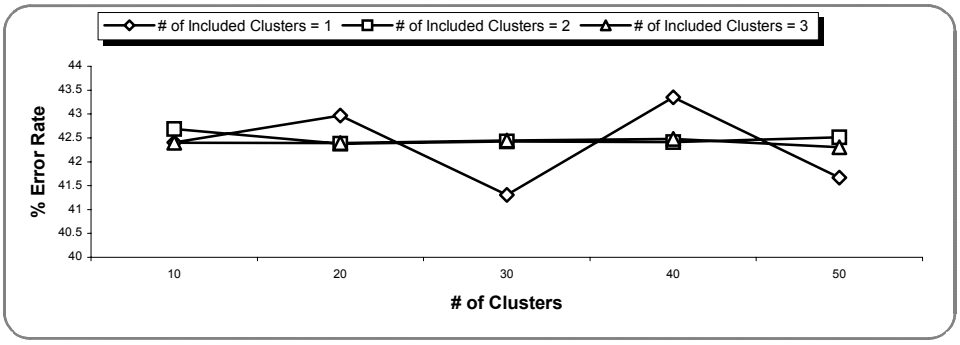

Figure 7: Experiments with the Number of Included Clusters

Figure 8 shows the situation when the HCC model with the number of included clusters may not provide good results where the test samples are located on the border of clusters or between clusters. The points marked with "**a**", "**b**", and "**c**" are known points, and the points marked with "**i**" and "**j**" are test or unknown points. The HCC model with the number of included clusters 1 can find the closest neighbors for points marked with "**i**". When finding the nearest points for the points marked with "**j**", more than one cluster nearest to the points need to be searched.

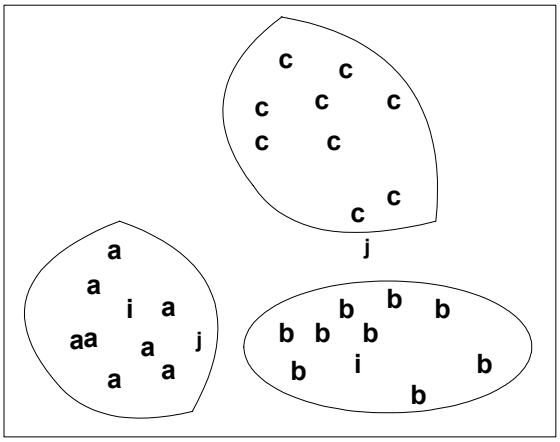

Figure 8: Example of Test Samples Located on Border or Between Clusters

Test Case 3: Experiments with the number of nearest neighbors

The number of nearest neighbors 3, 5, 7, and 10 are selected in the test experiment. The purpose of the experiment is to evaluate the effects of changing the number of nearest neighbors without changing any other input parameters. The following figure summarizes the finding of each test in this category. Figure 9 represents four series for 3-Nearest Neighbors, 5-Nearest Neighbors, 7-Nearest Neighbors, and 10-Nearest Neighbors respectively. 7-NN and 10-NN provide slightly better results than 3-NN. The 5-NN HCC model is the worst that is consistent with our evaluation using the PC model.

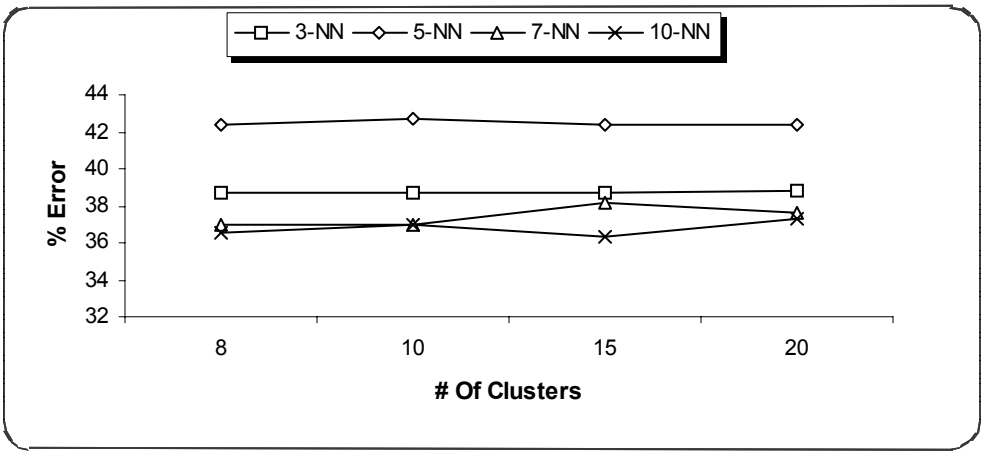

Figure 9: Experiments with the Number of Nearest Neighbors

4.3 Comparison of HCC model and Spectra's GAP model

GAP module, Spectra's software for demand forecasting and product allocation application, is based on pure statistics techniques. This module takes one store size parameter, the number of households, and nine shopper's demographic attributes as the inputs. Two small experiments are performed to compare the accuracy between the GAP model and the HCC model. One random account and product is selected for each test experiment. Table 4 shows the results of each test experiment. The HCC model shows impressive Root Mean Square Error (RMSE) improvement in the first experiment over the GAP model. The HCC model provides slightly better results than the GAP module in the second experiment. It is not feasible to run the GAP and HCC models for each test account and product due to lack of time and resources, but the two random experiments shows that the data mining based project provides better or slightly better results than the traditional statistical approach.

Account and Product Names	GAP RMSE	PROJECT RMSE
Walgreens (East and Central, 157 stores) Tropicana (Fr Jc-Org-O/Cont : Jc)	62.8565257	50.56068
Food Lion Store (South, 194 stores) Kraft Miracle Whip Light (SD-Miracle Whip : Sld Drsg)	9.38493318	9.16515

Table 4: RMSE Comparison between HCC and GAP Models

4.4 Comparison of PC and HCC models and discussions

The test experiments of the PC model and the HCC model are analyzed separately in Sections 3.3 and 4.2 respectively. This section compares the results of both models in terms of accuracy as well as performance. The HCC model provides the same or slightly better accuracy with significant running time improvement over PC model. Table 5 shows the side-by-side comparison of both models for each evaluation measure for the same test experiment. Table 6 shows that the HCC model not only provides better results but also achieves a significant performance improvement over the PC model.

Evaluation Measures Test Samples = 500, # NN = 3	Pure Classification (PC)	Hybrid Clustering Classification (HCC)		
		n = 1	n = 2	n = 3
Root Relative Square Error	38.74%	38.76%	38.73%	38.72%
Relative Absolute Error	0.28034	0.28272	0.27855	0.27907
Predicted within 5% diff.	14%	14%	14%	14%
Predicted within 6 to 10% diff.	8%	8%	9%	9%
Predicted within 11 to 25% diff.	26%	26%	27%	26%
Predicted within 26 to 50% diff.	28%	28%	27%	28%
Predicted with > 50% diff.	24%	24%	23%	23%

Table 5: Prediction Accuracy Comparison between PC and HCC Models

Test Experiments	PC Model %Error	HCC Model % Error	Running Time Improvement
Test Samples = 500 # Nearest Neighbors = 5	42.38%	42.38%	46%
Test Samples = 4500 # Nearest Neighbors = 5	50.93%	50.92%	56%

Table 6: HCC Model's Performance Improvement over PC Model

By comparing many test experiments, it appears that the Hybrid Clustering Classification technique is a better solution than the Pure Classification technique at every level, and a workable solution in practical situation with large data set and large number of attributes. In other words, the k-Mean Mode Clustering with the k-Nearest Neighbor method produces the true k-Nearest Neighbor Classification with a significant performance improvement.

The concept of combining clustering method with classification method works very well particularly for this type of demand forecasting and product allocations applications. The Clustering method partitions stores into clusters by considering only store attributes. Now similar stores are in one group and they are different from stores in other groups. But product sales may differ among stores in the same group because of differences in the demography of shoppers. The classification method classifies unknown product sales in stores by applying classification technique with the help of demographic

attributes of shoppers. Any clustering and classification techniques can be used in this hybrid concept. This project implemented k-Mean Mode clustering and k-NN classification techniques because they are ideal for experimental purposes. Hierarchical clustering or combination of hierarchical clustering and partition clustering can be used instead of only using partition clustering, and regression trees can be used as classification technique instead of k-NN technique. Usually data mining algorithms scale well for single relation mining, but the proposed HCC model maintains the same scalability on multiple relations with the hybrid concept.

5 Implementation Review

This section reviews the application from a programmer's perspective and from the end user's perspective. The application is designed to be easy to use, learn, and extend. Oracle 9i is the application database system. Oracle 9i provides a quite intelligent Cost Based Optimizer (CBT), and the CBT helped to improve the performance of many data intensive queries of this application. The application is written in Java 2 (JDK 1.4) and Swing. The application is divided into four layers, and Section 5.1 outlines the layered architecture including a description of some important classes of each layer. The user interface is very user friendly and light weighted. The output presentation provides two types of outputs, an on screen summary and detailed output in output file. Sections 5.2 and 5.3 discuss the input interface and output presentation with screen shots in detail.

5.1 Application Architecture Overview

All classes of the application are divided into four different layers based on their features. The layers are well defined and separated from each other. Each layer is outlined as shown in Figure 10. Some key classes and their concepts are covered briefly later in this section.

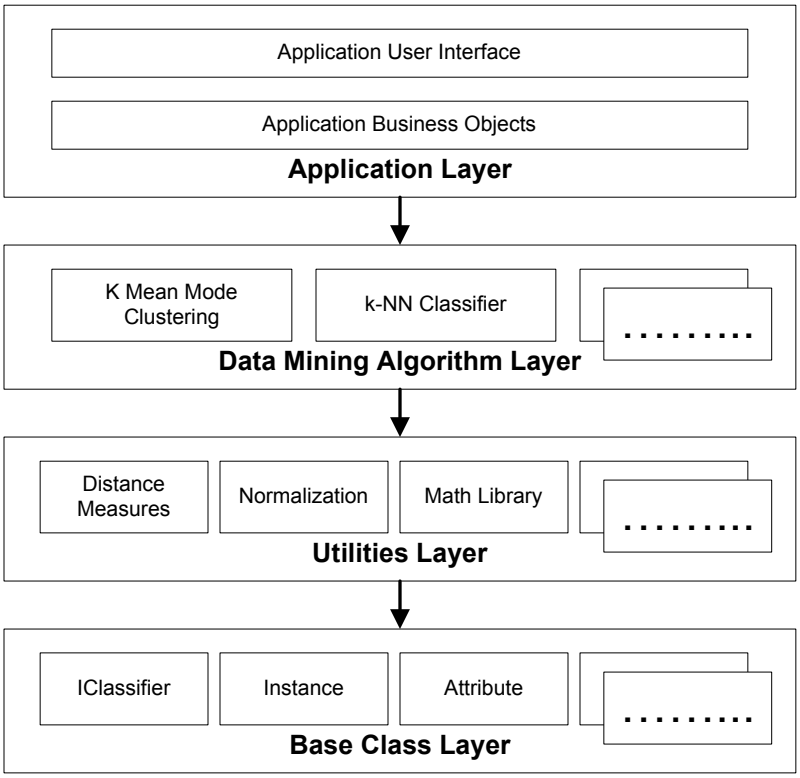

Figure 10: Application Layers Diagram

Base Class Layer: IClassifier, Instance, and Attribute are the three key classes in this layer. IClassifier is a simple and algorithm independent interface that provides definitions for building the classifier and classifying any unknown sample. Interface and Attribute represent each training/testing record and attribute, respectively.

Utilities Layer: This layer provides the supporting features such as all distance measures, normalization functions, math library, and database loading classes. Database Loader classes use Oracle indexing and hints to read data from multiple relations or tables.

Data Mining Algorithm Layer: k-Mean Mode Clustering and k-Nearest Neighbor models are part of this layer. Both classes implement a specific data mining algorithm. One important point is that these classes do not know anything about the business or application specific details.

Application Layer: Classes in this layer implement the business function. They know what underlying classes need to call to implement specific business solutions. This layer provides a customized user Interface to end-users.

The layered architecture helps not only to keep application logic separately from data mining techniques but also to design generic and extendable system. The three layers below the application layer can be reused in any other data mining projects. IClassifier, IFilter, and IDistance interfaces are three important interfaces, pure abstract classes, in this application. Figure 11 shows the interfaces and the classes providing particular functionality by implementing the interfaces. The method arguments are ignored in Figure 11. IClassifier, the abstract interface, defines two methods to build classifier and to classify unknown instance from classifier. KNNAlgorithm and KMeanModeAlgorithm provide specific data mining techniques by implementing the two methods of IClassifier interface. IFilter provides an interface for any filtering or normalization technique by just defining two simple methods, getFilteredValue(), and getOriginalValue(). IDistance defines one method, distance(from, to) which returns the distance between instance from and instance to. The EuclideanDistance class provides Euclidean Distance Measure by implementing the IDistance interface. Now it is quite easy to add new data mining technique or new normalization technique or new Distance measure just providing implementation of specific interface. New features can be added very easily without impacting any major changes in the application.

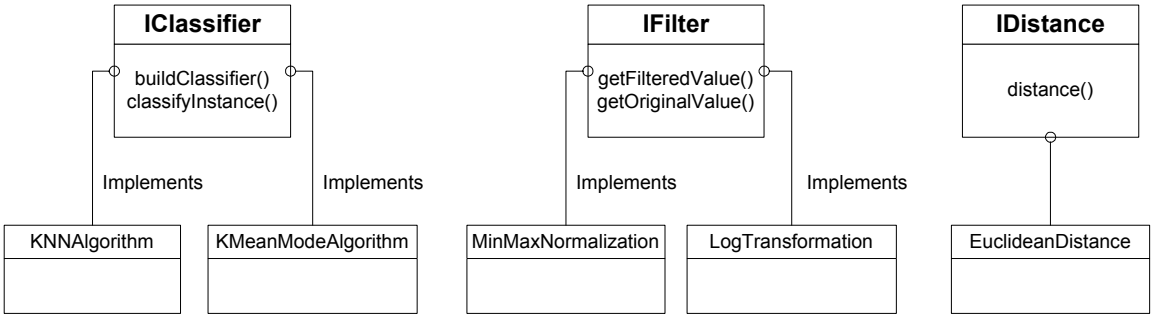

Figure 11: IClassifier, IFilter, and IDistance Interfaces

This section provided details from a programmer's perspective. The next two sections discuss input interface and output presentation from user's perspective.

5.2 Input Interface

The graphical user interface is simple and easy to understand. Understanding the business and data can help to drive the application more efficiently by adding/removing attributes and specifying attributes weight. The application provides all available input options to the users to run the PC and HCC models. There are three input tabs and one output tab in the application. This section briefly summarizes each input tab with the help of screen shots.

Setup Tab: The first tab is the setup window. Users can run the application either by choosing the Pure Classification (PC) model or by choosing the Hybrid Clustering Classification (HCC) model as shown in the Figure 12. Initially all other tabs are disabled. The Clustering tab becomes enabled only if the user selects the Hybrid Clustering Classification Model. The Classification tab is enabled regardless of which model the user selects. This tab also provides a prompt for the Distance Method that does not depend upon any specific model. The Exit button is available in all the tabs, and it allows the user to close the application after cleaning up open resources.

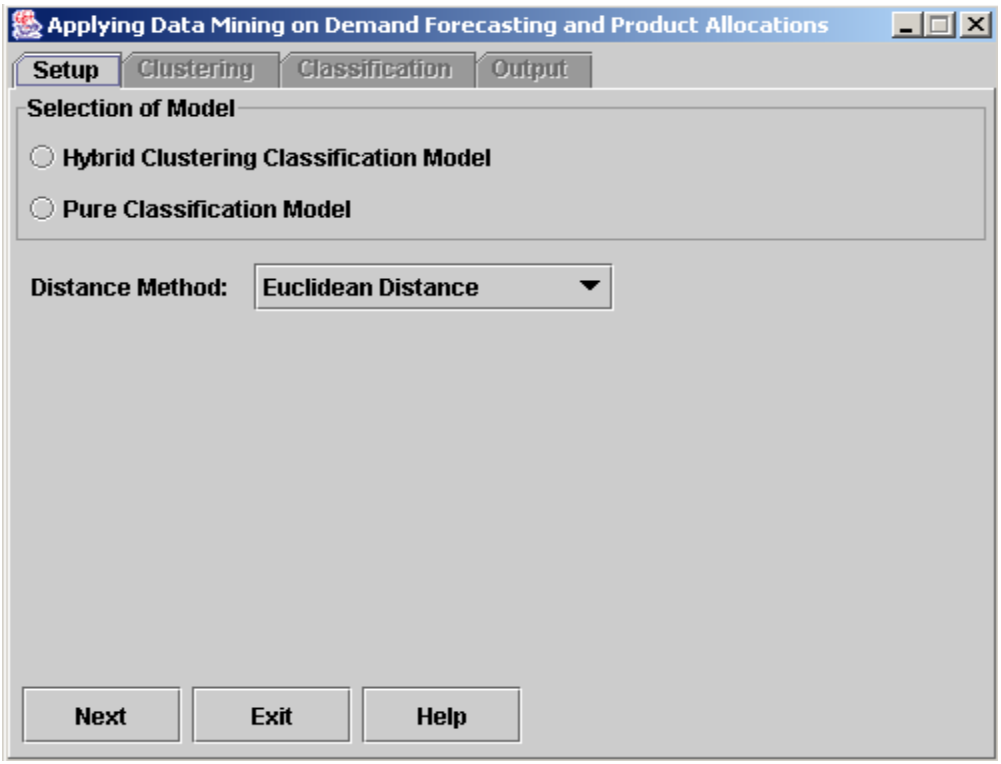

Figure 12: User Interface - Setup Tab

Clustering Tab: This tab provides the prompts to run the clustering technique as shown in Figure 13. The Clustering tab becomes enabled only if the user selects the Hybrid Clustering Classification model in the setup tab. The input prompts are the number of clusters and list of all attributes with weights as shown in the following screen. Select attributes prompt is basically a table with four columns. The first two columns, Attribute Type and Attribute Name, are read only columns. The user can override the default weight of attributes in the third column. The last column allows the user to select or unselect the attributes. Some attributes are selected by default, but the user can unselect the default attributes and select some other attributes.

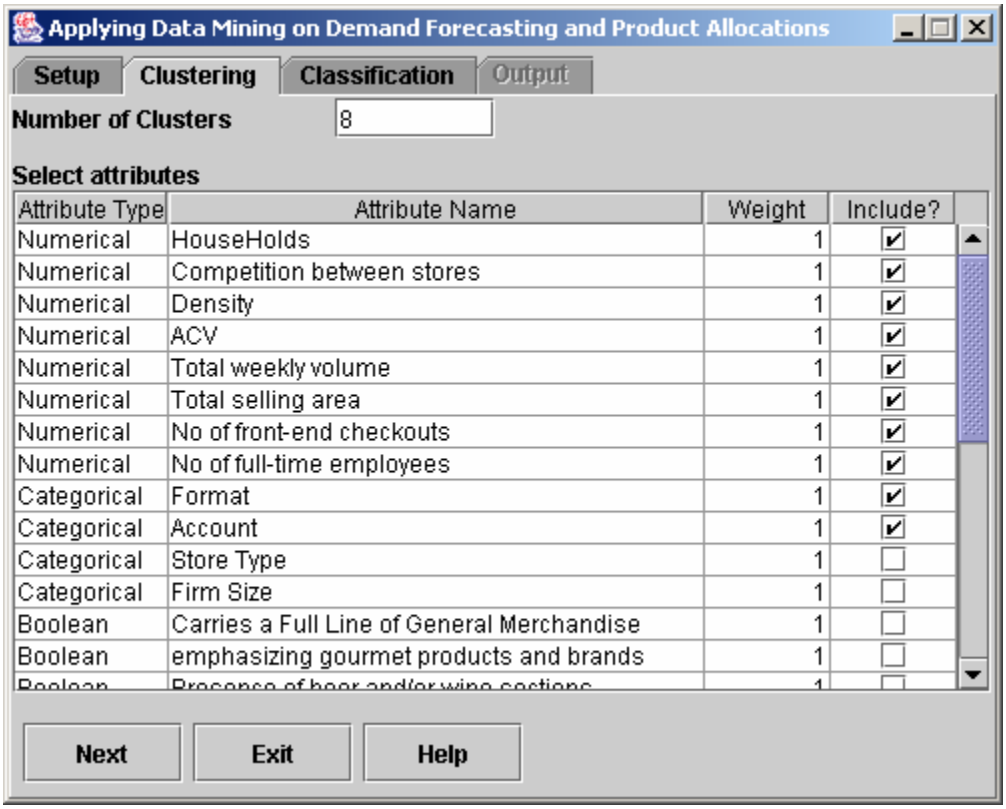

Figure 13: User Interface - Clustering Tab

Classification Tab: This tab allows users to select the input parameters related to the classification technique. The input parameters are the number of nearest neighbors, the number of included clusters, the size of the test data set, and a list of attributes with weight as shown in Figure 14. The classification tab is available regardless of what model is being selected in the setup tab. If the user selects the Pure Classification (PC) model, the application will hide the number of included clusters prompt. The number of clusters prompt is required only for Hybrid Clustering Classification (HCC) model. The user can also select either all test samples by checking the All Test Samples prompt or some percentage of the test samples with the help of Percentage of Test Samples prompt.

Select attributes table of this tab is the same as of that clustering tab. There are some common attributes in the Clustering tab and Classification tab. If the user selects some of them in the Clustering tab then the Classification tab will set them as selected by default. But the user can override the selection by unselecting them. Clicking on the Submit button will run the selected model.

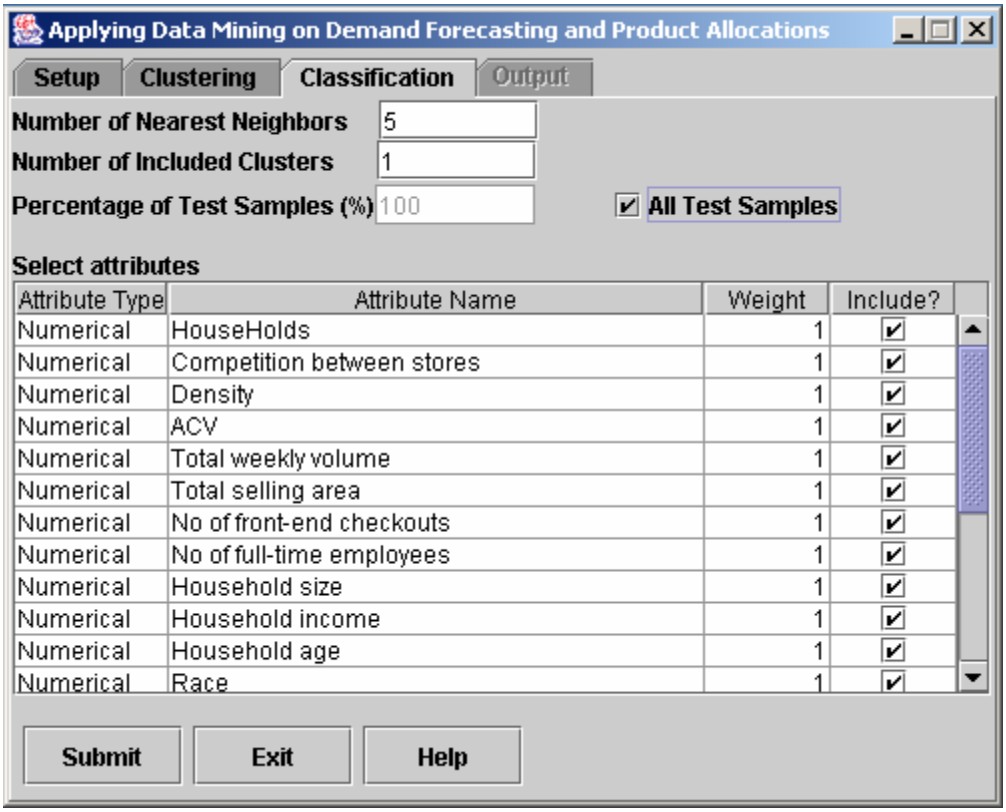

Figure 14: User Interface – Classification Tab

5.3 Output Presentation

The application presents output in two different ways: an on screen summary output and a detailed model output in an output log file. The output tab of the application provides the summary model status along with a progress bar indicating the running status of the model as shown in Figure 15. The on screen summary window displays all selected prompts, cluster quality measures if the HCC model is selected, evaluation measures of the selected model, and the total running time of the application.

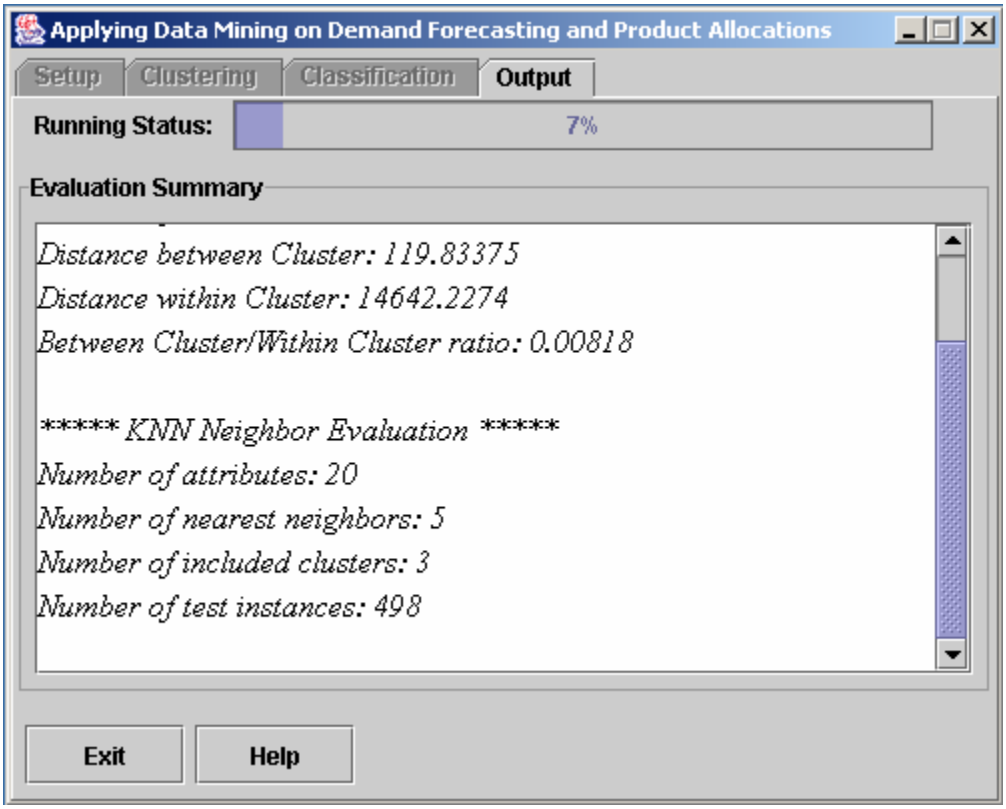

Figure 15: Model Evaluation Presentation

The application creates the output log file for each model run, and assigns a name to the log file using the current time-stamp for unique and easy identification. The log file contains the following information of each model run:

- All selected input prompts with values
- Clustering quality measures
    - Within-Cluster variation
    - Between-Cluster variation
- Description of each cluster
    - Number of instances
    - Mean and SD for each numerical attribute
    - Mode and its counts for each categorical attribute
    - True counts and False counts for each Boolean attribute
- Evaluation Measures (described as section 2.3)
    - Root Relative Square Error and Root Mean Square Error

- Prediction Accuracy Counts
- Running time

6 Conclusion

In this paper, we have proposed a data mining application for better demand forecasting and product allocations. The paper presented a complete data mining process from identifying business objective to evaluating models on multi-relation data sets. This project implemented two mining models, Pure Classification and Hybrid Clustering Classification. Both models are designed to be generic solutions for this type of multi-relation data mining. The main aim of Hybrid Clustering Classification is to divide the expensive classification task into simpler tasks by creating well-defined clusters in the training data set. Experimental results show that the Hybrid Clustering Classification provides better accuracy as well as scalability than the Pure Classification. The feature weighting, selection of attributes in models, and prompts for all possible input parameters allow users, especially domain experts, to run the data mining models more efficiently for better demand forecasting.

This application is designed to be extendable at every level in this system. New attributes can be added into the system without affecting anything in the application. One possible enhancement is to integrate hierarchical and partition clustering techniques to select the correct set of k points as initial k cluster centroids as well as to discover the exact number of clusters rather than estimating this manually. Another possible enhancement is to add regression trees, and then evaluate the results with the Hybrid Clustering Classification model.

Appendix A – Table Descriptions

Table: STORES [1]

Name	Datatype	Size	Scale	Nulls?	Description
TDLINX	VARCHAR2	7		No	Store's unique identification
NAME	VARCHAR2	50		No	Name of store
STREET	VARCHAR2	50		No	Street address of store
CITY	VARCHAR2	20		No	City of store
STATE	VARCHAR2	2		No	State of store
ZIP	NUMBER	10	0	No	Zip of store
COUNTY	NUMBER	8	0	No	County of store
BLOCKGROUP	VARCHAR2	32		No	Census block group of store
FORMAT	VARCHAR2	5		No	Format of store
CHANNEL	VARCHAR2	1		No	Store's channel
ACCOUNT_ID	VARCHAR2	32		Yes	Store's account identification
STORENUM	NUMBER	8	0	Yes	Store's number

Table: FACTS

Name	Datatype	Size	Scale	Nulls?	Description
ID	VARCHAR2	20		No	Unique identification of store's fact
NAME	VARCHAR2	100		Yes	Name of store's fact
DATATYPE	VARCHAR2	1		Yes	Data type of a fact ("N" or "S")
BUSINESSUSE	VARCHAR2	15		Yes	Business use ("NUMERICAL" or "CATEGORICAL" or "BOOLEAN")

Table: FACT_DATA

Name	Datatype	Size	Scale	Nulls?	Description
TDLINX	VARCHAR2	7		No	Store's unique identification (see STORES)
FACT_ID	VARCHAR2	20		No	Unique fact identification (see FACTS)
NUMVALUE	NUMBER	15	0	Yes	Numerical value of store's fact
TEXTVALUE	VARCHAR2	25		Yes	Categorical or Boolean value of store's fact

[1] Table definitions report is generated from Oracle 9i Enterprise Manager.

Table: PRODUCTS

Name	Datatype	Size	Scale	Nulls?	Description
ID	VARCHAR2	13		No	Product's unique identification
NAME	VARCHAR2	100		No	Description of a product
PRODUCT_GROUP	NUMBER	3	0	No	Group of a product
MODULE	NUMBER	4	0	No	Module name of a product
BRAND	NUMBER	6	0	No	Brand of a product

Table: DI_DATA

Name	Datatype	Size	Scale	Nulls?	Description
TDLINX	VARCHAR2	7		No	Store's unique identification (see STORES)
PRODUCT_ID	VARCHAR2	13		No	Product's unique identification (see PRODUCTS)
DI1	NUMBER	20	10	No	Household Size
DI2	NUMBER	20	10	No	Household Income
DI3	NUMBER	20	10	No	Household Age
DI4	NUMBER	20	10	No	Race
DI5	NUMBER	20	10	No	Age and Presence of Children
DI6	NUMBER	20	10	No	Housing Tenure
DI7	NUMBER	20	10	No	Household Education
DI8	NUMBER	20	10	No	Lifestyle
DI9	NUMBER	20	10	No	Nielsen County Size

Table: STORE_PRODUCT_SALES

Name	Datatype	Size	Scale	Nulls?	Description
TDLINX	VARCHAR2	7		No	Store's unique identification (see STORES)
PRODUCT_ID	VARCHAR2	13		No	Product's unique identification (see PRODUCTS)
AVGDOLLARS	NUMBER	5	0	No	Average weekly sales in dollars

Appendix B – List of attributes

Attribute	Datatype
Population density	NUMERICAL
Total households	NUMERICAL
Stores competition	NUMERICAL
All commodity value	NUMERICAL
# of checkouts	NUMERICAL
# of end-aisle for contract	NUMERICAL
Full time employees	NUMERICAL
Selling area in sq. ft.	NUMERICAL
Average weekly volume	NUMERICAL
Grocery selling area	NUMERICAL
Grocery weekly volume	NUMERICAL
Household Size	NUMERICAL
Household Income	NUMERICAL
Household age	NUMERICAL
Race	NUMERICAL
Age and presence of children	NUMERICAL
Housing tenure	NUMERICAL
Household education	NUMERICAL
Lifestyle	NUMERICAL
Nielsen county size	NUMERICAL
Store format	CATEGORICAL
Account of store	CATEGORICAL
Merchandising type	CATEGORICAL
Firm size of store	CATEGORICAL
Type of store	CATEGORICAL
Product brand name	CATEGORICAL
Presence of an in-store bakery	BOOLEAN
Presence of a full service bank	BOOLEAN
Presence of beer and/or wine sections	BOOLEAN
Presence of a bulk food section	BOOLEAN
Presence of coffee grinders	BOOLEAN
Presence of an in-store service-deli	BOOLEAN
Presence of a fast food franchise	BOOLEAN
Presence of an in-store seafood counter	BOOLEAN
Presence of an in-store floral center	BOOLEAN
Carries a full line of general merchandise	BOOLEAN
Presence of gasoline	BOOLEAN

Continued on next page

Attribute	Datatype
Supermarket emphasizing gourmet products and brands	BOOLEAN
supermarket appealing to Hispanic customers	BOOLEAN
Limited assortment (<1500 items)	BOOLEAN
Store sells lottery tickets	BOOLEAN
Presence of an in-store restaurant	BOOLEAN
Presence of an in-store salad bar	BOOLEAN
Presence of a seasonal aisle	BOOLEAN
Presence of an in-store cosmetic counter	BOOLEAN
Presence of a specialty cheese center	BOOLEAN
Presence of an in-store service meat/butcher	BOOLEAN
High volume cigarette sales	BOOLEAN
Presence of an in-store video rental department	BOOLEAN
Store sells videos	BOOLEAN
Indicating store remodeled since last survey	BOOLEAN
Presence of prepared food section	BOOLEAN

References

[1] Joe Mckendrick. “Reversing the Supply Chain”. Teradata Magazine - Applied Solutions. 2003. Volume 3 - No. 3. <http://www.teradatamagazine.com>

[2] Jaiwei Han and Micheline Kamber, Data Mining: Concepts and Techniques, Morgan Kaufman Publishers, San Francisco, 2001.

[3] Dan Steinberg and Scott Cardell. “Improving Data Mining with New Hybrid Methods”. Online. 1998. <http://www.salford-systems.com>.

[4] J. Jin, A. Keiichi. “Improvement of Decision Tree Generation by using Instance-Based Learning and Clustering Method”. IEEE International Conference. Volume 1. 14-17 Oct. 1996.

[5] Steven Salzberg and Arthur L. Delcher. “Best-Case Results for Nearest-Neighbor Learning”. IEEE Transactions on Pattern Analysis and Machine Intelligence. Volume 17, No 6, June 1995.

[6] Jorma Laaksonen and Erkki Oja. “Classification with Learning k-Nearest Neighbors”. IEEE International Conference. Volume 3. 3-6 June 1996.

[7] David Hand, Heikki Mannila and Padhraic Smyth. Principle of Data Mining. MIT Press. Cambridge. 2001.

[8] Ian Witten and Eibe Frank. Data Mining Practical Machine Learning Tools and Techniques. Morgan Kaufman Publishers. San Francisco. 1999.