Adaptive Algorithms for Real-World Transactional Data Mining

Edward Tersoo Apeh

A thesis submitted in partial fulfillment of the requirements of Bournemouth University for the degree of Doctor of Philosophy

October 2012

Bournemouth University

Abstract

The accurate identification of the right customer to target with the right product at the right time, through the right channel, to satisfy the customer's evolving needs, is a key performance driver and enhancer for businesses. Data mining is an analytic process designed to explore usually large amounts of data (typically business or market related) in search of consistent patterns and/or systematic relationships between variables for the purpose of generating explanatory/predictive data models from the detected patterns. It provides an effective and established mechanism for accurate identification and classification of customers. Data models derived from the data mining process can aid in effectively recognizing the status and preference of customers - individually and as a group. Such data models can be incorporated into the business market segmentation, customer targeting and channelling decisions with the goal of maximizing the total customer lifetime profit. However, due to costs, privacy and/or data protection reasons, the customer data available for data mining is often restricted to verified and validated data, (in most cases, only the business owned transactional data is available). Transactional data is a valuable resource for generating such data models. Transactional data can be electronically collected and readily made available for data mining in large quantity at minimum extra cost. Transactional data is however, inherently sparse and skewed. These inherent characteristics of transactional data give rise to the poor performance of data models built using customer data based on transactional data. Data models for identifying, describing, and classifying customers, constructed using evolving transactional data thus need to effectively handle the inherent sparseness and skewness of evolving transactional data in order to be efficient and accurate. Using real-world transactional data, this thesis presents the findings and results from the investigation of data mining algorithms for analysing, describing, identifying and classifying customers with evolving needs. In particular, methods for handling the issues of scalability, uncertainty and adaptation whilst mining evolving transactional data are analysed and presented. A novel application of a new framework for integrating transactional data binning and classification techniques is presented alongside an effective prototype selection algorithm for efficient transactional data model building. A new change mining architecture for monitoring, detecting and visualizing the change in customer behaviour using transactional data is proposed and discussed as an effective means for analysing and understanding the change in customer buying behaviour over time. Finally, the challenging problem of discerning between the change in the customer profile (which may necessitate the effective change of the customer's label) and the change in performance of the model(s) (which may necessitate changing or adapting the model(s)) is introduced and discussed by way of a novel flexible and efficient architecture for classifier model adaptation and customer profiles class relabeling.

Contents

List of Figures

List of Tables

Acknowledgements

I wish to start by expressing my deep gratitude to God for the wonder of Computational Intelligence that can inspire endless interest and a quest for knowledge discovery.

Formula 1, Football, And Basketball are Teams Sports - and so is Undertaking a Ph.D. My aspiration to pursue this research would not have seen the light of day without the valuable contributions of a number of supportive people, to whom I will always be grateful.

I am profoundly grateful for the guidance and support of my supervisor, Prof. Bogdan C. Gabrys. Bogdan's insightful comments and timely professional and personal advice made my entire time at Bournemouth University an enthralling and enriching experience. I thank Bogdan for his support, for sharing his endless supply of ideas and feedback, and for his friendship. I could not have had a better advisor. Many thanks, Bogdan, for your tireless work and ceaseless encouragement. Thank you B!

I am also grateful to the remainder of my thesis committee: Prof. Trevor Martin of Bristol University, Dr. Amanda Schierz of the Institute of Cancer Research, Dr. Michael Barker of Screwfix Limited and Dr. Indrė Žliobaite of the INFER project at Bournemouth University. They were all generous with their time and provided me with useful insights and help with research and in preparing this thesis.

My gratitude also go to Kevin Loader, Rob Josling, and Paul Belshaw all of Screwfix Limited as well as Hugo Jaspers of Sligro Food Group N.V., for their time and for providing me with their data and domain knowledge without which the key research findings of this thesis will not have been substantiated. I am particularly grateful to Dr. Mykola Pechenizkiy of Eindhoven University of Technology for liberality with his profound research insight and help in getting the permission to use SLIGRO's data.

I must also extend my gratitude to my fellow researchers and the support staff at: the SMART Technology Research Centre, the school of Design, Engineering and Computing, and the entire Bournemouth University community, with whom I shared many coffee breaks, lunches around campus, laughs over the mundane issues of life, and many other ups and downs of being a Ph.D. researcher.

Lastly, but certainly not least, I must extended a great deal of love and appreciation to all my friends and family outside of the Bournemouth University community who have helped me along in my life. Special thanks are in order for my parents and siblings who always pushed me to achieve. To my late mother Caroline for recognizing and cultivating my interest and abilities in science and math at such an early age. To my father Joseph for providing an example of how hard work and patience can reap rewards. To my stepmother Magdalene for the continuous prayers, encouragement and support. Many thanks too to Sue Burt, Laurence and Gabrielle Hodge, Rob and Margaret Averill, Mike and Diane Reynolds, Steve and Mallin Ambore, and Jacqueline Msoma for their continuous support and friendship over the years.

Finally, I would like to thank Akin Sulaimon and Joyce Feese for being such great friends; and always having faith in my abilities and inspiring me to seek out my passions in work and life. Thank you all.

Declaration

The thesis is entirely my own work, and, except where otherwise stated, describes my own research.

Chapter 1

Introduction

Data mining algorithms have now gained prominence as core business profit enhancing tools, with enterprise database software systems now incorporating them as standard. A recent survey by Gartner [Herschel, 2008] and Forester Research [Kobielus, 2008] attribute the increased dependence and usage of data mining tools to businesses becoming more "information-driven".

This is more so, given that we currently live in a world where information overload is a pertinent issue. Data mining algorithms have proven to be invaluable tools for discovering hidden/unknown patterns in data, for predictive or descriptive purposes. For instance, the discovered patterns can be used by a product retailer to answer questions like "Who is likely to buy a certain product in the next six months?"; "What are the characteristics of these likely buyers?"

In order to adequately utilize the discovered patterns and respond to changing market forces or market growth, businesses tend to constantly change their processes with a resultant deterioration of the data mining models performance due to their becoming obsolete as soon as the business process they model changes.

To discover the hidden/unknown patterns, data mining algorithms employ statistical methods which operate on the data to produce statistical conclusions for specific patterns in the data. Huge transactional datasets containing millions of training examples with a large number of attributes (tall fat data) are relatively easy to gather for such data mining purposes [Raykar, 2005].

However, the inherent sparseness of transactional datasets and its constant changing nature makes the data mining process challenging; even more so for data mining algorithms which do not make any assumptions on the constantly changing form of the underlying function generating the data.

This seriously restricts the use of these data mining algorithms in the context of mining transactional datasets.

Interfacing the data mining algorithms with transactional databases as an integral part of an organization's data management plan is also made difficult as workarounds are

often required to deal with the issue of sparse transactional datasets.

1.1 Thesis Goal and Scope

This thesis presents the analysis, design and implementation of data mining algorithms that discover hidden/unknown patterns in transactional data and generate models that are adaptive to the change and uncertainty inherent in transactional data.

It presents investigations on how data mining techniques have been and can be used for knowledge discovery in transactional data. It proposes approaches that will make it easier for organisations to apply data mining to transactional data sets.

In particular, using data from Screwfix - a leading UK retail outfit, and SLIGRO Food Group N.V. - a group of food-retail and food-service companies selling to the Dutch food and beverages market as case studies, this thesis describes the application of robust adaptive data mining algorithms to real-world transactional datasets and the down-streaming of advanced modelling results to non-expert users.

Screwfix and Sligro have large quantities of customer and order data. The data is heavily sparse and in constant flux in that many of the customers purchase few items per transaction and constantly change their buying behaviour at different time points. For example, many Screwfix customers tend to buy different items depending on the job they are undertaking at a particular time, e.g. gardening in the spring and building in the summer.

The constantly changing buying behaviour of the customers complicates the process involved in identifying significant correlation between the different buying behaviour time points, making the analytical characterization of the customers challenging. Furthermore, sparseness and skewness of transactional data is compounded by the similarity of few items bought by the customers which makes predictability and distinction of the customers based on their buying behaviour challenging.

Data mining algorithms which can effectively handle the inherent sparseness and skewness of transactions to identify behaviour of customers over time, can thus be used to aid in the decision support process.

Screwfix and SLIGRO are examples of the type of data rich organisations that need to build data mining and analysis skills to be able to improve on how they use their data in their marketing and sales departments. Recent attempts by Screwfix and SLIGRO to analyse their transactional data have failed due to sparsity and the constantly changing nature of transactional data.

This thesis presents investigations of methods for adapting and changing data mining models as the nature of Screwfix and SLIGRO's transactional data grows and changes. Adaptation helps to prolong the useful life of learned predictive/inference models which is an important part of making models more useful to non-expert users.

1.2 Summary of Contributions

Concisely, the main contributions of this thesis are:

1. An in-depth critical analysis of the main research challenges of adaptivity, scalability and uncertainty encountered when mining real-world transactional data. In particular, the thesis details the problem of mining large transactional data and how sparsity and skewness results in uncertainty of inference from transactional data.

2. Clustering and sub-sampling are often chosen as the favoured data pre-processing approaches in data mining project. The thesis highlights the problem of transactional data pre-processing and shows the inappropriateness in mining transactional data using some of the clustering and sampling techniques proposed in the literature.

3. Investigations and implementations of adaptive algorithms for solving business problems using real-world transactional data. In particular, the thesis presents the analysis, design, implementation and interpretation of results for classifying and detecting as well as adapting to customer behaviour using robust adaptive models generated from transactional data.

4. Methods for the down streaming of advanced models of real-world transactional data to non-expert users. More specifically, the thesis presents the results for detecting, visualizing and adapting to the change in buying behaviour to customer transactions over time using the real-world transactional data provided by Screwfix and SLIGRO.

1.3 List of Publications

The research undertaken has thus far resulted in the following publications:

- Edward Apeh, Bogdan Gabrys, Amanda Schierz, "Customer Profile Classification: To Adapt Classifiers or To Relabel Customer Profiles?", Accepted for publication in the Special Issue: NaBIC2011 Neurocomputing Journal. Elsevier.

- Edward Apeh, Indrė Žliobaite, Mykola Pechenizkiy, Bogdan Gabrys, "Predicting Multi-Class Customer Profiles Based on Transactions: a Case Study in Food Sales", in The Proceedings of AI-2012 Thirty-second SGAI International Conference on Artificial Intelligence (SGAI).

- Edward Apeh, Bogdan Gabrys, "Detecting and Visualizing the Change in Classification of Customer Profiles based on Transactional Data", in the Special Issue of the Journal of Evolving Systems, October 2012.

- Edward Apeh, Bogdan Gabrys, "Change Mining of Customer Profiles based on Transactional Data", in Proceedings of the 11th IEEE International Conference on Data Mining Workshops (ICDMW 2011). IEEE, December 2011
- Edward Apeh, Bogdan Gabrys, Amanda Schierz, "Customer Profile Classification Using Transactional Data", in Proceedings of the Third World Congress on Nature and Biologically Inspired Computing (NaBIC2011). IEEE, October 2011
- Amanda Schierz, Marcin Budka, Edward Apeh, First and Second Winners' notes: "Using Multi-Resolution Clustering for Music Genre Identification" in the ISMIS 2011 Contest: Music Information Retrieval.
- Edward Apeh, Bogdan Gabrys, Amanda Schierz, "Robust Adaptive Algorithms for Relational Data Mining", in The Proceedings of the 3rd School of Design, Engineering and Computing Poster Conference, Bournemouth University, UK, May 2010.

1.4 Roadmap

The thesis continues in Chapter 2 with an overview of data mining together with a description of the problems encountered when mining transactional data for information. The main part of that chapter surveys techniques for handling the problem of uncertainty and adaptation and their applicability in mining transactional data.

Preprocessing of the data for the purpose of data mining is a recurrent challenge in data mining and is reported to take up to 80 percent of data mining and knowledge discovery projects' time [Adriaans and Zantinge, 1996, Han and Kamber, 2006, Kotsiantis and et al., 2006, Pyle, 1999]. For mining transactional data, a part of the preprocessing step involves constructing appropriate customer profiles. Chapter 3 provides a detailed description of customer profiles and the methods for their construction along with a statistical description of Screwfix's transactional data. The results obtained from using clustering as a preprocessing step for Screwfix's transactional data is also presented.

Chapters 4 and 5 contain the results of investigated approaches for uncertainty handling in classifying transactional data for a binary and multiclass case respectively. Chapter 6 then presents adaptive mechanisms for transactional data mining. Chapter 7 concludes this thesis with a summary and a discussion of open research problems pertaining to transactional data mining.

Figure 1.1 shows the overall structure of the thesis.

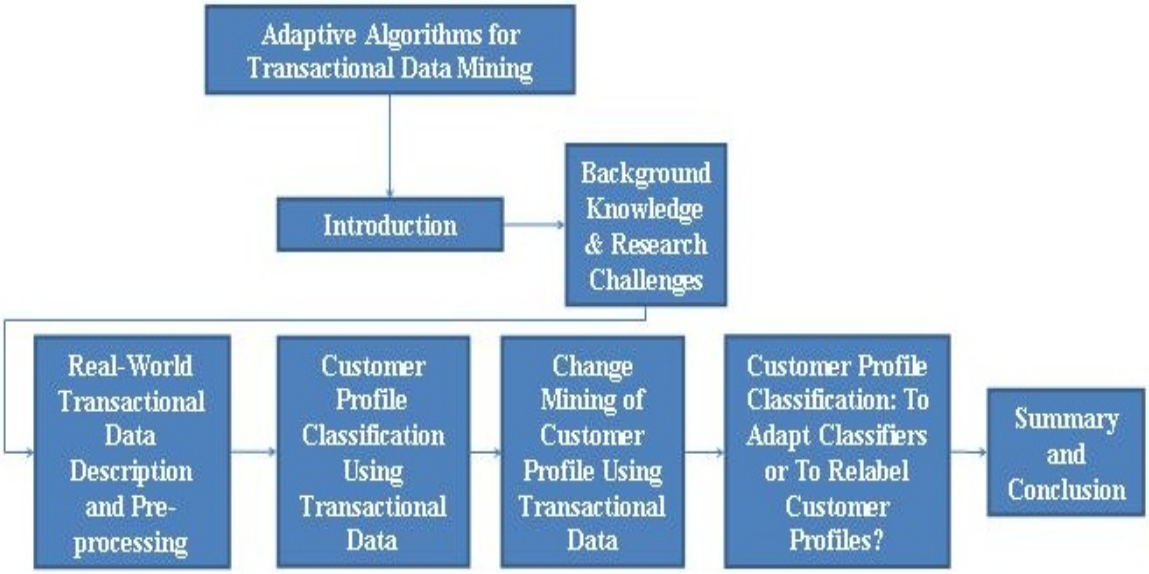


Figure 1.1: Graphical Representation of the Thesis Structure

Chapter 2

Background Knowledge and Research Challenges

The purpose of this chapter is to give a concise overview of the fundamentals of data mining, as well as challenges encountered in its application to transactional data.

It starts with an introductory overview of the field of data mining in Section 2.1. Data mining algorithms discover hidden/unknown patterns in data and generate models for the purpose of knowledge discovery and understanding. The discussion here covers the common data mining approaches and their use in mining transactional data.

The discipline of data mining has seen an explosion of interest over the last few years, and has been successfully applied across an extraordinary range of problem domains, in areas as diverse as finance [Kovalerchuk and Vityaev, 2000], medicine [Lavra and Zupan, 2005, Lavrac, 1999], engineering [Grossman et al., 2001], geology [Ester et al., 2001, Shekhar et al., 2003] and physics [Grossman et al., 2001, Sumathi and Sivanandam, 2006]. Indeed, anywhere that there are problems of prediction, classification or control, data mining techniques are being introduced. This increase in usage can be attributed to the ability of data mining techniques to contribute to the generation of new opportunities, by providing the following capabilities [Berry and Linoff, 2004]:

- **Automated prediction of trends and behaviours.** Data mining automates the process of finding predictive information in large databases. Questions that traditionally required extensive hands-on analysis can now be answered quickly, directly from the data. A typical example of a predictive problem is targeted marketing [Associates, 1999]. Data mining uses data on past promotional mailings to identify the targets most likely to maximize return on investment in future mailings. Other predictive problems include forecasting bankruptcy [Foster and Stine, 2004, Sung et al., 1999] and other forms of default [Kumar and Ravi, 2008, Phua et al., 2005], and identifying segments of a population likely to respond similarly to given events [Jiang and Tuzhilin, 2006].

- **Automated discovery of previously unknown patterns.** Data mining tools sweep through databases and identify in one step previously hidden patterns . An example of pattern discovery is the analysis of retail sales data to identify seemingly unrelated products that are often purchased together [Gutierrez, 2006]. Other pattern discovery problems include detecting fraudulent credit card transactions [Chan et al., 1999] and identifying anomalous data that could represent data entry keying errors [Margineantu et al., 2005].

The application of data mining techniques in mining transactional data have however been hampered due to the skewness and sparsity of transactional data. This will be elaborated upon in the later chapters of this thesis.

2.1 Overview of Data Mining

Data mining is defined as the explorative and non-trivial search for implicit, previously unknown, and potentially useful insights from data [Han and Kamber, 2006]. It is the automated process of discovering patterns in data [Fayyad, 1996]. It consists of different phases and incorporates a number of computer science fields such as Artificial Intelligence, Databases, and Machine Learning, as well as intellectual human capabilities such as curiosity and creativity.

The goal of data mining is to estimate (or learn) a useful model of an unknown system from available data. The generated model is subsequently often used in business applications for predictive or descriptive purposes. In distinguishing data mining from the other learning methods of predictive and statistical model estimation, Cherkassky and Mulier [1998] describe data mining as the learning methodology which attempts to extract a subset of data samples (from a given large data set) with useful (or interesting) properties. However, the apparent non-distinction of data mining from the other learning methods of predictive and statistical model estimation, arises from the 'seemly' absence of generally accepted theoretical frameworks for data mining which has resulted in data mining algorithms being initially introduced (by practitioners) and then 'justified' using formal arguments from statistics, predictive learning, and information retrieval.

The concept of data mining is, however, becoming increasingly popular as a business information management tool, where it is expected to reveal knowledge that can guide decisions in conditions of limited certainty [Hand, 1998, Hand et al., 2001].

2.1.1 Stages of the Data Mining Process

Generally, the process of data mining consists of four stages [Fayyad, 1996]:

1. the initial exploration,

2. model building or pattern identification with validation/verification,

3. deployment (i.e., the application of the model to new data in order to generate predictions or describe the phenomenon responsible for the data),

4. model management.

Stage 1: Exploration.

This stage usually starts with data preparation which may involve cleaning data, data transformations, selecting subsets of records and - in the case of data sets with large numbers of variables ("fields") - performing some preliminary feature selection operations to bring the number of variables to a manageable range (depending on the statistical methods being considered). Then, depending on the nature of the analytic problem, this first stage of the process of data mining may involve anywhere from a simple choice of straightforward predictors for a regression model, to elaborate exploratory analyses using a wide variety of graphical and statistical methods, in order to identify the most relevant variables and determine the complexity and/or the general nature of models that can be taken into account in the next stage.

The main goal of this stage as outlined in the visual guide to the Cross Industry Standard Process for Data Mining (CRISP-DM) methodology shown in Figure 2.1 is to understand the data to be mined as well as to assess and determine the business' data mining goal.

Stage 2: Model Development and Validation.

This stage involves processing the prepared data and considering various models from which the best one is chosen based on their predictive performance (e.g., explaining the variability in question and producing stable results across samples). There are a variety of techniques which have been developed to achieve this goal - many of which are based on so-called "competitive evaluation of models", that is, applying different models to the same data set and then comparing their performance to choose the best. These techniques - which are often considered the core of data mining for data model generation/selection/-validation - include: base classifiers (such as Decision Trees, Support Vector Machines, Neural Nets), ensemble methods (such as Bagging (Voting, Averaging), Boosting, Stacking (Stacked Generalizations)), and meta-learning. They are discussed in greater detail in [Hastie et al., 2009].

The data processing, modelling and evaluation tasks are usually repeated with the knowledge and feedback from stage 1 incorporated if needed, until the model that performs best on the data is determined, as depicted in Figure 2.1.

Stage 3: Model Deployment.

This penultimate stage of the initial data mining model development and deployment cycle involves using the model selected as best in the previous stage and applying it to new data, in order to generate predictions or estimates of the expected outcome.

Stage 4: Model Management.

Usually, model management implies periodic creation of models and replacement of current ones in an automatic or semi-automatic fashion. In this sense, model management is tightly coupled with model development and deployment steps. It can be seen as an outer-loop controlling these steps. When dealing with large numbers of models, a model management component, responsible for automating the creation and deployment of models, becomes a necessity. A useful approach for implementing model management is the champion/challenger testing strategy. In a nutshell, champion/challenger testing is a systematic, empirical method of comparing the performance of a production model (the champion) against that of new models built on more recent data (the challengers). If a challenger model outperforms the champion model, it becomes the new champion and is deployed in the production system. Challenger models are built periodically as new data are made available [Campos et al., 2005].

Another dimension of model management is the creation and management of metadata about models. Model metadata can be used to dynamically select, for scoring, models appropriate to answer a given question [Jain et al., 2008].

The model management stage usually encompasses the modelling, evaluation and deployment steps as also depicted in Figure 2.1.

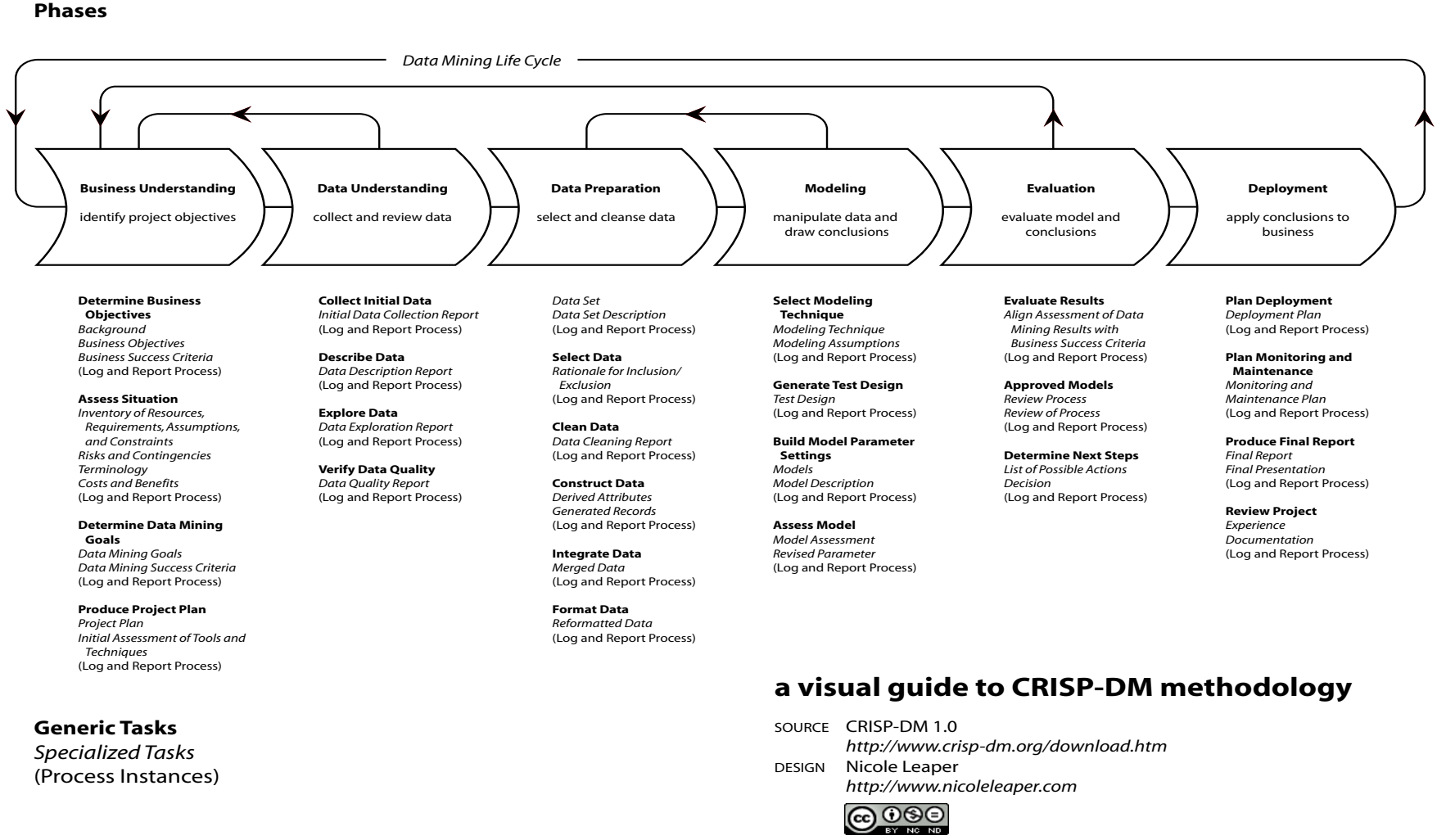


Figure 2.1: A Visual Guide to CRISP-DM Methodology [CRISP-DM.org and Leaper, 2009]

2.1.2 Data Mining Tasks

The different data mining techniques used in accomplishing the stages in Section 2.1.1 can be classified according to different groups, depending on the kinds of knowledge to be discovered (i.e. problem to be solved), the kinds of databases to be mined, and the computing tools available.

In terms of tasks, data mining techniques can be classified into two categories: **descriptive** and **predictive** [Han and Kamber, 2006]. Descriptive data mining tasks characterize the general properties of the data in the database. Predictive data mining tasks perform inference on the current data, in order to make predictions.

In terms of problems of intellectual, economic and business interest, data mining techniques can be grouped into the following four major categories:

Classification

Classification is concerned with arranging the data into predefined groups. It consists essentially of examining the features of a newly presented object and assigning it to one of a predefined set of classes. The objects to be classified are generally represented by records in a database table or a file, and the act of classification consists of adding a new column with a class code of some kind. For example, an email program might attempt to classify an email as legitimate or Spam.

The classification task is characterized by well-defined classes, and a training set consisting of pre-classified examples. The task is to build a model of some kind that can be applied to unclassified data in order to classify it.

Common classification algorithms like Nearest neighbour, Naive Bayes classifier and Neural network have been used to address classification tasks such as: [Berry and Linoff, 2000]

- Classifying credit applicants as low, medium, or high risk;
- Choosing content to be displayed on a Web page;
- Determining which phone numbers correspond to fax machines;
- Spotting fraudulent insurance claims;
- Assigning industry codes and job designations on the basis of free-text job descriptions.

All the examples given above have a limited number of classes/discrete outcomes, and the task is to assign any record into one of them.

Regression

Regression attempts to find a function which models the data with the least error. Unlike classification which deals with discrete outcomes, such as yes or no; regression deals with continuously valued outcomes which are estimated from a model learned from a set of training examples that include the numerical outcome. Given some input data, a regression model delivers a value for some unknown continuous variable such as income,height, or credit card balance.

In practice, regression is often used to perform a classification task [Berry and Linoff, 2004]. For example, a telecommunications company might build a model which assigns all of its customers, based on their lifetime value, into one of two classes of "low propensity to churn" and "high propensity to churn". An alternative approach will be to build a model which assigns each customer a "propensity to churn" score. These might be a value from 0 to 1 indicating the estimated probability that the customer will churn. The initial classification task now comes down to establishing a threshold score. Any customer with a score greater than or equal to the threshold is classified as belonging to the "high propensity to churn" class and any customer having a lower score is considered to belong to "low propensity to churn" class.

The regression approach has the great advantage that individual records can be ordered according to the estimate. To see the importance of this, suppose that the telecommunication company has budgeted for a loyalty reward programme as an incentive to keep 5000 of their likely to churn customers. If the classification approach is used and 15000 customers are identified as having a high propensity to churn, it might simply select 5000 customers from the identified 15000 customers. If, on the other hand, each customer has a propensity to churn score, it can instead include the 5000 most likely to churn customers in the loyalty reward program.

Examples of regression task include:

- A credit card company may estimate the amount of money an individual will spend in a year;
- A warranty provider may estimate the number of claims a particular product is likely to generate;
- A supermarket may estimate the number of customers in a particular location.

Data mining techniques well suited to regression tasks include: generalized, logistic, probit, multinomial or neural networks regression models [Dobson and Barnett, 2008].

Clustering

Clustering is like classification but the groups are not predefined, so the algorithm will try to group similar items together. It is the task of segmenting a heterogeneous population

into a number of more homogeneous subgroups or *clusters*, which are not predefined and without the use of any labels. The records are grouped together on the basis of similarity values, calculated using a number of possible similarity measures. It is up to the user to determine what meaning, if any, to attach to the resulting clusters. Clusters of symptoms might indicate different diseases. Clusters of customer attributes might indicate different market segments.

Clustering is often done as a prelude to some other form of data mining or modeling. For example, clustering might be the first step in a market segmentation effort; instead of trying to come up with a one-size-fits-all rule for "what kind of promotion do customers respond to best," first divide the customer base into clusters or people with similar buying habits, and then ask what kind of promotion works best for each cluster [Berry and Linoff, 2000].

There are many clustering methods available, and each of them may give a different grouping of a dataset. The choice of a particular method will depend on the type of output desired, the known performance of the method with particular types of data, the hardware and software facilities available and the size of the dataset. In general, clustering methods may be divided into two categories: *hierarchical and non-hierarchical methods*, based on the cluster structure they produce [Jain and Dubes, 1988].

- **The hierarchical methods** produce a set of nested clusters in which each pair of objects or clusters is progressively nested in a larger cluster until only one cluster remains. The hierarchical methods can be further divided into agglomerative or divisive methods. In agglomerative methods, the hierarchy is built up in a series of *N-1* agglomerations, or fusion, of pairs of objects, beginning with the unclustered dataset. The less common divisive methods begin with all objects in a single cluster and at each of the *N-1* steps divide some clusters into two smaller clusters, until each object resides in its own cluster.

- **The non-hierarchical methods** divide a dataset of N objects into M clusters, with or without overlap. These methods are sometimes divided into partitioning methods, in which the classes are mutually exclusive, and/or the less common clumping method, in which overlap is allowed. Each object is a member of the cluster with which it is most similar, however the threshold of similarity has to be defined.

Affinity Grouping or Association Rules

Affinity Grouping or Association Rules involves determining which things go together. It consist of searching for relationships between variables. It can be used to identify cross-selling opportunities and to design attractive packages or groupings of product and services. For example a supermarket might gather data of what each customer buys. Using association rule learning, the supermarket can work out what products are frequently

bought together, which is useful for marketing purposes. This is sometimes referred to as "market basket analysis" [Gutierrez, 2006]. This kind of market basket data analysis would enable the supermarket to bundle groups of items together as a strategy for maximizing sales. For example, given the knowledge that printers are commonly purchased together with computers, a supermarket could offer expensive models of printers at a discount to customers buying selected computers, in the hope of selling more of the expensive printers [Han and Kamber, 2006].

Affinity grouping can also be referred to as a simple approach to generating rules from data [Berry and Linoff, 2004]. For example, if two items, say biscuits and lemonade, occur together frequently enough, we can generate two *association rules* thus:

- People who buy biscuits also buy lemonade with probability **P1**,
- People who buy lemonade also buy biscuits with probability **P2**.

2.1.3 Data Mining Tasks covered in this Thesis

This thesis mainly covers the clustering and classification data mining tasks. Clustering is used in describing the transactional data and aids in accomplishing the business understanding, data understanding and data preparation phases of the data mining life cycle outlined in Figure 2.1. Classification models are also built, evaluated, deployed and adapted as highlighted in the modeling, evaluation and deployment phases of the data mining life cycle in Figure 2.1.

2.2 The Problem of Mining Transactional Data

In general, a transactional database consists of a file where each record represents a transaction [Berry and Linoff, 2004, Han and Kamber, 2006]. A transaction typically includes a unique transaction identity number (trans ID) and a list of the items making up the transaction (such as items purchased in a store). The transactional database may have additional tables associated with it, which contain other information regarding the sale, such as the date of the transaction, the customer ID number, the ID number of the salesperson and of the branch at which the sale occurred, and so on.

Transactional data are time-stamped data, collected over time, at no particular frequency. Some examples of transactional data are

- Web log data;
- Point of Sales (POS) data;
- Retail data;

- Inventory data;
- Call Center data;
- Trading data.

Transactional data apart from being traditionally large, is also inherently sparse, due mainly to the underlying process from which they are generated. For example, in retail transactional data, where it is usual for customers to purchase only a very small fraction of products, the average size of a basket (i.e., the collection of items that a customer purchases in a typical transaction) might contain just 3-4 products out of 1,000s of products in the retailer's catalogue/inventory. Such a transaction when represented in an attribute-vector representation will have an average of 3-4 out of 1000s of product attributes that are not null. This implies that the fraction of non-zero attributes on the table (i.e. the sparsity factor) will be 3/1000 - 4/1000, or 0.3 - 0.4%. Figure 2.2 depicts the distribution of a typical transactional dataset.

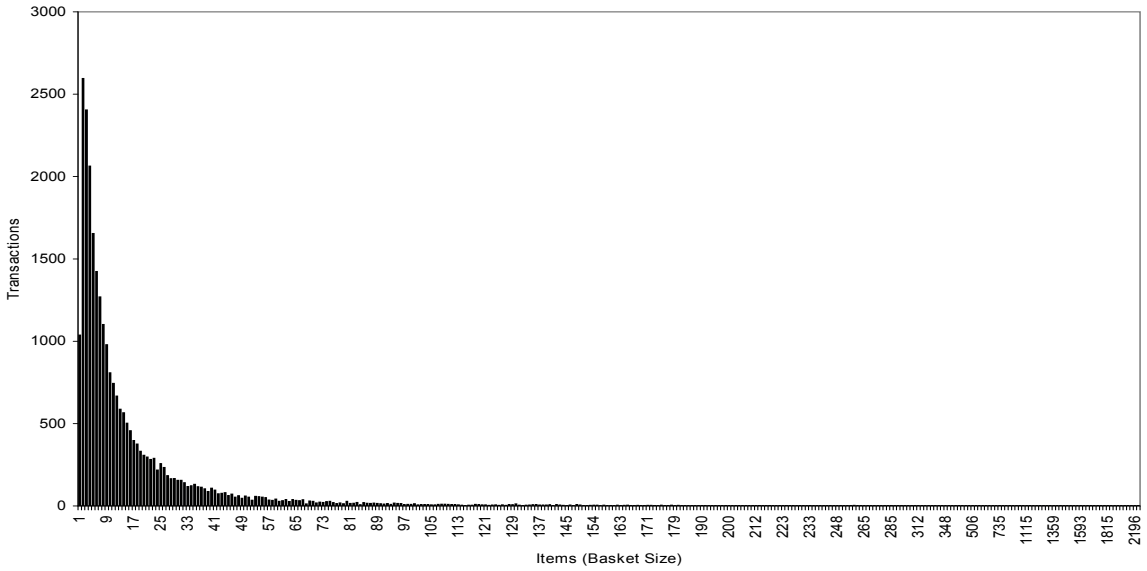


Figure 2.2: Plot depicting the distribution of items per transaction (Basket size) of a typical transactional dataset sourced from Screwfix Limited

Furthermore, the data mining process typically involves using the attribute-vector representation of the transaction, to build a predictive model consisting of the identified relevant independent variables which best minimize prediction error. Although processing power has continued to increase, performing the data mining process on the entire dataset available can be prohibitive in terms of time and finance.

This prohibition can be attributed either to the design structure of some data mining algorithms or the complexity of the problem to be solved [Nisbet et al., 2009]. Also, even though the data size, in terms of volume, might not be of great importance in solving data

mining problems; analysing all available variables is often computationally impossible in fields such as geodetics, bio-informatics, and finance [Mannila, 2000].

Pre-processing techniques are often employed to reduce the size of the dataset used in building and maintaining data mining based business models.

For sparse data, conventional sampling/feature reduction may not work well, because most of the samples are zeros [Church et al., 2006] while sampling or choosing fixed dataset columns/features from the dataset, as is done in some cases [Gemulla, 2008, Gemulla and Lehner, 2008], is also inflexible because different rows may have very different sparsity factors, leading to each sampled data instance conveying little or no information for accurate inference. The difficulty in performing an accurate inference with a reduced feature space of typical transactional dataset can be seen in Figure 2.3 where the data projection in 2D space is shown using the first 2 principal components. It can also be seen that the data is concentrated around the origin, reflecting the fact that a vast majority of customers buy only few items over long periods of time.

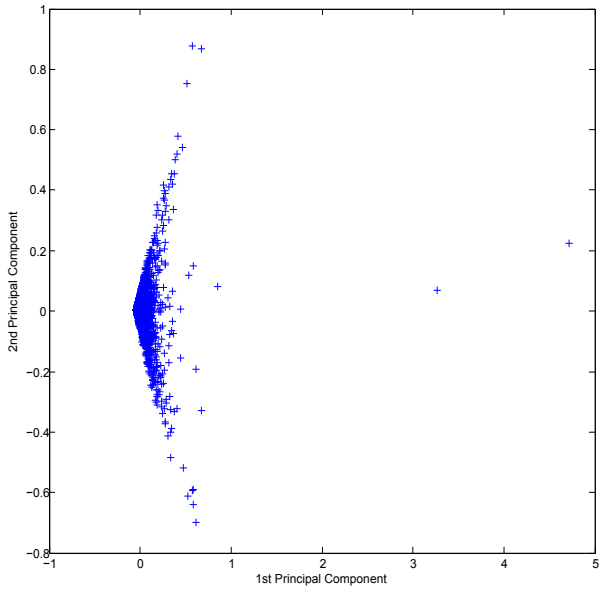


Figure 2.3: Plot showing the projection of the first two principal components of Customer Profiles based on the transactional data of Screwfix's Electricians and PlumbHeaters.

Likewise, automatically clustering the transactional datasets as proposed in [Giannotti et al., 2002, Wang and Karypis, 2004, Yan et al., 2006, 2010], into mutually exclusive partitions is unwieldy with a resultant high misclassification rate of the discovered clusters due to the inherent skewness of transactional datasets.

2.2.1 Issues Encountered when Mining Transactional Data

Companies that have been in business for decades accumulate masses of data about their customers, suppliers, products and services. Also, the rapid pace of e-commerce means

that Web startups can become huge enterprises in months, not years, amassing proportionately large databases as they grow.

Most data mining techniques are, however, not well-suited to learn from evolving disparate data sources that may include multi-channel purchases, descriptive features, estimates based on partial data, video data, etc. as they are designed to operate on a static datasets of interest amassed over time. They take in, as input, a single homogeneous data set consisting of a fixed number of attributes and records, similar to a database relation or query result, with the assumption that the input data contains relatively few records which are generally static in nature.

For example, predictive data mining techniques employ statistical methods which operate on the whole data to produce statistical conclusions for specific patterns in the data. Huge data sets containing millions of training examples with a large number of attributes (tall fat data) are relatively easy to gather for such predictive data mining purposes. However, one of the bottlenecks for successful inference of useful information from the data is the computational complexity of some machine learning algorithms which do not make any assumptions on the form of the underlying function.

Most state-of-the-art non-parametric machine learning algorithms[1] -also known as memory based methods- such as Support Vector Machines, have a computational complexity of either $O(N^2)$ (for prediction) or $O(N^3)$ (for training in situations where solving the quadratic problem and choosing the support vectors directly involves inverting the kernel matrix [Bordes et al., 2005]), where N is the number of training examples [Burges, 1998, Gray and Moore, 2001, Kearns, 1990]. The computational bottleneck at the heart of these algorithms is the multiplication of a structured matrix with a vector, referred to in the literature as Sparse matrix vector product (MVP) [Agarwal et al., 1992].

This seriously restricts the use of these data mining algorithms in the context of mining transactional datasets, as the Sparse matrix vector product requires that all of the available data be retained while making the inference. Interfacing the data mining algorithms with business databases, as an integral part of an organization's data management plan, is also made challenging, as workarounds are often required to deal with the issues of scalability, uncertainty and adaptivity.

2.2.2 Scalability

Most data mining techniques require all the data to be mined to be available in one table. This is counter-intuitive, especially as most real world applications describe complex objects in terms of properties and relations. For example, consider the one-to-many

[1] A learning algorithm is said to be non-parametric if the complexity of the functions it can learn is allowed to grow as the amount of training data is increased. The term non-parametric has been restricted in some publications to learning algorithms in which the learned function is expressed directly in terms of the training examples, e.g., the nearest-neighbor classifier.

depiction of a customer's many purchases at a store in Figure 2.4: A data mining task such

CustomerTable

CustomerID	Name	Gender	Address
1	Lee Jones	Male	FD, 1A High Street
2	Rob Smith	Male	First Dimensions, 1A High St
3	Dave Owens	Male	Opus Group, 200 Main Rd.
4	Bob Smith	Male	RBS, The Lodge
5	Alex Locke	Male	FD, 200 High Rd.
6	Jane Oshay	Female	Socx, 9 The Avenue

TransactionTable

TransactionID	CustomerID	Date of Purchase	ProductPurchased
1	1	01-03-06	Hammer
2	1	02-03-06	Screw driver
3	3	01-03-06	Drill bits
4	2	01-04-06	Water pipes
5	3	09-04-06	Screws
6	5	08-04-06	Nails

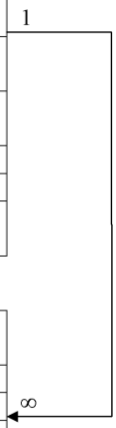


Figure 2.4: One-to-Many Customers Purchases

as basket analysis will require that the data be in a single table as depicted in Table 2.1 obtained by performing a join on the data in the Customer Table and Transaction Table in Figure 2.4.

CustomerID	DateOfPurchase	ProductPurchased	Gender
1	01-03-06	Hammer	Male
1	02-03-06	Screw driver	Male
3	01-03-06	Drill bits	Male
2	01-04-06	Water Pipes	Male
3	09-04-06	Screws	Male
5	08-04-06	Nails	Male

Table 2.1: Example of a Table of Customer Transactions

However, performing a join to form Table 2.1 restricts the meaning/usefulness of the table to just details of customer transactions and will result in the loss of meaning/usefulness of the resultant table if the detailed information (e.g. Address) on each unique customer who has made a purchase is required by the business.

Alternatively, we could aggregate the customer purchases in an effort to include more customer details whilst attempting to exclude redundant information to obtain Table 2.2. This however results in loss of information as whilst our aforementioned joining goals have been met and we know the number of purchases made; the exact dates of the individual purchases, together with the exact products purchased, are, however, unknown.

The work of putting all of an established retail organization's data into a single table is further made complex because current databases are much too large to be held in main

CustomerID	DateOfLastPurchase	NumberOfPurchases	Gender	Address
1	02-03-06	2	Male	1A High Sreet
2	02-04-06	1	Male	1A High St
3	09-04-06	2	Male	200 Main Rd
5	08-04-06	1	Male	200 High Rd

Table 2.2: Example of a Table of Customer-Details Transactions

memory. Retrieving data from disk is markedly slower than accessing data in RAM. Thus, to be efficient, the data-mining techniques applied to very large data[2] must be highly scalable[3]. An algorithm is said to be scalable if, given a fixed amount of main memory, its runtime increases linearly with the number of records in the input database.

Research on the scalability of mining algorithms tends to focus on either:

1. developing special-purpose scalable implementations of existing well-known algorithms that are guaranteed to return the same result as the original (naive) implementation, but that typically will run faster on large data sets or

2. derive new approximate algorithms that inherently have desirable scaling performance by virtue of relying on various heuristics based on a relatively small number of linear scans of the data.

There have been several clustering algorithms [Bradley et al., 1998, Lara and Barandela, 2005, Zhang et al., 1996], association rule algorithms [Agrawal and Shafer, 1996, Agrawal and Srikant, 1994, Zaïane et al., 2001] and classification algorithms [Shafer et al., 1996, Srivastava et al., 1997] that were designed to achieve scalability by processing the data points in an incremental manner, or by processing the data points in small batches. However, these algorithms still treat all the objects of the data set the same way without making any distinction between old data and new data. Therefore, these approaches cannot possibly handle evolving data, where new concepts emerge, old concepts die out, and existing concepts change.

Ganti et al. [1999] survey a broad range of scalability issues which algorithms that address three data-mining problems: market basket analysis, clustering, and classification have to overcome. Grossman [2001], Grossman and Guo [2002] describe some approaches and specific techniques for scaling data mining algorithms to large data sets through parallel processing, that have been used to implement data mining algorithms that are

[2]The term "Large Data" is often used in the data mining literature to mean data that does not fit into the memory of a single processor.

[3]The term "scalable" is used in the data mining literature to refer to data mining algorithms that scale gracefully and predictably (e.g. linearly) as the number of records n and/or the number of variables p grow.

efficient in both time and space when dealing with very large data sets. However, the results obtained from these techniques assume the data being mined is static and does not represent a continuously changing environment.

2.2.3 Uncertainty

The rapid business changes, coupled with the rapidly growing, massive databases prevalent in retail, all add up to the increase of uncertainty involved in any problem-solving situation. This results from some information deficiency pertaining to the system within which the problem-solving situation is conceptualized. Information deficiencies which manifest themselves as incomplete, imprecise, fragmentary, unreliable, vague, or contradictory information become even more pronounced in data mining system models, where putting all the data into a single table, in compliance with the fundamental requirement of data mining algorithms, results in loss of information/meaning.

The uncertainty can also arise due to implementing the following workarounds in an attempt to achieve efficient implementation of data mining algorithms:

- Sampling and filtering of large data sets leads to uncertainty about how the samples differ from each other and the overall data distribution.
- The data mining task being undertaken can also lead to uncertainty. For instance, uncertainty about the nature of the data, such as the occurrence of missing data or latent data values.

In general, these various information deficiencies determine the type of the associated uncertainty [Klir, 2005].

Many conceptual approaches such as Probability and Fuzzy Logic together with their related concepts such as Probability Theory and Fuzzy Sets have been formulated to handle[4] uncertainty in data and models of imprecise knowledge discovered from using data mining techniques.

Proponents of the probabilistic approach to handling uncertainty assert that fuzzy logic, whilst having a moderately large following, remains rather controversial and lacks the sound theoretical backbone and widespread application and acceptance of probability.

Unwin [1986] in distinguishing between the two forms of uncertainty that arise in risk and reliability analyses, i.e.:

1. that due to the randomness inherent in the system under investigation and,
2. that due to the vagueness inherent in the assessor's perception and judgement of that system;

[4]The term "handle" is used here to refer to the ability to measure the amount of uncertainty obtained in generating the results pertaining to a given problem-solving situation. [Klir, 2005]

proposed that, whereas, the application of the probabilistic approach to the first variety of uncertainty is an appropriate one, the same may not be true of the latter. This is due mainly to the capability of fuzzy set theory to provide a formal framework for the representation of vagueness, through seeking to quantify the imprecision that characterizes our linguistic description of perception and comprehension.

In advocating *information mining*[5], Kruse et al. [1999] emphasize the appropriateness of fuzzy set methods in the data mining/knowledge discovery process as solutions obtained using fuzzy approaches are easy to understand and to apply due to their closeness to human reasoning.

We now present an overview of the fuzzy logic approach and probabilistic approaches of handling uncertainty.

Fuzzy Approach to Handling Uncertainty

Fuzzy logic is logic of fuzzy sets introduced by Lofti Zadeh in [Zadeh, 1965] to deal with variables with values in the interval [0, 1]. A Fuzzy set has, potentially, an infinite range of truth values between one and zero. Fuzzy sets provide a means of defining a series of overlapping concepts for a model variable, since it represents degrees of membership. The values from the complete universe of discourse for a variable can have memberships in more than one fuzzy set [Cox, 2005].

Propositions in fuzzy logic have a degree of truth, and membership in fuzzy sets can be fully inclusive, fully exclusive, or some degree in between. The fuzzy set is distinct from a crisp set[6], in that it allows the elements to have a degree of membership. In a fuzzy set, transition between membership and non-membership is gradual rather than abrupt; each member is given a degree of membership between 0 (non-membership) and 1 (full membership), such as 0.27 or 0.75.

Each fuzzy set is defined in terms of a relevant crisp universal set by a function known as a *membership function*, which is analogous to the characteristics function of crisp sets.

Fuzzy logic's capability of supporting, to a reasonable extent, human type reasoning in natural form, by allowing partial membership for data items in fuzzy subsets, has put it in good stead for integration with data mining techniques, to handle the challenges posed by the massive collection of natural data and modelling of imprecise and qualitative knowledge, as well as handling of uncertainty at various stages [Kruse et al., 1999].

The fuzzy approaches to handling uncertainty exploit concepts of fuzzy theory and assign data samples into classes with different degrees of belief. By so doing, fuzzy approaches to handling uncertainty consider that data samples belong to several classes at

[5]Information mining is the non-trivial process of identifying valid, novel, potentially useful,and understandable patterns in heterogeneous information sources.

[6]Boolean sets are often called *crisp sets* as a way of distinctly differentiating them from fuzzy sets, as well as a way of indicating the sharpness of crispness of their membership function

the same time with different degrees.

Probabilistic Approach to Handling Uncertainty

Probabilities are descriptions of the likelihood of some event occurring (ranging from 0 to 1).

Probability is usually thought of in terms of relative frequencies and subjective inference.

The relative frequencies view of probability takes the perspective that probability is an objective concept. In particular, the probability of an event is defined as the limiting proportion of times that the event would occur in repetitions of the essentially identical situations, e.g. the number of times head comes up in an unbiased coin toss.

The subjective inference view of probability takes the perspective that probability is an individual degree of belief that a given event will occur. Thus, probability in the subjective context is not an objective property of the outside world, but rather an internal state of the individual - and may differ from individual to individual.

The principles and methodologies for data analysis that derive from the subjective point of view are often referred to as Bayesian statistics and they form the central tenet for the explicit characterization of all forms of uncertainty in data analysis problems.

Using Bayesian statistics, subjective probability provides for a very flexible framework for modeling different forms of uncertainty, such as, uncertainty about any parameters estimated from the data, uncertainty as to which among a set of model structures are best or closest to "truth", uncertainty in any forecast made, etc. [Hand et al., 2001].

The probabilistic approach to handling uncertainty involves building a probability model and using it to estimate the probability that a data sample belongs to a class as follows:

For each data sample vector, $\boldsymbol{x}$, the probability that it belongs to each of the classes $C_i\,(i = 1, \ldots, n_c)\,,\ P(C_i\,|x\,)$, is estimated. The sample vector, $\boldsymbol{x}$, is then assigned to the class for which its probability of belonging is maximum [Vazirgiannis et al., 2003].

2.2.4 Adaptation

Due to the dynamic nature of doing business, businesses tend to constantly change their processes in order to adequately respond to changing market forces or market growth. Data mining models tend to, therefore, become obsolete as soon as the business process they model changes.

Data analyst or system administrators with knowledge of the domain of application can tune the models by adjusting the model parameters, or by entering specific patterns that will trigger notifications of change in the business environment. Unfortunately, determining which potential parameters or patterns will be useful is a time-consuming process

of trial-and-error. Moreover, the patterns of the retail business environment as determined by businesses and consumer behaviour in response to market forces are dynamic. Businesses and consumers constantly change their strategies and spending behaviour in response to constantly changing economic environment. By the time a data mining model/system is manually tuned, the economics driving the business function it models may have changed significantly.

For all these reasons, it is important that a data mining model adapt easily to new conditions. It should be able to notice new patterns of business/customer behaviour. It should also be able to modify its change notification, for example, as the business function modelled or customer behaviour profile changes.

Holland [1992] provides a nature-inspired conceptual framework for adaptive systems. He defines adaptive systems as a broad class of problem solving and data analysis techniques, that derive their inspiration from highly abstracted models of naturally occurring processes consisting of four components:

1. an environment or, more correctly, the input from the environment to which the adaptive system adapts. In data mining the data set is the input. The structures adapt themselves to the information contained within the data set, with the result that structures are produced that provide a good description or explanation of this information.

2. a set of structures which are progressively modified. These structures constitute the basis of the adaptive process, being largely determined by field of study. In data mining, the mined models are essentially the structures that form the basis of the analysis and adaptive process.

3. an adaptive plan which modifies the system structures, i.e. the models. The models/structures in an adaptive system are modified in response to input datasets (i.e. the environment) under the control of the adaptive plan. In data mining, the type of adaptive plan used depends on the type of models being used in the problem solving or data analysis activity. The adaptive plan results in a progressive, incremental and probabilistic modification of the system's structure. The field of evolutionary computation has extensively studied this type of plan [Bäck, 1995, Biethahn and Nissen, 1995]. Other forms of adaptive plans are analytic (on which multivariate statistics is based) [Friedman, 1991, Hastie et al., 2009], hill-climbing (used by standard neural networks) [Chalup and Maire, 1999], and plans embedded in the structure itself (as in ant colony optimization) [Dorigo and Di Caro, 1999]. More recently, social adaptation, based on communities of interacting agents, have led to new fields such as memetic algorithms [Krasnogor, 2008].

4. a measure of the performance of each structure/model, or fitness, which provides

feedback on how well the structure/model has represented the environment, solved the problem, or explained the nature of the input dataset. In data mining, this usually consists of maximizing or minimizing a cost or objective function by systematically choosing input values from within an allowed set and computing the value of the function to measure the error or decide which model is best.

According to Holland [1992], a system undergoing adaptation is largely characterized by the mixture of operators acting on the structures at each stage. The set of factors controlling this changing mixture - the adaptive plan - constitutes the workings of the system as far as its adaptive character is concerned. The adaptive plan determines just what structures arise in response to the environment, and the set of structures attainable, by applying all possible operator sequences which mark out the limits of the adaptive plan's domain of action. Since a given structure performs differently in different environments - the structure is more or less fit - it is the adaptive plan's task to produce structures which perform "well" (are fit) in the environment confronting it. "Adaptations" to the environment are persistent properties of the sequence of structures generated by the adaptive plan.

Thus, an adaptive system, as described by [Holland, 1992], undergoes a progressive modification of its component structures/models. The rate and direction of this modification is controlled by feedback indicating how well the structures/models are explaining the available data.

Within the data mining framework, depicted in Figure 2.1 of Section 2.1, adaptation implies implementing a model management in such a way that models are periodically evaluated in an automatic or semi-automatic fashion. In this sense, model management is tightly coupled with the modelling, evaluation and deployment phases of Figure 2.1. It can be seen as an outer-loop controlling these steps as shown in Figure 2.5.

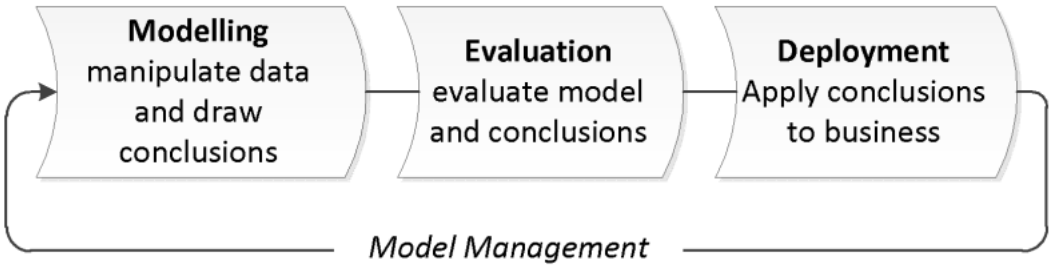


Figure 2.5: Model Management Cross Section of CRISP-DM

An approach for implementing model management is the champion/challenger testing strategy [Berry and Linoff, 2004, Nath, 2007, Nisbet et al., 2009]. This implementation involves comparing the performance of a model currently being used (the champion) against that of new models built on more recent data (the challengers). If the challenger

model outperforms the champion model, it becomes the new champion and is deployed in the production system. Challenger models are built periodically as new data are made available.

Another aspect of model management is the creation and management of metadata about models [Ari et al., 2008a,b, Jain et al., 2008]. Model metadata can be used to dynamically select, for scoring, models appropriate for handling a particular data mining task. An example of this aspect of model management is the encoding of models into XML files using Predictive Model Markup Language (PMML) [Pechter, 2009]. PMML files may contain one or more mining models along with their data dictionaries, model schemas, and data transformations.

A crucial shortcoming of the aforementioned approaches to adaptation in data mining is the lack of a precise definition of the time context in which the adaptation is performed.

Furthermore, the range of models (such as association rules, clusters, regression results and rule-based/tree models) that can be used for the data mining task vary vastly in their applicability and in their representation of the input dataset. Determining when and how to effectively adapt them in data mining activities is usually time consuming. Also, the other components of an adaptive system, the environment (i.e. the data set), the adaptive plan (i.e. the process of modification of models), and the fitness (the level of performance of the models), all add up to make adapting the models, whilst undertaking complex data mining tasks, such as learning under concept drift (e.g. a retail model of seasonal customer purchases) more challenging.

In machine learning, changing concepts are often handled by using a time window of fixed or adaptive size on the training data [Widmer and Kubat, 1996], or weighting data or parts of the learned model according to their age and/or usefulness for the classification task [Taylor et al., 1997].

For windows of fixed size, the choice of a "good" window size is a compromise between fast adaptability (small window) and good and stable learning results in phases without or with little concept change (large window) [Klinkenberg and Renz, 1998]. The basic idea of the adaptive window management, on the other hand, is to adjust the window size to the current extent of concept drift. Indicators such as performance measures (e.g. the accuracy of the most recent model), properties of the model (e.g. the complexity of the most recent tree structure) and properties of the data (e.g. class distribution) are usually monitored to detect concept drift.

Thus, most adaptive algorithms (such as the FLORA algorithms [Widmer and Kubat, 1996], Very Fast Decision Tree (VFDT) [Domingos and Hulten, 2000], Concept-adapting Very Fast Decision Tree (CVFDT) [Hulten et al., 2001]) in the machine learning literature, incorporate one or more of the following components:

- windows to remember recent examples;

- methods for detecting the change in complexity or distribution in the input dataset;
- methods for determining updated estimations for some statistics of the input dataset.

For the work in this thesis, the aforementioned components are viewed as the basis for solving the following three central adaptive system problems:

1. What needs to be remembered or forgotten?
2. How can the change(s) in the model be monitored and detected?
3. How can the change(s) in the input dataset be measured?

The goal is to demonstrate that by basing mining algorithms on well-designed, well-encapsulated modules for these tasks, more adaptive and more flexible solutions for mining transactional data can be obtained than by using ad-hoc data mining techniques.

2.3 Summary

This chapter presented background overview of approaches and techniques for data mining. The issues inherent in transactional data which makes mining them for knowledge challenging were also discussed. The chapter concluded with a discussion of the current research challenges of stability, uncertainty and adaptation encountered when mining transactional data. In Chapter 3, a detailed description of transactional data will be provided together with techniques for preprocessing them in order to construct effective customer profiles for the purpose of implementing business applications, such as, personalization and recommender systems. A detailed description of the Screwfix's transactional data used in this thesis is also given.

Chapter 3

Transactional Data Description and Preprocessing

The key goal in utilizing the collected transactional data for data mining, is to create a set of customer-centric data models (customer profiles) comprising the interests and behaviour of all customers, that can be used as input to a variety of data mining algorithms for knowledge discovery. The output from these algorithms, i.e., the discovered knowledge, can then be used both for describing and predicting the interests and behaviour of the customers. The exact representations of these customer models differ based on the approach taken to model the customers and the granularity of the information available. The data mining tasks therefore differ in complexity based on the expressiveness of the customer profile representation chosen and the data available.

One common approach to customer modelling is the grouping of customers with similar buying behaviour. In this way, if a customer is found to belong to a particular group of customers with similar buying behaviour, then that customer may be expected to have similar interests and behaviour with the rest of the customers belonging to the same group. Thus, for example, inferences can quickly be made on an individual customer based on the behaviour of other customers.

However, as outlined in Figure 2.1, before data mining techniques can be applied to create the customer-centric data models, the raw transactional data must undergo a series of preprocessing steps so as to determine the business' data mining goals as well as explore, describe and transform the raw transactional data into an appropriate format for subsequent processing. The typical preprocessing steps include [Han and Kamber, 2006]:

1. Feature extraction - to identify relevant attributes for a data mining task using techniques such as, event detection, feature selection, and feature transformation (including normalization and application of Fourier or wavelet transforms).

2. Data cleaning - to resolve data quality issues such as, discrepancies, noise, outliers, missing values, and mislabelling errors.

3. Data reduction - to improve the processing time or reduce the variability in data by means of techniques such as, statistical sampling and data aggregation.

4. Dimension reduction - to reduce the number of features presented to a data mining algorithm; principal component analysis (PCA), ISOMAP, and locally linear embedding (LLE), are some examples of linear and non-linear dimension reduction techniques.

Nevertheless, as discussed in Chapter 2, data preprocessing is a challenging task which tends to take up to 80% of the data mining task.

This chapter presents the results obtained from the initial analysis and preprocessing Screwfix's transactional data so as to gain insight into the business objective for data mining and the problem of mining transactional data to create customer-centric data models..

It commences in Section 3.1 with a detailed description of the real world transactional data obtained from Screwfix Ltd. The architecture of the data warehouse of the transactional data is then given, together with their method of collection and their basic statistics.

The results obtained from clustering and normalizing the transactional data are then presented in Section 3.2. Section 3.3 then presents the effect of conventional statistical sampling on the predictability of 4 classifiers

The chapter concludes with a summary and discussion of the findings obtained from the exploratory analysis of Screwfix's transactional data in Section 3.4.

3.1 Description of Screwfix's Transactional Database

Screwfix's transactional data going back to around 1999 is stored in a Sybase RDBDMS (i.e. The MIS in Figure 3.1).

A new system was recently introduced as part of Screwfix's goal for a better and higher quality data management. As part of the transition to the new system, a copy of each day's data is sent via an ETL system to the MDM oracle DB as shown in Figure 3.1. The new database has been optimized for quick access to aggregated data (e.g. number of orders taken, number of active customers, etc.) as well as data cleansing.

The data is also stored in Alterian - a database specially built for holding both transactional and non-transactional information on customers for marketing purposes.

The data held on the customers comprises of demographic details (i.e. name, gender, birth date, address, etc.) and transaction details (i.e. orders). These records are stored in the Customers, Addresses, Order and Order details tables.

For our experiments, only the customerID in the Customers table and the information from the Order and Order details tables were used.

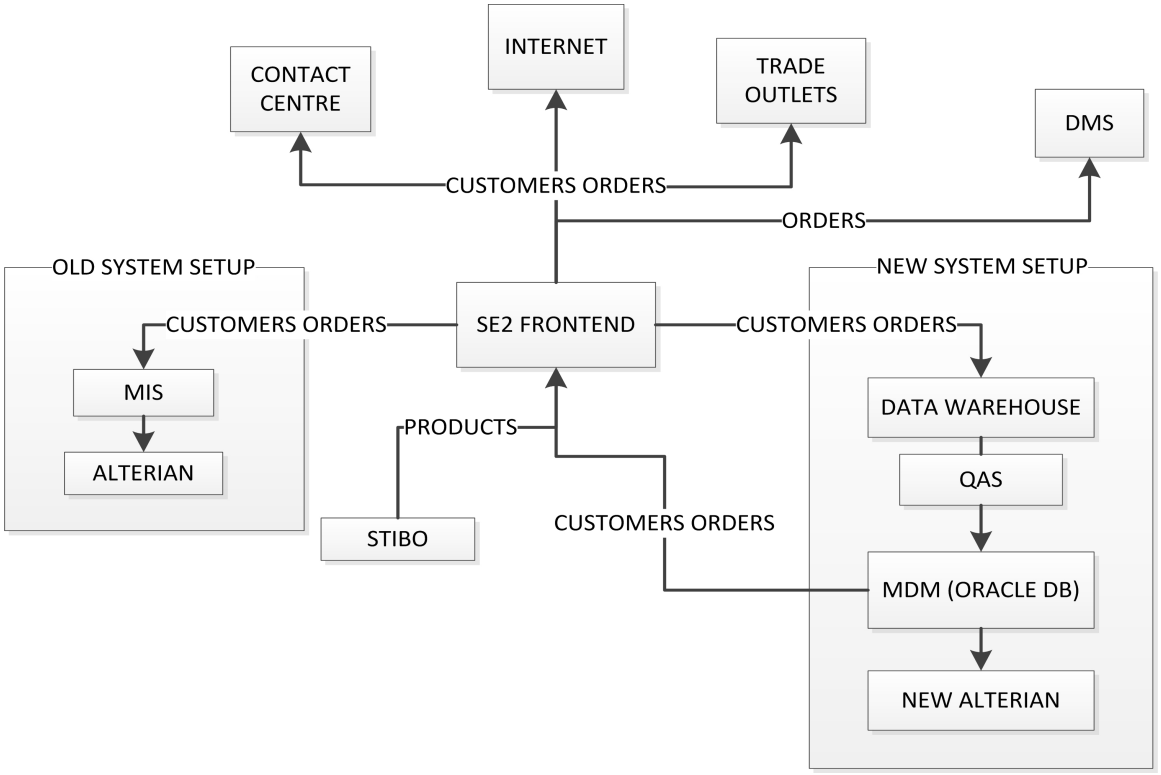


Figure 3.1: Screwfix's Systems Architecture framework, showing the old and new systems set-up

3.1.1 Description of Screwfix's Customers and Trade-types

Screwfix has approximately 7.7 million customers held on their system. These customers are segmented into unconverted, inactive and active. Unconverted (2.8M) and inactive (2.4M) customers make up approximately 5.2M of Screwfix customer base and include customers who have attended trade-shows, made enquiries or requested a catalogue but are yet to make any orders. 1 million of the remaining 2.5 million are considered active (have made at least an order within the last one year).

Screwfix customers are also conceptually grouped on the basis of the information they provide into the 12 trade-types as shown in Table 3.1.

Figure 3.2 shows the total number of each trade-type, categorized on the basis of the information provided by the customers. It can be seen that a large number of the customers (i.e. 293600 NTS trade-type) did not specify their trade-type, a sign that many of the respondents (in this case, customers) did not want to divulge information about themselves.

Two of the 12 trade-types, i.e., **Electrician** and **PlumbHeaters** were therefore verified with the third party professional bodies listed in Table 3.2. Figure 3.3 shows the total number of the verified Electrician and PlumbHeat trade-types.

Many of the verified **Electrician** and **PlumbHeat** customers' trade types where found not to reflect the type-trades provided.

S/N	Trade-Type	Abbreviation
1	Domestic/light commercial electricians	Electrician
2	Bathroom fitters	Bathroom
3	Decorators (Domestic/light components)	Decorator
4	Joiners	Joiner
5	Domestic kitchen fitters	Kitchen
6	Gardening and Landscapers	Landscaper
7	Multi-trade domestic small general contractor	MTD
8	Small facilities/maintainers	Maintenance
9	No Trade Specified	NTS
10	Other Trades	Other Trades
11	Specialist plasterers	Plasterers
12	Small plumbing and heating contractors	PlumbHeat

Table 3.1: Screwfix's Trade-Types

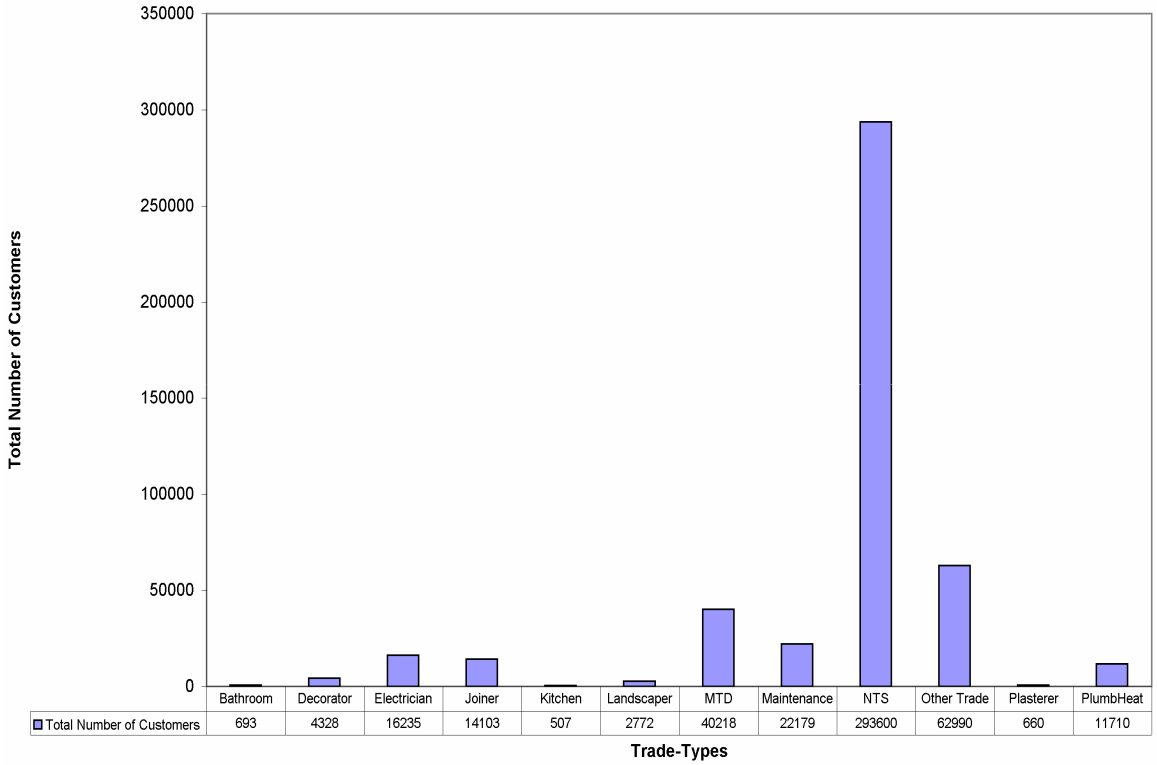


Figure 3.2: Total number of the customers categorized into Trade-Types that made at least one (1) order during the period between the 1st quarter of 2007 and the 2nd quarter of 2009

S/N	Professional Body	Trade-Type
1	Corgi	PlumbHeat
2	CIPHE	PlumbHeat
3	NICEIC	Electrician
4	ECA	Electrician
5	NAPPIT	Electrician

Table 3.2: Verified Trade-Types and their Professional Bodies

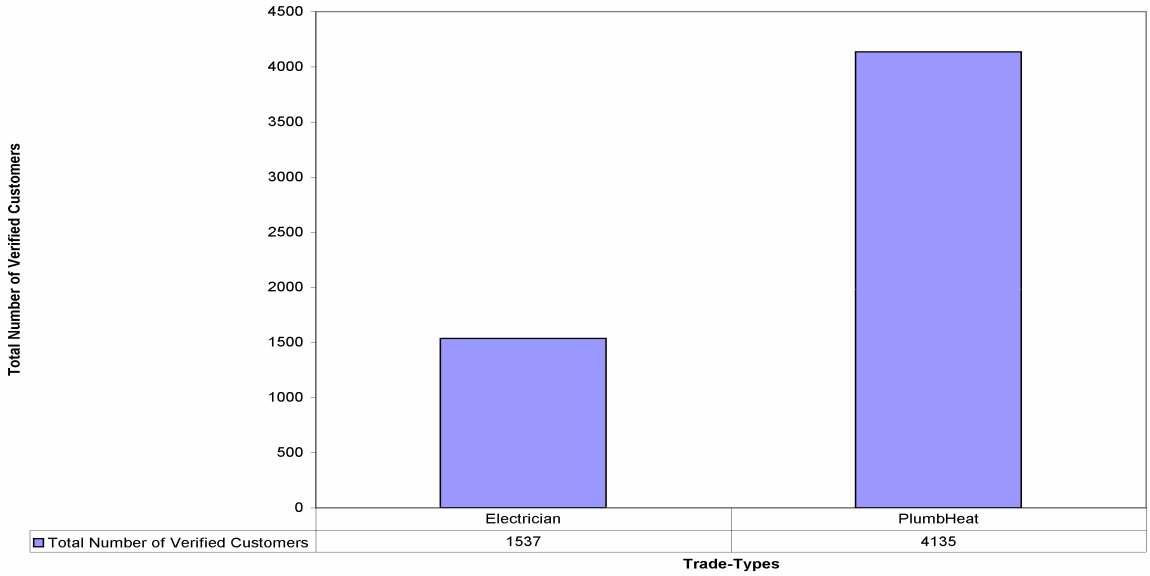


Figure 3.3: Total number of the Customers whose Trade-Types were verified with a third-party

Figures A.1 and A.3 of Appendix A, depict the discrepancies between the trade-type information provided and its verification for the **Electrician** trade-type; while Figures A.2 and A.6 of Appendix A depict the discrepancies for the **PlumbHeat**.

Figures A.4 and A.5 of Appendix A further show the discrepancies between the categorized and verified trade-types in terms of the number of items transacted in each order. The numerical composition in each segment of the items transacted can be found in Tables C.1 and C.2 of Appendix C, for the verified Electrician and PlumbHeat trade-types respectively.

These discrepancies discovered in the provided recorded values and the actual values could seriously affect the quality of the mining results. The inconsistencies contribute to the data being inaccurate for analysis and uncertainty in the results obtained. If the unverified data is used (without resolving the inconsistencies) for the mining process, many of the customers would possibly be put into wrong clusters or incorrectly classified. Also, the discrepancy can cause confusion for the model building procedure, resulting in unreliable models. Although, most mining routines have some procedures for dealing with incomplete or noisy data, they are not always robust to inconsistencies in the data [Fayyad, 1996, Han and Kamber, 2006]. Instead, they may concentrate on avoiding over-fitting the data to the function being modelled [Bishop, 1995, Bramer, 2007]. Section 3.2 further demonstrates the effects the identified discrepancies have on the result obtained from clustering Screwfix's transactional data.

3.1.2 Description of Products

Screwfix currently sell in excess of 15,000 different items. These are grouped hierarchically into 929 groups which are further grouped into 37 topics.

Figure 3.4 shows the columnar and row-wise aggregation of Screwfix's product items by items, groups and topics as defined by the business.

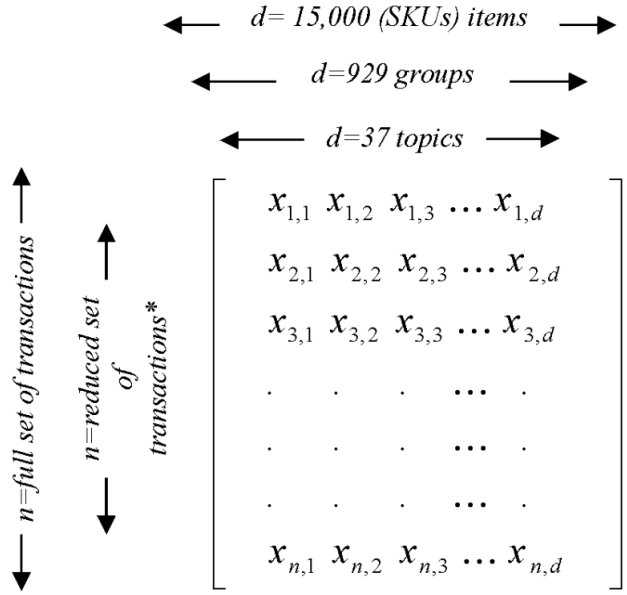


Figure 3.4: View of Screwfix's Defined Product Items.
*The reduced set of transactions is obtained by aggregating the total individual transactions made per customer in the full set of transactions.

The unverified customer trade-types have similar buying patterns as can be seen in the plots of the topics transaction patterns in Figure 3.5 and Figure 3.6. The similarity in the buying pattern of the unverified customer trade-types would make it difficult to build data mining models that will accurately distinguish the customers by their trade-types based on their transactions.

Further numerical details which show the similarity of the buying patterns for the unverified customer trade-types can be found in Tables C.3 and C.4 of Appendix C.

The buying patterns for the verified customer trade-types are, however, more distinguishable as can be seen in the plots of the average Topics transacted by each group of number of Topics per transaction in Figures 3.7 and 3.8; and from the plots of the proportions of the items transacted by the verified trade-types in Figure B.2 and B.1 of Appendix B. It can also be seen from the plots that the PlumbHeat trade-type consistently buy a large proportion of items under the **Plumbing** topic and the Electrician trade-type consistently buy a large proportion of items under the **Electrical** topic. Numerical details of the 'items by topics' transactions for the verified Electrician and PlumbHeat trade-types can be found in Appendix C.

However, the items transacted for the verified trade-types were found, in keeping with

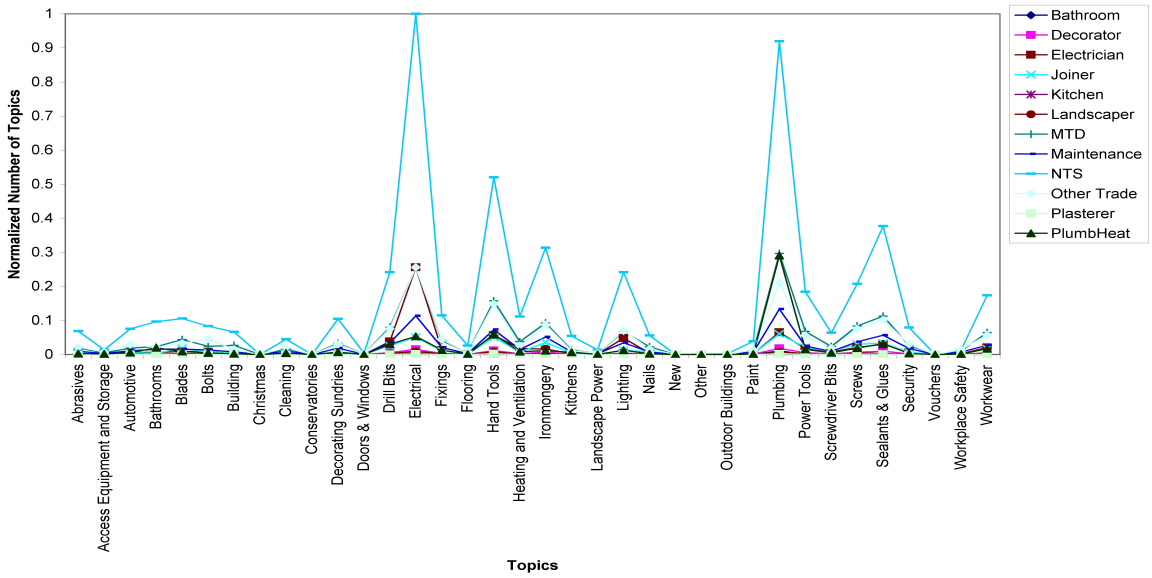


Figure 3.5: Plot showing transaction patterns of the Topics (normalized by scaling the aggregated total number of item topic bought in the range [0.0, 1.0]) for the unverified customer trade-types.

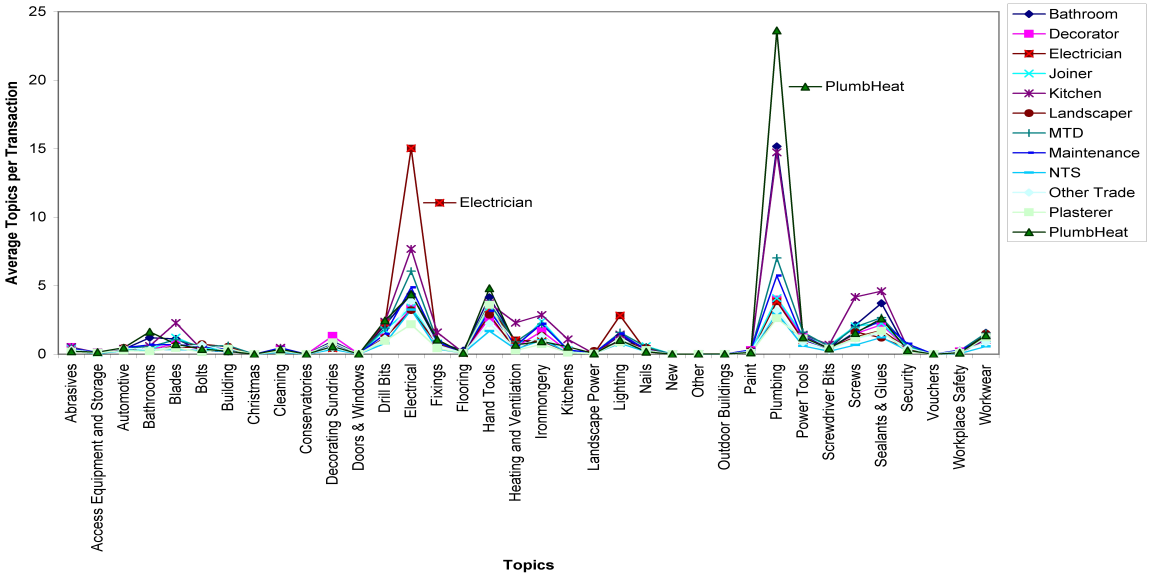


Figure 3.6: Plot showing the average Topics transacted per transaction for the unverified customers' trade-types.

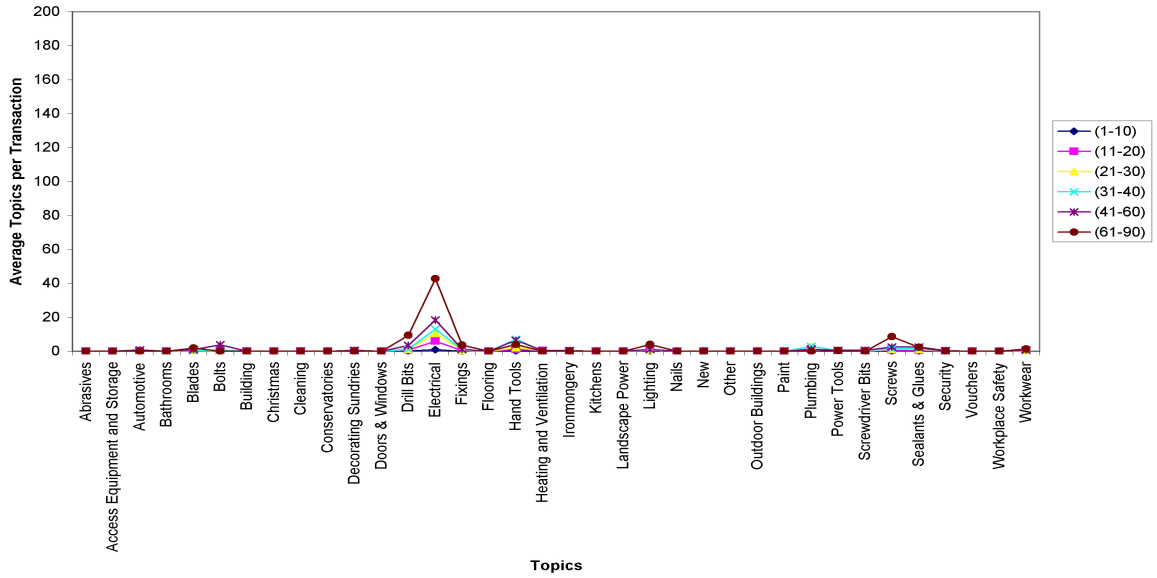


Figure 3.7: Plot showing the average Topics transacted per transactions by the verified Electrician trade-type.

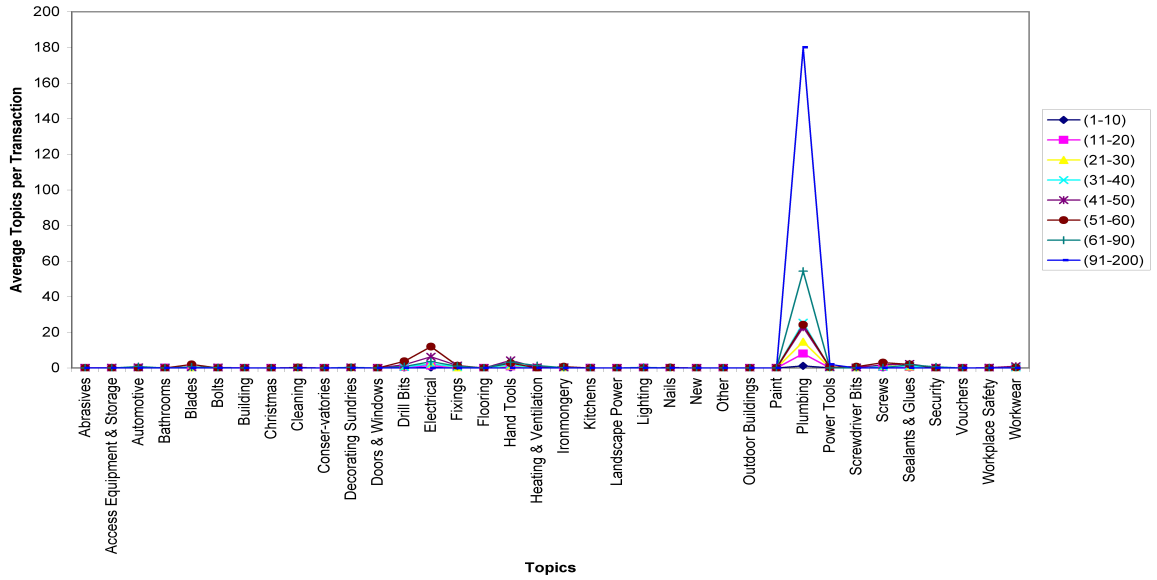


Figure 3.8: Plot showing the average Topics transacted per transactions by the verified PlumbHeat trade-type.

the inherent nature of transactional data, to be sparse and skewed with a high number of trade-types making few purchases as can be seen in the plots in Figures 3.9 and 3.10.

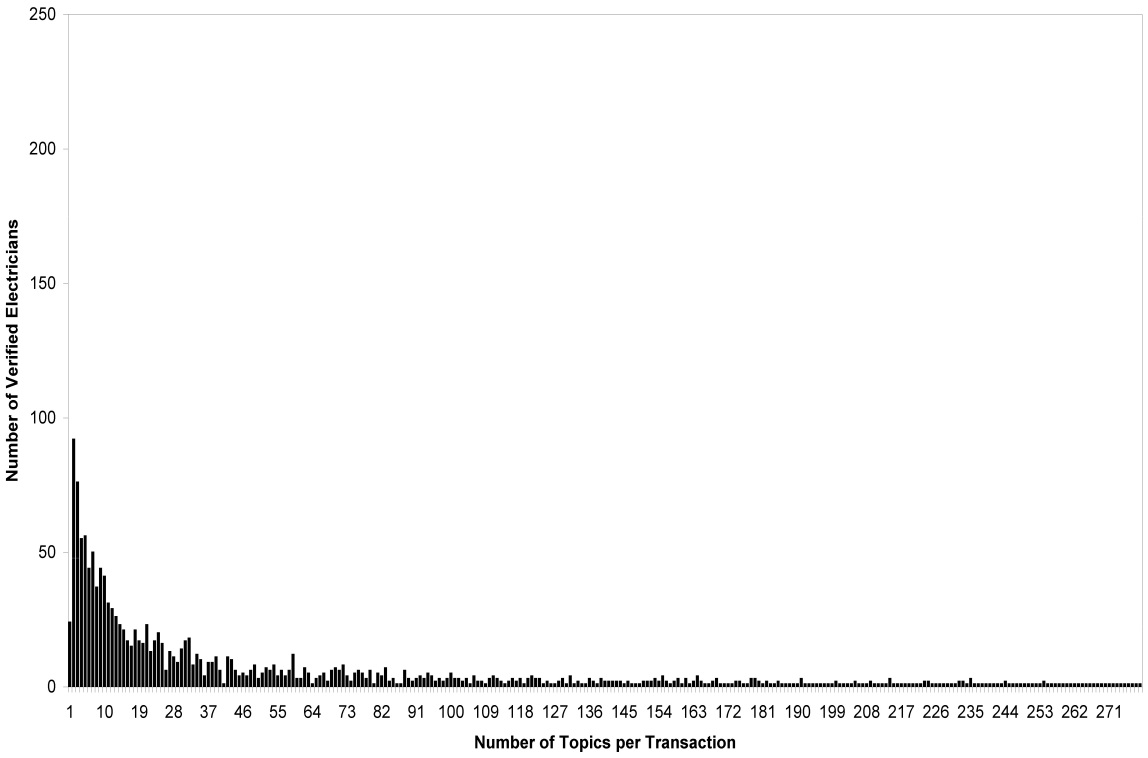


Figure 3.9: Plot showing the positive skewness of topics transacted by the verified **Electrician**

3.2 Exploration of Screwfix's Customer Transactional Data using Clustering Algorithms

In the context of utilizing collected transactional data, it is very helpful to acquire information about costumers' interests and then group customers with similar interests in products. Clustering algorithms are often used to generate descriptive customer models. This section presents results obtained from using Hierarchical, and K-means clustering algorithms to generate descriptive models from Screwfix's transactional data.

3.2.1 Hierarchical Clustering of Screwfix Transactional Data

To gain a better understanding of the relationship between Screwfix's customer Trade-types on the basis of their transactions; a hierarchical clustering analysis was undertaken using the topic transactions for 2007.

Figure 3.11 shows the result obtained from using the Hamming distance as a distance function to cluster the topics transactions data.

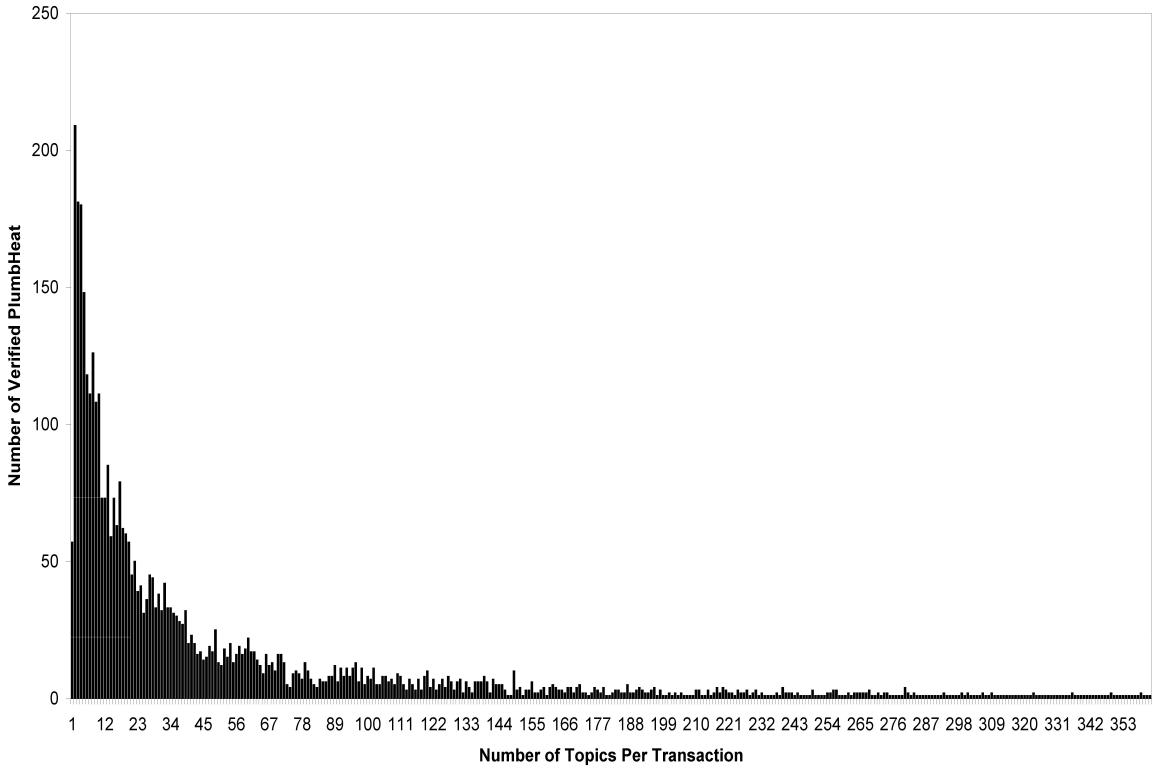


Figure 3.10: Plot showing the positive skewness of topics transacted by the verified **PlumbHeat** trade-type

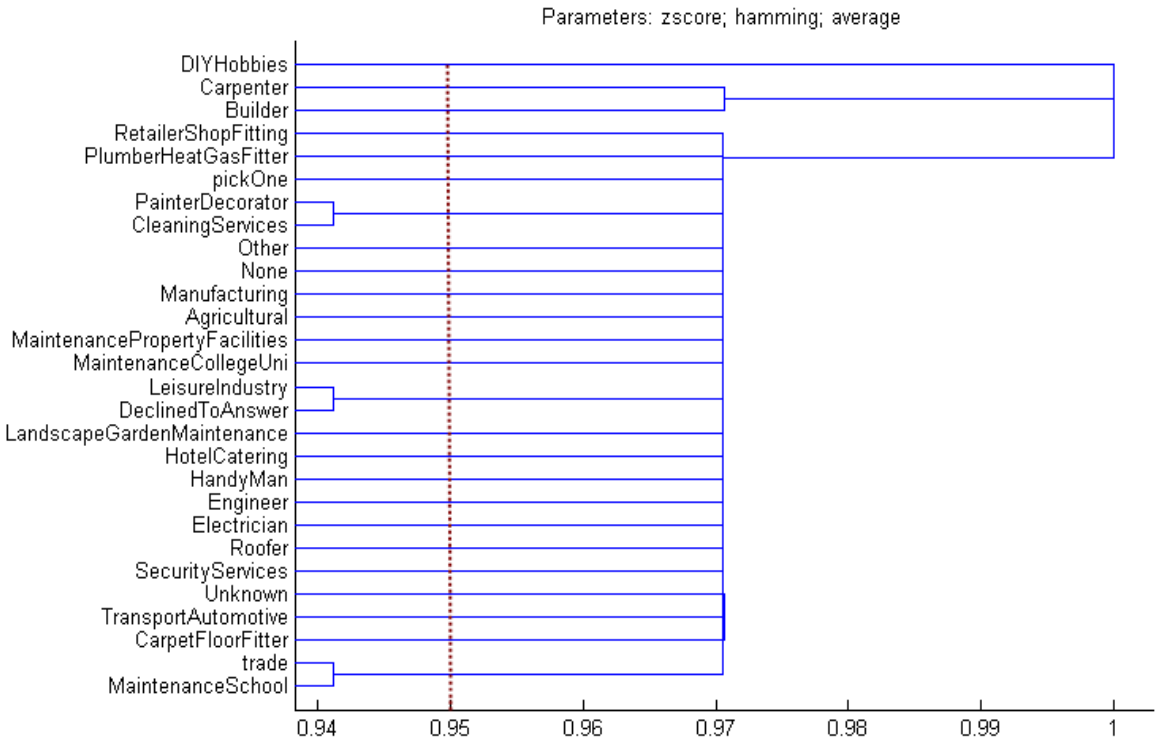


Figure 3.11: Best Dendrogram (with a cophenetic correlation coefficient of 1) of the Trade-Type Topic Transactions for 2007.

There are a few intuitive clusters formed from the data, with trade-types with similar buying patterns grouped together; i.e.:

1. a mixed cluster containing PainterDecorator and CleaningServices trade-types
2. a mixed cluster containing LeisureIndustry and DeclinedToAnswer trade-types
3. a mixed cluster containing Trade and MaintenanceSchool trade-types
4. 22 singleton clusters containing each of the remaining 22 trade-types and thus not revealing any new details about the data.

Each of the three clusters containing 2 trade-types are related by the topic transactions made by the clustered trade-types in 2007. The other 22 trade-types do not group into clusters when the cut is made at a distance of 0.95.

One issue that can be seen from looking at the dendrogram in Figure 3.11 is the inability to assess at a glance the quality of the clusters formed or to estimate the 'correct' number (based on the buying pattern) of trade-type groups in the Screwfix's transactional data.

The upper tail rule, which was developed by Mojena [1977], is one of the well-known criterion function distribution based indexes, which is, $a_{j+1} > {}_{\alpha} + c\sigma_{\alpha}$, where ${}_{\alpha}$ and σ_{α} is the mean and standard deviation of the distribution of clustering criterion value. It is often used to address the uncertainty of the clusters inherent in data. It finds the first biggest jump of the series of the clustering criterion values as the number of cluster, which is in the upper tail of the clustering criterion value distribution for hierarchical agglomerative clustering. If no such number can be found then there is only one cluster.

Applying MojenaŠs upper tail rule to the hierarchical cluster depicted in the dendrogram in Figure 3.11 results in the plot in Figure 3.12.

Whilst the plot in Figure 3.12 gives the likely number of trade-type clusters (i.e. any from 3 upwards) that are inherently in Screwfix's transactional data; the contents of the clusters can not be gleaned from the plot nor can the 'correct' number of trade-type clusters.

Another hierarchical clustering analysis was performed which produced the dendrogram in Figures 3.13 showing the relation between the topics, i.e. topics that were bought together by the Trade-types in 2007.

Table 3.3 shows the Topics in the 9 clusters (from left to right) formed by the dendrogram in Figure 3.13 when cut at 0.05 while Figure 3.14 depicts the upper tail rule's estimates of the Topics in Screwfix's transactional database.

The results obtained from the hierarchical clustering of the Topics, whilst providing insight into the Topics that are purchased together, still does not address the issue of uncertainty (i.e. uniqueness and distinctness) of the clusters formed.

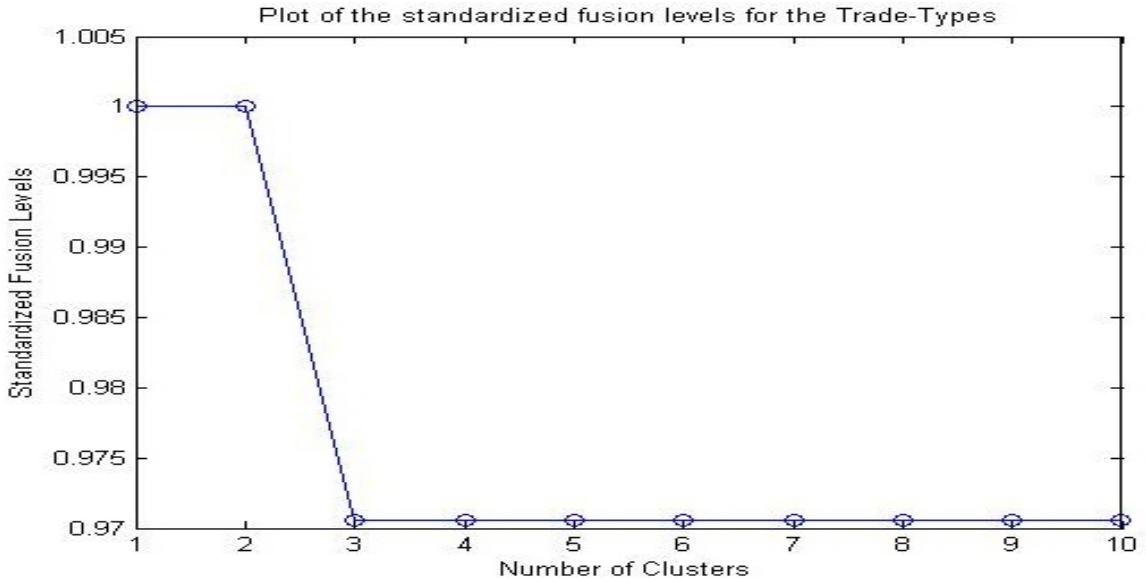


Figure 3.12: This figure, shows the plot of the standardized fusion levels. The 'elbow' in the curves indicates that no less than 3 clusters are reasonable for grouping the Trade-types.

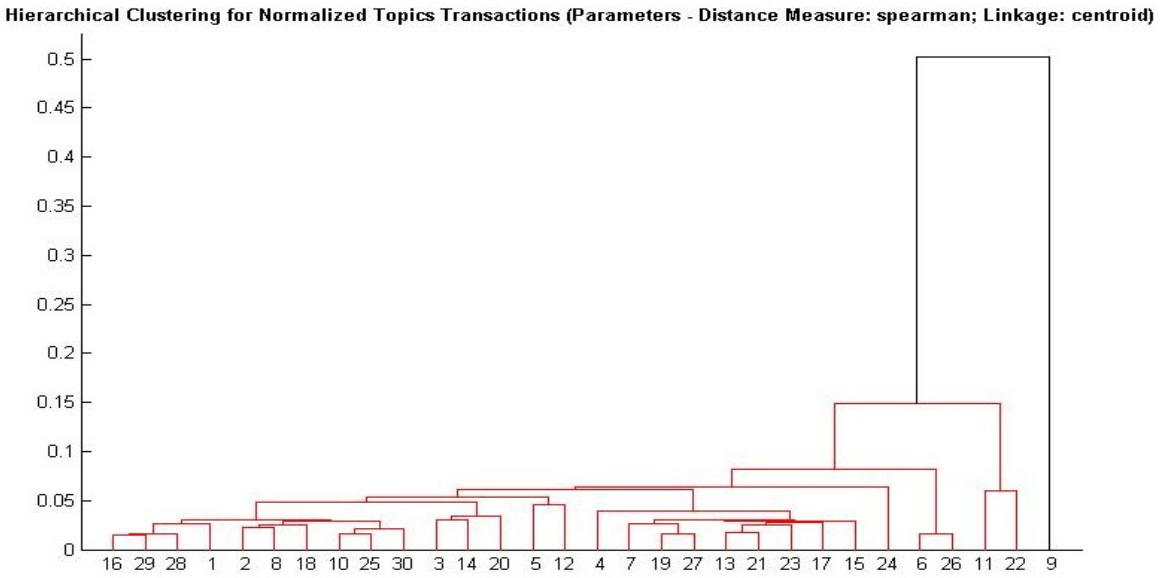


Figure 3.13: The dendrogram shows the results of Topics purchased together by the Trade-types obtained using Spearman distance and centroid linkage on the Topics transactional data for 2007.

Cluster	Leaf Nodes	Topics in Cluster
1	16, 29, 28, 1, 2, 8, 25, 18, 10, 25, 30	Hand Tools, Workwear, Screwdriver Bits, Abrasives, Access Equipment and Storage, Cleaning, Ironmongery, Decorating Sundries, Paint, Sealants Glues
2	3, 14, 20	Automotive, Fixings, Landscape Power
3	5, 12	Blades, Drill Bit
4	4, 7, 19, 27, 13, 21, 17, 15	Bathrooms, Building, Kitchens, Power Tools, Electrical, Lighting, Heating and Ventilation, Flooring
5	24	Outdoor Buildings
6	6, 26	Security, Workplace Safety
7	11	Doors & Windows
8	22	Voucher
9	9	Conservatories

Table 3.3: Description of 9 Clusters of Topics from Hierarchical Clustering.

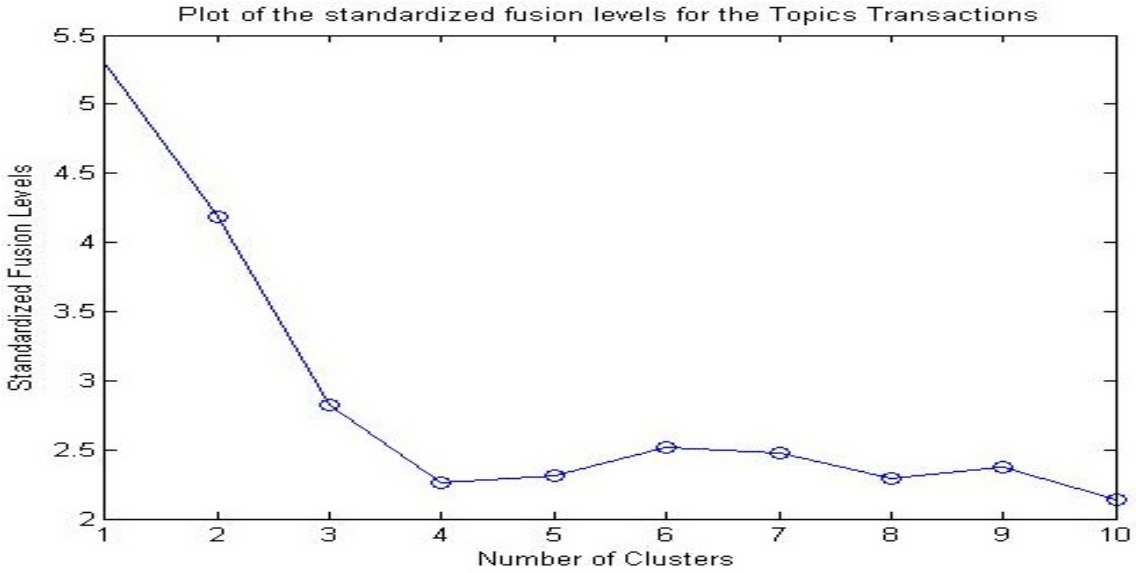


Figure 3.14: This figure, shows the plot of the Topics standardized fusion levels. The 'elbow' in the curves indicates that four clusters is reasonable for the topics transactions data. However, the other 'elbows' at 8 might provide interesting clusters, too.

Another major drawback that can be noticed with using hierarchical clustering techniques for the purpose of descriptive analysis, is that using the linkage techniques leads to confusing results, as can be seen in both Figures 3.11 and 3.13 where both of the average (weighted and unweighted) linkage methods have resulted in the decrease of merged points of some of the clusters formed.

3.2.2 K-means Clustering of Screwfix Transactional Database

In order to gain an alternative insight into the underlying nature of Screwfix's transactional data, a K-means analysis on how the Trade-types and Topics transactions cluster was performed.

Kaufman and Rousseeuw [1990] present the silhouette statistic as a way of estimating the optimal solution (i.e. the 'correct' number of underlying clusters) when using the K-means clustering algorithm. The silhouette for a given observation is computed as follows. For each observation i, compute a_i as the average dissimilarity of observation i with all objects in cluster c:

$$a_i = \bar{d}\,(i, c) \tag{3.1}$$

Let b_i be the minimum of the average dissimilarities $\bar{d}\,(i, c)$, computed in Equation 3.1. The silhouette for the i-th observation is:

$$sw_i = \frac{(b_i - a_i)}{max\,(a_i, b_i)} \tag{3.2}$$

The average silhouette width can be computed by averaging sw_i over all oberservations:

$$s\bar{w}_i = \frac{1}{n}\sum_{i=1}^{n} sw_i. \tag{3.3}$$

Heuristically, the silhouette measures how well matched an object is to the other objects in its own cluster versus how well matched it would be if it were moved to the next closest cluster. Observations with a large silhouette are well clustered, but those with small values tend to be ones that are scattered between clusters. The silhouette sw_i in Equation 3.2 ranges from -1 to 1. If an observation has a value close to 1, then the data point is closer to its own cluster than a neighbouring one. If it has a silhouette close to -1, then it is not very well clustered. A silhouette close to zero indicates that the observation could just as well belong to its current cluster or one that is near to it.

Kaufman and Rousseeuw [2005] used the average silhouette to estimate the number of clusters in the data set by using the partition with two or more clusters that yields the largest average silhouette width. They state that an average silhouette width greater than 0.5 indicates a reasonable partition of the data, and a value of less than 0.2 would

indicate that the data do not exhibit cluster structure.

This thesis uses the silhouette statistics here to investigate the number of clusters inherent in the Topics transacted and the customer Trade-types.

Figures 3.15 and 3.16 show the silhouette plots of the silhouette values for the Topics and Trade-types with each cluster ranked in decreasing order obtained from clustering Screwfix's transactional data for 2007.

In Figure 3.15, the top one indicates large values for cluster 1 and a few negative values for clusters 3. The second plot in Figure 3.15 with 7 clusters shows that there are no negative silhouette values with large values for cluster 1. These plots, like the preceding plot of the standardized fusion levels in Figure 3.14, indicate that cluster numbers above 3 are a fit to the Trade-types; with 7 clusters here being a better fit with mean silhouette value of **0.9269** compared to the mean silhouette value of **0.7687** for 4 clusters.

On the other hand, Figure 3.16 for the Topics K-means clusters indicate large values for cluster 1 and negative values for cluster 3 for the second silhouette plot of 8 clusters Topics.

It can be seen from observing the contents of the 3 clusters of Trade-Types found by the K-means and hierarchical clustering techniques in Figure 3.17 and Table 3.4 that although both the K-means and hierarchical clustering techniques indicate 3 clusters as the best fit for the Topics in Screwfix's transaction data; the composition of the clusters are different; leading to uncertainty about the composition of the grouping inherent in Screwfix's transactional data.

The uncertainty is further enhanced by the overlap of cluster contents in the hierarchical cluster case (e.g. the 'Carpenter' trade-type belongs in both clusters 2 and 3).

The same discrepancies between clusters found by the two clustering techniques can be observed from the Topics perspective as depicted in Table 3.5.

The uncertainty due to the sparseness of the transactional data can also be seen when the clustering process is performed on the verified customer profiles as can be seen in the plots in Figure 3.18.

3.3 Effect of Conventional Sampling Data Mining Algorithms' Predictability

Section 2.2.2 highlighted the information loss and subsequent fall in predictive accuracy encountered by data mining algorithms, due to the need to scale large data down for efficient data mining.

This can be more so for sparse transactional data, especially if an ad-hoc data reduction process is used. A practical analysis of the effect of conventional sampling on the predictability of 2 parametric (linear and quadratic discriminant functions) and 2 non-

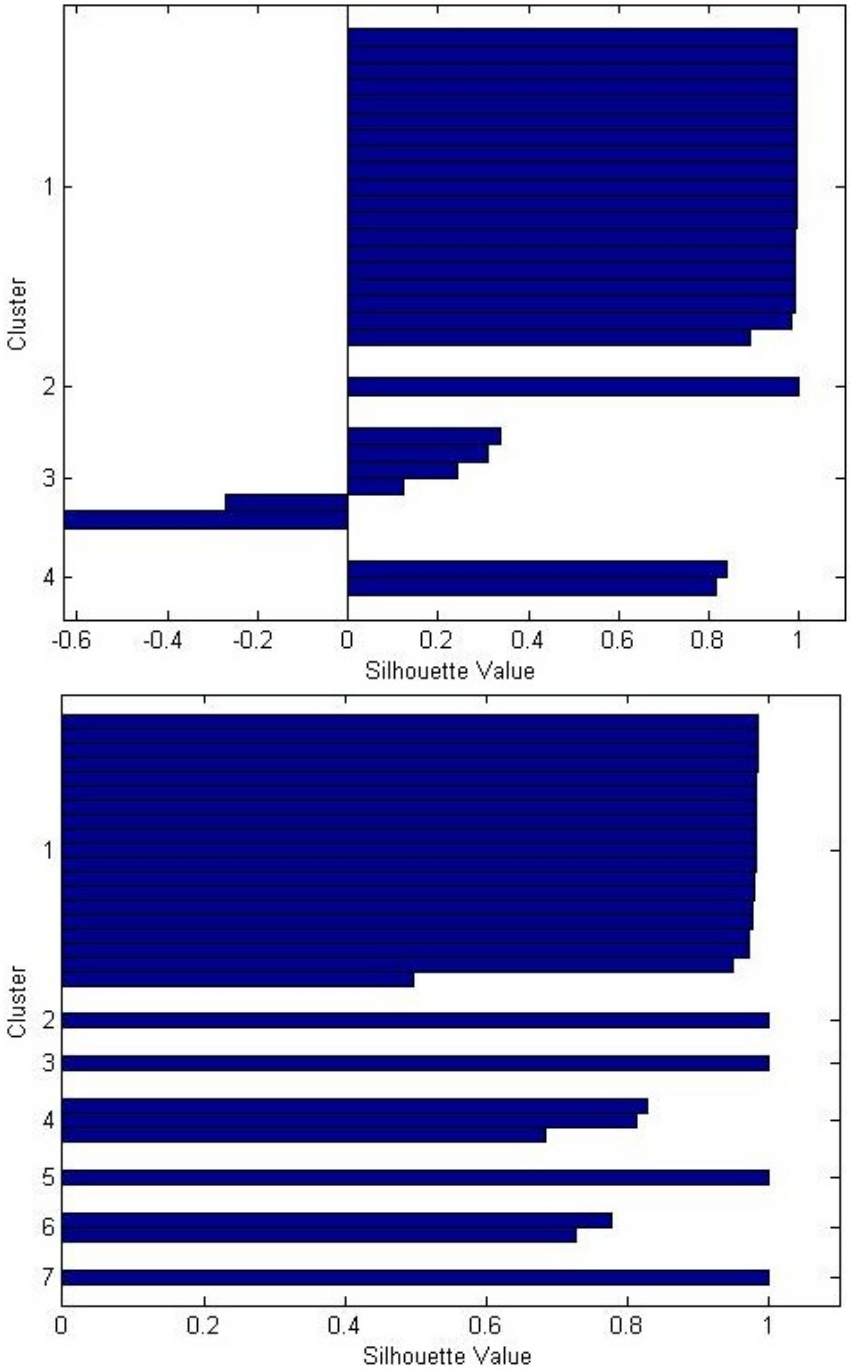


Figure 3.15: The silhouette plots for k = 4 and k = 7 clusters for the Trade-types in Screwfix's transactional data.

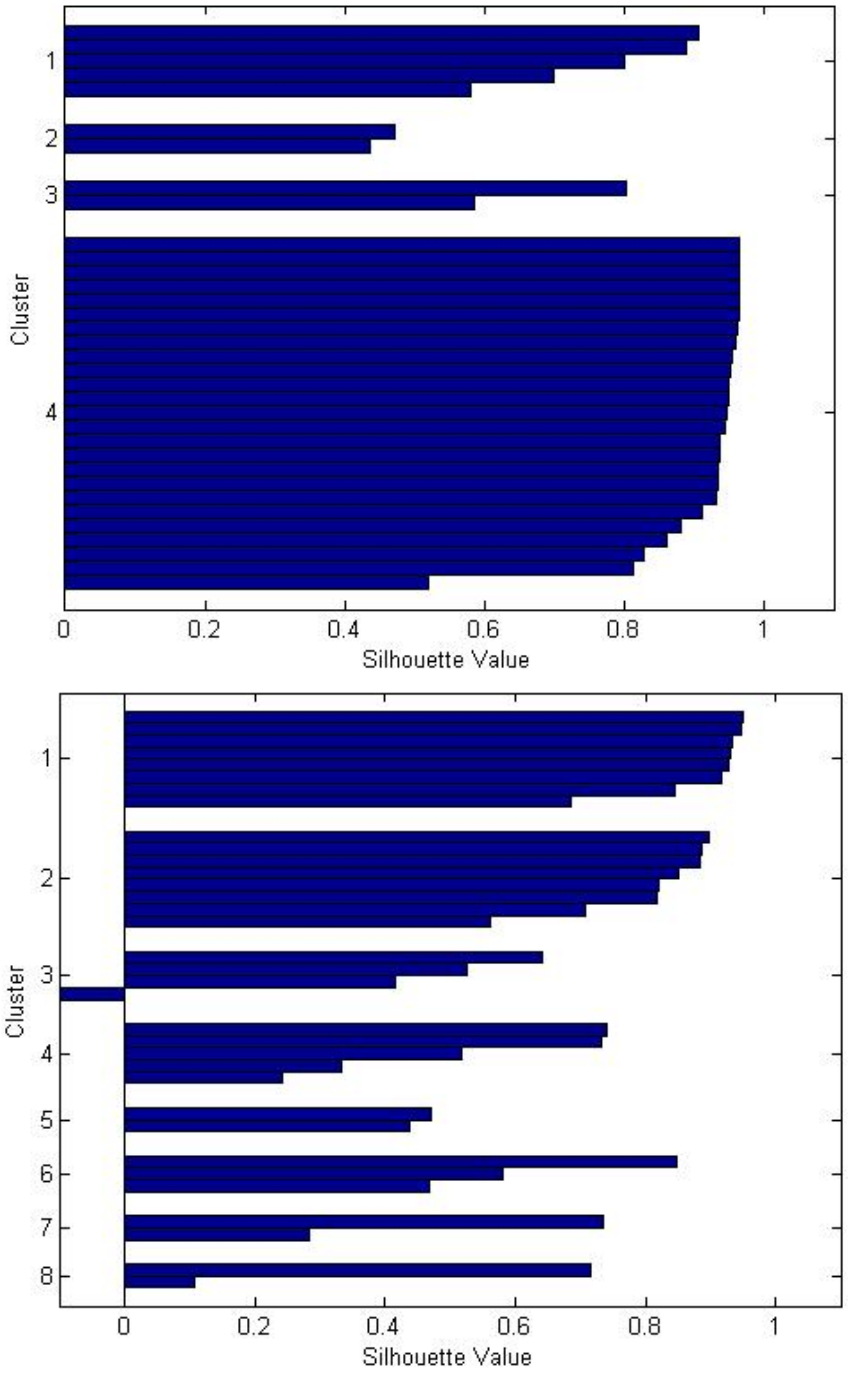


Figure 3.16: The silhouette plots for k = 4 and k = 8 clusters for the Topics in Screwfix's transactional data.

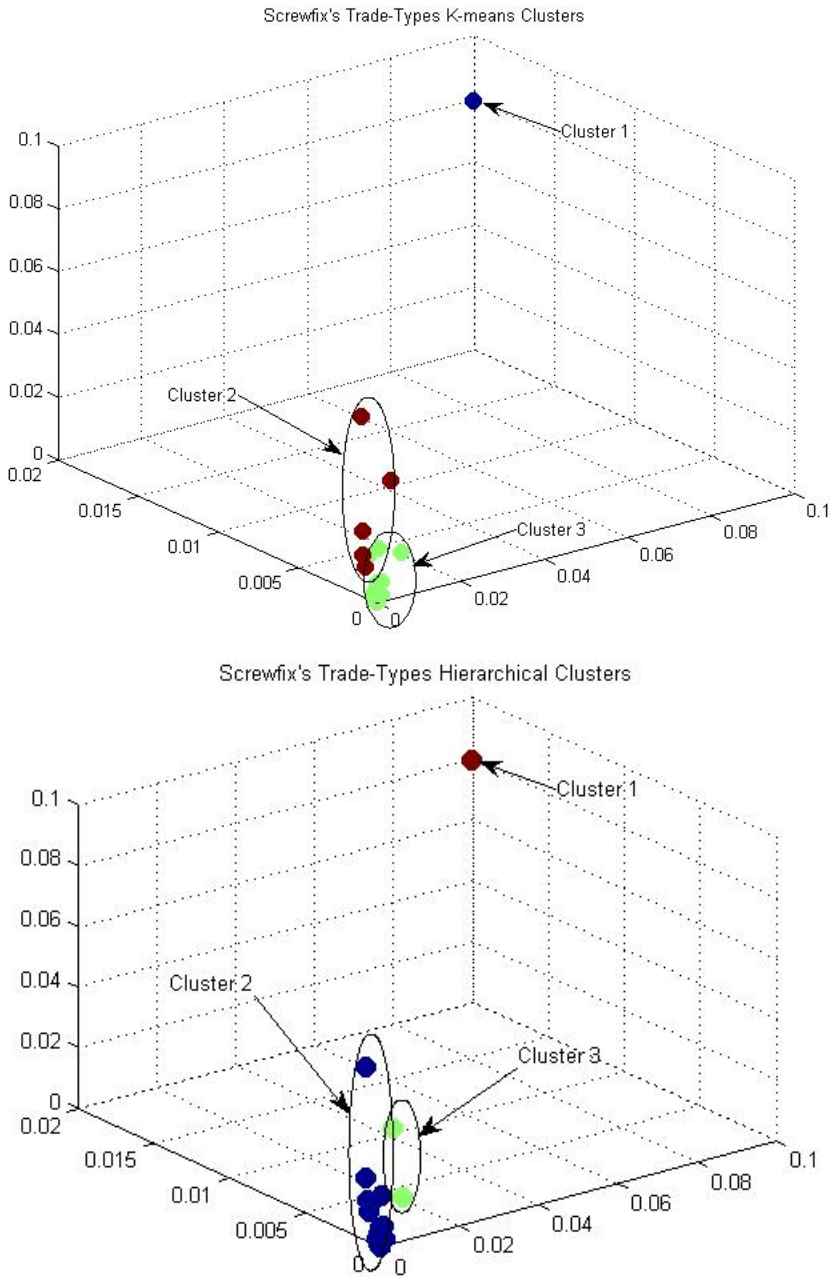


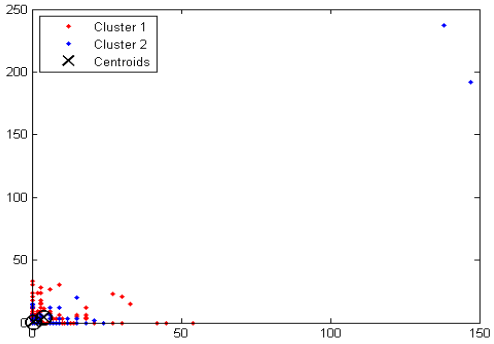
Figure 3.17: Plots showing the three (3) trade-type clusters found by K-means and hierarchical clustering techniques.

Trade-Type	K-Means Cluster No.	Hierarchical Cluster No.
Agricultural	2	2
Builder	2	2
Carpenter	2&3	2&3
CarpetFloorFitter	2	3
CleaningServices	2	2
DeclinedToAnswer	2	2
DIYHobbies	1	1
Electrician	2	2
Engineer	3	2
HandyMan	2	2
HotelCatering	2	2
LandscapeGardenMaintenance	2	2
LeisureIndustry	2	2
MaintenanceCollegeUni	2	2
MaintenancePropertyFacilities	2	2
MaintenanceSchool	2	2
Manufacturing	2	2
None	2	2
Other	3	2
PainterDecorator	2	2
pickOne	3	2
PlumberHeatGasFitter	2	2
RetailerShopFitting	2	2
Roofer	2	2
SecurityServices	2	2
Trade	2	2
TransportAutomotive	3	2
Unknown	2	2

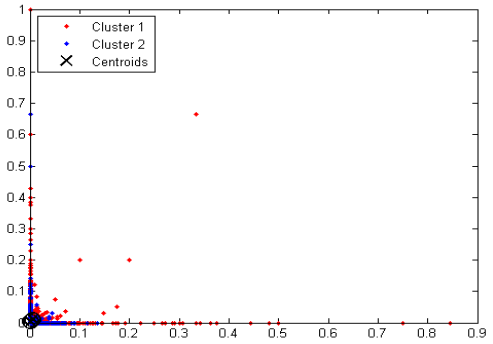
Table 3.4: K-means and Hierarchical Clustering Techniques' Cluster Groupings of Screwfix's Trade-Types

Topics	K-Means Cluster No.	Hierarchical Cluster No.
Abrasives	4	2
Access Equipment and Storage	4	2
Automotive	4	2
Bathrooms	4	2
Blades	4	2
Bolts	4	2
Building	4	2
Cleaning	4	2
Conservatories	4	4
Decorating Sundries	4	2
Doors & Windows	4	3
Drill Bits	1	2
Electrical	2	2
Fixings	4	2
Flooring	4	2
Hand Tools	3	2
Heating and Ventilation	4	2
Ironmongery	1	2
Kitchens	4	2
Landscape Power	4	2
Lighting	1	2
Nails	4	2
Other	3	2
Outdoor Buildings	4	2
Paint	4	2
Plumbing	2	2
Power Tools	1	2
Screwdriver Bits	4	2
Screws	4	2
Sealants & Glues	1	2
Security	4	1
Vouchers	4	3
Workplace Safety	4	1
Workwear	4	2

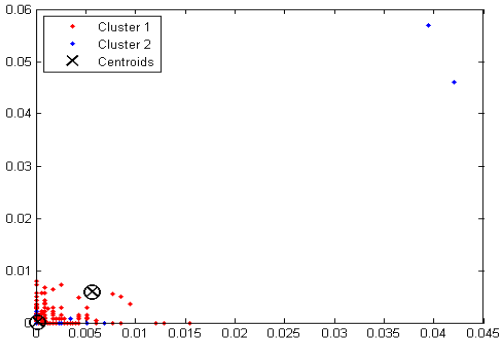
Table 3.5: Screwfix's Topics K-means and Hierarchical Clustering Techniques Cluster Groupings



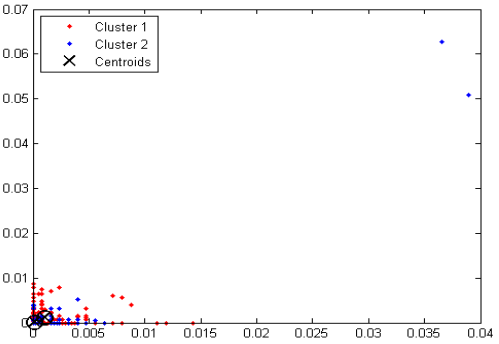
Clustering of Absolute values of Verified Customer Profiles' Transactional Data



Clustering of Min-Max Normalized applied on Rows of Verified Customer Profiles' Transactional Data



Clustering of Min-Max Normalized applied on Columns of Transactional Data



Clustering of Min-Max Normalized applied on Entire Verified Customer Profiles' Transactional Data

Figure 3.18: Plots showing the results obtained from clustering the absolute values of transactional data as well as those obtained from applying the Min-Max normalization method on the individual customer profile transactions (row-wise), on the items transacted (column-wise), and on the entire transactional data. It can be seen that due to the sparseness of the transactions per customer normalization has little effect on separability as would otherwise be expected from the clustering process.

parametric (k-nearest neighbour and decision tree) classifiers further asserts this drawback.

To perform the analysis, 10,000, 50,000 and 100,000 were uniformly sampled from Screwfix's 2007 and 2008 transactional data with the 28 Trade-types as targets (i.e. 28 classes). Figures D.1, D.2 and D.3 of Appendix D show the distributions of targets for each of the year 2007 data samples whilst Figures D.4, D.5 and D.6 of Appendix D show the distributions of targets for each of the year 2008 data samples.

The analysis was undertaken using **ldc** (- *Linear discriminant classifier assuming normal densities with equal covariance matrices*), **qdc** (- *Quadratic discriminant classifier assuming normal densities (sigma unconstrained)*), **Knnc** (- *K-Nearest Neighbour classifier with K= 3*) and **stumpc** (- *Decision stump tree classifier*) which are PRTools [Duin et al., 2007] implementations of the 2 parametric (linear and quadratic discriminant classifiers) and 2 non-parametric (k-nearest neighbour and decision tree) classifiers being studied.

The standard 10-fold cross-validation, in which each part is held out in turn and the classifier trained on the remaining nine-tenths with the misclassification rate calculated on the hold out, was used to evaluate the classifiers. The cross-validation was done 30 times repeatedly to obtain the performance of the classifiers depicted in the plots in Figure 3.19 on the 10000, 50000 and 100000 datasets uniformly sampled from Screwfix's 2007 and 2008 transactional data with the 28 Trade-types as targets (i.e. 28 classes). The Matlab code for the experiment can be found in Appendix F.

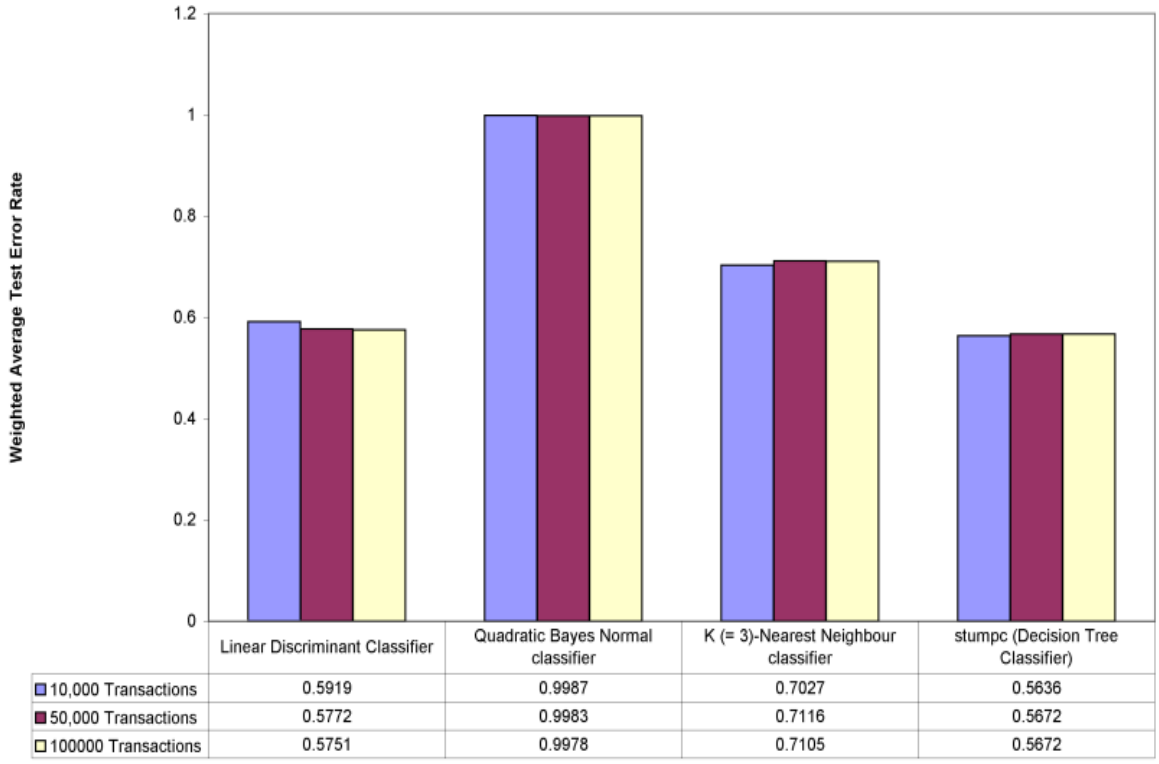
Looking at the performance results in Figure 3.19, one can see that the attribute-value based data mining classifiers perform very poorly with the best classifier, *ldc* achieving an average error of 58% on all three dataset samples.

Screwfix's transactional data is inherently sparse as described in Section 3.1 and depicted in Table E.5 .

This kind of data is typical of transactional data due to customers typically purchasing only a very small fraction of products as discussed in Section 2.2.

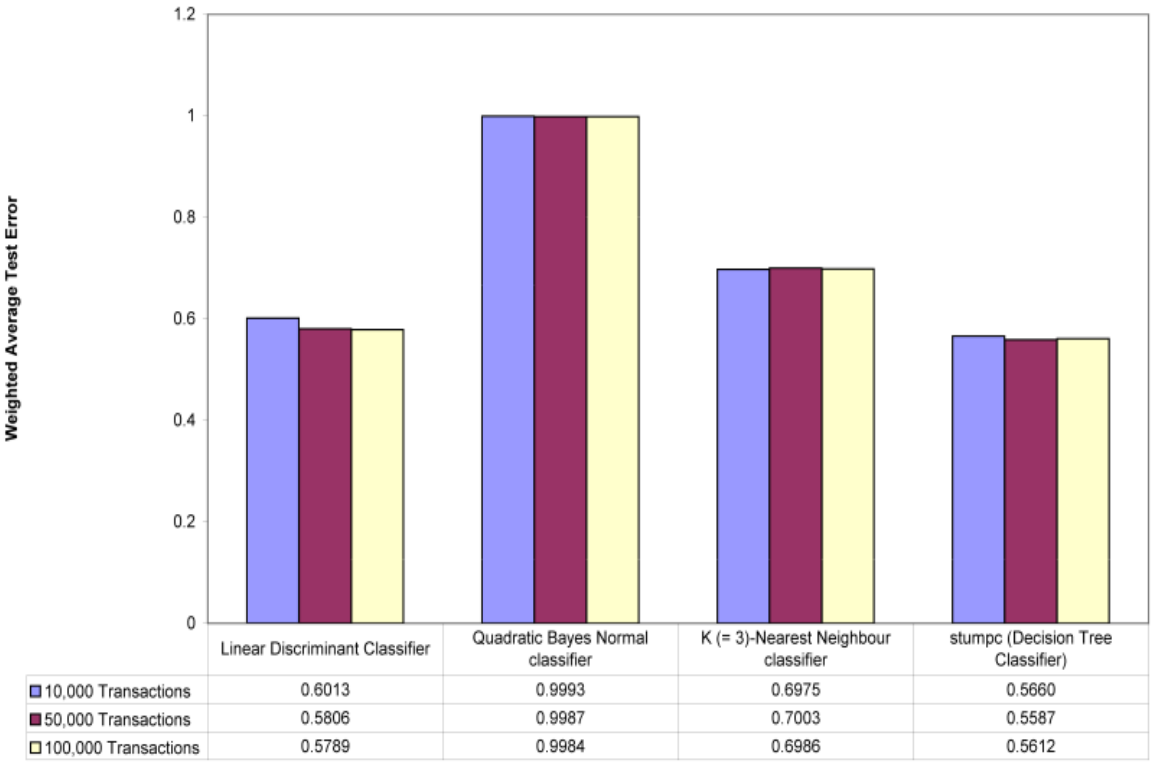
For sparse data, conventional random sampling may not work well because most of the samples are zeros [Church et al., 2006]. Sampling fixed dataset columns (as we have done here with Screwfix's Topics transactional data) from the dataset is also inflexible because different rows may have very different sparsity factors. Thus, each sampled data instance conveys little or no information for the aforementioned classifiers (especially **qdc** and **knn**) to adequately distinguish one Trade-Type from the other on the basis of Topics transacted.

In practice the set of data mining algorithms that are said to be best at handling sparse data are those that process the training set data into trees of related patterns [Quinlan, 1986, Schaffer, 1992] or build association models [Agrawal and Shafer, 1996, Agrawal and Srikant, 1994, Gouda and Zaki, 2001] as they are designed to use sparse data. This accounts for the relatively better performance of the decision tree classifier (stumpc)



	Linear Discriminant Classifier	Quadratic Bayes Normal classifier	K (= 3)-Nearest Neighbour classifier	stumpc (Decision Tree Classifier)
10,000 Transactions	0.5919	0.9987	0.7027	0.5636
50,000 Transactions	0.5772	0.9983	0.7116	0.5672
100000 Transactions	0.5751	0.9978	0.7105	0.5672

Classification Performance on Sampled Screwfix's Transactions for 2007



	Linear Discriminant Classifier	Quadratic Bayes Normal classifier	K (= 3)-Nearest Neighbour classifier	stumpc (Decision Tree Classifier)
10,000 Transactions	0.6013	0.9993	0.6975	0.5660
50,000 Transactions	0.5806	0.9987	0.7003	0.5587
100,000 Transactions	0.5789	0.9984	0.6986	0.5612

Classification Performance on Sampled Screwfix's Transactions for 2008

Figure 3.19: Plots showing the performance of ldc, qdc, 3-Nearest Neighbour and Decision Tree Classifiers on Sampled Screwfix's Transactional Data

compared to the other classifiers on the sampled datasets.

Similarly, the **ldc** performed much better than **qdc** and **knn** due its being a linear discriminant analysis based classifier [McLachlan, 2004], which searches for a linear combination (i.e. association) of features that characterize or separate the 28 Trade-Types.

However, data that has a significantly higher sparsity factor can require extremely large amounts of temporary space to build associations even if small datasets of the data are sampled [Ahmed et al., 2004, Han et al., 1997]; thus defeating the reason for sampling for data mining purposes i.e. to efficiently utilize and save memory space [Brin and Page, 1998].

This thesis presents a more guided data reduction approach in Section 4.4.2, which uses the results from the K-mean's algorithm to extract prototypes that are the most representative of transactional data that have been aggregated and binned.

3.4 Summary of Exploratory Analysis and Conclusion

The chapter commenced with an informal and formal description of transactional data together with their use in building customer profiles. A detailed description of Screwfix's transactional data was then presented, along with a statistical analysis that uncovered discrepancies in the transactional data. This occurred mainly due to the reluctance of the customers to divulge information about their trade-type and inaccurate labels for those who declared their line of business.

The identified discrepancies were shown to result in the unreliability of the customer labels for the 12 trade-types in Screwfix's transactional data with the "No Trade Specified (NTS)", "Multi-trade domestic small general contractor (MTD)", and "Other trade (OT)", which make up the specified trade-types, being a mixed bag of all the trade-types.

To handle the identified discrepancies, a necessary third party verification of the Electrician and PlumbHeat was undertaken. Further statistical analysis on the verified trade-type were shown to have more distinguishable buying patterns.

The results of the statistical analysis of Screwfix's transactional data covering the period between the 1st quarter of 2007 and the 2nd quarter of 2009 are presented in Section 3.1.1 and Section 3.1.2. The total number of each of the unverified trade-types was provided, along with those for the two verified trade-types. Plots comparing the items transacted by each of the trade-types were also presented as well as plots showing the positive skewness of the buying behaviour of the trade-types.

The following key findings can be gleaned from the plots:

1. There are significant discrepancies between the unverified and the third-party verified trade-type information for the Electrician and PlumbHeat trade-type as can be

seen in Figures A.1, A.2, A.3, A.4, A.5 and A.6.

These discrepancies, which were found to have come about mainly due to the customers' unwillingness to divulge information about themselves and mislabelled customer profiles, increase the uncertainty of the transactional data. This increased uncertainty makes it challenging to effectively distinguish the customers using just their transactional data. The challenge in differentiating the unverified customers' trade-type based on their transactions is further compounded by the similarity of their buying behaviour as shown in Figure 3.5 and 3.6. The patterns of the buying behaviour for the verified customer trade-types are however more distinguishable in terms of numbers/type of items purchased as can be seen in Figures 3.7 and 3.8 for the Electrician and PlumbHeat trade-types respectively. Furthermore, the verified Electrician trade-type were found to consistently purchase a large proportion of the Electrical topic while the verified PlumbHeat consistently purchase a large proportion of the Plumbing topic as can be seen in Figures 3.6, B.1 and B.2. The verified customer trade-types are thus more viable, due to their consistence, for building models that will more effectively distinguish the Electrician trade-type from the PlumbHeat trade-type.

2. The transactional data is highly sparse and positively skewed with a large proportion of the trade-types making very few purchases and a very small proportion of the trade-types making high purchases as can be seen in the plots in Figures 3.9,and 3.10 where the mass of the trade-type distribution is concentrated on the left of the plots with relatively few high values of items transacted.

   This means that the transactional data has very few items, which will make it difficult for mined models to accurately distinguish the customers based on their transactional data. The transactional data will need to be transformed or consolidated into forms appropriate for mining.

The use of clustering for grouping transactional data was then investigated but found to be inadequate for transforming the transaction data in a form appropriate for mining. The following list summarizes the key findings obtained from applying hierarchical and K-means clustering techniques to Screwfix's topics transactional data.

1. Both cluster evaluation techniques, i.e. upper-tail rule for hierarchical clustering and silhouette statistics for the K-means clustering techniques; identified 3 clusters as the minimum number of trade-type groupings and 4 clusters for Topic groupings in Screwfix's transactional data for 2007. See Figures 3.12 and 3.14 for a graphical depiction of the Upper-Tail rule's cluster evaluation results; and Figures 3.15 and 3.16 for the silhouette statistics cluster evaluation plots. This is contrary to the

real-world setting as encapsulated by Screwfix's customer database which, in reality, contains a mixture of more than 3 customer trade-types and more than 4 Topic groupings. The sparseness inherent in transactional data, due to a large proportion of the customers buying few products, results in the single, large, inseparable cluster, even when the data is verified and normalized as can be seen in Figure 3.18.

2. Both cluster evaluation techniques for the two clustering techniques studied identified a unique cluster for 'DIY/Hobbies' but there were overlaps and discrepancies in the contents of the remaining trade-type clusters found by both techniques. See Table 3.4 for the details of the cluster contents and Figure 3.17 for the graphical depiction.

3. Both K-means and hierarchical clustering identified Topics that were purchased together however there were discrepancies between the contents of Topics grouped together by the clustering techniques. See Table 3.5 for details.

Thus, using the K-means and hierarchical clustering algorithms to explore both the verified and unverified customer profiles resulted in an indistinguishable cluster group. Both clustering algorithms were susceptible to the sparseness (due to a large proportion of the customers buying very few items) of the transactional data.

The indistinguishability of the trade-types due to the sparseness and skewness of the transactional data were further shown in Section 3.3 to adversely affect the prediction performance of 2 parametric (linear and quadratic discriminant functions) and 2 non-parametric (K-nearest neighbour and decision tree) classifiers using conventional sampled transactional data.

The transactional data used in this thesis was transformed by aggregating the customers' transactions over time to deal with the sparseness. The aggregated customer transactions were then partitioned by binning them based on the number of items bought to deal with the inherent skewness in the transactional data.

The next chapter presents the use of binning, to group the customer profiles based on the number of items bought and demonstrates the improvement in accuracy for classifying customer profiles built using transactional data. In particular the problem of the minimum number of items required to confidently classify a customer profile is investigated and the results obtained from applying the proposed approach on 4 classifiers are analysed for both the two-class classification problem and the multi-class classification problem.

Chapter 4

Customer Profile Classification

This chapter presents approaches for classifying customer profiles using transactional data.

A key business application of mined knowledge from transactional data is in building customer profiles for personalization and recommender systems. These business applications of transactional data mining differ, not only in the data mining algorithms they use to make inference or recommendations, but also in the way in which the customer profiles they use are constructed.

This chapter begins by presenting an overview of customer profiles and the construction of customer profiles from transactional data. A description of customer profiles is provided together with methods for building such profiles.

The chapter continues with a description of the problem of classifying customer profiles. A background overview of the different approaches used to address the problem are then provided together with their shortcomings.

Our approach to solving the problem for a two-class case and multi-class scenario are then presented, together with results from the experiments performed on real-world data provided by Screwfix and Sligro respectively. The chapter concludes with a discussion on the characteristics of the customer profiles which are confidently classified by the two approaches.

4.1 Overview of Customer Profile

A customer profile is an outline of the type of customer likely to purchase a product. Customer profiles may vary from product to product and are constantly updated with changing customer preferences. Well-developed customer profiles are an essential analysis tool, as they aid businesses in targeting their advertising and marketing to the right customer at the right time; thereby cutting advertising costs and saving time and money by concentrating on real potential customers rather than too wide a range of individuals.

Customer profiles can be *factual* or *behaviourally* based. A factual based customer profile consists of a set of characteristics (e.g. demographic information such as name,

gender, birth date, etc.) while a behaviourally based customer profile consists of what the customer is actually doing and is usually derived from transactional data [Bazik and Feltes, 1999, Yu, 1999].

Behaviourally based customer profiles are much stronger predictors of the future actions of a customer, as they encapsulate the dynamism of the customer more than demographically based customer profiles which are more or less static.

Furthermore, the information that makes up demographically based customer profiles is expensive to acquire by the business, while the information for behaviourally based customer profiles can be gleaned each time a customer makes a purchase.

4.1.1 Customer Profile Construction Methods

As mentioned in the preceding section, whereas factual customer profiles describe who the customer is, behaviourally based customer profiles describe what the customer does.

Behaviourally based customer profiles can be constructed using rules specified by human experts or extracted from transactional data using data mining methods [Meteren and Someren, 2000, Paulson and Tzanavari, 2003].

Customer profiles based on data mining require the collection of data that accurately reflect the interest of the customers and their interactions with the business, i.e. services and/or items. Customer profiles generated using data mining methods can be grouped based on the data used:

1. Interest-based: Customer profiles are built with features of items extracted from item descriptions, or relational attributes associated with items in the backend databases. The process of building such profiles involves two stages. First, the level of user interest is determined from a subset of items. This task may be accomplished implicitly, by passively observing the user and using various heuristics to classify items as interesting or non-interesting [Lieberman, 1995, Mladenic, 1999], or it can be based on explicit user judgement, assigning interest levels to items or manually identifying positive and negative examples [Lang, 1995, Pazzani and Billsus, 1997]. The second stage involves transforming each item into a bag of words (vector) representation, with each token being assigned a weight, using methods such as TF-IDF [Salton and McGill, 1986] or minimum description length. This approach to building customer profiles has a major disadvantage common to approaches based on an individual profile, in that it's unable to capture new and unexpected changes in general customer behaviour due to its focus on the individual customer's previous interests.

2. Rule-based: Customer profile-construction process usually consists of two main steps: rule discovery and rule validation. Various data mining algorithms such

as Apriori, FP-Growth, and CART, can be used for the rule discovery step. Aggarwal et al. [2002] proposed profile association rules, in which the left-hand side consists of customer profile information (e.g. age, salary, education) and the right-hand consists of customer behaviour information (e.g., buying nappies, milk products). Profile association rules are especially useful for segmenting users based on their transactional characteristics and for deriving customers' behavioural attributes from their transactional attributes. One of the problems with many rule discovery methods is the large number of generated rules, many of which, although statistically acceptable, are spurious, irrelevant, or trivial [Meteren and Someren, 2000, Paulson and Tzanavari, 2003]. Post-analysis is usually used to filter out irrelevant and spurious rules. This usually involves getting a domain expert to inspect and validate/reject the generated rules one-by-one. This manual inspection is, however, not scalable to a large number of rules and customer profiles.

3. Ratings-based: Customer profiles are generally represented as a vector or set of ratings providing the customer's preferences on a subset of items [Meteren and Someren, 2000, Mobasher et al., 2002].

Irrespective of the algorithmic approach used to build the customer profile, the data used for building the profile can be collected implicitly or explicitly.

Explicit collection usually requires the customer's active participation. For customer profiles that are based on factual information, customer involvement may take the form of taking part in surveys or providing personal and financial information at the time of a transaction.

Implicit collection on the other hand, involves monitoring and measuring customer behaviour, data such as customer's purchase and activity history, to create the customer profiles. Collecting data implicitly has the advantage of removing the burden associated with providing personal information from the user.

This thesis focuses mainly on customer profiles built from implicit customer feedback, collected automatically by monitoring customers' purchase histories. In other words, the work here is centred around the application of data mining techniques that attempt to learn individual and group customer profiles, using transactional data for the purpose of generating robust adaptive models, that can be used to more efficiently target customers.

4.2 The Problem of Classifying Customer Profiles

4.2.1 Problem Statement

Formally, given a set of transactions $\mathbf{T}$, containing N transactions categorized into d product item topics:

$$\mathbf{T} = \begin{bmatrix} t_{1,1} & t_{1,2} & \cdots & t_{1,d} \\ t_{2,1} & t_{2,2} & \cdots & t_{2,d} \\ \vdots & \vdots & \cdots & \vdots \\ t_{N,1} & t_{N,2} & \cdots & t_{N,d} \end{bmatrix}$$

we define a set of M customer profiles,

$$\mathbf{P} = \begin{bmatrix} p_{1,1} & p_{1,2} & \cdots & p_{1,d} \\ \vdots & \vdots & \cdots & \vdots \\ p_{i,1} & p_{i,2} & \cdots & p_{i,d} \\ \vdots & \vdots & \cdots & \vdots \\ p_{M,1} & p_{M,2} & \cdots & p_{M,d} \end{bmatrix}$$

with

$$p_{i,j} = \sum_{k=i_1}^{n_i} t_{k,j}, \ \forall_{i=1,\ldots,M} \ \forall_{j=1,\ldots,d}$$

where $k \in \{i_1, i_2, \ldots, i_{n_i}\}$ is a set of indexes referring to the n_i transactions of the i-th customer in the set of transactions $\mathbf{T}$. Table E.6 shows an excerpt of a computed Electrician and PlumbHeat profile.

Goal: We seek to build classifiers using distinctive groups (defined by the number of items purchased), for which the predictive error of classifying unseen customer profiles over time is minimal.

4.3 Background Knowledge and Related Work

Using a data mining algorithm to discover the best model for a business problem involves processing historical data with the goal of identifying the relevant independent variables which will best minimize the error for predicting unseen future instances.

Typically, the process of building the data mining model for classifying customers involves an initial pre-processing of the available data and then applying one (or a combination of) classification technique(s) to classify the customers.

The decision to apply a pre-processing technique may be driven by the need to: generate a model from a dataset that is too large to process in full (data reduction), handle missing values/inconsistent data (data cleansing), combine data from multiple sources into a coherent store (data integration), normalize data so that it can be more efficiently processed (data transformation), etc.

The classification process essentially consists of examining the features of a newly presented object and assigning it to one of a predefined set of classes. The objects to

be classified are generally represented by records in a database table or a file, and the act of classification consists of adding a new column with a class code of some kind. For example an email classifier might attempt to classify an email as legitimate or Spam.

Common classification algorithms like Nearest neighbour, Naive Bayes classifier and Neural network have been used to classify customer profiles. For instance, Allera and Horsburgh [1998], Gerbec et al. [2003] examine the classification of electricity customers based on their consumption. Their approach typically involved grouping the consumers according to their type of activity (e.g. residential and non-residential), using data mining techniques such as feature selection and clustering. However, Chicco et al. [2001, 2004] reported that better approaches are needed to classify electricity customer profiles as there are poor correlations between type of activity and electricity consumption of the consumer.

Other customer profile classification areas include: insurance fraud detection [Nisbet et al., 2009], web content management [Berry and Linoff, 2000, Markov and Larose, 2007], credit risk classification [Madeira et al., 2003, Soares et al., 2008], etc. All the aforementioned examples have a limited number of classes/discrete outcomes, and the task is to assign new records into one of them.

This Chapter presents an investigation of a data mining approach that combines an unsupervised data binning pre-processing technique with classification to identify different types of customer profiles using their transactions. Customers' with sparse transactions which tend to make up the bulk of transactional data are difficult to distinguish and accurately classify. This problem is even more pronounced when the sparse transactions are mixed with dense transactions, as the classifier performance tends to be biased towards the larger number of customers with sparse transactions.

4.4 Customer Profile Classification using Transactional Data

To overcome the problem of sparsity which is inherent in transactional data, our proposed approach groups customer profiles into bins on the basis of the number of items transacted. The ultimate goal is to determine the minimum number of items required to more accurately and confidently classify a customer profile into one of the identified distinctive groups. In situations where very large bins are returned by the binning process, prototype selection is then undertaken to obtain customer profiles which are most representative of each bin.

We now present a background overview of binning in Section 4.4.1 and prototype selection in Section 4.4.2. The classification process for a two-class classification is usually a straight forward assignment of the customer profile into one of two classes. The case

is, however, different for multi-class classification which involves assigning the customer profiles into one of many classes. The proposed approach for the multi-class classification of customer profiles is presented in Section 4.4.3.

4.4.1 Overview of Data Binning

Data binning is an unsupervised discretization method in which the data is grouped into either *Equal Interval Width* or *Equal Frequency Intervals*.

The equal-width data binning algorithm works by determining the minimum and maximum values of the attribute of interest and then dividing the range into a user-defined number of equal width bin intervals. This approach to data binning is, however, vulnerable to outliers that may drastically skew the range [Catlett, 1991].

The equal-frequency data binning algorithm, on the other hand, determines the minimum and maximum values of the attribute of interest, sorts all values in ascending order, and divides the range into a user-defined number of intervals, so that every interval contains the same number of sorted values.

Kerber [1992] asserts that since binning, like many unsupervised methods, do not utilize instance labels in setting partition boundaries, it is likely that classification information will be lost by binning as a result of combining values that are strongly associated with different classes into the same bin. This can result in effective classification being much more difficult to perform in some cases.

Chiu et al. [1991], Chmielewski and Grzymala-Busse [1995] use a variation of equal frequency intervals - maximal marginal entropy - to adjust the boundaries so as to minimise entropy at each interval. Holte [1993] presented an example of a simple supervised data binning approach, in which his *Information Retrieval* (IR) algorithm divides the domain of every continuous variable into pure bins, each containing a strong majority of one particular class, with the constraint that each bin must include at least some pre-specified number of instances. This approach appears to work reasonably well when used with the IR induction algorithm.

A typical data binning process broadly consists of four steps:

1. sorting the continuous values of the feature to be binned,
2. evaluating a cut-point for splitting or adjacent intervals for merging,
3. according to some criterion, splitting or merging intervals of continuous value, and
4. finally stopping at some point.

One key parameter of concern in the data binning process is determining the best "cut-point" to split a range of continuous values or the best pair of adjacent intervals to merge. Entropy based-and/or-statistical based evaluation function have been used to determine

an appropriate "cut-point" with varying results [Kerber, 1992, Kohavi and Sahami, 1996, Li and Wang, 2002, Liu and Wang, 2005, Witten and Frank, 1999].

As discussed in Section 4.4, transactional data tends to be skewed towards the large number of customers who make fewer purchases. This makes distinguishing them for classification purposes difficult.

In order to overcome the problem of skewness that is inherent in sparse transactional data as well as to ensure sufficient statistical power for inference, it is particularly important that the cut-point for splitting the range of the number of items bought, be such that each bin contains a proportional representation of the number of customer profiles for each class.

To meet this requirement for better classification performance, we regroup the customer transactions into bins defined by the number of items per transactions and choose a "cut-point" that is a large fraction of the total number of customer profile transactions. This heuristic ensures that the bins with the fewer items per transaction have a large enough representation of examples to make up for the sparseness of the transactions, $\mathbf{t_i} \in \mathbf{T_j}$. Our approach is outlined as follows:

Given a set of M customer profiles,

$$P = [\mathbf{p_1}, \mathbf{p_2}, \ldots, \mathbf{p}_M]$$

with each customer profile $\mathbf{p_i}$ having its aggregated d-dimensional transaction as defined in Section 4.2, re-ordered to obtain:

$$\widehat{\mathbf{P}} = \begin{bmatrix} \widehat{\mathbf{p}}_\mathbf{1} \\ \vdots \\ \widehat{\mathbf{p}}_\mathbf{M} \end{bmatrix}, \text{ where } \widehat{\mathbf{p}}_\mathbf{i} = [\widehat{p}_{i,1}, \ldots, \widehat{p}_{i,d}]$$

based on the sum total of the number of items transacted by each of the customers.

The corresponding vector $\widehat{\mathbf{s}}$, consisting of the total number of items bought by each of the M customers is:

$$\widehat{\mathbf{s}} = \begin{bmatrix} \sum_{i=1}^{d} \widehat{p}_{1,i} \\ \vdots \\ \sum_{i=1}^{d} \widehat{p}_{M,i} \end{bmatrix} = \begin{bmatrix} \widehat{s}_1 \\ \vdots \\ \widehat{s}_M \end{bmatrix}, \text{ where } \widehat{s}_1 \leq \widehat{s}_2 \leq \ldots \leq \widehat{s}_M$$

Given that $\widehat{\mathbf{P}}$ and $\widehat{\mathbf{s}}$ are sorted in ascending order, the bins can be easily determined. That is, for a given bin size of Q (in our case $Q = 40000$ instances) there will be $\lceil M/Q$ bins.

The training sets in the respective *b-1* bins will be:

$$B_i = \begin{bmatrix} \widehat{\mathbf{p}}_{\mathbf{1+(i-1)Q}} \\ \vdots \\ \widehat{\mathbf{p}}_{\mathbf{iQ}} \end{bmatrix}, \text{ for } i = 1, \ldots, b-1$$

with the minimum and maximum number of items transacted within each bin given by:

$$\left[\widehat{s}_{1+(i-1)Q} \ \widehat{s}_{iQ}\right]$$

For the (b-th) bin, the number of items can be smaller than Q; and the respective training set and range of total items bought is given by:

$$B_i = \begin{bmatrix} \widehat{\mathbf{p}}_{\mathbf{1+(i-1)Q}} \\ \vdots \\ \widehat{\mathbf{p}}_{\mathbf{M}} \end{bmatrix}, \text{ for } i = b$$

and $\left[\widehat{s}_{1+(i-1)Q} \ \widehat{s}_{M}\right]$.

4.4.2 The K-means Algorithm as a Prototype Selection Tool

The binning process, discussed in Section 4.4.1, whilst abating the problem of skewness in transactional data results in transactional data groups whose sparsity makes sampling them for classifier modelling unwieldy with a resultant poor performance of the classifier model as discussed in Section 2.2.

An alternative approach to random sampling is to carefully select prototypes that most represent each bin. Many methods have been developed for prototype selection. Some of them are aimed at minimizing the space and time needed for the classification of a dataset; while others attempt to improve accuracy.

Typical examples of the former include Edited Nearest Neighbour (ENN) [Wilson, 1972], Multi-edit [Ferri et al., 1999], Relative Neighbourhood Graph Edition (RNGE) [Sánchez et al., 1997], etc.; while the Decremental Reduction Optimization Procedure Family (DROP3) [Wilson and Martinez, 1997], Prototype Selection by Relative Certainty Gain (PSRCG) [Olvera-López et al., 2010] and Model Class Selection (MoCS) [Brodley, 1993] have been proposed as prototype selection for accuracy improvement.

Although different researchers have addressed the issue of prototype selection there is no research that suggests an automatic procedure for instance selection, which can be employed for any given classification algorithm and in a computationally efficient way, for sparse transactional data. This thesis presents an algorithm for prototype selection,

which exploits the K-means clustering algorithm. It is aimed at reducing the error rate compared to that obtained by using a simple sampling of the transactional data. The proposed approach in this thesis is analogous to the fuzzy clustering based approach proposed in [Bezdek and Kuncheva, 2009, Liu et al., 2002, Spillmann et al., 2006] in which the centroids are selected as prototypes.

K-means has a linear complexity in computation, i.e. for I iterations of the K-means algorithm performed on a dataset containing m instances each has n attributes, its complexity may be calculated as: $O(I * K * m * n)$. K-means, therefore, has a time complexity advantage, when used in decomposing large transactional data, in comparison to other clustering methods (e.g. hierarchical clustering methods), which have non-linear complexity with respect to the number of instances. Also, K-means is easy to interpret, simple to implement, has a fast speed of convergence and can be used on sparse data [Dhillon et al., 1999, Witten and Tibshirani, 2010.].

The Basic K-means Prototype Selection Algorithm

The basic K-means prototype selection uses the K-means algorithm for the purpose of space (i.e. bin) decomposition. It then uses silhouette statistic to select the instances that are closest to the centre (measured by the average silhouette width) of the decomposed bin.

First, the K-means algorithm is used to partition the transactional data in each bin into K groups, such that the within-group sum-of-squares is minimized. The K-means algorithm works by defining the within-bin scatter matrix given by:

$$S_W = \frac{1}{n} \sum_{j=1}^{g} \sum_{i=1}^{n} I_{ij} \left(X_i - \overline{X_j}\right) \left(X_i - \overline{X_j}\right)^T \tag{4.1}$$

where I_{ij} is one if X_i belongs to group j and zero otherwise, and g is the number of groups. The criterion that is minimized by the K-means algorithm is given by the sum of the diagonal elements of S_W, i.e., the trace of the matrix, as follows

$$Tr\left(S_W\right) = \sum S_{W_{ii}} \tag{4.2}$$

Minimizing the trace, is essentially equivalent to minimizing the total within-group sum of squares about the group means [Everitt et al., 2011].

In order to proceed with the decomposition of the unlabelled data, K-means requires the number of subsets, or in our case, groups, existing in the data. The K-means algorithm requires this parameter as input, and the results are affected by its value. Various heuristics attempt to find an optimal number of groups most of them refer to inter-cluster distance or intra-cluster similarity. Nevertheless, in this case as we know the actual class of each instance, we use the number of classes in the transactional data and employ sil-

houette statistic, described in Section 3.2.2, to determine and select the instances closest to the centre of each class as prototypes for classification.

4.4.3 Solution for Multi-class classification

Multi-class classification involves assigning one of many class labels to an input instance. Formally, given a training dataset of the form (x_i, y_i), where $x_i \in R_n$ is the ith example and $y_i \in \omega_A, \ldots, \omega_K$ is the ith class label, multiclass classification algorithms aim to learn a model H such that $H(x_i) = y_i$ for new unseen instances. [Han and Kamber, 2006, Vapnik, 1995]

Classifiers such as k-nearest neighbours and multi-layered perceptrons can directly deal with multi-class problems. However, for complex classification problems involving a large number of classes, it has been often observed [Hsu and Lin, 2002, Vapnik, 1995], that obtaining a classifier that discriminates between two classes, outperforms the one that simultaneously distinguishes among all classes.

Thus, techniques such as the One-Vs-All method [Vapnik, 1995], the All-Vs-All [Friedman, 1996, Hastie and Tibshirani, 1998], and the Error Correcting approaches [Allwein et al., 2001, Crammer and Singer, 2002, Dietterich and Bakiri, 1995], which decompose the outer-space are often used in solving the multi-class classification problem.

The proposed approach in this thesis, involves binning the customer profiles based on the number of items purchased and selecting prototypes of each of the classes in the discovered bin. Multi-class models are then built using 10-fold cross-validation using the discovered prototypes.

Algorithm 1 formally describes the procedure for training the predictors.

New customer profiles are predicted by first determining their closest bin based on the number of transactions and then using the classifier trained within that bin for the prediction as formally described in Algorithm 2.

4.5 Two-Class Classification Experiments and Analysis of Results

To evaluate the performance of the proposed approach on a two-class real-world transactional data, a series of experiments were performed using transactional data provided by Screwfix of **Electricians** and **PlumbHeaters** covering a period of 30 months.

Many of the recorded customers' trade types do not, however, reflect the "true" trades of the customers mainly due to changes in the transaction behaviour of customers over time.

Table 4.1 shows the total number of the verified Electricians and PlumbHeaters tradetypes used for the evaluation experiment, together with the number of transactions and

Input: Training set P
Output: Predictors C_B, bins cluster centres μ_k^B
Initialize: $S = 0.6$

Bin P into B bins based on the number of items per transaction.;
for $b \leftarrow 1$ **to** B **do**
    Cluster into $K_{Classes}$ groups using K-means, where $K_{Classes}$ is the number of classes in T;
    Record bin cluster centre μ_k^b;
    for *each instance* x_i *in bin* b **do**
        for *each group* $k \in K_{classes}$ **do**
            Compute the Silhouette Statistics sw_i^k;
            if $sw_i^k > S$ **then**
                include x_i into the set of prototypes $P_{kb}^{\star}$
            end
        end
    end
    Train a predictor C_b on instances in $P_{kb}^{\star}$;
end

Algorithm 1: Train predictors

Input: new customer profile x_i, predictors C_B, bins cluster centres μ_k^B
Output: predicted class k_i

Assign x_i to the closest bin $B_p : p = \arg\min \left\| x - \mu_k^b \right\|$;
classify x_i using the predictor C_b;

Algorithm 2: Predict class of new customer profile

Table 4.1: Customer Profile Data

Profile Name	No. Customers	No. Transactions	No. Items Transacted
Electricians	1537	32063	111730
PlumbHeaters	4135	68715	230542

items transacted by the aforementioned trade-types over the period under consideration. Figures 4.1 shows the individual items topics contribution to total number of items topics transacted by the Electricians and PlumbHeaters trade-types over the 30 months as discovered by the binning process.

It can be seen from the plots in Figure 4.1 that the number of instances in each of the discovered bins is statistically sufficient and not too large for efficiently inducing a classifier. The prototype selection algorithm outlined in Section 4.4.2 was therefore not used.

4.5.1 Bin Evaluation using the AUC score

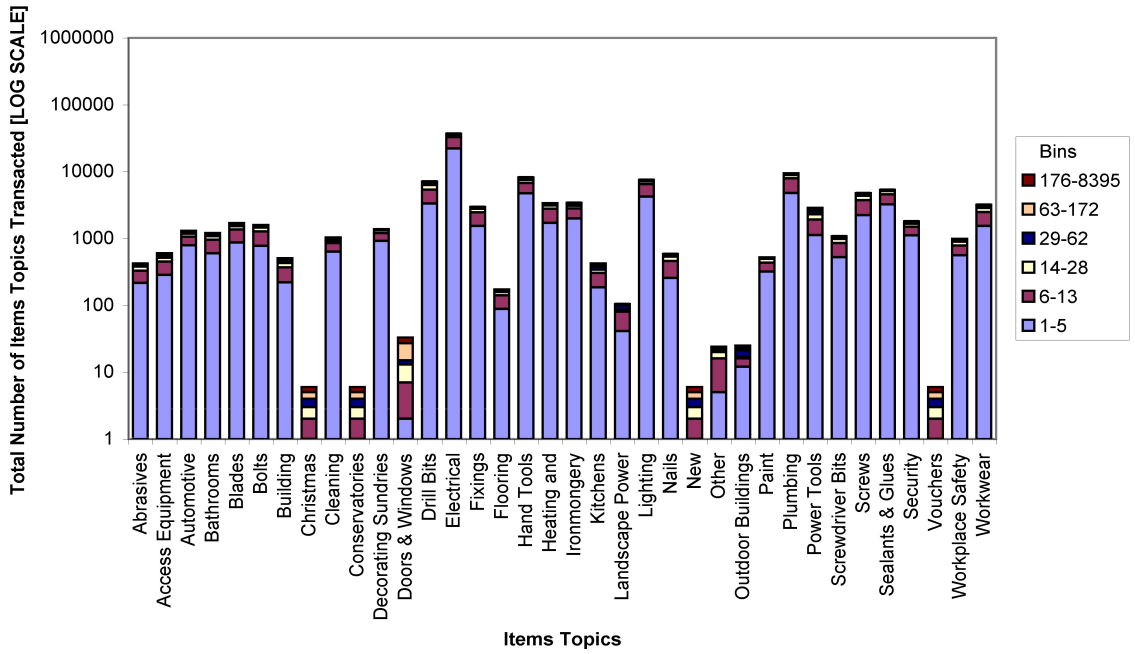
As can been seen in Table 5.1 the Electricians and PlumbHeaters are imbalanced by a ratio of approximately 1:3. The performance of data mining algorithms is typically evaluated using predictive accuracy. However, this is not appropriate when the data is imbalanced and/or the costs of different errors vary markedly [Japkowicz, 2000, Provost, 2000]. This is mainly because the large difference in representation between the classes can lead to a bias in which even a simple default strategy of guessing would give a high predictive accuracy to the majority class [Chawla, 2005].

The Receiver Operating Characteristic (ROC) curve is a standard technique for summarizing classifier performance over a range of trade-offs between true positive and false positive error measures. It is not influenced by decision biases and prior probabilities, and it places the performance of diverse systems on a common, easily interpreted scale [Swets, 1988].

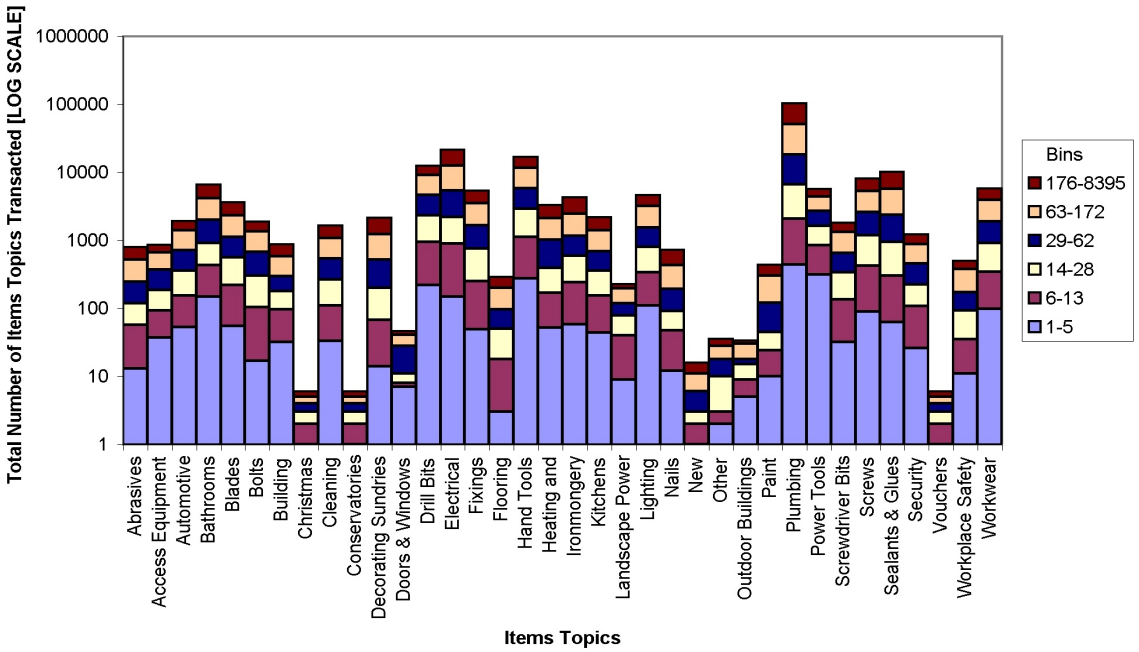
The Area Under the Curve (AUC) is an accepted traditional performance metric for a ROC curve [Bradley, 1997, Duda et al., 2000, Lee, 2000]. The ROC convex hull can also be used as a robust method of identifying potentially optimal classifiers [Provost et al., 1999] and is therefore used in our experiments.

4.5.2 Discussion of Experiment Results

The goal of the experiment was to identify a range of required items per transaction in order to more accurately classify unseen customers to a customer profile. The transactional binning part algorithm outlined in Section 4.4.1 was implemented using Matlab Version 7.9.0.529 (R2009b) on Intel Core2 Duo machine running Microsoft Windows XP while an evaluation of the ROC performance on the identified bins was performed using Weka's implementation of C4.5 [Quinlan, 1993] (implemented as J48 in Weka), linear discriminant classification [Frank et al., 1998] (implemented as Classification via Regression in Weka), Naive Bayes [Pernkopf, 2007] and SVM [Boser et al., 1992, Cristianini and Shawe-Taylor, 2000](implemented as SMO in Weka).



Electricians Items Topics Bar Graphs



PlumbHeaters Items Topics Bar Graphs

Figure 4.1: Stacked Bar graphs showing the contribution of the individual items topics' to the total number of items topics transacted over the 30 months period.

Table 4.2 shows the classification performance on the identified bins while Table 4.3 shows the classification performance on the randomly sampled data of customers. Figure 4.2 shows the comparative ROC performance of the 4 classifiers on both the identified bins and the randomly sampled customers.

Table 4.2: Classification Performance on Identified Bins[a].

(a) C4.5 Decision Tree

Bins	No. Electricians	No. Plumb-Heaters	Bin Size	TP (E)	FP (E)	TP (P)	FP (P)	Accuracy	ROC
1-5	303	775	1078	0.221	0.132	0.868	0.779	0.686	0.599
6-13	302	805	1107	0.325	0.199	0.801	0.675	0.671	0.594
14-28	249	784	1033	0.442	0.18	0.82	0.558	0.729	0.648
29-62	256	761	1017	0.516	0.184	0.816	0.484	0.74	0.687
63-175	*288*	*713*	*1001*	*0.59*	*0.147*	*0.853*	*0.41*	*0.777*	*0.702*
176-8395	139	297	436	0.748	0.101	0.899	0.252	0.851	0.813

(b) Linear Discriminant Classification

Bins	No. Electricians	No. Plumb-Heaters	Bin Size	TP (E)	FP (E)	TP (P)	FP (P)	Accuracy	ROC
1-5	303	775	1078	0.096	0.041	0.959	0.904	0.716	0.649
6-13	302	805	1107	0.129	0.037	0.963	0.871	0.735	0.704
14-28	*249*	*784*	*1033*	*0.289*	*0.07*	*0.93*	*0.711*	*0.775*	*0.791*
29-62	256	761	1017	0.402	0.075	0.925	0.598	0.794	0.836
63-175	288	713	1001	0.573	0.069	0.931	0.427	0.828	0.879
176-8395	139	297	436	0.633	0.047	0.953	0.367	0.851	0.92

(c) Naive Bayes

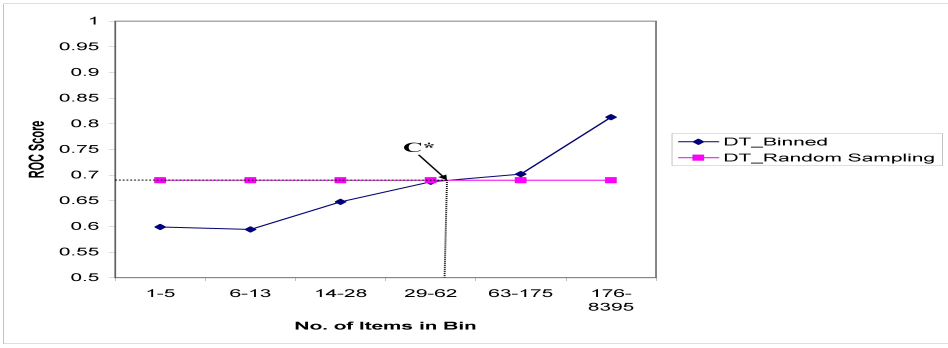
Bins	No. Electricians	No. Plumb-Heaters	Bin Size	TP (E)	FP (E)	TP (P)	FP (P)	Accuracy	ROC
1-5	303	775	1078	0.294	0.183	0.817	0.706	0.67	0.617
6-13	*302*	*805*	*1107*	*0.351*	*0.217*	*0.783*	*0.649*	*0.665*	*0.666*
14-28	249	784	1033	0.522	0.162	0.838	0.478	0.762	0.76
29-62	256	761	1017	0.59	0.171	0.829	0.41	0.769	0.795
63-175	288	713	1001	0.625	0.143	0.857	0.375	0.79	0.846
176-8395	139	297	436	0.41	0.047	0.953	0.59	0.78	0.892

(d) SVM

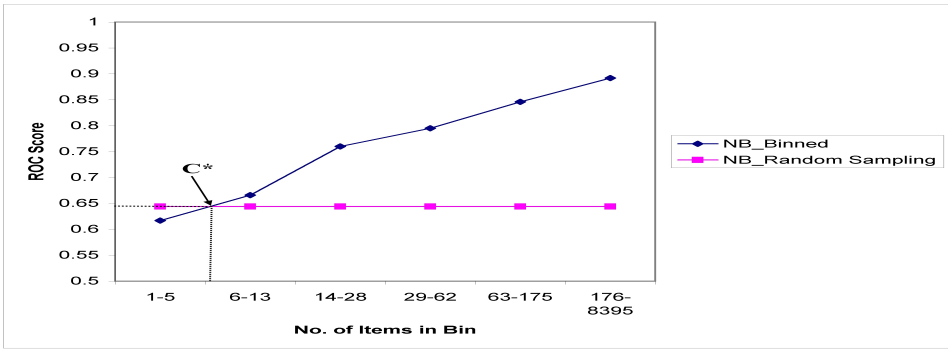
Bins	No. Electricians	No. Plumb-Heaters	Bin Size	TP (E)	FP (E)	TP (P)	FP (P)	Accuracy	ROC
1-5	303	775	1078	0.04	0.017	0.983	0.96	0.718	0.535
6-13	302	805	1107	0.066	0.024	0.976	0.934	0.728	0.616
14-28	249	784	1033	0.309	0.071	0.929	0.691	0.779	0.773
29-62	*256*	*761*	*1017*	*0.445*	*0.072*	*0.928*	*0.555*	*0.806*	*0.821*
63-175	288	713	1001	0.625	0.077	0.923	0.375	0.837	0.877
176-8395	139	297	436	0.784	0.061	0.939	0.216	0.89	0.95

[a]The gray rows represent the performance obtained for the bins at and above the "critical point" which denotes the minimum number of items per transaction required by the classifiers studied to confidently identify and distinguish a customer profile.

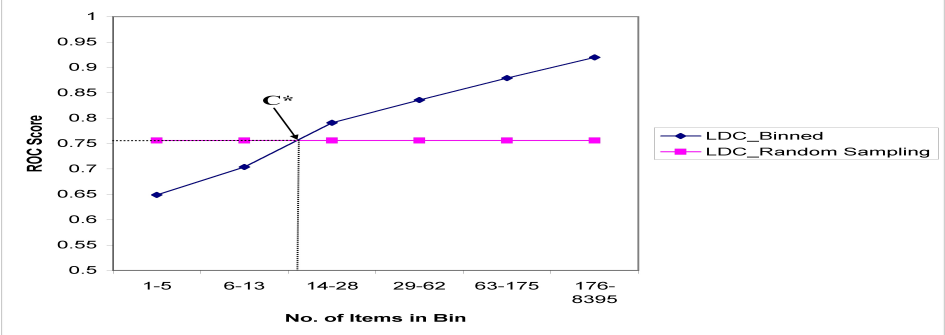
The difficulty in classifying customers with few transactions can be observed from Table 4.2 and Figure 4.2 in which the ROC classification performance values increase as the number of items in the bins increase. The effect of the sparsity and skewness of the transactional data on classification performance can also be seen from Tables 4.2



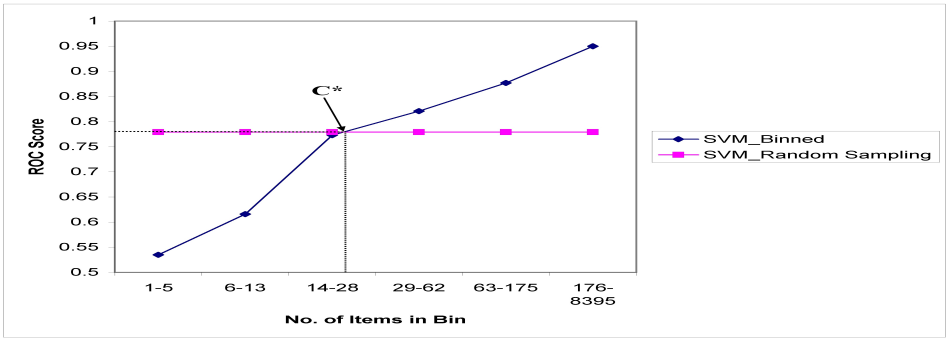
Decision Tree Two-class Classifier Performance



Naive Bayes Two-class Classifier Performance



Linear Regression Two-class Classifier Performance



Support Vector Machine Two-class Classifier Performance

Figure 4.2: Plots showing the two-class 10 fold cross-validation classification performance of Decision Tree, Naive Bayes, Linear Discriminant, and Support Vector Machine on the selected SLIGRO prototypes of customer profiles based on transactional data.

Table 4.3: Baseline Classification Performance on Randomly Sampled Customer Profiles

Classifiers	No. Electricians	No. Plumb-Heaters	Total No. Instances	TP (E)	FP (E)	TP (P)	FP (P)	Accuracy	ROC
C4.5	1537	4135	5672	0.472	0.163	0.837	0.528	0.738	0.69
LDC	1537	4135	5672	0.114	0.007	0.993	0.886	0.755	0.756
Naive Bayes	1537	4135	5672	0.181	0.049	0.951	0.819	0.742	0.644
SVM	1537	4135	5672	0.24	0.033	0.967	0.76	0.77	0.779

and 4.3 where the classification performance is drawn more to the majority PlumbHeaters class, as reflected by the true positive rate measures. Thus, a classifier built using the entire transactional data will be less accurate and less confident in its classification of the customers with larger items per transaction, as can be seen in Figure 4.2. The bins for which the ROC classification performance becomes better than that obtained from the baseline, (bin 63-175 for C4.5, bin 14-28 for LDC, bin 6-13 for Naive Bayes and bin 29-62), can be interpreted as the "critical point" at which the minimum number of items per transaction required for the aforementioned classifiers to confidently identify and distinguish a customer profile, given a dataset of highly sparse and skewed transactions.

4.6 Multi-class Classification Experiments and Analysis of Results

To evaluate the performance of the proposed approach on a two-class customereal-world transactional data, a series of experiments were performed using transactional data provided by SLIGRO Food Group N.V. SLIGRO Food Group N.V. encompasses food retail and foodservice companies, selling to the Dutch food and beverages market.

The provided data consists of 408,625 aggregated SLIGRO customer transactions, collected over three consecutive years. Each aggregated customer transaction record contains information about the customer number, the item number, the number of items purchased and the customer category as stipulated by SLIGRO.

In total 148,601 SKU products were transacted by 65 customer categories. Tables G.1 lists the categories along with their transactions. Figure 4.3 shows the number of distinct top selling items purchased per customer transaction, while Figure 4.4 shows the plots of the mean, standard deviation and maximum number of distinct top selling items purchased per customer transaction.

4.6.1 Experiment Methodology and Results

The main goal of the experiments was to validate the proposed approach on a real-world case study. More specifically, the experiments were aimed at determining the effect of the number of items bought by each category on the classification performance of four (4)

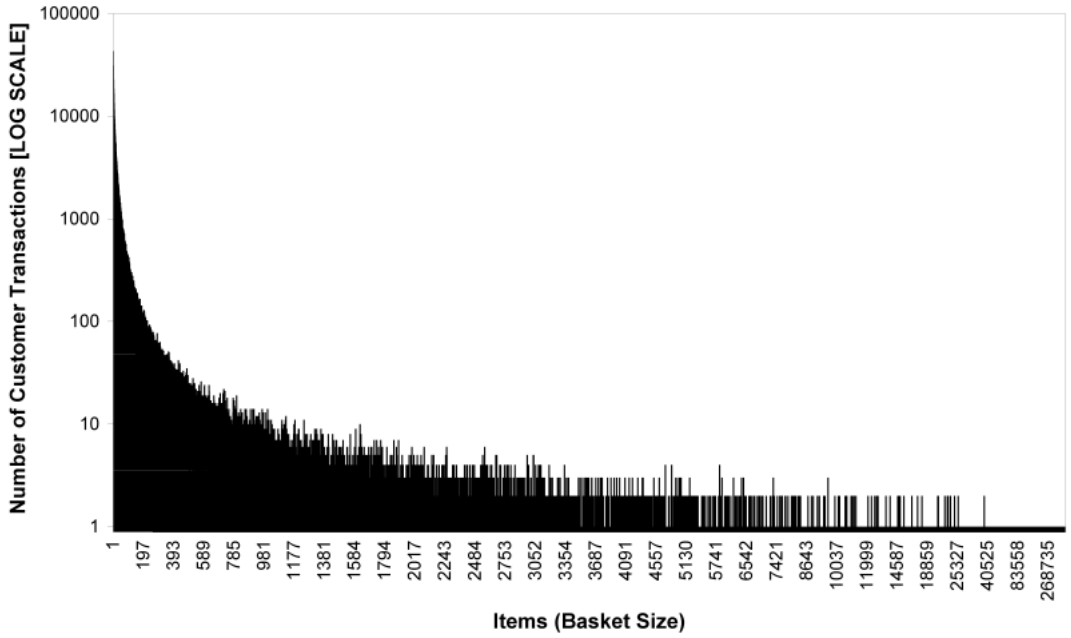


Figure 4.3: Plot showing the distribution of Items per Transaction (Basket size) of SLIGRO's Transactional Dataset

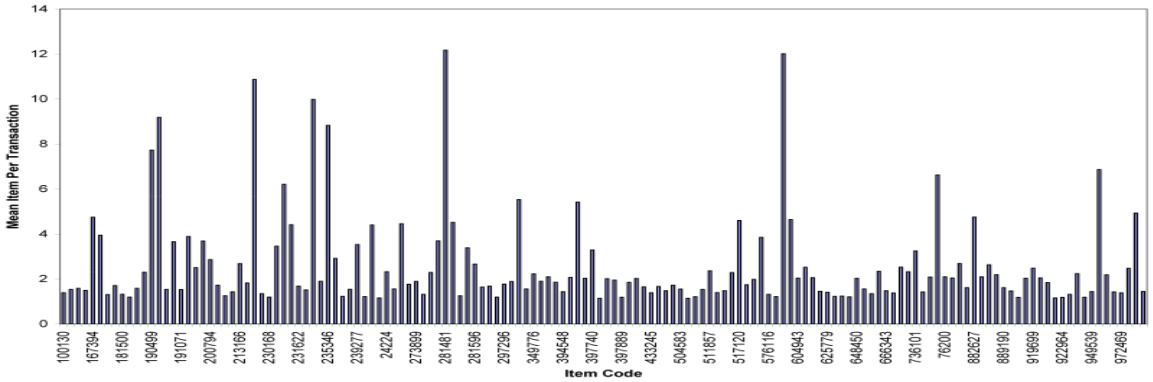
multi-class classifiers. We compared the performance of the approach with the random sampling baseline, which randomly sub-samples customer profiles for classification.

To perform the experiments, customer profiles (i.e. SLIGRO categories) with great-than or equal to 3,000 transactions over the 3 year period were first binned into the groups shown in Table 4.4.

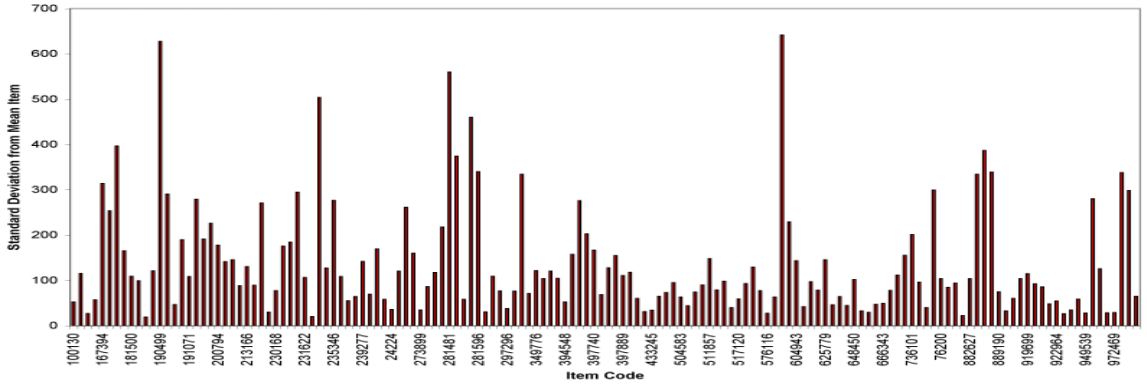
Table 4.4: Identified Data Bins in SLIGRO's Transactional Data.

Categories	Bins				
	1-5	6-20	21-127	128-894222	Total
100	551	470	699	1439	3159
190	4443	3957	2677	300	11377
230	9284	7963	6358	1125	24730
300	1454	1765	2697	2676	8592
310	972	1117	2069	3083	7241
331	970	1052	1713	3235	6970
360	898	957	1064	948	3867
380	714	768	1116	1037	3635
390	974	1071	1814	1593	5452
391	789	935	1140	586	3450
590	1088	1075	1334	900	4397
620	891	1104	1457	1189	4641
800	8718	8749	7904	1701	27072
820	1518	1431	1105	134	4188
840	4051	4396	4089	609	13145
890	1443	1513	1364	217	4537
900	2941	1928	1420	206	6495
Total	41699	40251	40020	20978	142948

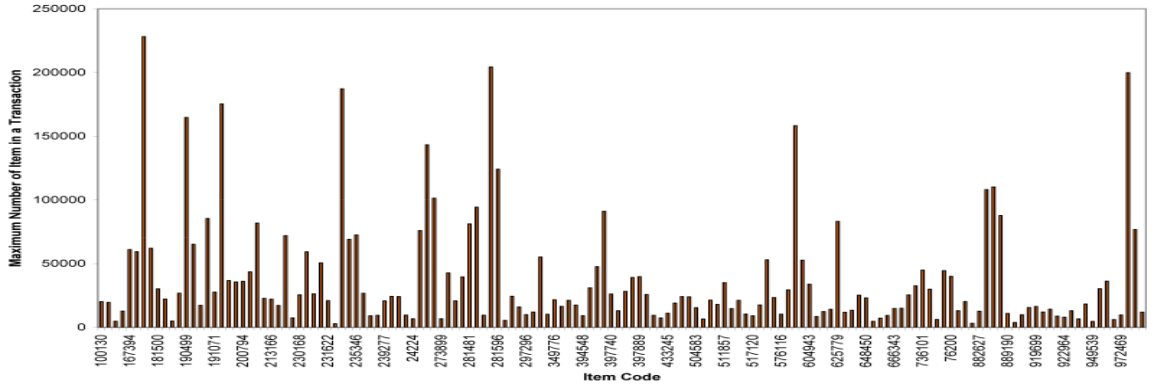
It can be seen that the size of each of the discovered bins was large for the efficiently inducing classifiers. Therefore, the prototypes as numbered in Table 4.5 for each of the categories were then selected using the K-means prototype selecting algorithm outlined in Section 4.4.2.



Mean number of distinct items purchased per transaction



Standard Deviation from Mean of distinct items purchased per transaction



Maximum item purchased in a Transaction

Figure 4.4: Plots showing the Mean, Standard Deviation, and Maximum number of items purchased in a Transaction.

Table 4.5: Number of Selected Prototype per Category per Identified Data Bins in SLI-GRO's Transactional Data.

Categories	Bins				
	1-5	6-20	21-127	128-894222	Total
100	23	17	12	115	167
190	105	185	47	31	368
230	246	381	94	112	833
300	80	72	82	268	502
310	55	36	7	305	403
331	39	39	23	320	421
360	30	43	12	93	178
380	24	22	13	103	162
390	36	42	21	158	257
391	41	42	16	59	158
590	42	56	25	90	213
620	18	60	21	109	208
800	287	328	101	171	887
820	38	51	11	14	114
840	121	199	43	61	424
890	25	69	21	22	137
900	72	68	17	21	178
Total	1282	1710	566	2052	5610

Experimental comparisons were performed on 4 classifiers in WEKA [Hall et al., 2009] using the selected prototypes from each bin. For each of the selected prototypes 10-fold cross-validation was repeated 10 times. To compare classifier performance on the entire multi-class transactional dataset, we use the weighted average AUC, where each target class c_i is weighted according to its prevalence thus: $AUC_{weighted} = \sum_{\forall c_i \in C} AUC\,(c_i) \times p\,(c_i)$

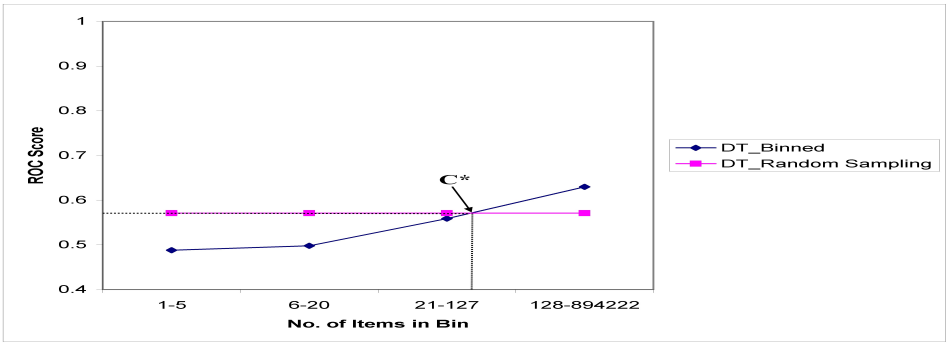
Figure 4.5 shows the plots of the ROC performance results obtained from the compared Decision Tree, Logistic Regression, Naive Bayes, and SVM multi-class classifier models trained using WEKA's built-in OVA and ECOC on the selected prototypes.

4.6.2 Discussion of Results

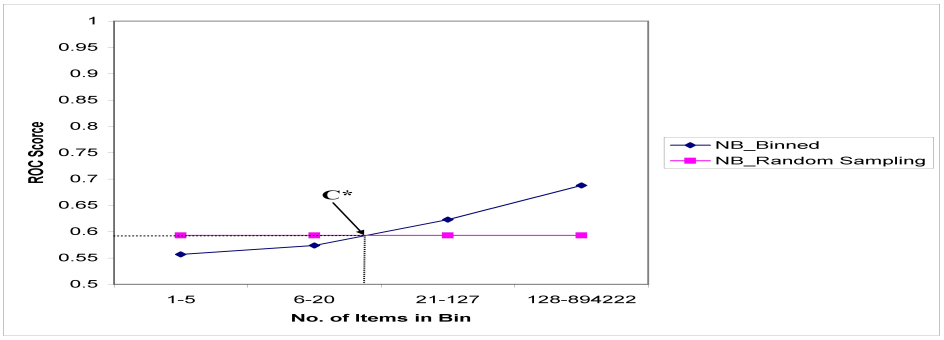
It can be seen from all 4 plots that there exists a critical point, based on the number of items purchased, at which the overall ROC classification performance is higher than that obtained from the standard data mining approach, of random sampling of customer profiles based on transactional data.

From a business perspective, the customers with profiles whose classification fall above the critical point can be prime candidates for direct interactive/one-to-one marketing campaigns while customers whose profiles fall below the critical point can be candidates for general market campaigns.

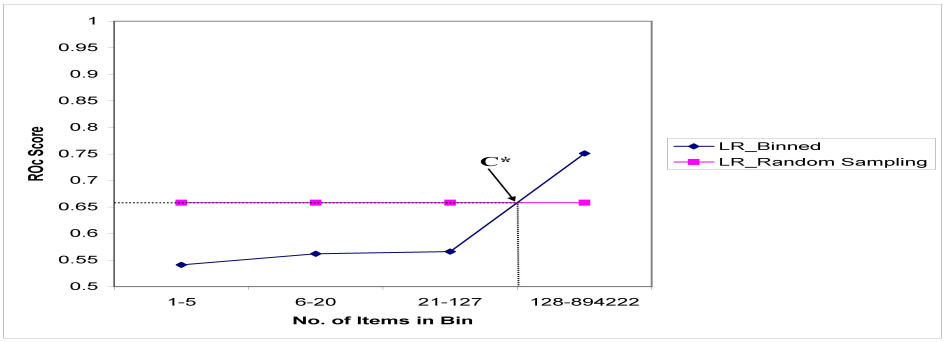
Also, the differences in classification performance on individual categories across the bins provide insight that can be valuable for developing better relationship with the customers. For instance, it can be gleaned from the top 10 items for the category codes 100 and 310 across the four (4) bins in Tables 4.6 and 4.7 that the highlighted product codes have a strong influence on the classification of the customer profiles based on



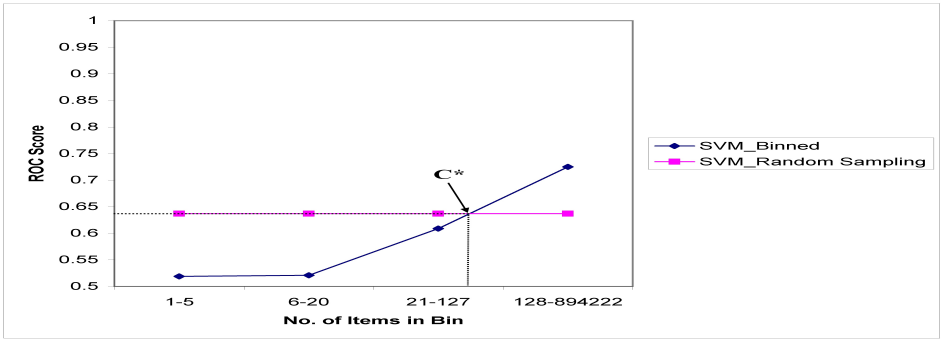
Decision Tree Multi-class Classifier Performance



Naive Bayes Multi-class Classifier Performance



Logistic Regression Multi-class Classifier Performance



Support Vector Machine Multi-class Classifier Performance

Figure 4.5: Plots showing the comparative multi-class classification performance of Decision Tree, Naive Bayes, Logistic Regression, and Support Vector Machine on the selected SLIGRO prototypes of customer profiles based on transactional data.

transactional data in that category. These products could be included in product promotions and customer targeting programmes as incentives for product growth and customer retention.

Table 4.6: Top 10 Product Code Purchased by Category Code 100 (Supermart/rijdende winkel) per Identified Data Bins in SLIGRO's Transactional Data.

Bin1		Bin2		Bin3		Bin4	
Product Code	Total	Product Code	Total	Product Code	Total	Product Code	Total
93456	42	190855	56	190855	193	190855	1710
184532	22	93456	42	882627	180	882627	1657
81491	11	936251	36	936251	156	190499	765
663523	10	24224	12	93456	151	190960	696
637263	2	652378	12	900360	42	235346	677
284099	1	591881	9	397805	8	192653	618
882627	1	64787	8	736177	6	900360	601
190855	0	637263	3	226054	6	438554	557
432257	0	432257	2	433245	4	432257	537
900360	0	226054	2	81491	4	81491	500

Legend: Details of Highlighted Products

Product Code	Product Name
81491	MARK.FRANS STOKBR. 220GR 91038
882627	MARLBORO BOX 19STLB090
190855	HEINEKEN PILS 24X30CL 1 00142
432257	AA DRINK HIGH ENERGY 33CL69733
900360	MARLBORO GOLD 19STLX090

4.7 Summary and Conclusion

This Chapter has presented two approaches for efficiently and confidently identifying and classifying two-class and multi-class customer profiles.

The two-class approach involved combining binning and classification in order to more confidently classify customer profiles using their transactions over time. The use of the proposed approach to discover the buying patterns of Screwfix's Electricians and Plumb-Heaters was presented, together with a discussion on the minimum number of items transacted required to more confidently classify a customer profile into one of two classes. Results shown in tables and plots in Section 4.5 show that there exist a "critical-point" for which the ROC performance for classifying customer profiles, based on transactions, significantly outperforms randomly sampling and classifying customer profiles.

The problem of multi-class classification of customer profiles was also highlighted together with the methods used in addressing it. An approach that involves binning, prototyping and 10 fold-cross validation was then presented together with experiments on using the proposed approach, to classify multi-class customer profiles using real-world transactional data provided by SLIGRO. The results gleaned from the tables and plots in Section 4.6, show that the classification performance, for all the studied classifiers on

Table 4.7: Top 10 Product Code Purchased by Category Code 310 (Cafetaria/shoarma/-fastfood) per Identified Data Bins in SLIGRO's Transactional Data.

Bin1		Bin2		Bin3		Bin4	
Product Code	Total	Product Code	Total	Product Code	Total	Product Code	Total
184532	111	184532	119	297296	146	192653	5062
93456	60	93456	70	93456	92	184532	4782
81491	8	64787	65	882627	63	190902	3689
394548	2	652378	36	936251	60	882627	2329
637263	1	190855	26	24224	54	265943	2130
192653	1	936251	24	736177	13	432257	1868
269117	1	591881	20	591881	11	255710	1578
476997	1	87353	9	282241	8	516140	1569
736177	1	432257	4	900360	7	231708	1483
882627	0	882627	3	192653	6	401963	1450

Legend: Details of Highlighted Products

Product Code	Product Name
184532	T.D.SERV 1L 33X33 WI500ST19180
192653	COCA-COLA REGULAR 33CL 2069
882627	MARK.FRANS STOKBR. 220GR 91038

the prototypes from the binned transactional datasets, depends on the number of distinct items bought as well as the number of the customer category classified in the sampled bin. A rise in either numbers across any of the bins results in a rise in performance.

The results obtained from the experiments validate the proposed approaches on two difficult real world problems. The predictive performance was shown to be consistent across different base classifiers, with the overall accuracy improving with the number of items purchased.

The analysis demonstrates that it is possible to find a critical number of items to be purchased to ensure accurate classification. Knowing this point allows for the filtering of customers and for focused marketing activities to be undertaken on the ones where better predictive accuracy can be expected. The results from the case studies also illustrated that the proposed approach can be used not only for the prediction of new customer profile classes, but also for business analysis, as closer insights can be gleaned from the predicted customer profiles, thereby enabling a better understanding of the customers of the business.

In Chapter 5, we present the investigations of mechanisms for mining change in Customer profiles over time.

Chapter 5

Change Mining of Customer Profile Classifications

Customer profiles based on real-world transactional data tend to change over time as customers change their buying behaviour in reaction to the change in their circumstances, the market, the business, etc. For instance, consider a retail shop with the customer type: Plumber. The buying behaviour of the Plumber customer type may be different in summer and winter e.g., increase in the transaction of plumbing tools during winter to meet an increase in demand for plumbing work (e.g. replacing freezing pipes, suction tubes, etc.); and in the summer to meet an increase in demand for gardening jobs (e.g. hose-pipes, water pumps, etc.).

Monitoring, detecting and understanding such changes in customer behaviour over time enables businesses to gain better insights into the relationship between their customers' buying behaviour and their attitude towards the business over time.

Change mining is a recent paradigm that encompasses mechanisms that monitor models and patterns over time, compare them, detect changes and quantify them on their interestingness [Böttcher, 2011, Böttcher et al., 2008, 2010, Kruse et al., 2010]. In essence, change mining concentrates on understanding the changes themselves. This includes detecting when change occurs in the population under observation, describing the change and pro-acting towards it.

This chapter presents the use of change mining to monitor, detect and visualize the change in the classification of customer profiles built using transactional data.

The chapter begins by providing a background overview of Change Mining together with related work. The investigation of mining change of customer profiles overtime is then presented in Section 5.2. The work on change detection is then presented in Section 5.1. The chapter concludes with a discussion on deciding what to do with the results from change mining and change detection. In the context of customer profile classification using transactional data, this will mean laying the foundation for tackling the important question of whether to adapt (or change) the model or incorporate a new labelling scheme

which is the subject of Chapter 6.

5.1 Background Overview of Change Mining and Related Work

Traditionally, data mining has focused on synthesising knowledge from a static world, in which data instances are collected, stored, and analysed, to derive models for the purpose of making decisions based on the knowledge inferred from them.

The real world is however dynamic and data instances tend to constantly change. Mining and analysis, therefore need to be done on the fly. Research in data stream mining have shown adaptive classifiers to be resilient to high prediction bias as the data change [Klinkenberg, 2004, Widmer and Kubat, 1996].

However, the challenge does not only lay in adapting the models to the evolving data but also to analysing how the models change and when they do so. Change mining is a recent approach for mining evolving data and encompasses methods that capture the process of change, analyse how models have changed and proscribe pro-acting measures on changes that have been discovered [Böttcher, 2011, Böttcher et al., 2010, Kruse et al., 2010].

Change mining involves processing a temporal sequence of datasets with the goal of inferring the changes experienced by their models during the elapsed time.

Change mining has found applicability in fast paced business domains such as retail [Böttcher et al., 2009, Song et al., 2001], manufacturing [Günther et al., 2006], telecommunication [Jin and Zhu, 2007], etc. where it is crucial to quickly detect emerging trends and make proactive rather than reactive decisions as early as possible. For instance, Günther et al. [2006] use a change mining approach to mine change logs in adaptive process management systems. Their approach provides an aggregated overview of all changes that happened up to the point of the change mining process, which can serve as a basis for process improvement actions. Similarly, Böttcher et al. [2009] use an approach based on the discovery of frequent itemsets and the analysis of their change over time. This results in a change-based notion of segment interestingness, to detect arbitrary segments and analyse their temporal development.

Formally, let $T = \langle t_0, \ldots, t_n \rangle$ be a sequence of time points and let D_i be the dataset accumulated during the interval $(t_{i-1}, t_i]$, where D_i may be a static dataset, whose records do not possess timestamps themselves, or a stream of records. Furthermore, let $f\,()$ be a decay function which determines which data contribute in the learning process[1] and with which weights. For example, $f\,()$ can express a sliding window W of length w, so that all

[1]The learning process alluded to here is the "learning" of the class of the customer profile as outputted by the static classifier in each of the varying time windows.

data in the interval $[t_{i-w}, t_0]$ are used for learning, (or $f\,()$ may be an exponential function of ageing, which assigns weights to the individual records on the basis of their age). At each sliding window $W_i, i > 0$, we observe a model (or a set of local models) Ξ_i, inferred from the dataset $\widehat{D}_i := f\left(\cup_{j=i-w}^{i} D_j\right)$.

Change Mining is defined as encompassing:

1. methods which describe the changes of Ξ_i to $\Xi_j, j > i > 0$ and

2. methods which build a predictive model over the sequence $< \Xi_1, \ldots, \Xi_n >$.

Hence, just like conventional data mining, Change Mining has a descriptive and a predictive subcategory of algorithms.

For Change Mining the description of changes among models involves two core tasks:

1. deciding on whether the models are indeed different and quantifying the difference,

2. semantically describing and interpreting the differences.

The prediction of changes in a sequence of models, on the other hand, implies building a higher order model, which determines whether the next member of the sequence will be different from the members seen thus far.

Change Mining essentially constitute a methodological process of four (4) generic tasks [Böttcher et al., 2008]:

1. Determining the goals of Change Mining:

   - Deciding between description of change or prediction of change, with change description being a prerequisite for change prediction.
   - Determining whether the interest lays with the result of change or with the process of change itself

2. Specifying a model of time:

   - Determining the type(s) of model to be studied
   - Determining the granularity level of change to be studied. That is deciding between the study of a whole model (such as a classifier) and of its components (such as individual classification rules, clusters, association rules, profiles) - these are the objects of change
   - Identifying the types of change that can occur on the selected objects of change

3. Designing a monitoring mechanism:

- Designing a method for tracing models or their components over time, depending on the specified objects of change
- Designing an algorithm for the identification of model (or model component) changes
- Designing an algorithm that captures the changes in the model (components)
- Extracting interesting changes
- Semantically interpreting change

This thesis focuses on the descriptive aspect of change mining. It presents an investigation of using change mining to monitor and evaluate the classification of dynamic customer profiles by a two-class decision tree ensemble over varying time windows using real-world time-stamped transactional data.

In particular, using the classifications from the generated decision tree ensembles in Section 5.2.1, the thesis achieves the following change mining objectives:

Objective 1 The discovery and description of the evolution of the class of customer profiles over time windows.

Objective 2 The discovery and description of differences in the classification in adjacent and/or non-adjacent time windows.

In this work, we define the classification of the customer profile built using the customer transactions as the object of change and assume that the customer transactions are aggregated in 3, 6, 9 and 12 monthly intervals. Furthermore, we define $f\,()$ to be a decay function which determines which data contribute in the inference process to be a sliding window of length w, so that all the data preceding the current time window are included in the inference process, i.e. $\widehat{D}_i := f\left(\cup_{j=i-w}^{i} D_j\right)$.

Thus, in terms of the change mining Objective 2, the time axis is partitioned into 3, 6, 9 and 12 monthly intervals. At the end of each monthly interval, the decision tree ensemble classifies each customer profile, Ξ_i based on their aggregated transactions within the time window.

5.2 Proposed Approach for Change Mining of Transactional Data

Performing the aforementioned Change Mining tasks in a transactional data mining setting requires paying particular attention to the way change is detected over time. This is mainly because the varying sparsity and skewness inherent in transactional data makes it challenging to detect the change in the distribution of the input. This is particularly

challenging for unstable learning algorithms such as multi-layer perceptrons and decision trees for which small changes in the input training samples may cause dramatic changes in the resulting models [Li and Belford, 2002]. Thus, large changes in the learned classifier structure may be observed even if the underlying population remains unchanged.

To overcome the above stated challenges, this thesis proposes an approach that involves choosing a time window and classifying customer profiles, based on their accumulated transactions in the chosen time window. The change in the classifications of the customer profiles over time are then monitored and quantified, using our introduced stability measure. The introduced stability measure is focused on quantifying the change in the classification of customer profiles by way of their classification over time, by the classifiers and not the change due to the change in the structure of the trained classifiers themselves.

The following sections describe the proposed approach in terms of a system consisting of preprocessing, training, inference, and change mining.

5.2.1 Change Mining System

Here we present an approach for mining change in customers' time-variant transaction behaviour, for the purpose of monitoring a classifier's performance, with the goal of maintaining robust decision support over time.

Given the nature of transactional data and the problems as described in Chapter 2, in Chapter 4 of this thesis proposed and evaluated an approach based on developing separate classifiers for different groups of customers, depending on the number of items bought. We observed from experiments, using the proposed approach on real-world data, that the discriminative power of the classifiers heavily depends on the need for a sufficient number of items to be bought in order to be able to create informative, and sufficiently discriminative profiles.

In this section, the proposed approach builds on the previous work and is designed to be applied in a dynamic retail environment. Combined with data binning, (related to the procedures described in Chapter 4), the proposed approach has resulted in a classification system that incorporates multiple time windows in which customers are classified based on their accumulated transactions in the respective time windows, and the different classification decisions from the different time windows optimally combined.

The classification system consists of two phases of training and inference as outlined in Figure 5.1.

5.2.2 Training Phase

In the training phase, each customer's transactional data in the training dataset, accumulated over a long period of time (in our case 30 months), is aggregated and classifier

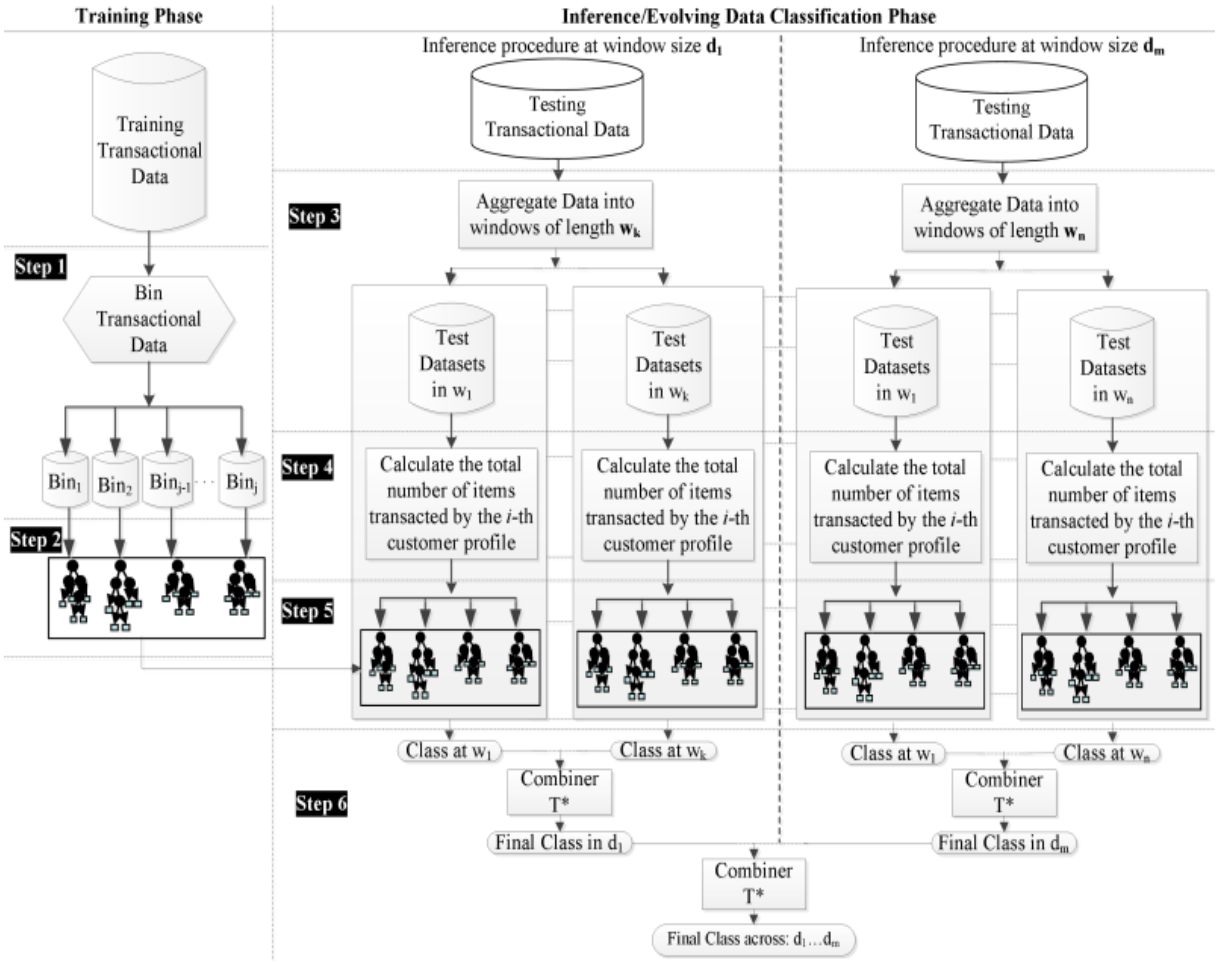


Figure 5.1: Architecture for change mining a classifier ensemble over time. The highlighted steps are described in Sections 5.2.2 and 5.2.3

models derived for change mining of evolving transactional data. The phase comprises of the following steps:

Step 1: Binning of data for training purposes

In order to overcome the problem of skewness that is inherent in sparse transactional data, we use the equal-frequency data binning algorithm outlined in Section 4.4.1 of Chapter 5 to partition/group customer transactions by the number of items purchased.

Step 2: Training individual classifiers on respective bins

Classifier models are then independently generated from each of the bins returned by the data binning process. Different base classifiers can be used. A number of them were evaluated in Section 4.5.2 of Chapter 4 within the framework shown in Figure 5.1 on the left, and the decision tree learner was chosen as the base classifier to perform the classification process in this thesis, due to its relatively similar performance to the other classifiers evaluated and its ease of interpretation.

5.2.3 Inference/Evolving Data Classification Phase

The inference phase involves using the classifier models derived from the training phase to classify and mine previously unseen customer transactions for change, using a set of user specified time window sizes. It includes the following steps:

Step 3: Representing the transactional data for the customers in different sizes of sliding time windows

In this step, previously unseen transactional data of customers over time are aggregated and represented in different window sizes each of which of which have moving time windows of varying lengths. For instance, in our case, the customer profiles for the two classes (i.e. Electricians and PlumbHeaters) studied, were aggregated over sliding windowing sequences on the basis of their previous 3, 6, 9 and 12 months transactions, which resulted in 28, 25, 22, and 19 sliding windows dataset partitions respectively.

Step 4: Directing the aggregated transaction data in each of the different sequence of time windows to appropriate classifiers.

The profiles based on the aggregated transactional data for each customer in each sequence of time windows are then directed to the suitable classifier, based on the number of items purchased in a given time window.

Step 5: Classification of the customer profiles using the appropriate classifier in each of the different time windows.

The previously unseen aggregated transaction data in each of the different time windows, from Step 4, are then classified by the appropriate (based on the number of items transacted) trained classifier from Step 2.

Step 6: Combining the outputs from the different time windows.

In an evolving transactional data setting, it is common for the features of the datasets to continually change as the customer buying behaviour change with time.

Using isolated classifications of individual time windows may give a distorted picture of the customers' true class and an increase in uncertainty especially in cases where the customer's buying behaviour is constantly changing.

One way of reducing the uncertainty due to misclassification or to introduce additional flexibility to the system (if one would like to focus on detecting change), is to combine the outputs from more than one time window using a combiner such as majority voting, gating, etc. [Polikar, 2006, Ruta and Gabrys, 2000]. This can result in more certain/stable classification over time, as well as enhance the ability to detect changes by focusing on or selecting classifiers that are able to follow changes in behaviour.

The strategy used in combining the classifiers are often grouped as [Kuncheva et al., 2001, Subramanian et al., 2010]:

1. *trainable* vs. *non-trainable*; or
2. *class label* vs. *class-specific continuous* outputs.

The parameters or weights used by *trainable* combiners are usually obtained through a separate training algorithm (e.g. the maximum likelihood estimates from the EM algorithms in mixture of experts model, Multilayer perceptron (MLP) networks trained by back propagation, etc.).

The *non-trainable* combiners (e.g. majority vote, min, max, etc.) on the other hand, use the classifiers' outputs without incorporating any other information, i.e. there is no separate training involved, beyond that used in inducing the ensemble members.

In the case of the class label combiners, like behaviour knowledge space, weighted majority voting, majority voting, etc., only the classification decision from the classifiers is needed, while methods like algebraic combiners, decision templates, weighted average, etc., need the continuous-valued outputs of individual classifiers. These values often represent the degrees of support the classifiers give to each class.

This thesis investigates the use of majority Voting, weighted majority voting, weighted average voting and minority voting combiners from the second grouping in an evolving transactional data setting.

Majority Voting Combiner

In our setting, the majority voting combiner focuses on the more frequent class. We define the decision to choose the j-th class at window $W_i, i > 0$ as $h_{W_i,j}$ for a system of W_n class decisions: $H = \{H_1, \ldots, H_{W_n}\}$ and C classes: $\omega = \{\omega_1, \ldots, \omega_C\}$. If the class at W_i-th window is ω_j, then $h_{W_i,j} = 1$, and 0, otherwise. Thus, in our case, the class with the most representation across the windows studied was chosen as the final class ω_d, if:

$$\omega_d = \arg \max_{j=1}^{C} \sum_{i=1}^{n} h_{W_i,j} \tag{5.1}$$

Ties were resolved arbitrarily.

Weighted Majority Voting Combiner

In the case of the weighted majority voting combiner, the class values in each window were assigned weights θ_{W_i} based on the accuracy value of the classifier performance, obtained on the training dataset in each bin as an estimate of the classifier's future performance. That is, formally, if the total weighted vote received by ω_d across the windows is higher than the total vote received by any other class, then ω_d will be chosen, if:

$$\omega_d = \arg \max_{j=1}^{C} \sum_{i=1}^{n} \theta_{W_i} h_{W_i,j} \tag{5.2}$$

Weighted Average Combiner

For the weighted average combiner, each classifier in each window were assigned weight θ_{W_i} in order to determine the chosen class. Thus, we have a class-conscious combination [Kuncheva et al., 2001] of a total of $\theta_{W_n} \times H_{W_n}$ weights that are class and classifier specific, with the final class ω_d chosen if:

$$\omega_d = \arg \max_{j=1}^{C} \sum_{i=1}^{n} \theta_{W_i,j} h_{W_i,j} \tag{5.3}$$

where $\theta_{W_i,j}$ is the weight of the i-th classifier for classifying class ω_d. In our case, the weight $\theta_{W_i,j}$ was obtained by combining the True Positive rate values for each class with the accuracy of the classifier performance obtained from the training phase, i.e. for example using the performance values in Table 5.3, the $\theta_{W_i,j}$ for Electricians who bought 1-5 items will be 0.15 while $\theta_{W_i,j}$ for PlumbHeaters who bought 1-5 items will be 0.6.

Minority Voting Combiner

The aforementioned majority vote based combiner rules focus more on the regular classes (in terms of numbers) obtained by the base classifiers and normally have a stabilizing and

improving effect on the classification performance. They, thus,provide for the monitoring of the more common behaviour of a customer over time.

However, rare events, such as a customer's change in buying behaviour occasioned by environment stimuli, might result in the temporal change in the classification of the customer within the time period of the environment stimuli. These rare events, which could be of interest for gaining insights into how customers' respond to external environment stimuli, would be missed by combiners, which are more focused on monitoring or detecting frequent events. Such rare events can be useful insight for better product positioning as well as getting a holistic view of a customer's buying behaviour over time, for better customer relationship management.

To monitor or detect the changes occasioned by the rare, or less common, changes in customer behaviour over time we propose Minority Voting.

Given a set of votes on C classes: $\omega = \{\omega_1, \ldots, \omega_C\}$ by W_n base classifiers: $H = \{H_1, \ldots, H_{W_n}\}$, we focus on the least common class. If all the base classifiers vote for a class ω_j, we simply output ω_j as the final class. If the majority vote is for a class ω_j, the class ω_l having the lowest number of votes is chosen as the final class. We expect this voting strategy to increase the likelihood of detecting a rare, or less common, change in the behaviour of the customer at the expense of accurate classification of the customer's profile. Formally, the vote of class ω_d will be chosen, if

$$\omega_d = \arg\min_{j=1}^{C} \sum_{i=1}^{n} h_{W_i,j} \tag{5.4}$$

Ties are resolved arbitrarily.

This combination scheme, therefore, aims to detect and highlight differences, rather than compensate for errors, as would be the case with majority vote. It is included here to help with change detection, analysis and mining.

5.2.4 Measurement of Classifier Stability Over Time

Stability is one of the two statistical tests (the other being trend) used in change mining to detect and analyse change [Böttcher et al., 2009].

To quantify and assess the stability of the classifier's predictions over time, we defined the prediction stability measure S for the *kth* customer in a time window w as:

$$S_k = \frac{n - m}{n + m} \tag{5.5}$$

where n is the number of times the *kth* customer was correctly classified using the customer profile generated from the transactional data within time window w, and m is the number of times the *kth* customer was wrongly classified. We define S as a measure of the classification stability normalized between -1 and +1.

The prediction stability is +1 in the case of perfect positive (correct) predictions over time, -1 in the case of a perfect (incorrect) classification over time - indicating a mislabelled customer; and some value between -1 and 1 in all other cases, indicating the level of change in classification, which highlights potential changing buying behaviour over time. As it approaches zero there is less stability in the classification and indicates equal number of classification for both classes over time. The closer the prediction stability is to either -1 or 1, the stronger the stability of prediction with only 1 class being predominantly chosen.

5.3 Experimental Evaluation

5.3.1 Objective of Experiments

This chapter aims to quantitatively capture and analyse the evolution of customer profiles based on their transactions. The analysis is based on monitoring the classification performance of customer profiles by a set of classifiers over time. To achieve this goal, experiments were conducted to:

1. Investigate the effect of varying time windows on the prediction accuracy of customer profiles over time; and
2. Investigate the prediction stability of customer profiles in varying transaction time windows.

To obtain and analyse the change customer profiles modelled, using real-world transactional data, experiments were performed using transactional data, provided by Screwfix of **Electrician** and **PlumbHeater** groups of customers covering a period of 30 months. Table 5.1 shows the total number of the verified Electrician and PlumbHeaters trade-types used for the evaluation experiment, together with the total number of transactions and items transacted by the aforementioned trade-types over the period under consideration.

Table 5.1: Customer Profile Data

Profile Name	No. Customers	No. Transactions	No. Items Transacted
Electricians	1537	32063	111730
PlumbHeaters	4135	68715	230542

5.3.2 Experimental approach and Analysis of Results

Data Partitioning using Data binning

To obtain the training and test dataset used for the experiments, we ran the dataset summarized in Table 5.1 through Step 1 (i.e. the data binning process) of the process outlined in Figure 5.1 in order to obtain the data bins shown in Table 5.2.

Table 5.2: Identified Data Bins

Bin	Bin Size	No. Electricians	No. PlumbHeaters
1-5	1078	303	775
6-13	1107	302	805
14-28	1033	249	784
29-62	1017	256	761
63-175	1001	288	713
176-8395	433	136	297

The first 5 bins were then sub-selected as the training dataset and the last bin as the test dataset for the experiments. The decision to sub-select the binned dataset in this manner was driven by the results, shown in Table 5.3, which were obtained from inducing C4.5 decision trees on each of the 6 data bins using a 10-fold cross validation process.

Table 5.3: C4.5 Decision Tree Classification (10-fold Cross Validation) Performance on Identified Bin

Bin	Bin Size	TP(E)	FP(E)	TP(P)	FP(P)	Accuracy
1-5	1078	0.22	0.13	0.87	0.78	0.69
6-13	1107	0.33	0.20	0.80	0.68	0.67
14-28	1033	0.44	0.18	0.82	0.56	0.73
29-62	1017	0.52	0.18	0.82	0.48	0.74
63-175	1001	0.59	0.15	0.85	0.41	0.78
176-8395	433	0.75	0.10	0.90	0.25	0.85

As can be seen, and as one would expect, the larger the number of transacted items the more accurate the classifier. More details can be found in our previous investigation on customer profile classification in Chapter 4.

Inducing Decision Trees from Binned Data

Using the algorithm outlined in Figure 5.1, we derived five (5) Decision Tree classifier models from the training datasets generated from binning the data. Weka's implementation of C4.5 [Quinlan, 1993](implemented as J48 in Weka) was used to induce the decision trees [Hall et al., 2009].

Effect of Window Size on Prediction Accuracy

To experimentally determine the effect of the time window size on the quality of the prediction, we regrouped the 30 months transactions for the 433 high purchasing customers discovered by the data binning process in Section 5.3.2 into 3, 6, 9 and 12 months sliding window snapshots, which resulted in 28, 25, 22, and 19 dataset partitions respectively. The aggregated transactions for each customer, in each of the partitions, were then passed

through the inference phase of Figure 5.1 to obtain the decision tree ensemble's prediction of each customer. Table 5.4 illustrates the classification output obtained for two customers: an Electrician and a PlumbHeater.

Table 5.4: An example illustrating the Classifications of an Electrician and a PlumbHeater over Time by the Decision Tree Ensemble outlined in Figure 5.1

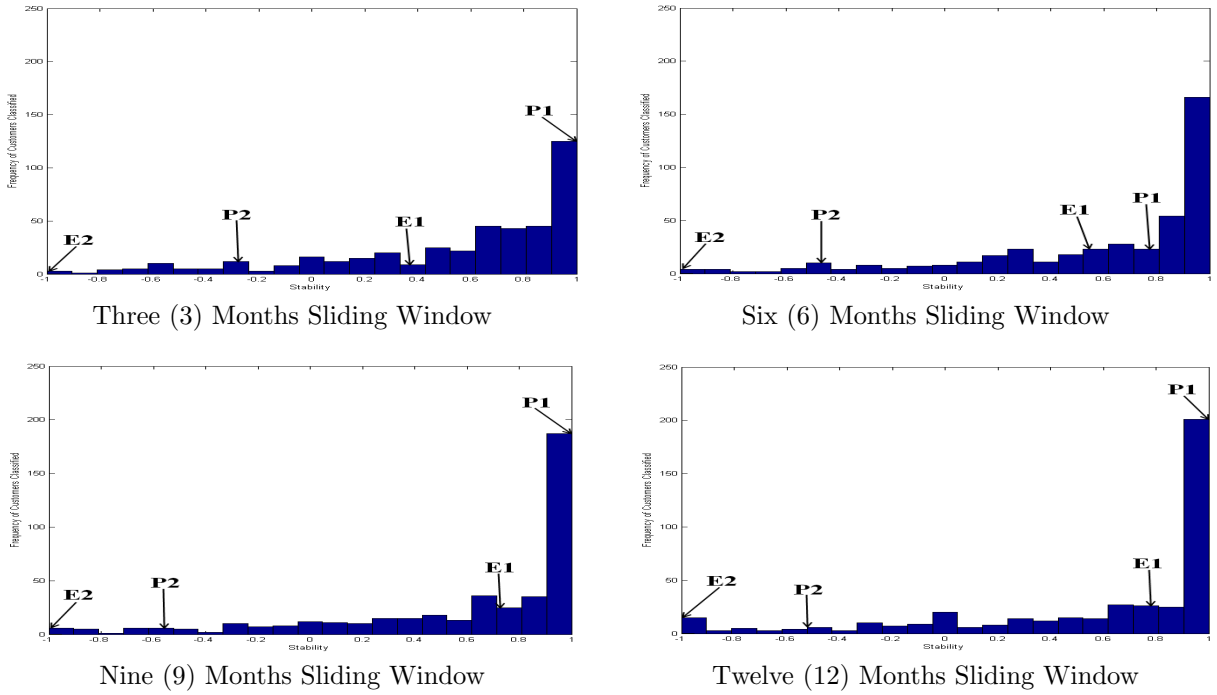
	Electrician (E)								PlumbHeater (P)							
Sliding Window Dataset Partitions	Class in 3 Months Window	Class in 6 Months Window	Class in 9 Months Window	Class in 12 months Window	Majority Vote Across Windows	Weighted Majority Vote Across Windows	Weighted Average Across Windows	Minority Vote Across Windows	Class in 3 Months Window	Class in 6 Months Window	Class in 9 Months Window	Class in 12 months Window	Majority Vote Across Windows	Weighted Majority Vote Across Windows	Weighted Average Across Windows	Minority Vote Across Windows
1	E				E	E	E	E	P				P	P	P	P
2	E				E	E	E	E	P				P	P	P	P
3	P				P	P	P	P	P				P	P	P	P
4	P	E			E	E	P	E	E	P			P	E	P	E
5	E	E			E	E	E	E	P	P			P	P	P	P
6	P	E			P	E	P	E	P	P			P	P	P	P
7	E	E	E		E	E	E	E	P	E	P		P	P	P	E
8	E	E	-		E	E	E	E	P	P	P		P	P	P	P
9	E	E	E		E	E	E	E	P	P	P		P	P	P	P
10	P	E	E	-	E	E	P	P	P	P	P	P	P	P	P	P
11	E	E	E	-	E	E	E	E	P	P	P	P	P	P	P	P
12	P	P	E	-	P	P	P	E	P	P	P	P	P	P	P	P
13	E	P	E	-	E	E	P	P	P	P	P	P	P	P	P	P
14	P	P	E	-	P	P	P	E	P	E	P	P	P	P	P	E
15	E	P	E	-	E	E	P	P	P	E	P	P	P	P	P	E
16	E	E	E	-	E	E	E	E	P	P	P	P	P	P	P	P
17	E	E	P	-	E	P	P	P	P	P	P	P	P	P	P	P
18	P	E	P	-	P	P	P	E	P	P	P	P	P	P	P	P
19	E	E	P	-	E	P	P	P	P	P	P	P	P	P	P	P
20	E	E	E	P	E	E	P	P	P	P	P	P	P	P	P	P
21	E	E	E	E	E	E	E	E	P	P	P	P	P	P	P	P
22	E	E	E	E	E	E	E	E	P	P	P	P	P	P	P	P
23	E	E	E	E	E	E	E	E	P	P	P	P	P	P	P	P
24	E	P	E	E	E	E	P	P	P	P	P	P	P	P	P	P
25	E	E	E	E	E	E	E	E	P	P	P	P	P	P	P	P
26	P	P	E	E	E	E	P	P	P	P	P	P	P	P	P	P
27	P	E	E	E	E	E	P	P	P	P	P	P	P	P	P	P
28	E	E	E	E	E	E	E	E	P	P	P	P	P	P	P	P
Majority Vote in Windows	E	E	E	E	E				P	P	P	P	P			
Weighted Majority Vote in Windows	E	E	E	E		E			P	P	P	P		P		
Weighted Average in Windows	P	P	P	P			P		P	P	P	P			P	
Minority in Windows	P	P	P	P				P	E	E	-	-				E
Accuracy Stability Measure	0.36	0.52	0.71	0.78	0.64	0.71	-0.07	0.21	0.93	0.76	1	1	1	0.93	1	0.71

A comparative look at the corresponding prediction stability measures in Table 5.4 show that the classification of the Electrician is less stable over time than that of the PlumbHeater for the 4 time windows studied. The sporadic change in classification of the Electrician in the 3 and 6 months intervals heighten the uncertainty attached to making a proactive decision in reaction to the change in customer's profile. This is mainly because the change in the customer's profile could be due to a change in buying behaviour (and thus a real change, worth reacting to) or due to a misclassification by the decision tree ensemble.

Individual Classifier Stability and Accuracy Over time

To measure and analyse the prediction stability and accuracy of the classification over time, the time-stamped transactions for the last bin in Table 5.3 (i.e. the 433 verified high purchasing customers spanning 30 months) were aggregated into 3, 6, 9 and 12 months time windows. Each of the sub-samples were then passed through the inference phase part of the algorithm outlined in Figure 5.1.

Figure 5.2 shows the prediction stability distribution while Table 5.5 shows the overall results obtained for the 4 tested time windows.



Three (3) Months Sliding Window

Six (6) Months Sliding Window

Nine (9) Months Sliding Window

Twelve (12) Months Sliding Window

Figure 5.2: Plots showing the prediction stability of the Decision Tree Ensemble models within the 3, 6, 9 and 12 months sliding windows.

It can be seen from the plots in Figure 5.2 and from Table 5.5 that the average prediction stability values (calculated as a mean of absolute stability values for the dataset) increase with the window size. Also, it can be seen in Table 5.5 that the values of the average misclassification rates fall over time, with the lowest value occurring for the 12 months window size.

Table 5.5: Individual Classifier Average Prediction Stability and Average Misclassification Rate Over Time

Sliding Windows Months	Average Prediction Stability	Average Misclassification Rate
3	0.66	0.18
6	0.67	0.17
9	0.68	0.15
12	0.69	0.13

This means, in terms of change detection, that the ability to quickly detect changes in the customer behaviour in the shorter time windows comes at the expense of lower accuracy, with the longer windows being more stable and accurate with regard to the accuracy calculated with the *a priori* class labels.

This information would be valuable, for instance, in determining the length of time a customer transactional behaviour should be observed before including the customer in a direct marketing campaign.

Furthermore, the information from the prediction stability over time can be useful for monitoring the changing behaviour of customers and improving the decision support provided by the decision tree ensemble. For instance, the following statements can be made about the four (4) customers with prediction stability shown in Table 5.6 (and highlighted in Figure 5.2):

Table 5.6: Typical Examples of Prediction Stability Distribution Over Time

Customers	Windows				Majority Vote Across Windows	Weighted Majority Vote Across Windows	Weighted Average Across Windows	Minority Voting
	3 Months	6 Months	9 Months	12 Months				
Electrician (E1)	0.38	0.52	0.71	0.78	0.64	0.71	-0.07	0.21
PlumbHeater (P1)	1	0.76	1	1	1	0.93	1	0.71
Electrician (E2)	-1	-1	-1	-1	-1	-1	-1	-1
PlumbHeater (P2)	-0.24	-0.44	-0.54	-0.5	-0.24	-0.24	0.05	0.05

- The increase in the prediction stability in each of the 4 sliding windows for Electrician (E1) reflects an increase in the number of items transacted by the said customer overtime. The values for Electrician (E1) in Table 5.7 further support this interpretation, as it can be seen that over time, the classification is mainly being done using classifiers derived from the larger bins.

- On the other hand, the drop in the prediction stability value of PlumbHeater (P1) into the stability bin ranged: 0.7 to 0.8 in the 6 months time window before falling back into the bin ranged: 0.9 to 1 with an increased stability value of +1 in the 9 and 12 month windows, indicates a temporal change in buying behaviour of the customer in the 6 months window. The true positive (TP) performance values for PlumbHeater (P1) for the 6 months in Table 5.7 further supports the temporal change in behaviour when compared with the expected true positive (TP) performance (in Table 5.3) of the decision tree ensemble.

- The -1 prediction stability value of Electrician (E2) for all 4 time windows is an indication of a strong discrepancy between the customer's buying behaviour and

their *a priori* label; a consistent discrepancy that can be taken to mean that the *a priori* label is incorrect and should be changed.

- Lastly, the negative values of the PlumbHeater (P2) (at the lower 25% percentile) in all 4 windows studied reflects the intermittent buying behaviour of the said customer with a resultant movement between different classes.

Table 5.7: An illustration of the typical data bin allocation and classification performance over time of the Decision Tree Ensemble outlined in Figure 5.1 on two Electricians and two PlumbHeaters.

	Percentage Aggregated Transactions Allocated Per Bin						True Positive Accuracy Rate Per Bin				
		Bins					Bins				
	Sliding Window Months	1-5	6-13	14-28	29-62	63-175	1-5	6-13	14-28	29-62	63-175
E1	3	-	-	21.43	57.14	21.43	-	-	0.57	0.75	0.67
	6	-	-	-	28.00	72.00	-	-	-	1.00	0.67
	9	-	-	-	-	95.45	-	-	-	-	0.86
	12	-	-	-	-	47.36	-	-	-	-	0.89
P1	3	10.71	25.00	50.00	14.29	-	0.67	1.00	1.00	1.00	-
	6	-	8.00	20.00	72.00	-	-	1.00	0.80	0.89	-
	9	-	-	13.64	22.73	63.64	-	-	1.00	1.00	1.00
	12	-	-	-	21.05	78.95	-	-	-	1.00	1.00
E2	3	-	-	3.57	39.29	-	-	-	0.00	0.00	-
	6	-	-	4.00	8.00	36.00	-	-	0.00	0.00	0.00
	9	-	-	4.55	9.09	40.91	-	-	0.00	0.00	0.00
	12	-	-	5.26	10.53	36.84	-	-	0.00	0.00	0.00
P2	3	10.71	14.29	3.57	25.00	21.43	1.00	0.00	0.00	0.29	0.50
	6	4.00	12.00	-	12.00	44.00	1.00	0.00	-	1.00	1.00
	9	4.55	13.64	-	4.55	36.36	1.00	0.00	-	1.00	0.13
	12	-	5.26	-	10.53	26.32	-	0.00	-	1.00	0.40

Effect of Combiners on Classification Over Time

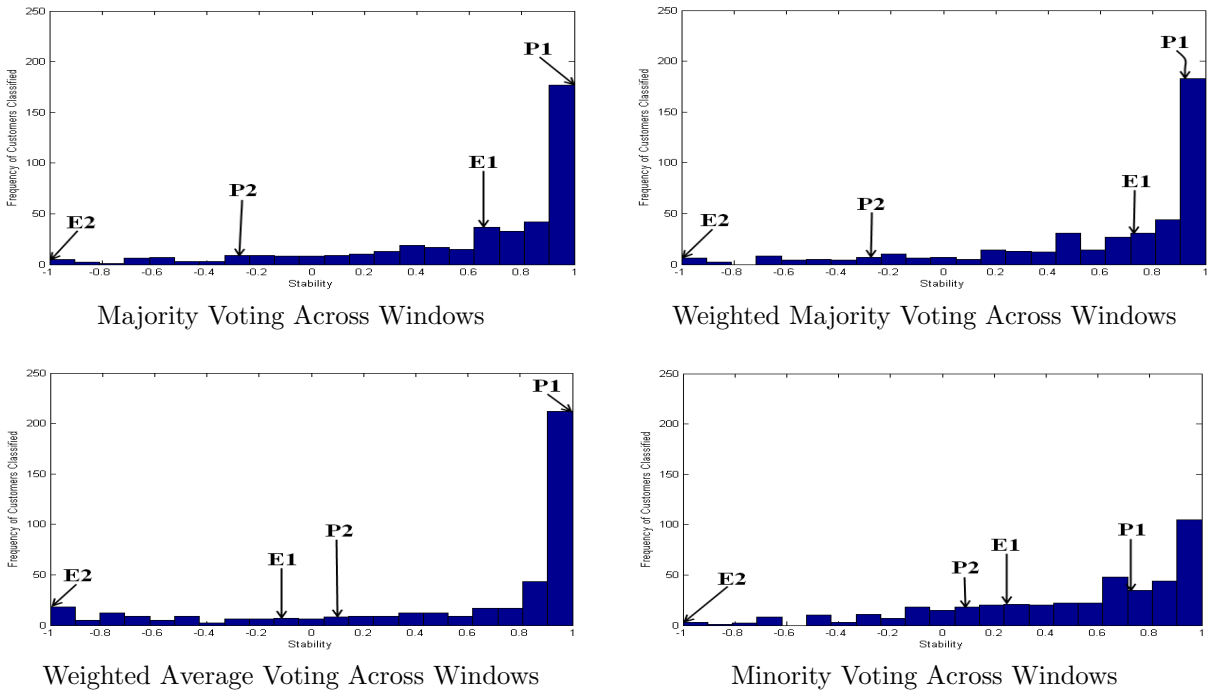
To gauge the performance of the classifiers across the 4 time windows, an analysis of majority voting, weighted majority voting, weighted average and minority voting was performed.

The results, as can be seen in Table 5.8 and Figure 5.3, show that the prediction stability values over time for the first three (3) combiners, which are focused on the frequent class, are higher than those obtained from all the 4 individual time windows studied in Section 5.3.2 as shown in Table 5.5, with the misclassification rate of the weighted average voting scheme at par with that obtained in the 9 months time window. Thus, the three (3) frequent class focused combiners, by incorporating the performance at the shorter time windows can be used to monitor and detect early changes in the customer behaviour as well as provide the prediction stability of the longer time windows; although at the expense of accuracy with regards to the *a priori* class labels.

Furthermore, the prediction stability over time of the three (3) combiners, which are focused on the frequent class, can be crossed checked with that obtained from the minority voting combiner, to gauge the significance of the detected change over time. For instance, it can be seen from Table 5.6 that the higher positive prediction stability values for the customers E1 and P1, for both the frequent class focused combiners and the minority class combiner, are higher than those obtained for E2 and P2 over time. The higher prediction

stability values obtained for the customers E1 and P1 can be interpreted as reflecting temporary changes, which would not necessitate a change in classification labels, while the lower prediction stability values for E2 and P2 can be interpreted as significant changes requiring a re-labelling of the customers (samples) or a re-evaluation of the classifiers responsible for their classification over time.

Also, the class-conscious combination [Battiti, 1994] of the weighted average can be used to monitor the changes in class imbalance between the Electrician class and the PlumbHeater class to which it is biased as can be seen in Table 5.6.



Majority Voting Across Windows

Weighted Majority Voting Across Windows

Weighted Average Voting Across Windows

Minority Voting Across Windows

Figure 5.3: Plots showing the distribution of prediction stability over time for the majority Voting, weighted majority voting, weighted average voting, and minority voting Combiners Across the 3, 6, 9 and 12 month windows.

Table 5.8: Combiners' Average Prediction Stability and Average Misclassification Rate Over Time

Combiner	Average Prediction Stability	Average Misclassification Rate
Majority Voting	0.73	0.16
Weighted Majority Voting	0.73	0.15
Weighted Average Voting	0.79	0.18
Minority Voting	0.61	0.21

5.4 Summary and Conclusion

This Chapter presented a change mining approach for evaluating and analysing a proposed robust technique for classifying time-variant customers based on their transactions over time. The approach consists of two phases of training and inference. The training phase involves inducing a decision tree ensemble from aggregated and binned customer transactions. The inference phase uses the induced decision tree ensemble from the training phase to classify time aggregated and binned customer transactions within a user specified time windows.

Experiments were performed to evaluate the prediction accuracy and stability of our technique within 4 different time windows on real-world time-stamped transactional data obtained from Screwfix Limited.

Overall, it was observed from the prediction stability tables and plots, that the classification obtained by looking at longer time windows rather than shorter windows improves the stability of the classification though reduces the ability to detect and react to short time changes in buying behaviour. However, using any (depending on the application) of the combiners which are focused on the frequent class in conjunction with the proposed Minority Voting combiner, provides the flexibility of monitoring, detecting and verifying short and long term changes in customers' behaviour.

These results show that monitoring classifier accuracy and stability over time can be used for change mining purposes.

Furthermore, the change mining process also provides information that can be valuable in determining the time to confidently target a customer with a direct marketing campaign as well as determining the usefulness of a classifier in a dynamic business domain.

From a company perspective, our work here is very useful as it helps in identifying customers who:

1. Change their buying behaviour frequently moving between different classes;
2. May have been assigned a wrong label in the verification process; and
3. Exhibit stable behaviour with only an occasional change of classification, which may be due to temporary behaviour change or misclassification.

The introduced simple stability measure, the evaluated minority voting (as an example of the decision combination rule focusing on detecting even minor differences), and the visualisation of classification over time, makes identification of these groups much easier.

Chapter 6

Customer Profile Classification: To Adapt Classifiers or To Relabel Customer Profiles?

In Chapter 5, static classifier models, built using data aggregated over 30 months, were used to monitor, detect and describe the change in the classification of customer profiles aggregated over 3, 6, 9 and 12 month sliding windows.

The decision to use static classifier models built using customer profiles aggregated over 30 months, to perform the work in Chapter 5 was based on the findings in Chapter 4, where it was shown that classifier models based on customers who bought more items performed better than classifier models based on customers who bought fewer items.

A classification stability measure was introduced and used to measure the change in classification of customers' over time with the customers whose stability measure was consistently at -1 identified and recommended for relabeling.

However, it is often the case that businesses needs to identify their customers as soon as possible. They might not have the luxury of time to build classifier models of customer profiles based on transactional data accumulated over a long period of time.

Meeting the need for a timely identification of reliable and stable customers is challenging, due to the dynamic nature of customer profiles built using transactional data.

The chapter commences with an overview of the problem of concept drift and label switching in the context of customer profile classification. A background overview of the techniques for adaptation and relabeling, together with related works, is then presented in Sections 6.2.1 and 6.3 respectively. The experiments in which the effect of model adaptation on classification performance, in contrast to the effect of re-labelling is then presented in Section 6.5. A comparative analysis of the results obtained from the experiments is also presented and discussed. The chapter concludes with a discussion on the problem of model adaptation, versus instances re-labelling, in a dynamic transactional data setting. In the context of customer profile classification using transactional data, this introduces

the important research question of whether to adapt (or change) the model or incorporate a new labelling scheme.

6.1 Concept Drift and Customer Profile Class Switching Problem.

As discussed in Chapter 5, customer profiles based on real-world transactional data tend to change overtime as customers change their buying behaviour in reaction to the change in their circumstances, the market, the business, etc.

Classifier models induced from such real-world evolving transactional data thus, tend to deteriorate over time in their predictive accuracy .

This may be due to the change in the distribution of the target variable the model is trying to predict. In this case, the *a priori* unknown change in the statistical properties of the target variable is referred to in the literature as concept drift [Tsymbal, 2004]. Concept drift complicates the classification task as the model needs to be able to track and adapt quickly, to unanticipated changes.

The change experienced may be abrupt (*concept shift*) or gradual (*concept drift*) [Stanley, 2003, Widmer and Kubat, 1996]. Both have deteriorating effects on the performance of a classifier designed and trained under the assumption that the customer profiles are fixed. Such dynamic customer profiles require robust adaptive classifier models in order for their classification to remain accurate over time.

Formally, given a continuous stream of data instances $x_1, x_2, \ldots$ with each instance being an m-dimensional vector in some pre-defined vector space $\aleph = \Re^m$, the problem of concept drift detection and analysis involves, at every time point p, splitting the instances in a set $\underline{X}$ of $\underline{n}$ recent instances and a set $\overline{X}$ containing the $\overline{n}$ instances that appeared prior to those in $\underline{X}$. The goal is to determine whether or not the instances in $\overline{X}$ were generated by the same distribution as the ones in $\underline{X}$. The standard tools for drift detection are methods from statistical decision theory [Nisbet et al., 2009]. These methods usually compute a statistic from the available data, which is sensitive to changes between the two sets of instances. The measured values of the statistic are then compared to the expected value, under the null hypothesis that both samples are from the same distribution. The resulting p-value can be seen as a measure of the strength of the drift. A good statistic must be sensitive to data properties that are likely to change by a large margin between samples from differing distributions. This means that it is not enough to look at means or variance-based measures, because distributions can differ significantly, even though mean or variance remain in the same range [Dries and Rückert, 2009].

Rank-based measures such as the Mann-Whitney or the Wald-Wolfowitz statistics are usually used instead. However, these statistics depend on a fixed set of characteristics

(e.g. mean and variance) of the underlying distribution [Lehmann and D'Abrera, 2006]. Thus, they work well in scenarios where the change in the underlying distribution affects the properties measured by the statistic, but they perform below par, if the drift influences the characteristics caught by the test statistic only to a small degree [Dries and Rückert, 2009].

This is usually the case for transactional data used in building customer profiles, where the inherent skewness and sparsity of the transactional data make the problem of detecting concept drift non-trivial for dynamic customer profiles [Hahsler, 2006, Hsu et al., 2004, Xiong et al., 2003].

Methods for detecting concept drift in dynamic customer profiles therefore need to be independent of the underlying distribution of the transactional data used in constructing the customer profiles.

Furthermore, the change in an individual customer's profile classification might not necessarily be due to the change in the distribution of the target variable representing the customers with similar profiles, but a temporary or permanent change in the individual customer's behaviour. That is, for example, the change in the number/type of items bought by an individual Electrician, to that bought by an individual Plumb-Heater may not necessarily be due to the entire set of Electricians' change in buying behaviour.

In such a scenario, the "misclassified" output by the classifier may not necessarily be "wrong" and the customer profile will need to be relabelled to reflect the customer's new buying behaviour.

6.2 Adaptive Customer Profile Classification

The current approaches for classifying customer profiles over time require the entire dataset to obtain the desired models that adequately represent the patterns inherent in the database. However, the occurrence of changes in the business environment which are often reflected in the training database as data operations such as: *additions* (customers making more orders of an item), *deletions* (customers returning an item within a 28 days stipulated period), *edits* (customers swapping items), etc.; often require that the training database be rescanned and retrained so the generated models reflect the changes done. Such business driven database changes, as highlighted in Chapter 2, make the costs of the required rescan each time the training dataset is modified prohibitive.

Algorithms for the adaptive learning that involves keeping of descriptive parameters of past mining results and operating only on data records that have been updated, are a more efficient approach which can lead to substantial savings in tracking and keeping in line with the changing customer behaviour over time.

6.2.1 Overview of Adaptive Systems

Much of the work for adaptive mining has been applied mainly to data streams where the developed algorithms tend to incorporate one or more of the elements shown in Table 6.1 to address the accompanying problems.

Table 6.1: Challenges and Methods for Adaptive Systems

Challenges	Methods
What needs be to remembered or forgotten?	Methods for determining which data contribute to the learning process.
When should the model be upgraded?	Methods for change detection.
How should the model be upgraded?	Methods for monitoring and updating estimations for some statistics of the input.

The algorithms are usually integrated into adaptive systems consisting of three modules as shown in Figure 6.1 (adapted from [Bifet and Gavaldà, 2009, Schon et al., 2006]).

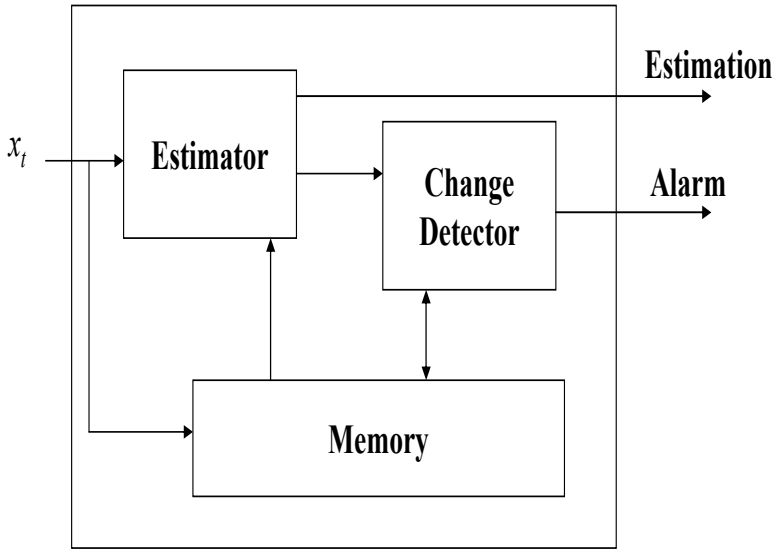


Figure 6.1: Framework of a typical adaptive system

Generally, the input to these algorithms is a sequence $x_1, x_2, \ldots, x_t, \ldots$ of data items whose distribution varies over time in an unknown way. The outputs at each time step are:

- an estimation of some important parameters of the input distribution, and
- a signal alarm indicating that distribution change has recently occurred.

In the data stream setting, all the x_t are real values. The desired estimation in the sequence of x_i is usually the expected value of the current x_t, and less often another

distribution statistics such as the variance. The only assumption on the distribution is that each x_t is drawn independently from each other. Thus, they are concerned with one-dimensional items. While the data streams often consist of structured items, most adaptive mining algorithms are not interested in the items themselves, but in a collection of real-valued (sufficient) statistics derived from the items; thus the input data stream is taken as decomposed into possibly many concurrent data streams of real values, which will be combined by the adaptive mining algorithm.

Memory is the component where the algorithm stores the sample data or summary that is considered relevant at current time, that is, for example, its description of the current data distribution.

The Estimator component is an algorithm that estimates the desired statistics on the input data, which may change over time. The algorithm may or may not use the data contained in the Memory. The simplest Estimator algorithm for the expected output value is the linear estimator, which simply returns the average of the data items contained in the Memory. Other examples of efficient estimators are Auto-Regressive, Auto Regressive Moving Average, and Kalman filters.

The change detector component outputs an alarm signal when it detects change in the input data distribution. It uses the output of the Estimator, and may or may not in addition use the contents of Memory.

The output from the estimator and change detector components are used in evaluating the models of the adaptive system at each time step. The methods used for evaluating the adaptive learning model in a stream context include [Gama et al., 2009]:

- Holdout which uses an independent test set [Han and Kamber, 2006]. It applies the current decision model to the test set at regular time intervals (or set of examples). The loss estimated in the holdout is an unbiased estimator.

- Predictive Sequential method [Dawid, 2008] which involves computing the error of a model from the sequence of examples. For each example in the stream, the actual model makes a prediction based only on the example attribute-values. The prequential-error is computed based on an accumulated sum of a loss function between the prediction and observed values:

$$S = \sum_{i=1}^{n} L\left(y_i, \hat{y}_i\right)$$

It is to be noted that, in the prequential framework, we do not need to know the true value y_i for all points in the stream. The framework can be also used in situations of limited feedback by computing the loss function and S_i for points where y_i is known.

The mean loss is given by: $M = \frac{1}{n} \times S$. For any loss function, we can estimate a

confidence interval for the probability of error, $M \pm \epsilon$, using Chernoff bound [Castillo and Gama, 2005]:

$$\epsilon_c = \sqrt{\frac{3 \times \bar{\mu}}{n} ln\,(2/\delta)}$$

where δ is a user defined confidence level. In the case of bounded loss functions, like the 0-1 loss, the Hoeffding bound [Hoeffding, 1963] can be used:

$$\epsilon_h = \sqrt{\frac{R^2 ln\,(1/\delta)}{2N}}$$

where R is the range of the random variable X (for a probability, R is one, and for an information gain, it is log c, where c is the number of classes).

Both bounds use the sum of independent random variables and give a relative or absolute approximation of the deviation of the random variable from its expectation. They are independent of the distribution of the random variable.

Langford [2005] provides an extensive review of different types of distribution free bounds that are prevalent in Machine Learning and categorizes each bound into one of two categories: (1) test set bound and (2) training set bound.

The test set bound bounds the error over the test set by considering that the error has a binomial distribution. A good approximation to this bound is the Chernoff bound [Chernoff, 1952] and the related Hoeffding bound [Hoeffding, 1963].

A training set bound, on the other hand, bounds the training error to the generalization error. Well known examples of this type of bound are the Vapnik-Chervonenkis bounds (VC bounds) [Vapnik, 1998], Probably Approximately Correct Bayes bounds (PAC Bayes bounds) [McAllester, 1999], OccamŠs Razor bounds [Blumer et al., 1987], Sample Compression bounds [Floyd and Warmuth, 1993] and Rademacher Complexity bounds [Bartlett et al., 2002].

Langford [2005] infers from the comparison of these two categories that test set bounds are generally much tighter than training set bounds and are a superior tool in reporting error rates.

The independence of the Hoeffding bound from the distribution of the random variable is particularly useful in our setting where the inherent sparsity and skewness of the input transactional data has adverse effect on the performance of standard (i.e. non-adaptive) classifier models.

An alternative approach to model adaptation is the champion/challenger testing strategy. In a nutshell, champion/challenger testing is a systematic, empirical method of comparing the performance of a production model (the champion) against that of new models built on more recent data (the challengers) [Han and Kamber, 2006]. If a challenger model outperforms the champion model, it becomes the new champion and is deployed in the production system. Challenger models are built periodically as new data are made

available [Campos et al., 2005].

6.2.2 Related Work

A number of adaptive algorithms, which mainly relate to association mining, have been proposed in the literature for transactional data. These include the re-running of the association mining algorithms on the updated database, as proposed by Tsai et al. [1999] which has a drawback in not utilizing existing information, such as the previously obtained support count.

The Fast Update Algorithm (FUP) [Cheung et al., 1996] and FUP2 [Cheung et al., 1997] overcome the aforementioned limitation, but they do this by rescanning the original database. Chang et al. [2005] proposed the New Fast Update Method (NFUP) which manages to overcome both limitations but does so under the assumption that the items have same "exhibition" time-line and thus those that fall in the same partition as the previously stored database are identical.

The algorithm proposed in this thesis is analogous to the aforementioned approaches in that they do not adequately address the problem considered here, namely, how to accurately learn the buying behaviour of changing customer profiles, based on their transactional data over time without having to rescan the entire database.

6.3 Relabeling as a Solution to the Customer Profile Class Switching Problem

Relabeling algorithms have been used to address the problem of label switching in Bayesian analysis using mixture models where the data are thought to belong to one of C classes (or components) but whose individual class memberships are unavailable [Nobile and Fearnside, 2007].

However, while Bayesian analysis using mixture models provide a flexible way to model heterogeneous data, the sensitivity of the posterior distribution to changes in the prior distribution for the parameter limits its applicability in inferring the class of dynamic customer profiles built using transactional data over time.

Alternative relabeling techniques, such as the partitioning of the solution space using pattern filtering [Fürnkranz, 1999, Zhu et al., 2003] and using majority/consensus to isolate stable and noisy instances [Brodley and Friedl, 1999] have been used to relabel instances. However, stable instances are often removed along with noisy instances in the case of the latter, which causes the solution space to be incomplete, with the resultant lost of valuable information; while the arbitrary partitioning of the solution space, by the pattern filtering approach, tends not to take into account the reduction in confidence due to mining subspaces. Thus, the classifiers built from the resulting datasets may

not accurately represent the actual problem and lead to increase in the generalization errors [Mantas et al., 2002].

The relabeling approach used in this thesis is analogous to the aforementioned approaches, in that the misclassified customer profiles in each time window are relabelled independent of the change in the distribution or nature (i.e. noisy or stable) of the customer profiles over time.

6.4 Proposed Classifier Model Adaptation and Relabeling for Customer Profile Classification

The proposed adaptation and relabeling algorithm consists of the three key phases of: pre-processing, inference/evolving data classification and customer profile class adaptation and relabeling.

6.4.1 Pre-processing Training Data Phase

Step 1: Aggregating and Binning data for training purposes

As discussed in Section 4.4, transactional data tends to be sparse and skewed towards the large number of customers who make fewer purchases. This makes distinguishing them for classification purposes difficult.

In order to handle the sparsity problem, the pre-processing phase involves aggregating the verified customer profiles in sliding time windows (3, 6, 9 and 12 month sliding windows were used for the work in this thesis).

Binning process outlined in Section 4.4.1 is then applied to the aggregated transaction to regroup them based on the number of items bought.

6.4.2 Inference/Evolving Data Classification Phase

The pre-processing phase is then followed by an inference/evolving data classification and adaptation/relabeling phase consisting of the following four (4) steps.

Step 2: Aggregating individual customers transactions in different sizes of sliding time windows

In this step, the test datasets of individual customer transactions in each window are aggregated based on the number of items bought.

Step 3: Deriving classifier models from the binned customers from Step 1

During this step, classifier models are derived for each of the discovered bins from Step 1 and used to classify previously unseen test data made of the aggregated number of individual customer items transacted in time window W_i from Step 2.

Step 4: Adaptation of classifier models/relabeling of customer profile

This step is comprised of two alternative components of (1) relabeling and (2) adaptation:

1. The relabeling component compares each test customer profile's classification in the current window W_i against the classification obtained in the previous window $W_{(i-1)}$. If the i-th customer profile classification in the current window W_i is the same as the classification obtained in the previous $W_{(i-1)}$, then the customer profile's classification is left unchanged; otherwise the customer profile is relabelled. The relabeling algorithm assumes that the customer profiles used in training the static classifier models for each of the bins have been verified and established with a third-party to truly represent the class of the customer profiles.

2. The adaptation component computes each classifier model's classification accuracy on the test data. If the performance rate is within the change bound (i.e. error bound in the case of the Hoeffding bound guided change detection and accuracy for the champion/challenger approach), the classifier model is left unchanged; otherwise a new classifier model is built using the most recent (i.e. current window's) training dataset and a new change bound on the test set computed.

Step 5: Combining the class outputs from the different time windows

As discussed in Section 5.2.3, in an evolving transactional data setting, it is common for the features of the datasets to continually change as the customer buying behaviour change with time. Using isolated classifications of individual time windows may give a distorted picture of the customers' true class and an increase in uncertainty especially in cases where the customer's buying behaviour is constantly changing. In other to reduce the uncertainty due to misclassification or to introduce additional flexibility to the system (if one would like to focus on detecting change), this step combines the outputs from the different time windows using a combiner such as majority voting, gating, etc. [Polikar, 2006, Ruta and Gabrys, 2000]. This can result in more certain/stable classification over time as well as enhance the ability to detect changes by focusing on or selecting classifiers that are able to follow changes in behaviour.

Figure 6.2 outlines the proposed architecture for adapting classifier models and relabeling customer profiles.

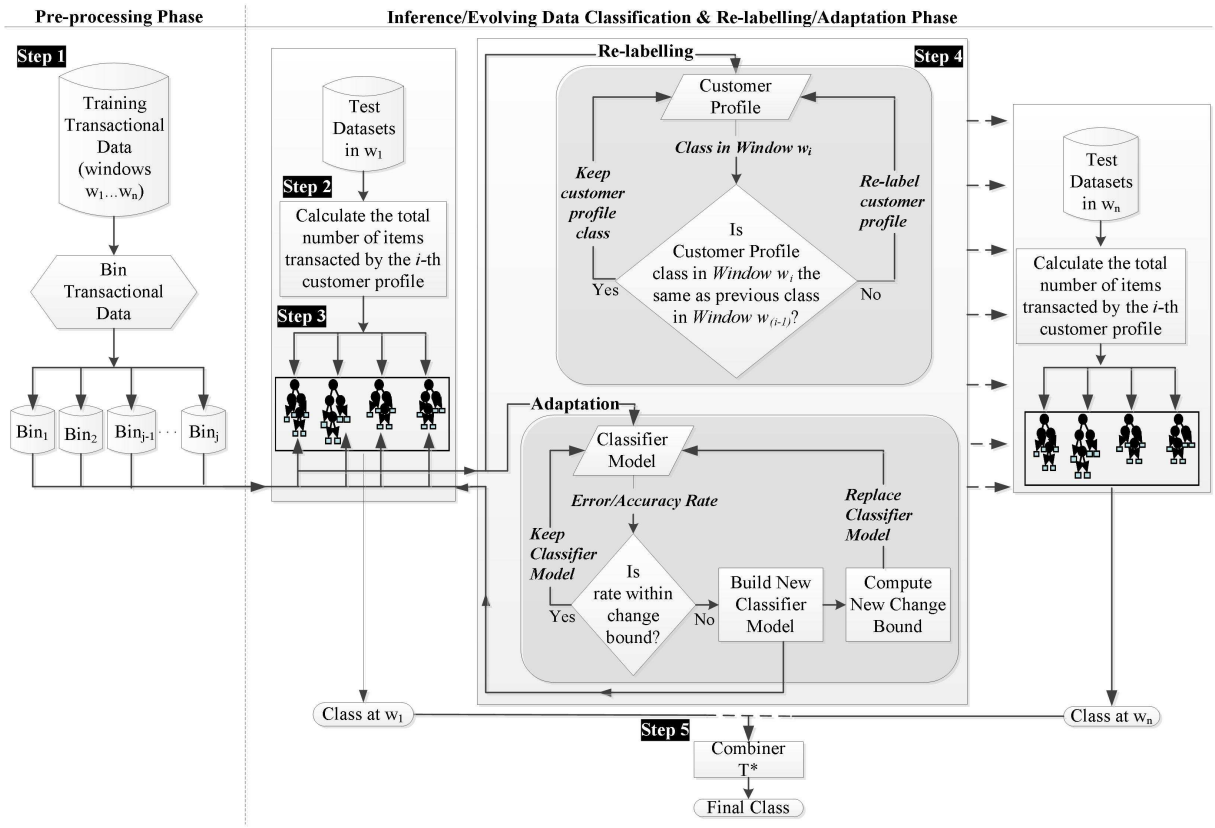


Figure 6.2: Architecture for adapting classifier models and relabeling misclassified customer profiles. The highlighted steps are described in Section 6.4.

6.5 Experiment Setting, Results and Evaluation

The goal of our experiment was to perform a comparative analysis of the performance obtained from relabeling the customer profiles classifications of previously unseen customer transactions over time window against the performance obtained from: (1) static classifier models, (2) adaptive classifier models based on the champion/challenger testing strategy and (3) adaptive classifier models based on the Hoeffding bound change detection strategy.

The customer profiles built, using the transactional data of **Electricians** and **PlumbHeaters** covering a period of 30 months provided by Screwfix, were used for the experiments. The detailed description of the data can be found in Chapter 3 of this thesis.

To perform the experiments, the customers, whose total transactions were identified in Chapter 4 to be in bins 1 to 5 over the 30 months transactions, were used as the training set, while the customers with transactions in bin 6 were used as the test set. The training and test set were regrouped into 3, 6, 9 and 12 months sliding windows, which resulted in 28, 25, 22 and 19 dataset partitions (each with 5 bins) respectively. The resulting dataset partitions were such, that each training dataset was paired with a corresponding test dataset. Figures H.1 and H.2 in Appendix H respectively show the distribution of the top 10 products transacted by the Electricians and PlumbHeaters in the training and

test datasets in the 3, 6, 9 and 12 months sliding windows.

WEKA's [Hall et al., 2009] implementation of: (1) the decision tree algorithm (J48), (2) Naive Bayes, (3) Linear Discriminant Classification and (4) Support Vector Machines (SVM) was then applied to each of the bins in the training dataset partitions. The algorithms were chosen in such a way as to enable the comparative analysis to cover parametric, non-parametric, linear and kernel-based classification methods respectively.

For the relabeling experiments, the customer profiles in each bin of the test windows were classified, using the classifier model derived from the corresponding training bin and window, as outlined in Figure 6.2. The label of the misclassified customer profile at each window were relabelled before the subsequent window classification was performed.

The procedure outlined in Figure 6.2 was used for the adaptive classifier experiments. Each of the bin derived classifier models were evaluated at each time window and a decision to keep it or change it made, based on the accuracy rate (in the case of the challenger/champion strategy) or the error rate (in the case of the Hoeffding bound strategy).

6.5.1 Effect of Window Length on Classification Accuracy

Comparative experiments were undertaken to gauge the effects of the chosen sliding window size on the customer profile classification over time. Table 6.2, as well as Figures I.1, I.2, I.3 and I.4 in Appendix I, show the comparative results obtained from using the four (4) classification approaches on ensembles of Weka's implementation of C4.5 Decision Trees (J48), Naive Bayes, Linear Discriminant Classifiers (LDC) and Support Vector Machines (SVM) for the 3, 6, 9 and 12 month windows respectively. It can be seen, from the tables, that the classification performance of the relabeling and adaptive approaches, tend to improve as the window size increases, with the LDC and SVM ensembles generally performing better on the vector representation of the customer profiles. It can also be seen from the lower accuracy rates in the 3 and 6 months that discriminating the class of the customer profiles is challenging in the shorter windows (3 Months especially), a reflection of the skewness and sparseness of the customer transactions due to few items bought.

6.5.2 Stability of Customer Profile Classification Over Time

Observing the stability in the classification of a customer profile over time, provides insight into the customer's buying behaviour over time. To comparatively assess the stability of the customer profile classification, Equation 5.5 from Chapter 5 was used to compute the stability of the customer profile classification over time.

It can be seen from the comparative customer profile classification stability values in Table 6.3, as well as Figures J.1, J.2, J.3 and J.4 in Appendix I, for all the 4 time window sizes studied, that relabeling the "misclassified" individual customers profile classifications

Table 6.2: Comparative Classification Performance of 4 Classifiers in 4 Sliding Time Window Sizes

(a) Three (3) Months Sliding Window

	Champion/Challenger Adaptation	Hoeffding Bound Adaptation	No Adaptation	Relabeling
C4.5	0.77	0.73	0.73	0.82
Naive Bayes	0.81	0.73	0.74	0.79
LDC	0.82	0.76	0.76	0.83
SVM	0.81	0.77	0.75	0.82

(b) Six (6) Months Sliding Window

	Champion/Challenger Adaptation	Hoeffding Bound Adaptation	No Adaptation	Relabeling
C4.5	0.79	0.68	0.68	0.83
Naive Bayes	0.81	0.75	0.75	0.85
LDC	0.84	0.82	0.82	0.92
SVM	0.83	0.79	0.79	0.90

(c) Nine (9) Months Sliding Window

	Champion/Challenger Adaptation	Hoeffding Bound Adaptation	No Adaptation	Relabeling
C4.5	0.79	0.73	0.73	0.89
Naive Bayes	0.81	0.80	0.80	0.90
LDC	0.85	0.85	0.83	0.93
SVM	0.84	0.84	0.82	0.93

(d) Twelve (12) Months Sliding Window

	Champion/Challenger Adaptation	Hoeffding Bound Adaptation	No Adaptation	Relabeling
C4.5	0.77	0.76	0.71	0.89
Naive Bayes	0.82	0.81	0.79	0.92
LDC	0.86	0.85	0.85	0.94
SVM	0.85	0.85	0.83	0.95

(in contrast to adapting the classifier models), leads to more stable (i.e. invariant) customer profile classifications in the longer time window sizes (i.e. 9 and 12 months sliding windows); while adapting the classifiers using the Accuracy Bound Adaptation approach leads to more stable (i.e. invariant) customer profile classifications in the shorter time window sizes (i.e. 3 and 6 months sliding windows).

As was discussed in Section 4.4, the customer profiles tend to be sparse in the shorter time window sizes, due to fewer items bought. This makes it more challenging to distinguish them in the shorter time window sizes. Therefore, relabeling the customer profiles in the shorter time window sizes has little impact in the classification stability, compared to the champion/challenger approach, where the classifiers relearn and adapt to the challenging sparse solution space. In the longer time window sizes, the customers tend to buy more items, making the solution space more stable and thus, the changes in classification are more likely to be due to change in the customer's buying behaviour.

Table 6.3: Comparative Customer Profiles Classifications Stability for the 4 Classifiers in the 4 Sliding Time Window Sizes

(a) Three (3) Months Sliding Window

	Champion/Challenger Adaptation	Hoeffding Bound Adaptation	No Adaptation	Relabeling
C4.5	0.67	0.64	0.66	0.66
Naive Bayes	0.68	0.54	0.56	0.58
LDC	0.73	0.65	0.64	0.68
SVM	0.69	0.64	0.59	0.65

(b) Six (6) Months Sliding Window

	Champion/Challenger Adaptation	Hoeffding Bound Adaptation	No Adaptation	Relabeling
C4.5	0.72	0.57	0.57	0.66
Naive Bayes	0.73	0.61	0.61	0.70
LDC	0.78	0.76	0.76	0.84
SVM	0.77	0.73	0.73	0.81

(c) Nine (9) Months Sliding Window

	Champion/Challenger Adaptation	Hoeffding Bound Adaptation	No Adaptation	Relabeling
C4.5	0.70	0.64	0.67	0.78
Naive Bayes	0.77	0.75	0.71	0.79
LDC	0.81	0.80	0.81	0.86
SVM	0.79	0.79	0.79	0.86

(d) Twelve (12) Months Sliding Window

	Champion/Challenger Adaptation	Hoeffding Bound Adaptation	No Adaptation	Relabeling
C4.5	0.64	0.66	0.66	0.77
Naive Bayes	0.81	0.80	0.77	0.83
LDC	0.83	0.83	0.84	0.88
SVM	0.84	0.84	0.83	0.89

Furthermore, at the individual level, it can be seen from the tables in Table 6.4, as well as the tables in Tables K.1, K.2, K.3 in Appendix K, illustrating the classification of an Electrician and PlumbHeater, that the changing classification of the customer profiles is closely related with the window size, with the classification stability increasing with the window size as the customers buy more.

Thus, viewing this perspective in tandem with the classification performance in Table 6.2, it can be seen, that the reduction in performance due to the change in class of the customer profiles as detected by the adaptive methods, in the longer time window sizes, may not necessarily be due to the deterioration in the performance of the learnt classifier models (i.e. due to concept drift), but due to the change in the individual customer buying behaviour over time. Relabeling such customer profiles might be the more appropriate thing to do, rather than changing the classifier model and keeping the class unchanged.

6.5.3 Effects of Combiners on the Customer Profile Classification Over Time

As was discussed in Chapter 5, combiners are often used to improve the flexibility of classifiers over time and to monitor customer buying behaviour across different time window sizes. To gauge the effect of combiners on the customer profiles classification stability by the 4 approaches over time, the majority, weighted majority, weighted average majority and minority voting combiners were applied on the individual customer profile classifications. It can be seen from the results obtained in the Table 6.4 for the Linear Regression Classifier and in Table 6.5, as well as the stability plots in Figures L.1, L.2, L.3,and L.4 in Appendix L, that using the combiners with the adaptive and relabeling approaches leads to more stable customer profile classifications over time with the best stability being obtained from the weighted average majority combiner.

6.6 Conclusion

This Chapter has investigated and presented a comparative analysis of 4 approaches for customer profile model management over time. A relabeling approach, which relabelled "misclassified" customer profiles, was presented and analysed alongside a static and two adaptive classification approaches. The results obtained on an individual and group customer profile classification level showed that the relabeling approach led to more stable and robust customer profiles over time.

The finding could be useful to businesses which need to identify their different customer types and their buying behaviour, using transactional data within a short time frame.

In terms of future research work, a lot of work for developing adaptive techniques has been directed at monitoring concept drift and improving classifier performance [Dries and

	Electrician (E)								PlumbHeater (P)							
Sliding Window Dataset Partitions	Class in 3 Months Window	Class in 6 Months Window	Class in 9 Months Window	Class in 12 months Window	Majority Vote Across Windows	Weighted Majority Vote Across Windows	Weighted Average Across Windows	Minority Vote Across Windows	Class in 3 Months Window	Class in 6 Months Window	Class in 9 Months Window	Class in 12 months Window	Majority Vote Across Windows	Weighted Majority Vote Across Windows	Weighted Average Across Windows	Minority Vote Across Windows
1	E				E	E	E	-	P				P	P	P	-
2	E				E	E	E	-	P				P	P	P	-
3	E				E	E	E	-	P				P	P	P	-
4	E	E			E	E	E	-	P	P			P	P	P	-
5	P	E			E	E	P	E	P	P			P	P	P	-
6	E	E			E	E	E	-	P	P			P	P	P	-
7	E	E	E		E	E	E	-	E	P	P		P	P	P	E
8	E	E	-		E	E	E	-	P	P	P		P	P	P	-
9	E	E	E		E	E	E	-	P	P	E		P	P	P	E
10	E	E	E	-	E	E	E	-	P	P	P	P	P	P	P	-
11	E	E	E	-	E	E	E	-	P	P	P	P	P	P	P	-
12	E	E	E	-	E	E	E	-	P	P	P	P	P	P	P	-
13	E	E	E	-	E	E	E	-	P	P	P	P	P	P	P	-
14	E	E	E	-	E	E	E	-	P	P	P	P	P	P	P	-
15	E	E	E	-	E	E	E	-	P	P	P	P	P	P	P	-
16	E	E	E	-	E	E	E	-	P	P	P	P	P	P	P	-
17	E	E	E	-	E	E	E	-	P	P	P	P	P	P	P	-
18	E	E	E	-	E	E	E	-	P	P	P	P	P	P	P	-
19	E	E	E	-	E	E	E	-	P	P	P	P	P	P	P	-
20	E	E	E	E	E	E	E	-	P	P	P	P	P	P	P	-
21	E	E	E	E	E	E	E	-	P	P	P	P	P	P	P	-
22	E	E	E	E	E	E	E	-	P	P	P	P	P	P	P	-
23	E	E	E	E	E	E	E	-	P	P	P	P	P	P	P	-
24	E	E	E	E	E	E	E	-	P	P	P	P	P	P	P	-
25	E	E	E	E	E	E	E	-	P	P	P	P	P	P	P	-
26	E	E	E	E	E	E	E	-	P	P	P	P	P	P	P	-
27	E	E	E	E	E	E	E	-	P	P	P	P	P	P	P	-
28	E	E	E	E	E	E	E	-	P	P	P	P	P	P	P	-
Majority Vote in Windows	E	E	E	E	E				P	P	P	P	P			
Weighted Majority Vote in Windows	E	E	E	E		E			P	P	P	P		P		
Weighted Average in Windows	E	E	E	E			E		P	P	P	P			P	
Minority in Windows	P	-	-	-				-	E	-	E	-				-
Stability	0.93	1	1	1	1	1	0.93	1	0.93	1	0.91	1	1	1	1	-1

(a) Adaptive Linear Regression Classifiers

	Electrician (E)								PlumbHeater (P)							
Sliding Window Dataset Partitions	Class in 3 Months Window	Class in 6 Months Window	Class in 9 Months Window	Class in 12 months Window	Majority Vote Across Windows	Weighted Majority Vote Across Windows	Weighted Average Across Windows	Minority Vote Across Windows	Class in 3 Months Window	Class in 6 Months Window	Class in 9 Months Window	Class in 12 months Window	Majority Vote Across Windows	Weighted Majority Vote Across Windows	Weighted Average Across Windows	Minority Vote Across Windows
1	E				E	E	E	-	P				P	P	P	-
2	E				E	E	E	-	P				P	P	P	-
3	E				E	E	E	-	P				P	P	P	-
4	E	E			E	E	E	-	P	P			P	P	P	-
5	E	E			E	E	E	-	P	P			P	P	P	-
6	E	P*			E	P	P	E	P	P			P	P	P	-
7	E	P	E		E	E	P	P	P	P	P		P	P	P	-
8	E	E*	-		E	E	E	-	P	P	P		P	P	P	-
9	E	E	E		E	E	E	-	P	P	P		P	P	P	-
10	E	E	E	-	E	E	E	-	P	P	P	P	P	P	P	-
11	E	E	E	-	E	E	E	-	P	P	P	P	P	P	P	-
12	P*	E	E	-	E	E	P	P	P	P	P	P	P	P	P	-
13	P	E	E	-	E	E	P	P	P	P	P	P	P	P	P	-
14	E*	E	E	-	E	E	E	-	P	P	P	P	P	P	P	-
15	E	E	E	-	E	E	E	-	P	P	P	P	P	P	P	-
16	P*	E	E	-	E	E	P	P	P	P	P	P	P	P	P	-
17	E*	E	E	-	E	E	E	-	P	P	P	P	P	P	P	-
18	E	E	E	-	E	E	E	-	P	P	P	P	P	P	P	-
19	E	E	E	-	E	E	E	-	P	P	P	P	P	P	P	-
20	E	E	E	E	E	E	E	-	P	P	P	P	P	P	P	-
21	E	E	E	E	E	E	E	-	P	P	P	P	P	P	P	-
22	E	E	E	E	E	E	E	-	P	P	P	P	P	P	P	-
23	E	E	E	E	E	E	E	-	P	P	P	P	P	P	P	-
24	E	E	E	E	E	E	E	-	P	P	P	P	P	P	P	-
25	P*	E	E	E	E	E	P	P	P	P	P	P	P	P	P	-
26	P	E	E	E	E	E	P	P	P	P	P	P	P	P	P	-
27	P	E	E	E	E	E	P	P	P	P	P	P	P	P	P	-
28	P	E	E	E	E	E	P	P	P	P	P	P	P	P	P	-
Majority Vote in Windows	E	E	E	E	E				P	P	P	P	P			
Weighted Majority Vote in Windows	E	E	E	E		E			P	P	P	P		P		
Weighted Average in Windows	P	P	P	P			P		P	P	P	P			P	
Minority in Windows	P	P	-	-				P	-	-	-	-				-
Stability	0.64	0.84	1	1	1	0.93	0.36	-0.78	1	1	1	1	1	1	1	-

(b) Static Linear Regression Classifiers With Misclassified Customer Profiles Relabelled (* Indicates point of relabeling)

Table 6.4: Tables illustrating the comparative stability of an Electrician (E1) and a Plumb-Heater(P1) customer profiles classifications by the Linear Regression Ensemble over time

Table 6.5: Comparative Customer Profiles Classifications Stability for Four (4) Combiners

(a) Majority Voting

	Champion/Challenger Adaptation	Hoeffding Bound Adaptation	No Adaptation	Relabeling
C4.5	0.74	0.67	0.65	0.65
Naive Bayes	0.73	0.65	0.64	0.64
LDC	0.78	0.76	0.76	0.76
SVM	0.78	0.75	0.73	0.73

(b) Weighted Majority Voting

	Champion/Challenger Adaptation	Hoeffding Bound Adaptation	No Adaptation	Relabeling
C4.5	0.74	0.67	0.64	0.64
Naive Bayes	0.74	0.67	0.66	0.66
LDC	0.79	0.77	0.77	0.77
SVM	0.78	0.76	0.75	0.75

(c) Weighted Average Voting

	Champion/Challenger Adaptation	Hoeffding Bound Adaptation	No Adaptation	Relabeling
C4.5	0.84	0.86	0.86	0.86
Naive Bayes	0.74	0.70	0.72	0.72
LDC	0.80	0.77	0.78	0.78
SVM	0.79	0.78	0.75	0.75

(d) Minority Voting

	Champion/Challenger Adaptation	Hoeffding Bound Adaptation	No Adaptation	Relabeling
C4.5	0.59	0.38	0.40	0.48
Naive Bayes	0.47	0.48	0.48	0.51
LDC	0.36	0.60	0.61	0.51
SVM	0.39	0.61	0.58	0.53

Rückert, 2009, Kelly et al., 1999, Minku et al., 2010, Žliobaitė, 2009]. However, the work in this chapter has shown that the change in the target variable the classifier model is attempting to infer, might not necessarily be due to concept drift, which will need the model being adapted or changed, but might be due to a temporal change, independent of the distribution of the target variable.

Thus, in the context of customer profile classification using transactional data, deciding whether to adapt (or change) the model or to relabel the customer profiles is a non-trivial and significant decision which requires further research.

For instance, the work in this chapter used a simple relabeling scheme based on the assumption that static classifier models initially built, (i.e. in the first time window), adequately represent the solution space of the target variables (i.e. Electricians and Plumb-Heaters). Future work needs to be directed at developing relabeling schemes, where the assumption that observed change might not necessarily be due to change in the distribution target variable, does not hold true, i.e., techniques for relabeling under concept drift. One approach, currently being investigated, involves attaching a decay constant, which is a function of the number of individual target variables in the training set, to each the customer profiles. That is a customer profile is only relabelled if its rate of change is lower than the decay rate of its last held target value.

Furthermore, the work here involved monitoring and identifying dynamic customer profile classification for a two-class decision parameter. Future work will be directed at developing relabeling schemes for multi-class settings.

Chapter 7

Thesis Summary, Conclusion and Proposed Direction for Future Work.

7.1 Summary and Conclusion

This thesis investigated the problem of mining real-world transactional data. In particular, the work undertaken was centred around the performance of data mining algorithms for the exploration/description and inference of retail transactional data.

Data mining algorithms and techniques were investigated with the main aim of addressing the research challenges of:

1. handling *uncertainty* arising from mislabelled profiles based on transactional data,
2. handling *scalability* and efficiently processing large transactional data in order to improve the inference process, and
3. *adapting* to the dynamic nature of transactional data representing continuously changing environment.

The inherent skewness and sparseness of transactional data, which make exploratory and predictive data mining challenging, were highlighted in Chapter 2, together with the research challenges of scalability, uncertainty and adaptation.

The challenge of transactional data exploration and modelling were further explored in Chapter 3, where techniques for describing and pre-processing transactional data were highlighted, together with their appropriateness for mining transactional data. In particular,the need for the use of third-party varied labels, as a way of handling the discrepancy due to non-disclosure and mislabelling, was highlighted in Section 3.1; while the effect of conventional sampling on the performance of classifiers on inherently sparse transactional data was empirically shown in Section 3.3.

Chapter 4 presented investigations on the minimum number of items required to accurately classify a customer profile. It commenced with a description of customer profiles

and methods for their construction in Section 4.1. An approach for handling the problem of customer profile classification scalability, accuracy and uncertainty using transactional data was then presented. The proposed approach combines binning and 10-fold classification to discover 'critical point' at which classifiers can more accurately classify a customer profile based on transactional data. In particular, the proposed approach involved developing a classifier for accurately classifying customer profiles based on transactional data using the following four stage approach:

1. Binning the transactional dataset based on the number of items bought,
2. Bin decomposition and prototype selection using K-means algorithm,
3. Generating classifier models for each bin, and
4. Using the generated classifier models to classify future input data.

The key advantage of this heuristic is that it enables the aggregation and grouping of like for like customer profiles based on the number of items bought and so enables the building of classifier models that would more accurately predict future instances on the basis of their transactions. Experiments, using Screwfix's transactional data, showed the existence of a critical point, based on the number of items bought, for which the ROC performance of classifiers improved on a two-class customer profile classification problem as discussed in Section 4.5. An approach for solving the multi-class classification version of the problem, in the context of transactional data, was also presented. Using the real-world multi-class transactional data provided by SLIGRO, an approach which involves binning, prototype selection and 10-fold multi-class classification, was investigated in Section 4.6. Experiments showed that the proposed approach can be used to effectively assign customer profiles, based on the number of items bought to categories.

Chapter 5 presented the work undertaken to design a novel change mining mechanism that detects and visualizes the change in the classification of customer profiles based on transactional data over time. An approach, which uses the concept of change mining to monitor, detect and visualize the changing classification of customer profiles, as returned by a classifier model over time, was proposed. The proposed algorithm encompasses the following steps:

1. Binning of data for training purposes
2. Training individual classifiers on respective bins
3. Representing the transactional data for the customers in different sizes of moving time windows
4. Directing the aggregated transaction data in each of the different time windows to appropriate classifiers.

5. Combining the outputs from the different time windows.

Experiments were then undertaken, in which the classification performance of a two-class decision tree ensembles, built using the data binning process based on the number of items purchased, was monitored over a varying 3,6,9 and 12 months time windows. The changing class values of the customer profiles were analysed and described. The proposed approach to change mining provides decision support, by helping to identify, for example, customers who changed their behaviour, who may have been assigned a wrong label in the verification process or who only temporarily changed their behaviour.

Chapter 6 then presented the results obtained from investigating methods for tracking customer buying behaviour and more confidently classifying customer profiles over time, with the goal of quickly identifying when customers change their buying behaviour and move from one profile class to another. Much of the work in the literature has focused on improving the classification performance of evolving data by building adaptive mechanisms for detecting concept drift. However, the deterioration in performance may not be due to concept drift but due to mislabelling or a temporal change in an individual customer's buying behaviour. This chapter highlights and introduces the directions for future work in addressing this important research problem by presenting a comparative analysis on deciding whether to adapt (or change) the classifier model or incorporate a new labelling scheme. Two adaptation methods were comparatively investigated, alongside a proposed relabeling approach, within the context of classifying customer profiles based on transactional data over time. The experimental results obtained from 4 classifiers, as analysed and discussed in Section 6.5, showed that the proposed approach for relabeling misclassified customer profiles between time windows, leads to more accurate and stable classification of customer profiles in the longer time window sizes; while the adaptive approaches are better suited for the shorter time window sizes.

The remaining part of this chapter now presents directions for the use of relational data mining algorithms as an alternative approach to the work undertaken in this thesis.

7.2 Suggestions for Future Research

7.2.1 Relational Data Mining as an alternative to Attribute-Value Data Mining Techniques

Relational Data Mining (RDM) is the multi-disciplinary field, dealing with knowledge discovery from relational databases consisting of multiple tables. To emphasize the contrast to typical data mining approaches that look for patterns in a single relation of a database, the name Multi-Relational Data Mining (MRDM) is often used as well [Dzeroski and Lavrac, 2001]. Mining data which consists of complex/structured objects also falls within

the scope of this field. The field aims at integrating results from existing fields such as inductive logic programming, KDD, data mining, machine learning and relational databases; producing new techniques for mining multi-relational data; and practical applications of such techniques.

The fundamental difference between the attribute-value based data mining approach used in this thesis and relational data mining techniques lies in the way they represent knowledge[1].

Many attribute-value based data mining algorithms, employ attribute-value representations, (which are essentially propositional logic), to identify subgroups [Han and Kamber, 2006].

In this representation, the central database of examples is given as a single relation table/flat-file with rows (or records) corresponding to data instances, and columns corresponding to attributes with no relationships between individual instances. One attribute is designated as the class attribute, and the learning task is to construct a learner that can predict the value of the class attribute from the values of the other attributes.

Formally, in the attribute-value representation, an attribute-value signature or *AV-signature* is a finite set of attributes $\{A_1, \ldots, A_n\}$. An attribute-value literal or *AV-literal* is an expression of the form $A_i = a_{ij}$, where A_i is an attribute and a_{ij} is an associated value. An attribute-value conjunction *AV-conjunction* is a conjunction of AV-literals, such that each attribute occurs at most once; the AV-conjunction is *complete* if each attribute occurs exactly once.

The attribute-value based data mining algorithms, which use attribute-value representation, extensively search for interesting subgroups within the central database of examples, using a pattern language of choice, that defines a search space of patterns as the starting point for the search process.

In order to traverse this space in a sensible, guided and efficient manner, the attribute-value based data mining algorithms require a means of judging the interestingness of a given pattern (and corresponding subgroup). In general terms, such a means of searching the space of patterns is referred to as a score function. Typically, a score function considers the database and acquires statistics about the pattern at hand, which in turn produces a score. This score helps the algorithm to make informed decisions about the progress and direction of the search [Flach, 1999].

Most of the attribute-value based data algorithms use *a priori* information about the kind of data instances that are known to exist in the database, as well as statistical

[1]The goal of Knowledge Representation is to represent knowledge in a manner that facilitates inferencing (i.e. drawing conclusions) from knowledge. It involves analysing how to formally represent patterns - i.e. how to use a symbol system to represent a domain of discourse, along with functions that allow inference (formalized reasoning) about the objects [Brachman and Levesque, 2004]. Thus, the representation of the domain of discourse determines the inference method employed for a knowledge discovery problem and an understanding of the output from the inference method is fundamental to understanding the inference method.

information from the database, to guide the search.

The simple tabular structure, that forms the basis of the attribute-value based data mining approach, has been a key reason for its popularity, but at the same time its weakness in terms of expressiveness and dealing with commercial/industrial problems beyond the scope of attribute-value learning [Maimon and Rokach, 2005]. In many large commercial databases, data elements exist as structured data instances, that consist of parts that may be connected in a variety of ways, rather than as vectors of attribute-value data. This makes such commercial databases too complex to analyze with an attribute-value based data mining algorithm without losing important information, adversely impacting on scalability and resulting in the quality of results from the data mining process.

Relational data mining algorithms overcome the limitations of propositional data mining, by using a first-order logic representation [Knobbe, 2005, Wrobel, 2000].

The nature of the first order representation used by relational data mining algorithms tend to differ, depending on whether the relational data mining algorithm is a new relational learning algorithm, systematically developed by creating a single table from a multi-relational database [Kramer et al., 2000, Krogel et al., 2003] or a first-order upgrade of propositional learning algorithms [Laer and Raedt, 2000, Raedt, 1997].

As an alternative to the work undertaken in this thesis, future research will involve investigating the first of the two aforementioned approaches to relational data mining; with the aim of providing scenarios where they can be used as an alternative to -or in combination with- data mining techniques, to address the research problems of adaptation, scalability and uncertainty when mining transactional databases.

In particular, future research work should cover:

1. The application of the relational data mining approach to clustering transactional data;

2. Sampling and search strategies for handling scalability and reducing uncertainty;

3. Adaptive mechanism(s) for relational data mining;

4. The application and down-streaming of robust market segmentation and basket analysis models to non-experts.

The following Sections further outlines the proposed future research work.

7.2.2 Proposed Application of Relational Data Mining Approach to Clustering Transactional Data

In Chapter 3, this thesis highlighted the uncertainty that comes from clustering for explorative purposes using attribute-value based data mining techniques.

The expressiveness of relational data mining techniques have been exploited successfully, in addressing the incomprehensibility of non-predictive or descriptive data mining tasks in the areas of molecular biology (including drug design, protein structure prediction, and functional genomics), environmental sciences, traffic control, and natural language processing. An overview of the aforementioned applications have been provided by Džroski [2000]. However, very little work has been done to use the background knowledge provided by relational data mining techniques, to more adequately explore and describe transactional data.

As part of future research work, we propose an investigation of the application of relational data mining to cluster transactional data. In particular, we suggest the application of relational data mining to cluster and explore the features that best describe transactional data, in terms of identifying typical customer profiles transactions and their product buying behaviour.

Findings from this work could then be analytically compared with the knowledge gleaned from Chapter 3, to assess and present the benefit(s) and drawback(s) in terms of the effect(s) of uncertainty, when clustering transactional data for similar customer profiles transactions and their product buying behaviour.

7.2.3 Proposed Future Research on Sampling and Search Strategies for Handling Scalability and Reducing Uncertainty

This proposed future research work involves investigating feature construction and aggregation processes, with the aim of showing how they impact on the performance of both attribute-value based and relational data mining techniques in terms of handling scalability and reducing uncertainty of mined models.

In Chapter 2, it was discussed that, in order to meet the requirement of attribute-value based data mining algorithms for flat data, time-consuming processes (such as: data manipulation and integration techniques, such as data cleansing, exact matching of identical records, etc.) are often performed, which results in datasets that are too large for many attribute-value based data mining algorithms. Sampling techniques, such as those described by Guha et al. [2000], are often used to reduce the size of the dataset.

However, as shown by the performance of the baseline classifiers in Chapters 4 and 5, the models built, using sampled data, tend to perform poorly on transactional data, due to their sparse nature and the inadequate/indirect attribute-value representation of the relationships between the data instances in the underlying relational data from which the datasets were sampled.

Several relational data mining systems have been developed which employ various search strategies and hypothesis evaluation criteria, in order to cope with intractably large search spaces and to be able to generate high-quality patterns [Knobbe, 2005].

Formally, the relational search process can be stated as:
Given:

- Background knowledge **B**
- Theory Description Language ***T***
- Positive examples P (class +)
- Negative examples N (class -)
- A covering relation **covers(B,T,e)**$\Leftrightarrow$ B $\cup$ T $|-e$

Find: a theory (i.e. a set of rules) that covers

- all positive examples (completeness)
- no negative examples (consistency)

Some of the more efficient of these search strategies for relational data mining include [Blockeel et al., 2003]:

1. Depth-first exploration which has been implemented in Java Expert System Shell (JESS) [Friedman-Hill, 1997] and Foil [Quinlan, 1993] and Tilde [Blockeel and De Raedt, 1998], retains and refines only the current best hypothesis.

2. Beam Search, implemented in Aleph [Srinivasan, 2000] and ML-SMART [Bergadano et al., 1989] avoids the limitations of greedy myopic optimization, that can arise in depth-first search [Russell and Norvig, 2003] by retaining and refining a limited number of the best current hypotheses [Bergadano et al., 1988].

3. Stochastic, population-based exploration of the hypothesis space using evolutionary computation and genetic algorithms (GAs) [Bäck, 1996, Goldberg, 1989]

There is a need to investigate the aforementioned search strategies in a retail data context and compare their performance in terms of the quality of knowledge discovery with those of attribute-value based data mining algorithm on sampled transactional data.

For attribute-value based transactional data mining investigations, the non-conventional sampling techniques proposed in [Church et al., 2006] and [Budka and Gabrys, 2010] can be used; whilst the depth-first and beam search strategies implemented in Aleph and JESS can be investigated for performing the search.

7.2.4 Proposed Research of Adaptive Mechanism(s) for Transactional Data

Chapter 6 of this thesis showed the change in the classification of customer profiles based on their buying patterns over time and highlighted the need for the incorporation of adaptive mechanism(s) to predictive models, so as to enable them to be robust to the change of different data characteristics (e.g., underlying distribution, skewness, mean, etc.).

Often the cause of change is hidden, not known *a priori*, or may be due to the patterns of customers' buying preferences, that may change with time, depending on the current season, availability of alternatives, inflation rate, etc. Either way, these changes make the data mining task more complicated, as they can induce more or less radical changes in the target concept - a phenomenon known as *concept drift* [Widmer and Kubat, 1996]. An effective data mining algorithm should be able to track such changes and quickly adapt to them.

More research work will be required to investigate measures for detecting and handling concept drift in transactional data.

In essence, the research should be aimed at identifying the adaptive mechanism (e.g. incremental learning, monitor/input of appropriate background knowledge, etc.) that best suits transactional data.

7.2.5 Design and implement user-friendly interfaces of advanced modelling software for non-expert users

Businesses, whilst performing their fundamental daily activities, such as managing slim margins and tenuous customer loyalty, as well as deciding where to locate stores, what products to stock, which customers to retain, how to effectively communicate with them and meet their diverse and changing needs; tend to accumulate an incredible amount of data on demographics, product sales based on seasons, transactions, etc.

Business organizations are increasingly using sophisticated modelling and optimization tools that utilize this data, to provide analytical support, which helps to address the business issues and painful points faced by retailers, whilst performing essential retail industry business functions.

These processes tend to be time consuming with models becoming out-of-date with the business by the time they are successfully built.

Future research work can use the knowledge gained from the work in this thesis, to design and build robust adaptive data mining systems, which address key business management problems of market segmentation (usually addressed using clustering or association mining techniques [Berry and Linoff, 2000, 2004]), accurate trade-type identification

(usually addressed with an appropriate classifier [Berry and Linoff, 2000, 2004]) and product bought together (-usually addressed using basket analysis via association mining or clustering [Agrawal and Imielinski, 1993]) without the need to export the data into a flat-file.

Appendices

Appendix A

Figures showing the discrepancies between the trade-type information provided and the verified trade-type.

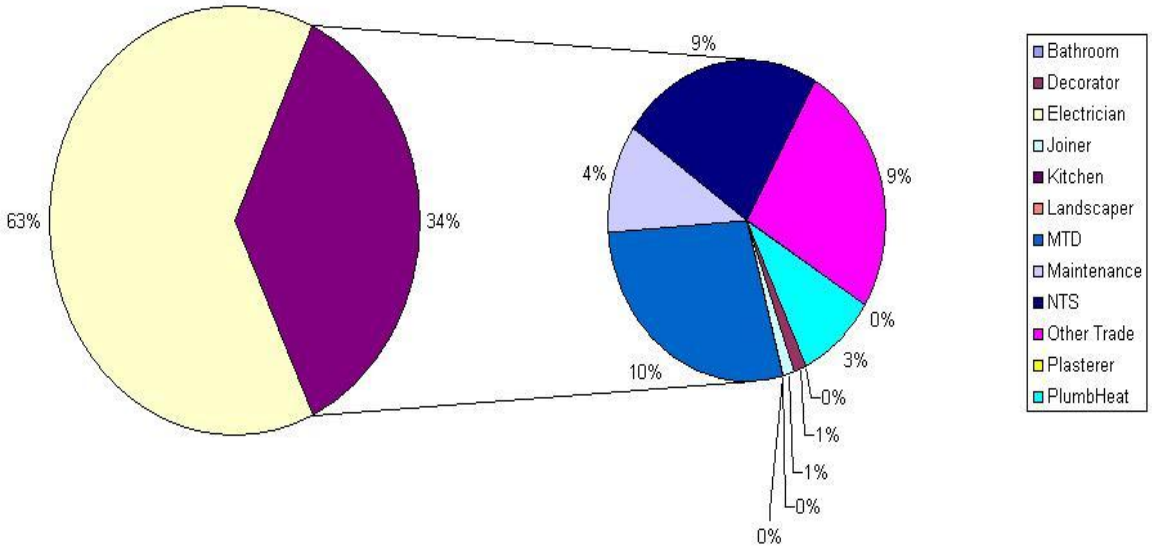


Figure A.1: Pie of Pie Chart showing the discrepancies in the number of Customers who were categorized as "Electrician" and were verified to be "Electrician" (63%); and those made by Customers who were categorized as belonging to one of the other 11 trade-types and were verified to be "Electrician" (34%).

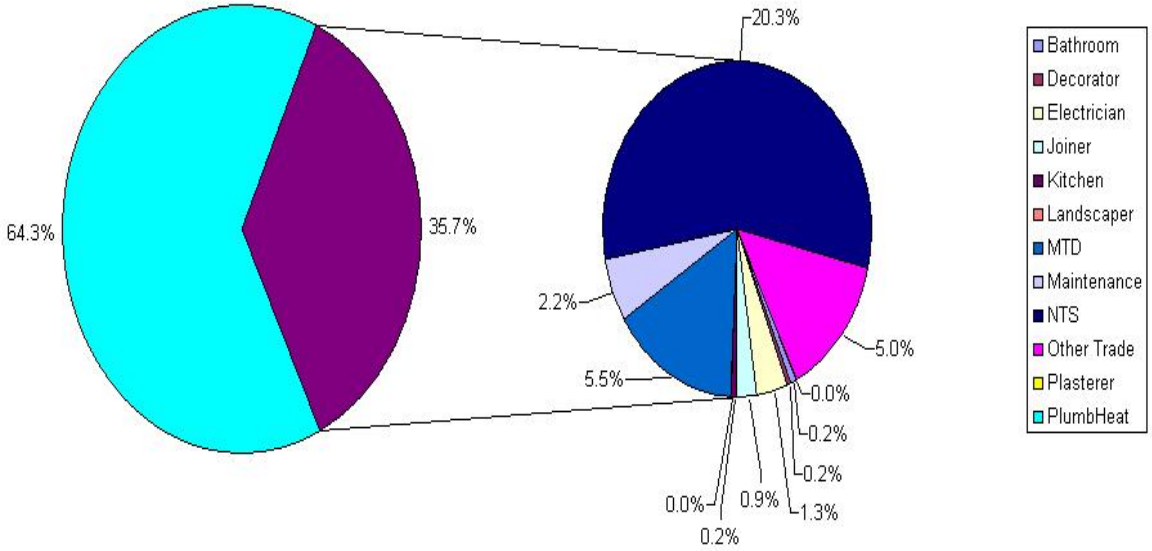


Figure A.2: Pie of Pie Chart showing the discrepancies in the **number of Customers** who were categorized as **PlumbHeat** and were verified to be under **PlumbHeat** (64.3%); those made by Customers who were categorized as belonging to one of the other 11 trade-types and were verified to be **PlumbHeat** (35.7%)

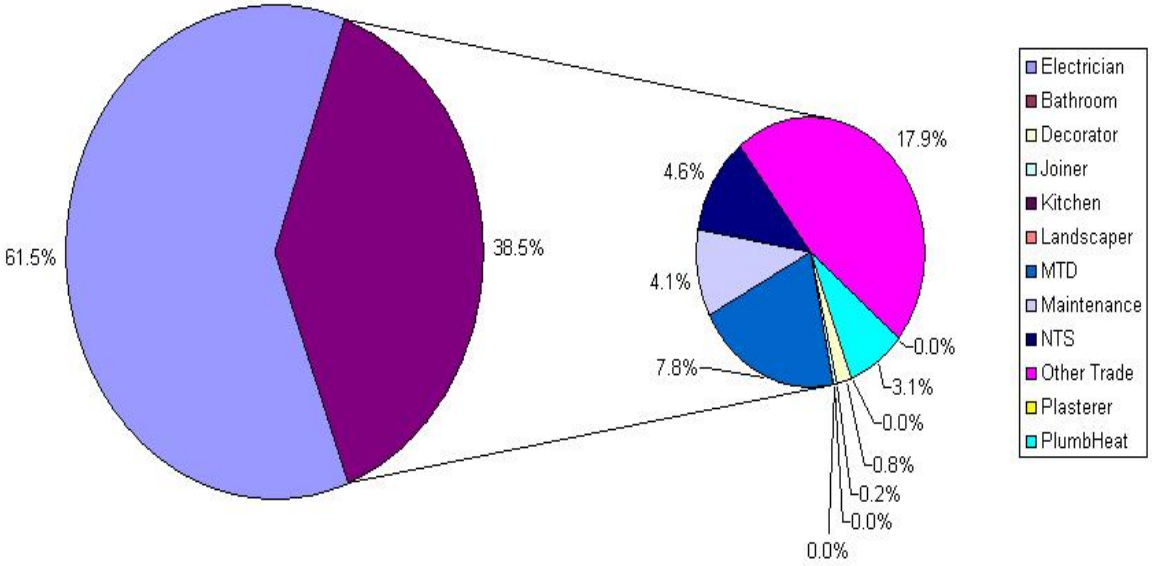


Figure A.3: Pie of Pie Chart showing the discrepancies in the **Orders made by the Customers** who were categorized as "Electrician" and were verified to be "Electrician" (61.5%); and those made by Customers who were categorized as belonging to one of the other 11 trade-types and were verified to be "Electrician" (38.5%).

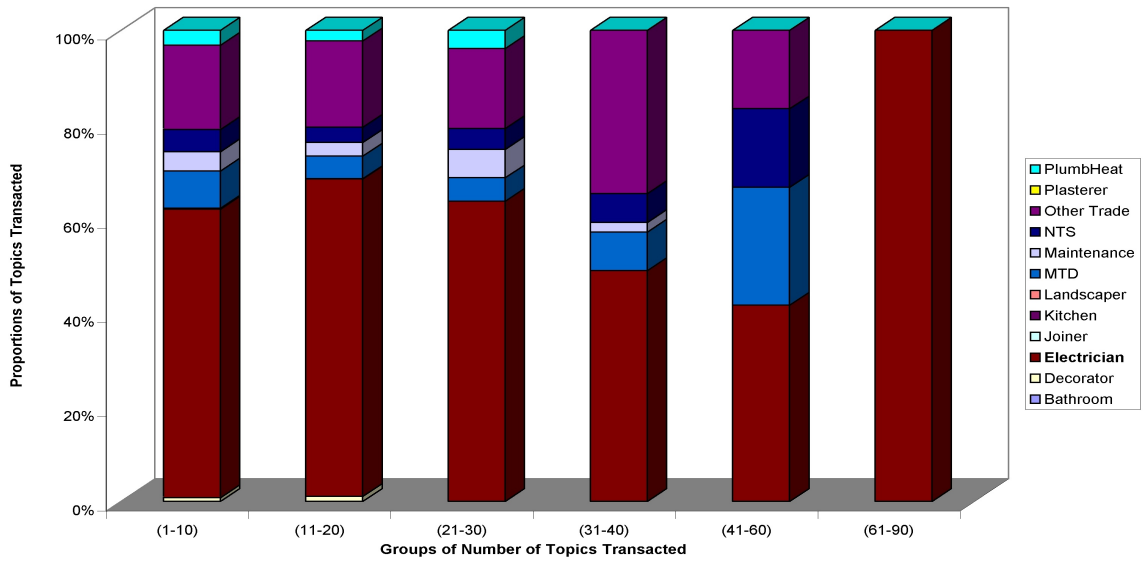


Figure A.4: Plot showing the discrepancies between Customers who were categorized as **Electrician** and were verified to be **Electrician**; and those who were categorized as belonging to one of the other 11 trade-types and were verified to be **Electrician**; segmented in terms of the number of items they bought.

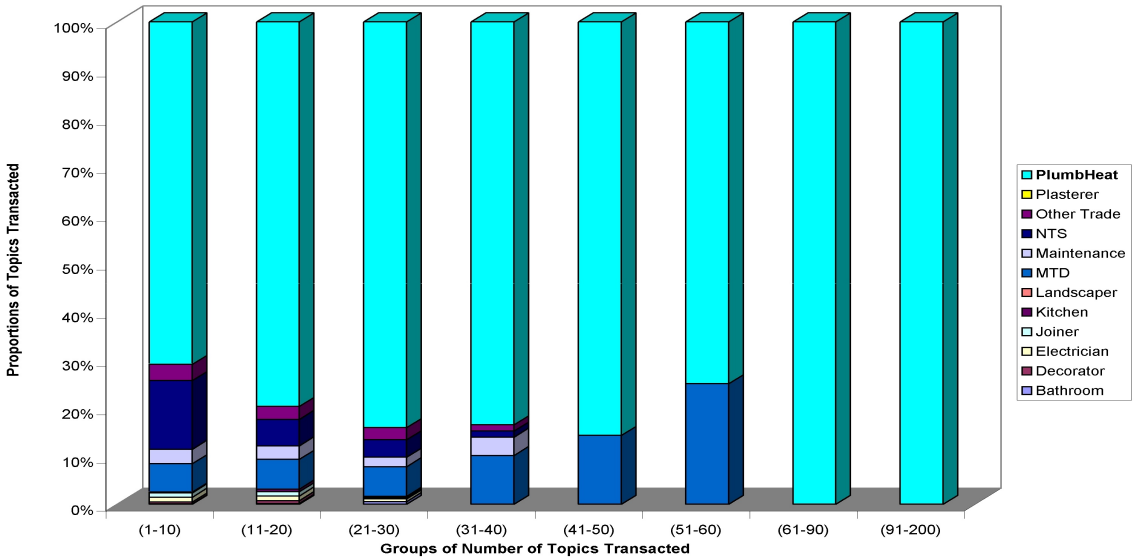


Figure A.5: Plot showing the discrepancies between Customers who were categorized under **PlumbHeat** and were verified to be under **PlumbHeat**; and those who were categorized as belonging to one of the other 11 trade-types and were varified to be **PlumbHeat**; segmented in terms of the number of items they bought.

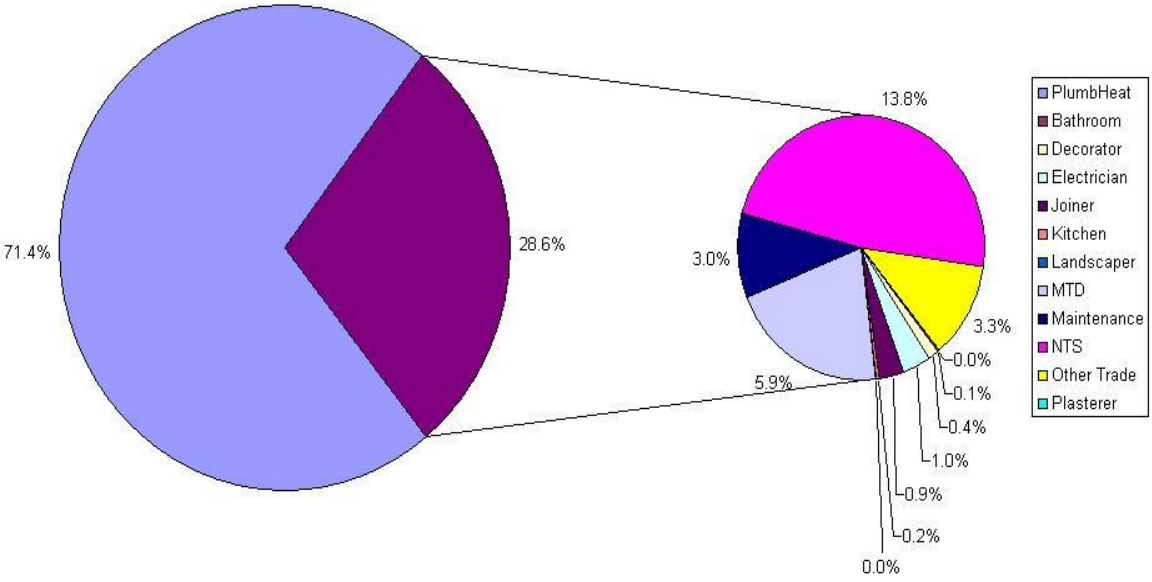


Figure A.6: Pie of Pie Chart showing the discrepancies in the **Orders made by the Customers** who were categorized as **PlumbHeat** and were verified to be under **PlumbHeat** (71.4%); those made by Customers who were categorized as belonging to one of the other 11 trade-types and were verified to be **PlumbHeat** (28.6%)

Appendix B

Figures showing the proportion of items transacted by the verified trade-type.

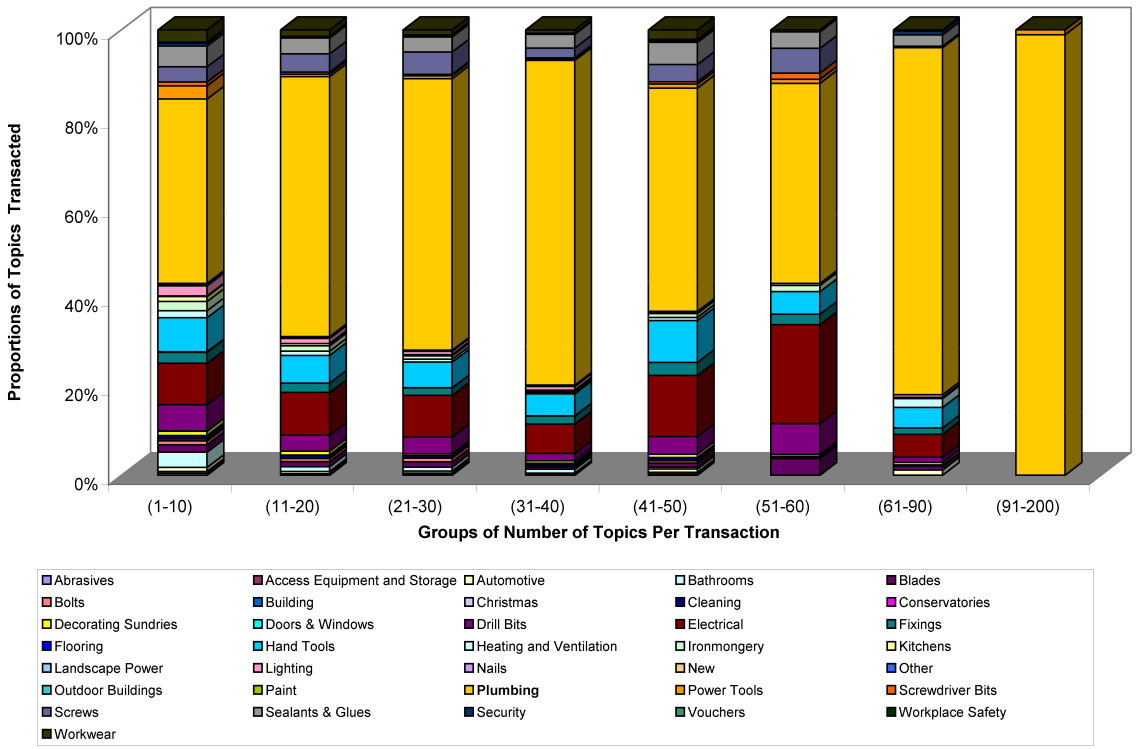


Figure B.1: Plots showing the proportions of topics transacted by the verified **Plumb-Heat** trade-types

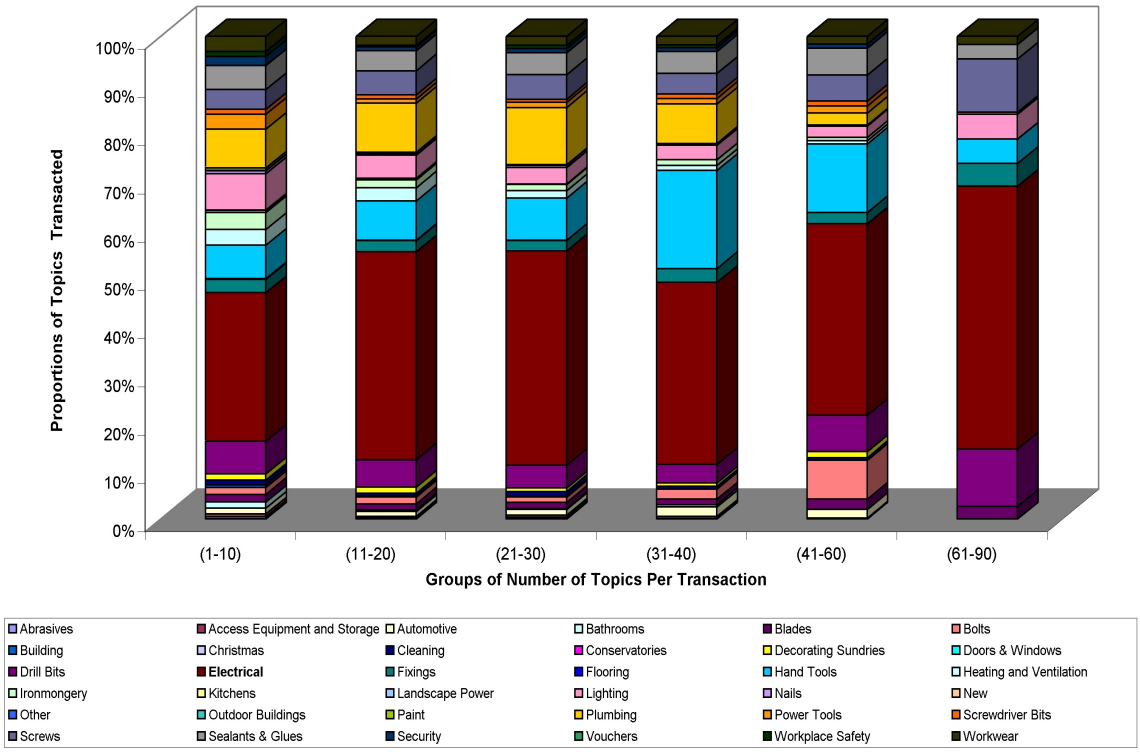


Figure B.2: Plots showing the proportions of topics transacted by the verified **Electrician** trade-types

Appendix C

Tables detailing the numerical composition of the discrepancies between categorized and verified Electrician and PlumbHeat Trade-types

Table C.1: Table showing the segmented numerical discrepancies between the categorized and verified Electrician trade-type

Topics	No. Items Transacted						
	(1-10)	(11-20)	(21-30)	(31-40)	(41-60)	(61-90)	Total Per Trade-Types
Bathroom	0	0	0	0	0	0	0
Decorator	232	14	0	0	0	0	246
Electrician	18669	894	116	24	5	3	19711
Joiner	73	1	0	0	0	0	74
Kitchen	0	0	0	0	0	0	0
Landscaper	0	0	0	0	0	0	0
MTD	2405	64	9	4	3	0	2485
Maintenance	1250	38	11	1	0	0	1300
NTS	1430	43	8	3	2	0	1486
Other Trade	5461	243	31	17	2	0	5754
Plasterer	0	0	0	0	0	0	0
PlumbHeat	970	30	7	0	0	0	1007
Total Per Segment	30490	1327	182	49	12	3	32063

Table C.2: Table showing the segmented numerical discrepancies between the categorized and verified PlumbHeat trade-type

Topics	No. Items Transacted								
	(1-10)	(11-20)	(21-30)	(31-40)	(41-50)	(51-60)	(61-90)	(91-200)	Total Per Trade-Types
Bathroom	42	4	2	0	0	0	0	0	48
Decorator	266	15	0	0	0	0	0	0	281
Electrician	657	25	2	0	0	0	0	0	684
Joiner	592	22	1	0	0	0	0	0	615
Kitchen	95	14	1	0	0	0	0	0	110
Landscaper	14	0	0	0	0	0	0	0	14
MTD	3857	158	22	8	3	1	0	0	4049
Maintenance	1966	72	7	3	0	0	0	0	2048
NTS	9343	140	13	1	0	0	0	0	9497
Other Trade	2215	69	9	1	0	0	0	0	2294
Plasterer	0	0	0	0	0	0	0	0	0
PlumbHeat	46641	2039	302	66	18	3	5	1	49075
Total Per Segment	65688	2558	359	79	21	4	5	1	68715

Table C.3: Number of Items by Topics Transacted Per Trade-Type

	Bathroom	Decorator	Electrician	Joiner	Kitchen	Landscaper	MTD	Maintenance	NTS	Other Trade	Plasterer	PlumbHeat
Abrasives	142	2221	3477	6031	262	857	18032	8839	65196	20534	125	2528
Access Equipment & Storage	73	500	1878	1113	60	275	4775	3325	13245	7337	66	1811
Automotive	251	1324	6528	4774	201	1265	18000	10519	71796	27859	178	5463
Bathrooms	833	1490	5652	4239	302	652	22180	15026	91935	20693	139	19133
Blades	771	3325	8447	17042	1155	1721	41917	15537	100515	31782	292	8539
Bolts	174	1392	7817	6251	159	2013	22811	12502	79593	43356	103	4443
Building	141	1243	3545	4644	123	1509	25662	7772	63089	14628	266	2449
Christmas	0	0	0	0	0	0	0	0	0	0	0	0
Cleaning	287	1560	3733	3224	239	796	15677	10130	42672	19413	150	4134
Conservatories	0	0	0	0	0	0	0	0	0	0	0	0
Decorating Sundries	659	5768	6993	7579	288	1257	31467	18052	99457	28976	566	6655
Doors & Windows	10	76	159	293	4	25	1081	462	3691	854	5	152
Drill Bits	1475	5279	36753	24697	1203	3371	76197	32559	229872	72051	640	28756
Electrical	3040	14407	243960	48118	3895	8912	243713	108173	952974	243927	1436	51164
Fixings	632	2969	12562	11039	804	2102	39430	19855	108986	42225	279	12450
Flooring	177	394	1177	1591	52	241	5733	2565	24819	5318	58	1064
Hand Tools	2880	11461	53938	48664	1884	7969	150129	70084	495530	141525	2379	56396
Heating & Ventilation	626	2191	16550	6809	1166	854	35910	13918	106104	25257	179	7948
Ironmongery	680	7246	15368	34039	1452	3729	88607	49158	298416	83346	478	10996
Kitchens	212	987	2822	4144	550	347	12240	7144	52017	10590	69	6405
Landscape Power	35	224	779	686	16	656	3484	2131	13222	3863	25	576
Lighting	937	4257	45943	11636	726	2663	63822	33082	230194	66263	565	12186
Nails	145	1305	3415	8231	187	1297	21534	7356	53371	15683	193	2104
New	1	3	11	10	2	3	27	16	94	28	0	23
Other	3	35	149	101	6	27	290	219	1383	625	8	106
Outdoor Buildings	8	30	117	108	2	30	329	266	2107	597	4	109
Paint	72	1279	2194	2914	109	667	10324	6949	36816	13245	58	1532
Plumbing	10515	17206	62416	57722	7472	7336	282610	127019	876232	200827	1731	276797
Power Tools	948	5454	16836	19780	724	3028	64599	23739	175388	53791	751	14284
Screwdriver Bits	279	1757	6941	9035	355	1037	22369	8787	60796	18174	263	4759
Screws	1471	6484	20498	28818	2120	4507	79940	35517	197709	72141	778	18058
Sealants & Glues	2581	9347	28344	31882	2335	3326	107496	54268	358906	96848	1124	30395
Security	162	1738	6918	6965	226	1107	24660	16656	75137	28238	122	3360
Vouchers	0	0	0	0	0	0	0	0	0	0	0	0
Workplace Safety	98	851	2092	1344	56	480	10220	6896	12963	14607	53	1172
Workwear	1083	4802	19689	16367	539	4147	61516	25939	165393	54437	806	16117

Table C.4: Average 'Items by Topics' Transacted Per Trade-Type

	Bathroom	Decorator	Electrician	Joiner	Kitchen	Landscaper	MTD	Maintenance	NTS	Other Trade	Plasterer	PlumbHeat
Abrasives	0.20	0.51	0.21	0.43	0.52	0.31	0.45	0.40	0.22	0.33	0.19	0.22
Access Equipment & Storage	0.11	0.12	0.12	0.08	0.12	0.10	0.12	0.15	0.05	0.12	0.10	0.15
Automotive	0.36	0.31	0.40	0.34	0.40	0.46	0.45	0.47	0.24	0.44	0.27	0.47
Bathrooms	1.20	0.34	0.35	0.30	0.60	0.24	0.55	0.68	0.31	0.33	0.21	1.63
Blades	1.11	0.77	0.52	1.21	2.28	0.62	1.04	0.70	0.34	0.50	0.44	0.73
Bolts	0.25	0.32	0.48	0.44	0.31	0.73	0.57	0.56	0.27	0.69	0.16	0.38
Building	0.20	0.29	0.22	0.33	0.24	0.54	0.64	0.35	0.21	0.23	0.40	0.21
Christmas	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Cleaning	0.41	0.36	0.23	0.23	0.47	0.29	0.39	0.46	0.15	0.31	0.23	0.35
Conservatories	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Decorating Sundries	0.95	1.33	0.43	0.54	0.57	0.45	0.78	0.81	0.34	0.46	0.86	0.57
Doors & Windows	0.01	0.02	0.01	0.02	0.01	0.01	0.03	0.02	0.01	0.01	0.01	0.01
Drill Bits	2.13	1.22	2.26	1.75	2.37	1.22	1.89	1.47	0.78	1.14	0.97	2.46
Electrical	4.39	3.33	15.03	3.41	7.68	3.22	6.06	4.88	3.25	3.87	2.18	4.37
Fixings	0.91	0.69	0.77	0.78	1.59	0.76	0.98	0.90	0.37	0.67	0.42	1.06
Flooring	0.26	0.09	0.07	0.11	0.10	0.09	0.14	0.12	0.08	0.08	0.09	0.09
Hand Tools	4.16	2.65	3.32	3.45	3.72	2.87	3.73	3.16	1.69	2.25	3.60	4.82
Heating & Ventilation	0.90	0.51	1.02	0.48	2.30	0.31	0.89	0.63	0.36	0.40	0.27	0.68
Ironmongery	0.98	1.67	0.95	2.41	2.86	1.35	2.20	2.22	1.02	1.32	0.72	0.94
Kitchens	0.31	0.23	0.17	0.29	1.08	0.13	0.30	0.32	0.18	0.17	0.10	0.55
Landscape Power	0.05	0.05	0.05	0.05	0.03	0.24	0.09	0.10	0.05	0.06	0.04	0.05
Lighting	1.35	0.98	2.83	0.83	1.43	0.96	1.59	1.49	0.78	1.05	0.86	1.04
Nails	0.21	0.30	0.21	0.58	0.37	0.47	0.54	0.33	0.18	0.25	0.29	0.18
New	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Other	0.00	0.01	0.01	0.01	0.01	0.01	0.01	0.01	0.00	0.01	0.01	0.01
Outdoor Buildings	0.01	0.01	0.01	0.01	0.00	0.01	0.01	0.01	0.01	0.01	0.01	0.01
Paint	0.10	0.30	0.14	0.21	0.21	0.24	0.26	0.31	0.13	0.21	0.09	0.13
Plumbing	15.17	3.98	3.84	4.09	14.74	2.65	7.03	5.73	2.98	3.19	2.62	23.64
Power Tools	1.37	1.26	1.04	1.40	1.43	1.09	1.61	1.07	0.60	0.85	1.14	1.22
Screwdriver Bits	0.40	0.41	0.43	0.64	0.70	0.37	0.56	0.40	0.21	0.29	0.40	0.41
Screws	2.12	1.50	1.26	2.04	4.18	1.63	1.99	1.60	0.67	1.15	1.18	1.54
Sealants & Glues	3.72	2.16	1.75	2.26	4.61	1.20	2.67	2.45	1.22	1.54	1.70	2.60
Security	0.23	0.40	0.43	0.49	0.45	0.40	0.61	0.75	0.26	0.45	0.18	0.29
Vouchers	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.00
Workplace Safety	0.14	0.20	0.13	0.10	0.11	0.17	0.25	0.31	0.04	0.23	0.08	0.10
Workwear	1.56	1.11	1.21	1.16	1.06	1.50	1.53	1.17	0.56	0.86	1.22	1.38

Table C.5: Table showing the segmented numerical proportions of the Items by Topics transacted by the verified Electrician trade-type

Topics	No. Items Transacted						
	(1-10)	(11-20)	(21-30)	(31-40)	(41-60)	(61-90)	Total Per Trade-Type
Abrasives	347	60	17	1	1	0	426
Access Equipment & Storage	545	30	18	8	0	0	601
Automotive	1019	188	51	33	10	0	1301
Bathrooms	1125	68	6	8	0	0	1207
Blades	1384	221	56	20	12	6	1699
Bolts	1219	253	50	35	45	0	1602
Building	451	49	8	0	0	0	508
Christmas	0	0	0	0	0	0	0
Cleaning	884	99	42	10	3	0	1038
Conservatories	0	0	0	0	0	0	0
Decorating Sundries	1106	223	32	11	7	0	1379
Doors & Windows	33	0	0	0	0	0	33
Drill Bits	5853	1027	206	65	42	28	7221
Electrical	26679	7824	1934	639	222	128	37426
Fixings	2399	428	93	49	13	11	2993
Flooring	165	8	1	0	0	0	174
Hand Tools	6008	1472	381	344	79	12	8296
Heating & Ventilation	2776	505	67	18	4	0	3370
Ironmongery	3073	297	56	20	4	0	3450
Kitchens	363	56	4	0	0	0	423
Landscape Power	100	4	1	0	0	0	105
Lighting	6523	872	147	50	13	12	7617
Nails	514	47	21	7	1	0	590
New	2	0	0	0	0	0	2
Other	22	2	0	0	0	0	24
Outdoor Buildings	24	0	0	0	0	0	24
Paint	462	54	6	0	0	0	522
Plumbing	6979	1842	515	138	14	0	9488
Power Tools	2655	156	47	20	8	1	2887
Screwdriver Bits	878	161	28	15	6	0	1088
Screws	3549	907	222	72	30	26	4806
Sealants & Glues	4319	755	196	78	31	7	5386
Security	1632	141	40	13	5	0	1831
Vouchers	0	0	0	0	0	0	0
Workplace Safety	895	54	27	10	0	0	986
Workwear	2754	347	83	30	9	4	3227
Total Topic Per Segment	86737	18150	4355	1694	559	235	111730

Table C.6: Table showing the segmented proportions (averaged by number of orders) of the Items by Topics transacted by the verified Electrician trade-type

Topics	Average No. Items Transacted					
	(1-10)	(11-20)	(21-30)	(31-40)	(41-60)	(61-90)
Abrasives	0.011381	0.045215	0.093407	0.020408	0.083333	0
Access Equipment and Storage	0.017875	0.022607	0.098901	0.163265	0	0
Automotive	0.033421	0.141673	0.28022	0.673469	0.833333	0
Bathrooms	0.036897	0.051243	0.032967	0.163265	0	0
Blades	0.045392	0.166541	0.307692	0.408163	1	2
Bolts	0.03998	0.190656	0.274725	0.714286	3.75	0
Building	0.014792	0.036925	0.043956	0	0	0
Christmas	0	0	0	0	0	0
Cleaning	0.028993	0.074604	0.230769	0.204082	0.25	0
Conservatories	0	0	0	0	0	0
Decorating Sundries	0.036274	0.168048	0.175824	0.22449	0.583333	0
Doors & Windows	0.001082	0	0	0	0	0
Drill Bits	0.191965	0.773926	1.131868	1.326531	3.5	9.333333
Electrical	0.875008	5.896006	10.62637	13.04082	18.5	42.66667
Fixings	0.078682	0.322532	0.510989	1	1.083333	3.666667
Flooring	0.005412	0.006029	0.005495	0	0	0
Hand Tools	0.197048	1.109269	2.093407	7.020408	6.583333	4
Heating and Ventilation	0.091046	0.380558	0.368132	0.367347	0.333333	0
Ironmongery	0.100787	0.223813	0.307692	0.408163	0.333333	0
Kitchens	0.011906	0.0422	0.021978	0	0	0
Landscape	0.00328	0.003014	0.005495	0	0	0
Power Lighting	0.213939	0.657121	0.807692	1.020408	1.083333	4
Nails	0.016858	0.035418	0.115385	0.142857	0.083333	0
New	6.56E-05	0	0	0	0	0
Other	0.000722	0.001507	0	0	0	0
Outdoor Buildings	0.000787	0	0	0	0	0
Paint	0.015153	0.040693	0.032967	0	0	0
Plumbing	0.228895	1.388093	2.82967	2.816327	1.166667	0
Power Tools	0.087078	0.117558	0.258242	0.408163	0.666667	0.333333
Screwdriver Bits	0.028796	0.121326	0.153846	0.306122	0.5	0
Screws	0.116399	0.683497	1.21978	1.469388	2.5	8.666667
Sealants & Glues	0.141653	0.568953	1.076923	1.591837	2.583333	2.333333
Security	0.053526	0.106255	0.21978	0.265306	0.416667	0
Vouchers	0	0	0	0	0	0
Workplace Safety	0.029354	0.040693	0.148352	0.204082	0	0
Workwear	0.090325	0.261492	0.456044	0.612245	0.75	1.333333

Table C.7: Table showing the segmented proportions of the Items by Topics transacted by the verified PlumbHeat trade-type

Topics	No. Items Transacted								
	(1-10)	(11-20)	(21-30)	(31-40)	(41-50)	(51-60)	(61-90)	(91-200)	Total Per Topic
Abrasives	695	75	19	5	2	0	0	0	796
Access Equipment & Storage	793	38	18	1	4	0	0	0	854
Automotive	1687	172	43	6	7	0	4	0	1919
Bathrooms	6176	381	77	24	4	0	0	0	6662
Blades	3097	398	102	15	9	8	3	0	3632
Bolts	1565	249	72	10	5	0	1	0	1902
Building	800	61	9	0	1	0	0	0	871
Christmas	0	0	0	0	0	0	0	0	0
Cleaning	1387	213	39	13	6	1	0	0	1659
Conservatories	0	0	0	0	0	0	0	0	0
Decorating Sundries	1797	306	35	15	7	1	2	0	2163
Doors & Windows	43	2	0	0	0	0	0	0	45
Drill Bits	10762	1263	329	46	38	15	4	0	12457
Electrical	16947	3373	815	181	132	48	18	0	21514
Fixings	4441	723	143	50	28	5	5	0	5395
Flooring	272	14	5	1	0	0	0	0	292
Hand Tools	13907	2190	501	136	90	11	16	0	16851
Heating & Ventilation	2878	334	56	10	7	0	7	0	3292
Ironmongery	3805	415	63	8	8	3	1	0	4303
Kitchens	2004	182	24	5	0	0	0	0	2215
Landscape Power	220	6	1	0	0	0	0	0	227
Lighting	4115	395	65	23	3	0	2	0	4603
Nails	598	103	14	4	1	1	0	0	721
New	14	0	0	0	0	0	0	0	14
Other	31	3	1	1	0	0	0	0	36
Outdoor Buildings	33	1	0	0	0	0	0	0	34
Paint	388	39	8	4	1	0	0	0	440
Plumbing	75395	20536	5296	2011	481	97	272	180	104268
Power Tools	5383	201	58	7	9	2	0	2	5662
Screwdriver Bits	1600	165	29	8	5	3	0	0	1810
Screws	6220	1435	438	61	37	12	1	0	8204
Sealants & Glues	8493	1254	292	86	48	8	9	0	10190
Security	1139	57	24	3	3	0	3	0	1229
Vouchers	0	0	0	0	0	0	0	0	0
Workplace Safety	422	62	14	3	3	0	0	0	504
Workwear	5101	530	102	22	21	1	1	0	5778
Total Items Per Segment	182208	35176	8692	2759	960	216	349	182	230542

Table C.8: Table showing the segmented proportions (averaged by number of orders) of the Items by Topics transacted by the verified PlumbHeat trade-type

Topics	No. Items Transacted							
	(1-10)	(11-20)	(21-30)	(31-40)	(41-50)	(51-60)	(61-90)	(91-200)
Abrasives	0.01058	0.02932	0.052925	0.063291	0.095238	0	0	0
Access Equipment & Storage	0.012072	0.014855	0.050139	0.012658	0.190476	0	0	0
Automotive	0.025682	0.06724	0.119777	0.075949	0.333333	0	0.8	0
Bathrooms	0.09402	0.148944	0.214485	0.303797	0.190476	0	0	0
Blades	0.047147	0.15559	0.284123	0.189873	0.428571	2	0.6	0
Bolts	0.023825	0.097342	0.200557	0.126582	0.238095	0	0.2	0
Building	0.012179	0.023847	0.02507	0	0.047619	0	0	0
Christmas	0	0	0	0	0	0	0	0
Cleaning	0.021115	0.083268	0.108635	0.164557	0.285714	0.25	0	0
Conservatories	0	0	0	0	0	0	0	0
Decorating Sundries	0.027357	0.119625	0.097493	0.189873	0.333333	0.25	0.4	0
Doors & Windows	0.000655	0.000782	0	0	0	0	0	0
Drill Bits	0.163835	0.493745	0.916435	0.582278	1.809524	3.75	0.8	0
Electrical	0.257992	1.318608	2.270195	2.291139	6.285714	12	3.6	0
Fixings	0.067607	0.282643	0.398329	0.632911	1.333333	1.25	1	0
Flooring	0.004141	0.005473	0.013928	0.012658	0	0	0	0
Hand Tools	0.211713	0.856138	1.395543	1.721519	4.285714	2.75	3.2	0
Heating & Ventilation	0.043813	0.130571	0.155989	0.126582	0.333333	0	1.4	0
Ironmongery	0.057925	0.162236	0.175487	0.101266	0.380952	0.75	0.2	0
Kitchens	0.030508	0.071149	0.066852	0.063291	0	0	0	0
Landscape	0.003349	0.002346	0.002786	0	0	0	0	0
Power								
Lighting	0.062645	0.154418	0.181058	0.291139	0.142857	0	0.4	0
Nails	0.009104	0.040266	0.038997	0.050633	0.047619	0.25	0	0
New	0.000213	0	0	0	0	0	0	0
Other	0.000472	0.001173	0.002786	0.012658	0	0	0	0
Outdoor Buildings	0.000502	0.000391	0	0	0	0	0	0
Paint	0.005907	0.015246	0.022284	0.050633	0.047619	0	0	0
Plumbing	1.147774	8.028147	14.75209	25.4557	22.90476	24.25	54.4	180
Power Tools	0.081948	0.078577	0.16156	0.088608	0.428571	0.5	0	2
Screwdriver Bits	0.024358	0.064504	0.08078	0.101266	0.238095	0.75	0	0
Screws	0.09469	0.560985	1.220056	0.772152	1.761905	3	0.2	0
Sealants & Glues	0.129293	0.490227	0.81337	1.088608	2.285714	2	1.8	0
Security	0.01734	0.022283	0.066852	0.037975	0.142857	0	0.6	0
Vouchers	0	0	0	0	0	0	0	0
Workplace Safety	0.006424	0.024238	0.038997	0.037975	0.142857	0	0	0
Workwear	0.077655	0.207193	0.284123	0.278481	1	0.25	0.2	0

Appendix D

Figures showing the distributions of classess uniformly sampled from Screwfix's 2007 and 2008 transactional data.

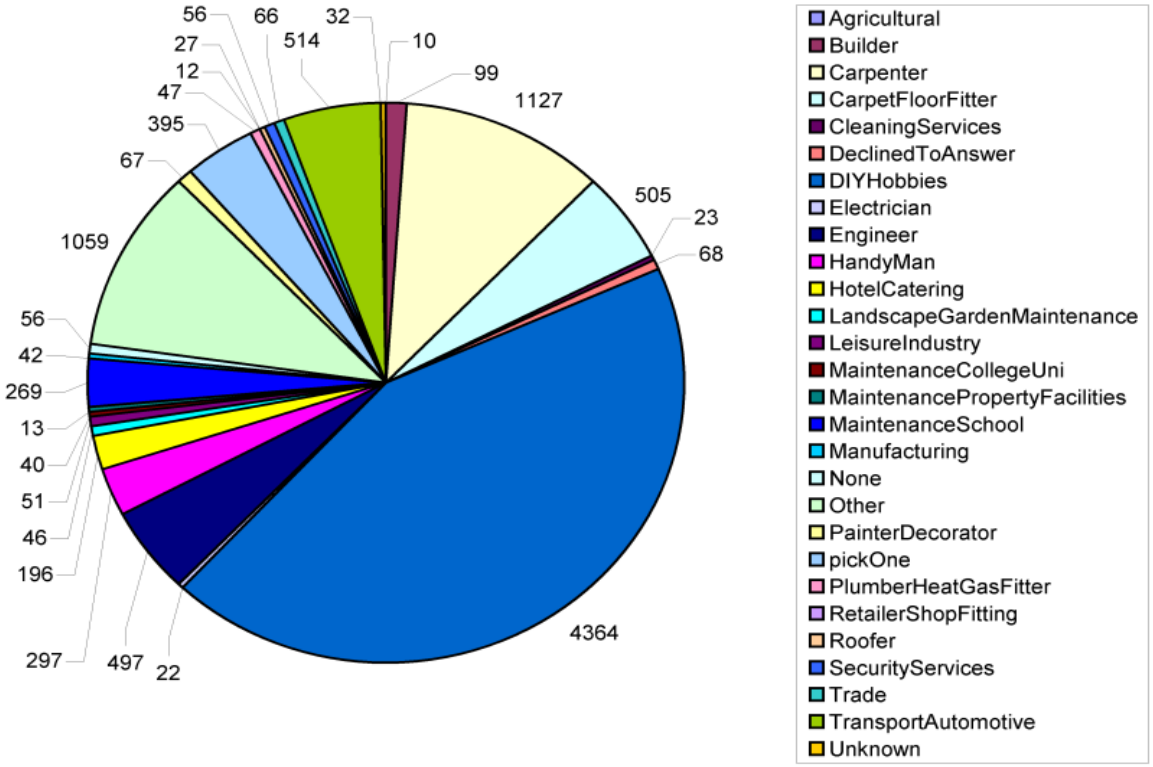


Figure D.1: Distribution of Classes of the Sampled 10000 Screwfix's Transactions for 2007

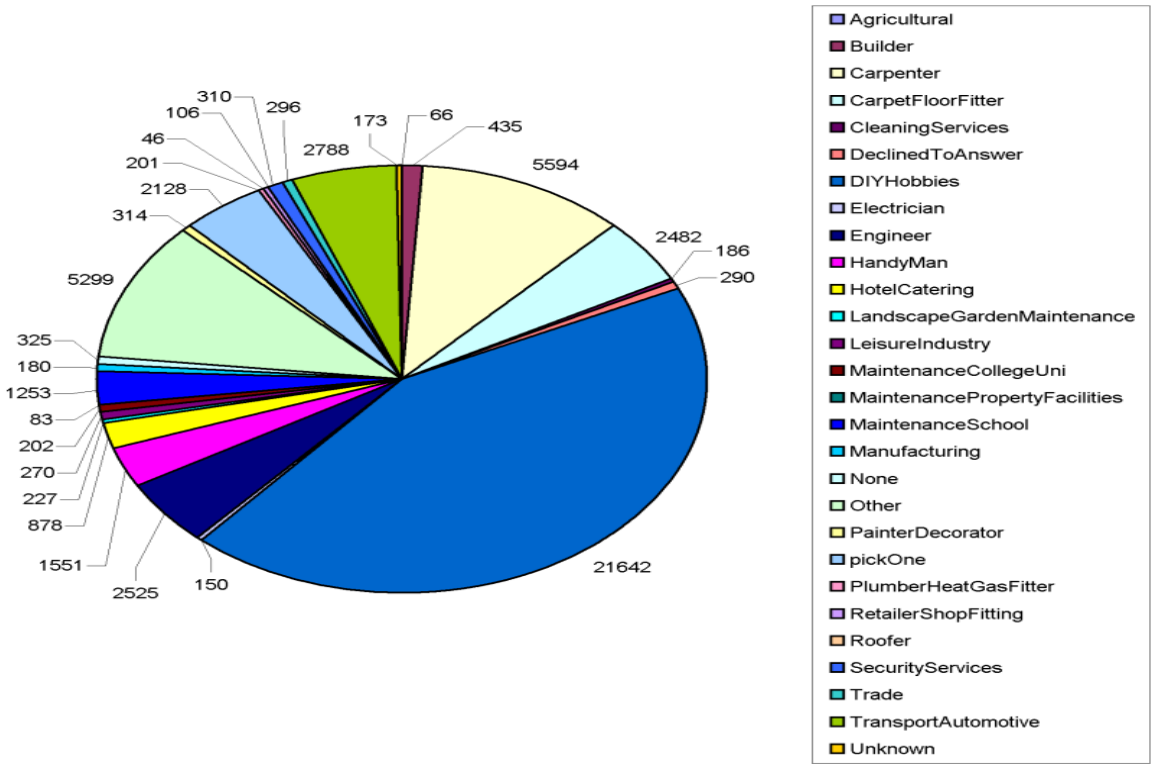


Figure D.2: Distribution of Classes of the Sampled 50000 Screwfix's Transactions for 2007

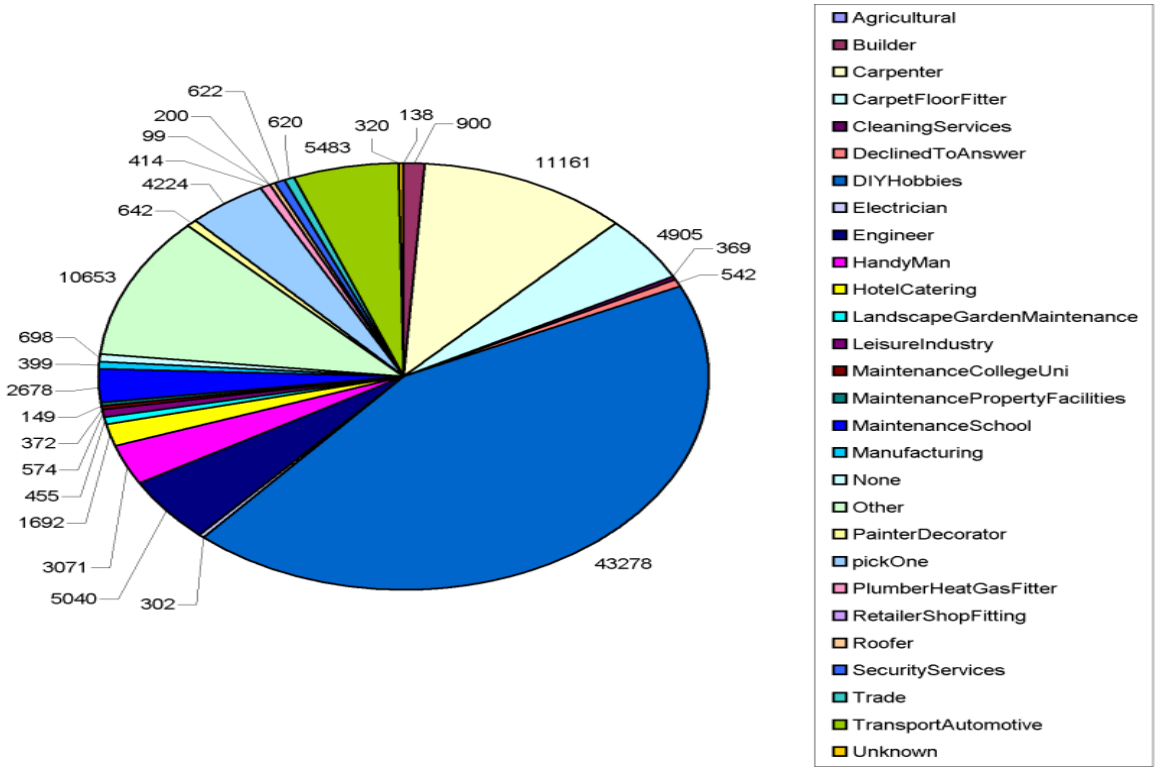


Figure D.3: Distribution of Classes of the Sampled 100000 Screwfix's Transactions for 2007

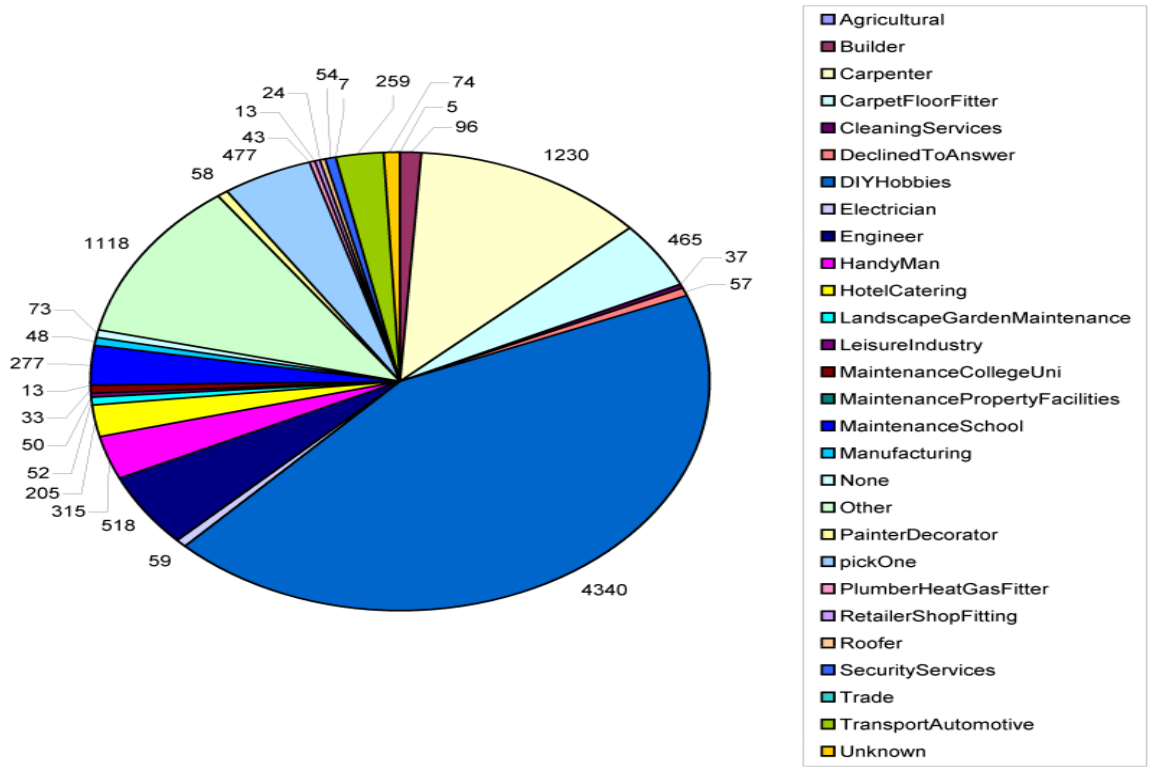


Figure D.4: Distribution of Classes of the Sampled 10000 Screwfix's Transactions for 2008

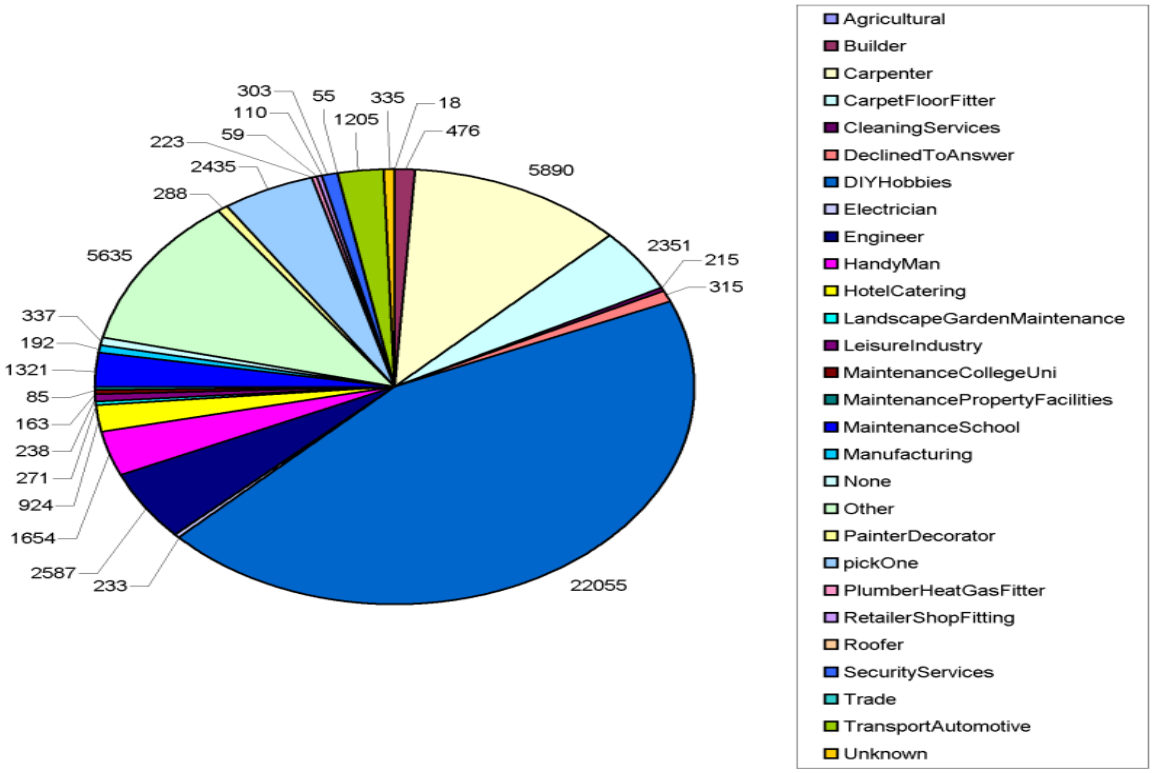


Figure D.5: Distribution of Classes of the Sampled 50000 Screwfix's Transactions for 2008

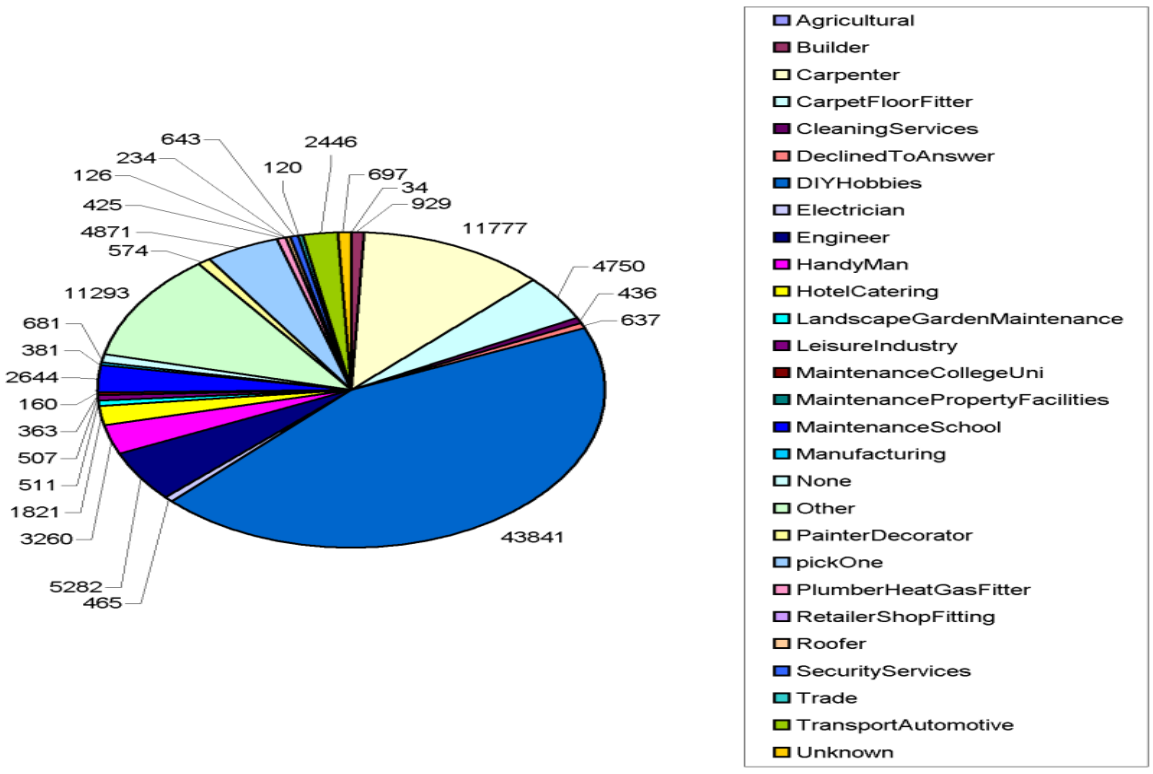


Figure D.6: Distribution of Classes of the Sampled 100000 Screwfix's Transactions for 2008

Appendix E

Tables of Screwfix's Transactional Data Attributes

Table E.1: List of Trade-types Identified in Screwfix's Transactional Database

SNo.	Trade-Type
1	Agricultural
2	Builder
3	Carpenter
4	CarpetFloorFitter
5	CleaningServices
6	DeclinedToAnswer
7	DIYHobbies
8	Electrician
9	Engineer
10	HandyMan
11	HotelCatering
12	LandscapeGardenMaintenance
13	LeisureIndustry
14	MaintenanceCollegeUni
15	MaintenancePropertyFacilities
16	MaintenanceSchool
17	Manufacturing
18	None
19	Other
20	PainterDecorator
21	pickOne
22	PlumberHeatGasFitter
23	RetailerShopFitting
24	Roofer
25	SecurityServices
26	trade
27	TransportAutomotive
28	Unknown

Table E.2: List of Screwfix's Topics

S.No.	Topics
1	Power Tools
2	Electrical
3	Building
4	Conservatories
5	Lighting
6	Bathrooms
7	Workplace Safety
8	Drill Bits
9	Doors & Windows
10	Abrasives
11	Hand Tools
12	Sealants & Glues
13	Fixings
14	Screwdriver Bits
15	Flooring
16	Landscape Power
17	Heating and Ventilation
18	Workwear
19	Decorating Sundries
20	Automotive
21	Other
22	Ironmongery
23	Kitchens
24	Cleaning
25	Security
26	Plumbing
27	Vouchers
28	Screws
29	Access Equipment and Storage
30	Paint
31	Outdoor Buildings
32	Nails
33	Bolts
34	Blades

Table E.3: Number of Screwfix's Topics Transacted in 2007

Topics	Number Transacted
Conservatories	4
Vouchers	5132
Outdoor Buildings	19446
Doors & Windows	25927
Landscape Power	72423
Access Equipment and Storage	94094
Flooring	116640
Workplace Safety	128710
Paint	230030
Kitchens	259634
Nails	277099
Bolts	283722
Cleaning	300071
Building	322305
Screwdriver Bits	343101
Fixings	347998
Abrasives	383090
Automotive	391122
Security	447075
Decorating Sundries	485104
Bathrooms	520249
Heating and Ventilation	536738
Blades	616555
Screws	653490
Workwear	769564
Power Tools	1068607
Drill Bits	1256343
Lighting	1347770
Ironmongery	1480908
Sealants & Glues	1804734
Hand Tools	2631306
Other	3207206
Plumbing	4412804
Electrical	4417499

Table E.4: List of Screwfix's Topics Transacted in 2008

Topics	Number Transacted
Vouchers	6617
Outdoor Buildings	12721
New	17300
Doors & Windows	24232
Landscape Power	113241
Flooring	118334
Access Equipment and Storage	132644
Workplace Safety	176976
Paint	246310
Kitchens	313625
Cleaning	379055
Nails	389439
Screwdriver Bits	425669
Abrasives	439529
Building	445770
Automotive	501030
Security	561060
Decorating Sundries	600562
Bathrooms	630736
Heating and Ventilation	681403
Blades	792016
Bolts	829047
Fixings	1015226
Workwear	1145322
Power Tools	1225188
Lighting	1540062
Drill Bits	1598675
Ironmongery	1974102
Screws	2151845
Other	2217159
Sealants & Glues	2242442
Hand Tools	3051240
Electrical	5637409
Plumbing	6146210

Table E.5: Cross-section of Sampled Screwfix's Transactional Data

OrderID	OrderDate	CustomerID	Trade-typeID	Abrasives	Access Equipment and Storage	Automotive	...
101137349	2052007	12515385	1	0	0	0	...
102803511	30052007	9345961	2	0	0	0	...
108159113	15082007	12805602	3	0	0	0	...
108808439	24082007	10094253	4	0	0	0	...
108777650	28082007	12814741	5	0	0	4	...
104721907	2072007	12223592	6	0	0	0	...
108213464	15082007	12814104	7	0	0	0	...
106542573	18072007	12593122	8	0	0	5	...
101787112	14052007	10943002	9	0	0	0	...
109156113	31082007	5991140	10	0	0	0	...
106937558	25072007	11158304	11	0	0	0	...
102720574	29052007	8439862	12	0	0	5	...
103203866	6062007	10535462	13	0	0	0	...
102052918	25052007	12065063	14	0	0	0	...
102178637	18052007	234430	15	0	0	0	...
106475196	17072007	1153170	16	0	0	0	...
112321650	18102007	235271	17	0	0	0	...
116797440	20122007	12001877	18	0	0	0	...
109755323	10092007	6458108	19	0	0	0	...
111450014	5102007	7459110	20	0	0	1	...
114984674	25112007	343463	21	0	0	0	...
109963893	13092007	10276919	22	0	0	0	...
109800902	10092007	10225504	23	0	0	0	...
109690201	8092007	10019128	24	0	0	0	...
115873124	6122007	12225400	25	0	0	0	...
111705644	9102007	7237994	26	0	0	0	...
112682790	23102007	11370869	27	0	0	0	...
113705188	7112007	782192	28	0	0	0	...

Table E.6: Cross-section of Computed Screwfix's Electrician and PlumbHeat Profiles.

CustomerID	Electrical	...	HandTools	Plumbing	...	WorkplaceSafety	Workwear	ClassId	ClassName
12515385	116	...	2	0	...	0	0	1	Electrician
11370869	8	...	10	44	...	1	6	2	PlumbHeat

Appendix F

Classification of Sampled Screwfix's Data Experiments Matlab Code

F.1 Classification: Cross-Validation Experiment Code

```
function [WeightedAverageTestError,TestErrorsPerClass,
    AssignedNumericLabels] = classificationExp(~)
%This experiment uses PRTools' implementation of cross-
    validation to compute the error estimates of
%ldc(), qdc(), knn(k=3), stumpc classifiers on
%uniformly sampled 10,000, 50,000 and 100,000 Screwfix's
    transaction data for 2007 and 2008
%The files are each imported from a specified CSV File location,
     the cross-validation computed
%and the estimated errors displayed on Matlab's command window.

sampled2007Orders = importdata('CSV File location for Sampled
    Screwfixs 2007 transaction Data');

sampled2008Orders = importdata('CSV File location for Sampled
    Screwfixs 2008 transaction Data');

A = sampled2007Orders(:,5:38);

B = sampled2008Orders(:,5:38);

Alabs = sampled2007Orders(:,4);

Blabs = sampled2008Orders(:,4);
```

```
C07 = dataset(A,Alabs);

D08 = dataset(B,Blabs);

data = {C07,D08};

classifiers = {ldc,qdc,knnc([],3),stumpc};
[WeightedAverageTestError,TestErrorsPerClass,
    AssignedNumericLabels]=crossval(data,classifiers,10,30);
end
```

Appendix G

SLIGRO Categories Data Description Table

Table G.1: Number of Transactions and Number of Items Transacted by SLIGRO Categories in the 3 Years Period.

Category	Category Code	No. Transactions	No. Items
Supermarkt/rijdende winkel	100	3163	45182082
Avondwinkel/toko	101	163	722835
Bakkerij/banketbakkerij	110	2467	899771
AGF/groentezaak/fruitspeczaak	120	1564	742059
Slagerij	130	1956	485898
Vishandel	140	1068	108297
Poelier/wild en gevogelte	150	291	30594
Spec.zaak voeding (toko/reform	160	611	147848
Slijterij / drankenhandel	170	1142	1571280
Zoetwaren/chocolade/tabakszaak	180	1473	230516
Kapsalon/drogisterij/apotheek	190	11381	277788
Dierenspeciaalzaak	200	672	18699
Tankstations	210	1552	1512982
Diverse detailhandel food	220	2807	712289
Diverse detailhandel non-food	230	24734	998751
Caf/zalencentrum/bar	300	8596	2053682
Cafetaria/shoarma/fastfood	310	7245	3831822
Brasserie/lunchroom/croiss.	320	2494	1733624
Restaurant Nederlands-Frans	331	6974	5236685
Restaurant Chin-Ind.-overig Az	332	2307	355479
Restaurant Italiaans	333	771	210404
Restaurant Grieks	334	368	143512
Restaurant Spaans	335	101	41688
Restaurant overig Europees	336	371	163331
Restaurant overig Zuid-Amerika	337	184	77211
Restaurant overig internationa	338	438	118360
Logiesverstrekkers (hotel/mote	350	2014	1714545
Recreatie (camping/app./bungal	360	3871	2268079
Party- of lokatiecatering	380	3639	1992469
Kantine sportver./sporthal	390	5456	3803329
Kantine vereniging	391	3454	428757
Kookclubs	500	1133	78399
Diverse Horeca	590	4401	1015822
Contractcatering (bedrijfsrest	600	1090	7026124
Kinderdagverblijf	611	1270	285554
Basisschool	612	2927	191446
Voortgezet onderwijs	613	2307	1423996
HBO / Universiteit	614	543	187208

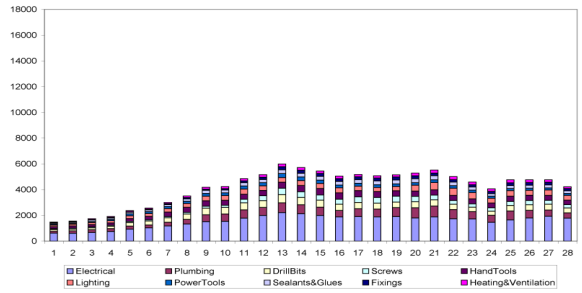
Continued on next page

Table G.1 – continued from previous page

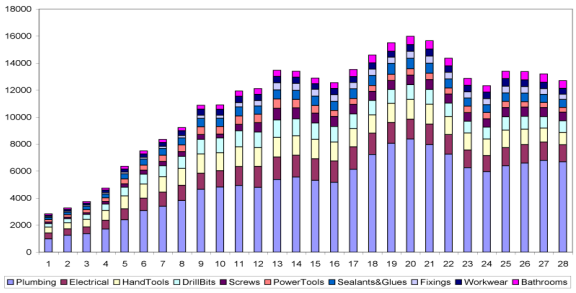
Category	Category Code	No. Transactions	No. Items
Schoolcatering	615	178	662068
Keuken instelling/tehuis/ziekh	620	4645	6632638
Defensie	630	244	40140
Crematorium	640	279	12229
Penitentiaire instellingen	650	184	376687
EXPORT	660	112	258888
Bedrijven < 50 werknemers	701	217796	10557705
Bedrijven 50-99 werknemers	702	4472	1342253
Bedrijven 100-499 werknemers	703	3140	2148643
Bedrijven >= 500 werknemers	704	1685	2494418
Vereniging / stichting	800	27076	1144063
Landbouw / tuinbouw / veeteelt	820	4192	112184
Fokkers / kennels / asiels	830	1910	47392
Vrije beroepen	840	13149	432466
Bijzondere relatie	890	4541	160806
Eigen personeel	900	6498	155559
Sligrovestiging/-dochter	910	46	151281
Personeel Freshpartners	920	180	3711
EM-T	951	109	39532348
Inversco CD Magazijn	952	18	42192
SLIGRO LOCATIES	953	12	1396
Diversen	959	1	3694
Restaurants eigen vestigingen	961	67	187172
Inversco CD Institutioneel	963	365	106778
Inversco CD Horeca	964	547	83450
INVERSCO DIR Horeca	966	11	46439
Geen klantenkaart	999	169	2442217

Appendix H

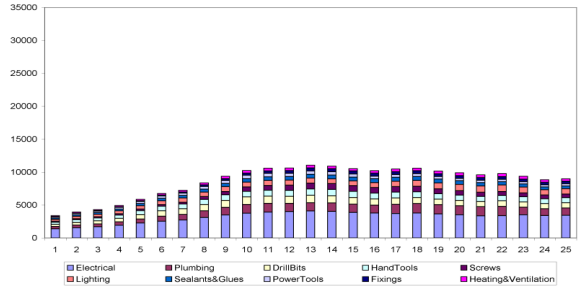
Figures Showing the Distribution of the Top 10 Products Transacted by the Electricians and PlumbHeaters in the Training and Test Datasets in the 3, 6, 9 and 12 Months Sliding Windows.



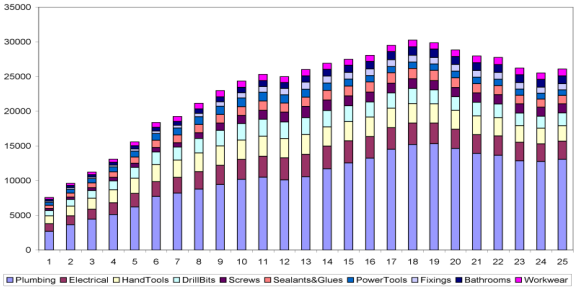
Electricians 3 Months Sliding Windows Transactions



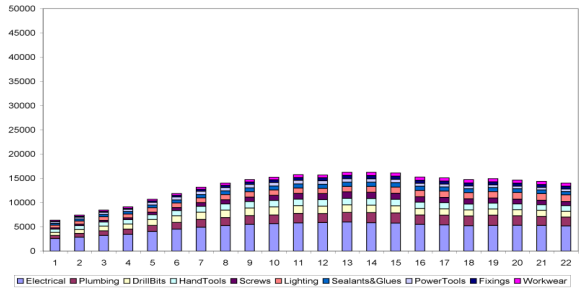
PlumbHeaters 3 Months Sliding Windows Transactions



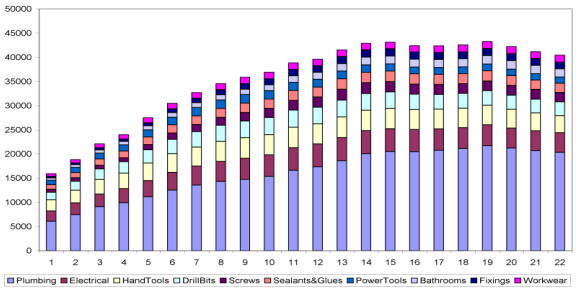
Electricians 6 Months Sliding Windows Transactions



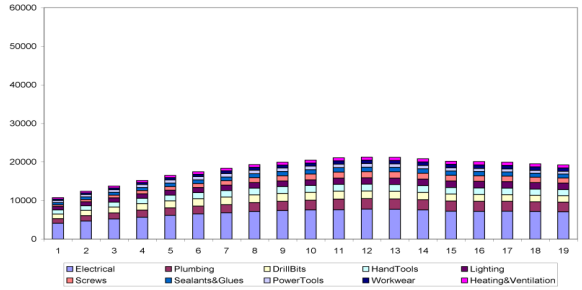
PlumbHeaters 6 Months Sliding Windows Transactions



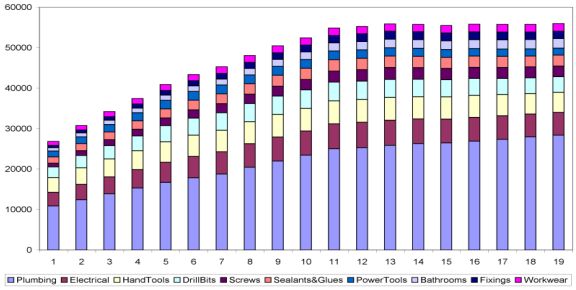
Electricians 9 Months Sliding Windows Transactions



PlumbHeaters 9 Months Sliding Windows Transactions

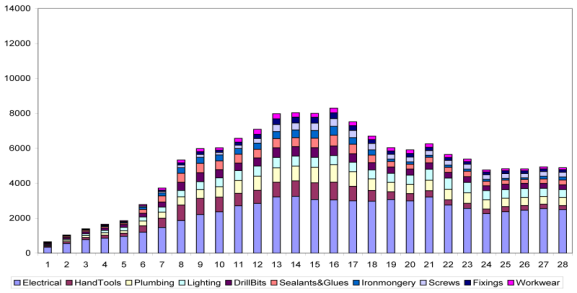


Electricians 12 Months Sliding Windows Transactions

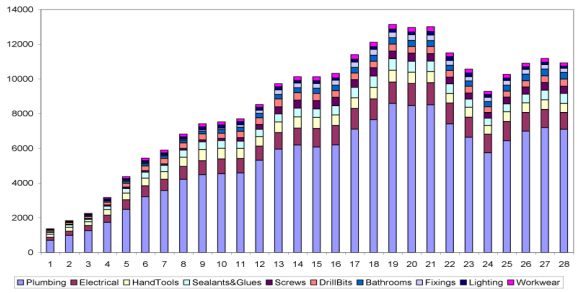


PlumbHeaters 12 Months Sliding Windows Transactions

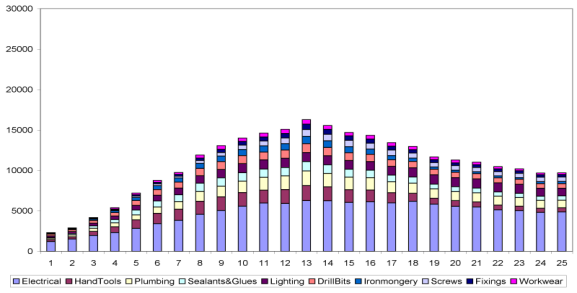
Figure H.1: Plots showing the Top 10 Products Transacted in Training Datasets by the Electricians and the PlumbHeaters in the 3, 6, 9 and 12 Months Sliding Windows.



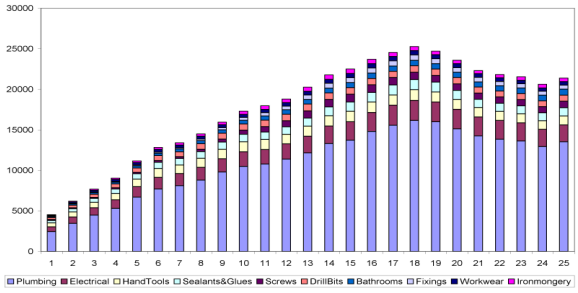
Electricians 3 Months Sliding Windows Transactions



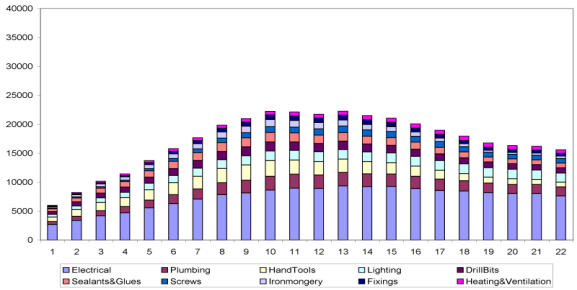
PlumbHeaters 3 Months Sliding Windows Transactions



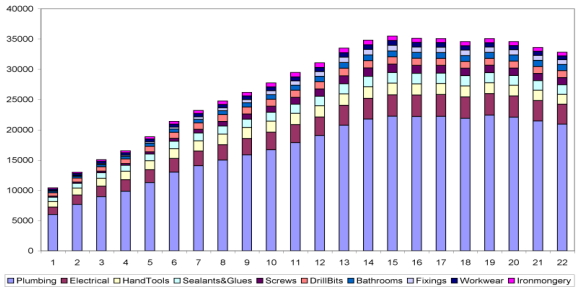
Electricians 6 Months Sliding Windows Transactions



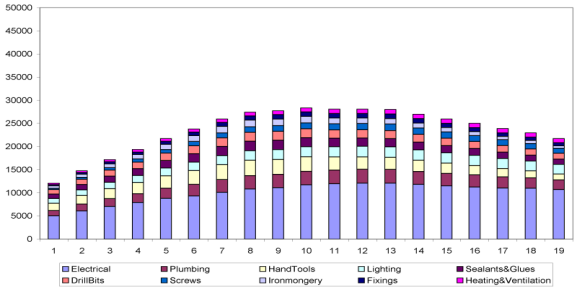
PlumbHeaters 6 Months Sliding Windows Transactions



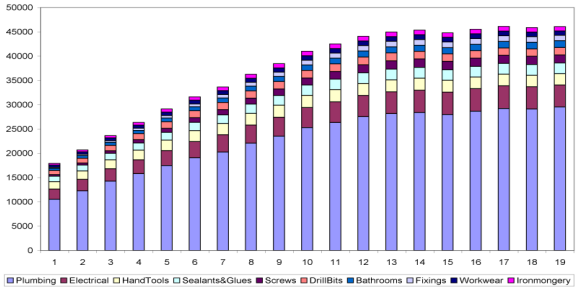
Electricians 9 Months Sliding Windows Transactions



PlumbHeaters 9 Months Sliding Windows Transactions



Electricians 12 Months Sliding Windows Transactions

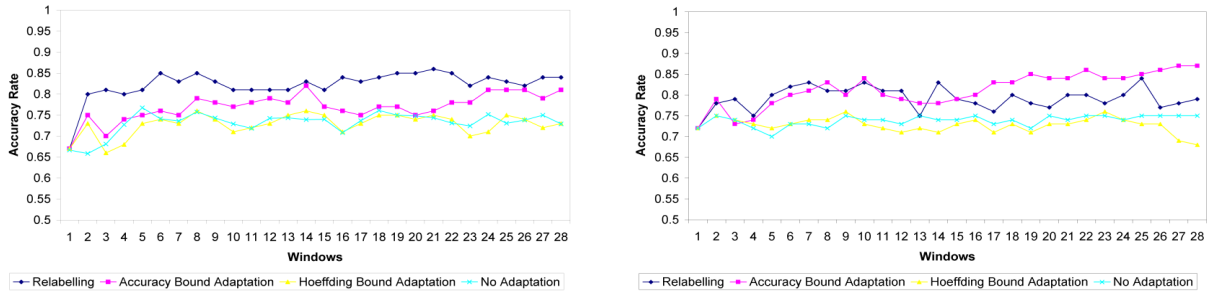


PlumbHeaters 12 Months Sliding Windows Transactions

Figure H.2: Plots showing the Top 10 Products Transacted in Test Datasets by the Electricians and the PlumbHeaters in the 3, 6, 9 and 12 Months Sliding Windows.

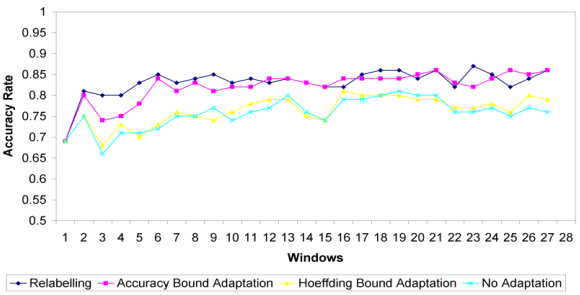
Appendix I

Figures Showing Comparative Performance of Adaptation Strategies on Ensembles of Decision Trees (J48), Naive Bayes, Linear Regression and Support Vector Machines for the 3, 6, 9 and 12 Months Sliding Windows.

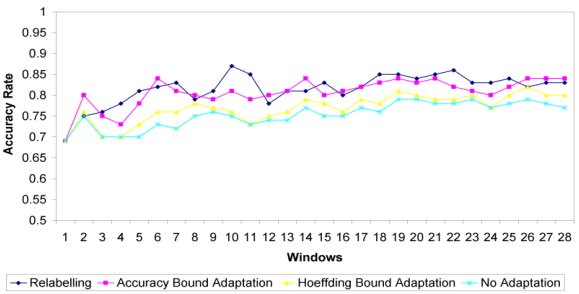


Decision Tree

Naive Bayes

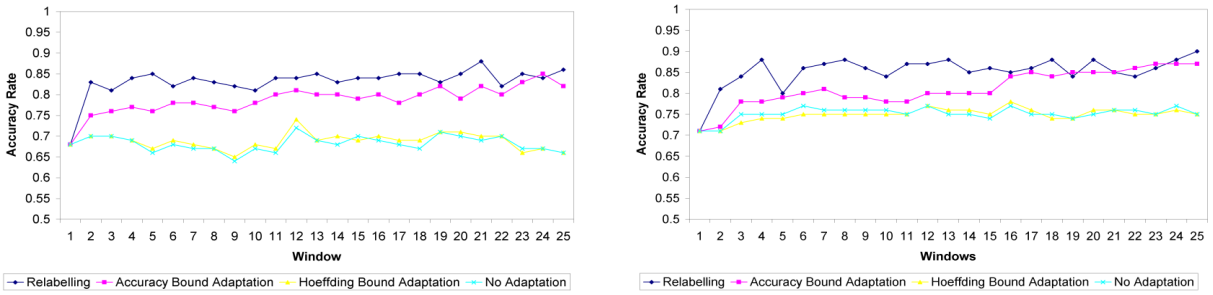


Linear Regression



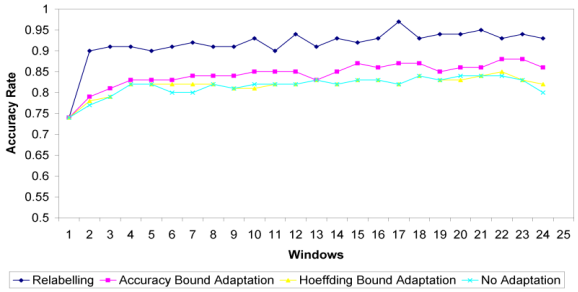
Support Vector Machines

Figure I.1: Plots showing the Comparative Performance of Decision Trees, Naive Bayes, Linear Regression and Support Vector Machines classifiers in the Three (3) Months Sliding Windows.

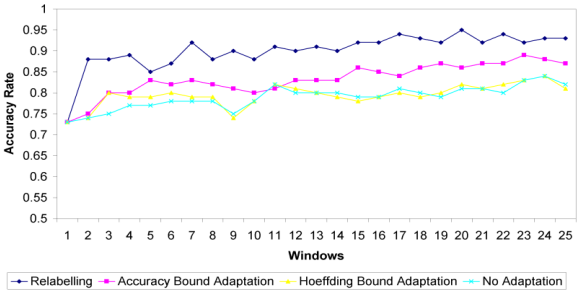


Decision Tree

Naive Bayes

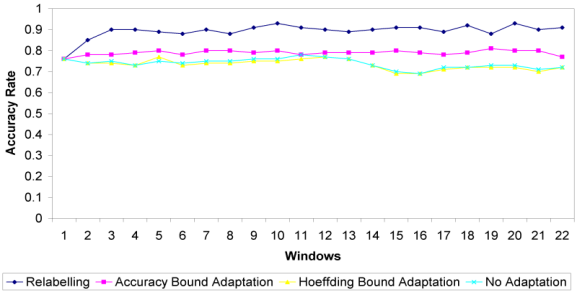


Linear Regression

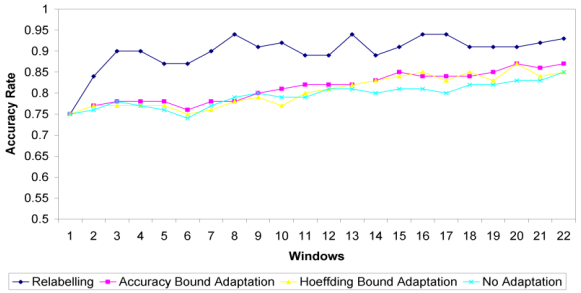


Support Vector Machines

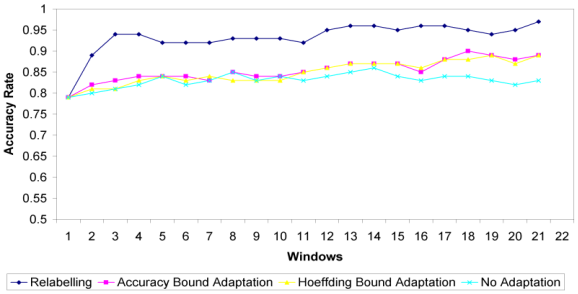
Figure I.2: Plots showing the Comparative Performance of Decision Trees, Naive Bayes, Linear Regression and Support Vector Machines classifiers in the Six (6) Months Sliding Windows.



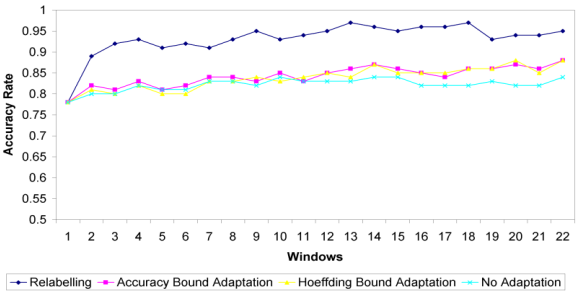
Decision Tree



Naive Bayes

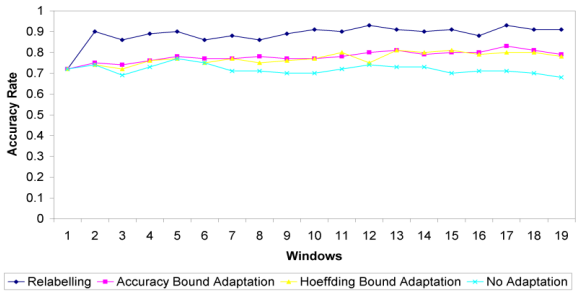


Linear Regression

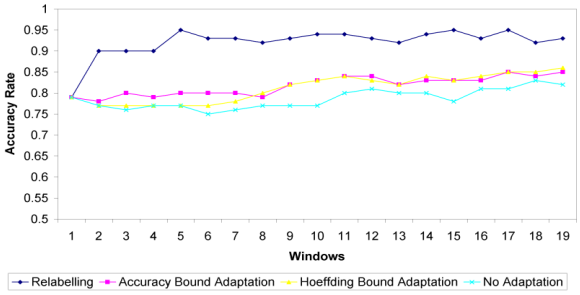


Support Vector Machines

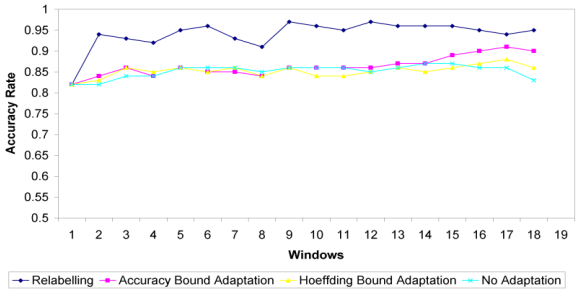
Figure I.3: Plots showing the Comparative Performance of Decision Trees, Naive Bayes, Linear Regression and Support Vector Machines classifiers in the Nine (9) Months Sliding Windows.



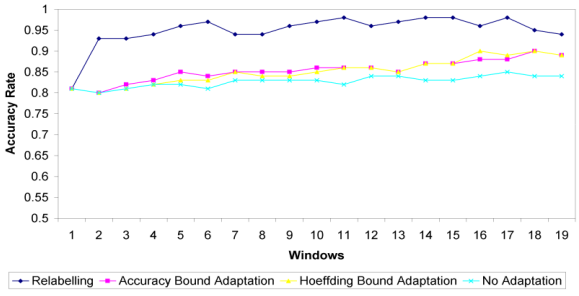
Decision Tree



Naive Bayes



Linear Regression

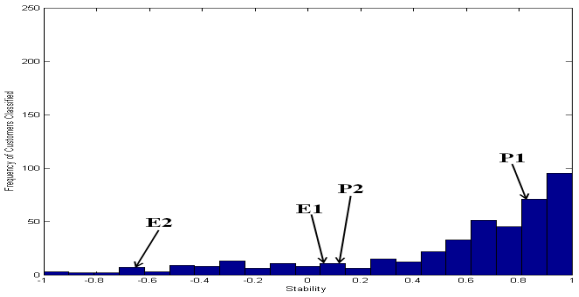


Support Vector Machines

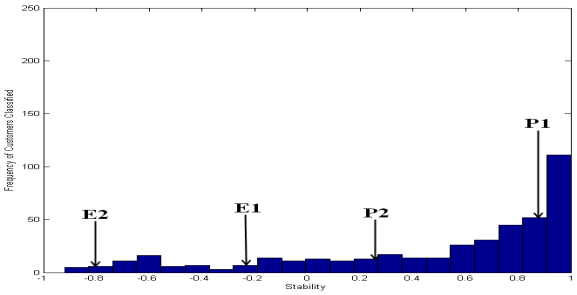
Figure I.4: Plots showing the Comparative Performance of Decision Trees, Naive Bayes, Linear Regression and Support Vector Machines classifiers in the Twelve (12) Months Sliding Windows.

Appendix J

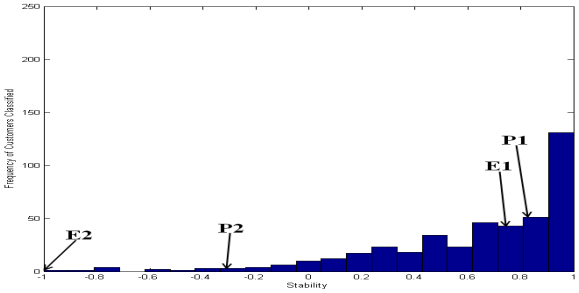
Figures Showing Comparative Customer Profile Classification Stability for the Decision Trees (J48), Naive Bayes, Linear Regression and Support Vector Machines for the 3, 6, 9 and 12 Months Sliding Windows.



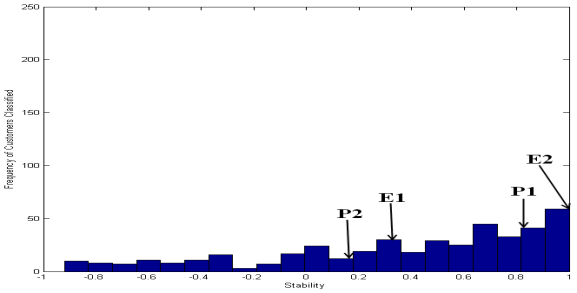
Adaptive Decision Tree Classifiers



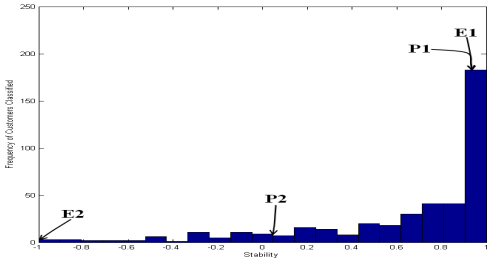
Static Decision Tree Classifiers With Misclassified Customer Profiles Relabelled



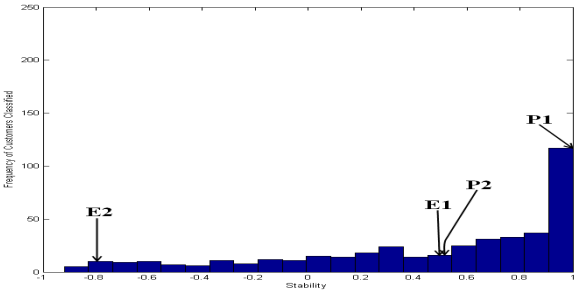
Adaptive Naive Bayes Classifiers



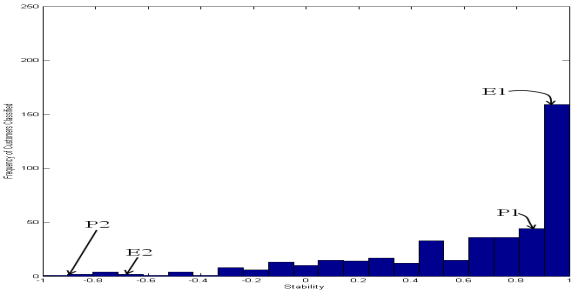
Static Naive Bayes Classifiers With Misclassified Customer Profiles Relabelled



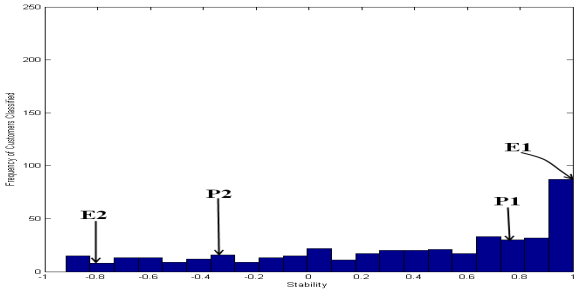
Adaptive Linear Regression Classifiers



Static Linear Regression Classifiers With Misclassified Customer Profiles Relabelled

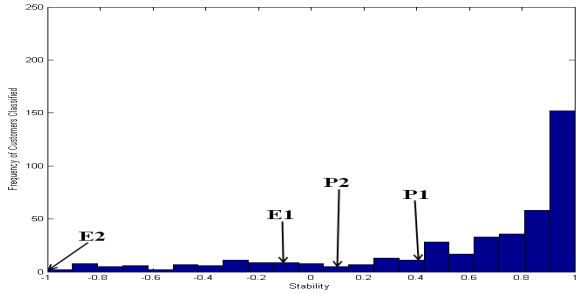


Adaptive SVM Classifiers

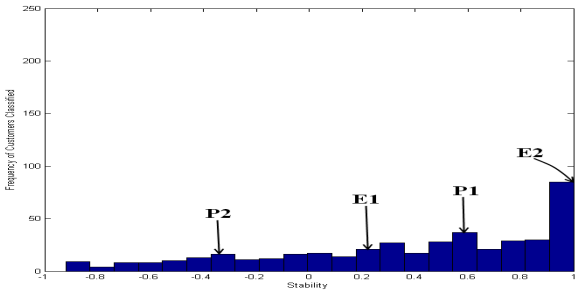


Static SVM Classifiers With Misclassified Customer Profiles Relabelled

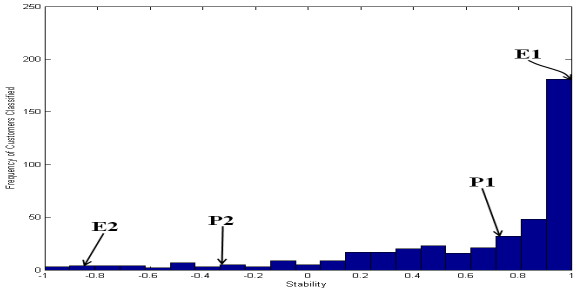
Figure J.1: Plots showing the comparative customer profiles classifications stability for the Decision Trees, Naive Bayes, Linear Regression and Support Vector Machines classifiers in the Three (3) Months Sliding Windows.



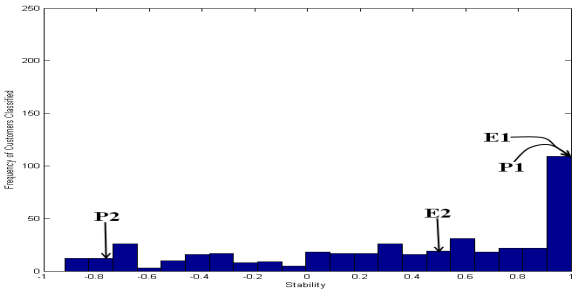
Adaptive Decision Tree Classifiers



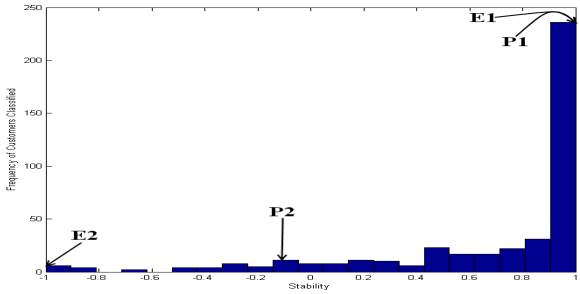
Static Decision Tree Classifiers With Misclassified Customer Profiles Relabelled



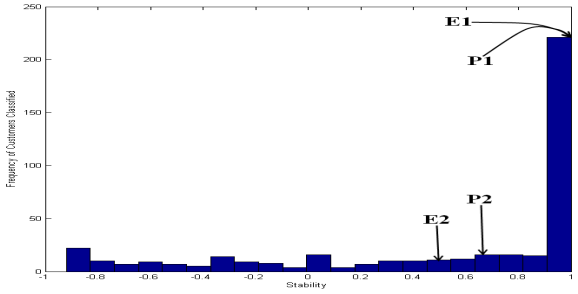
Adaptive Naive Bayes Classifiers



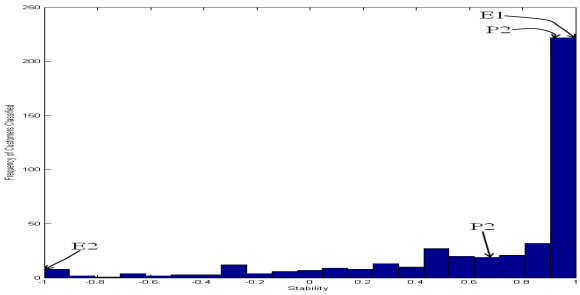
Static Naive Bayes Classifiers With Misclassified Customer Profiles Relabelled



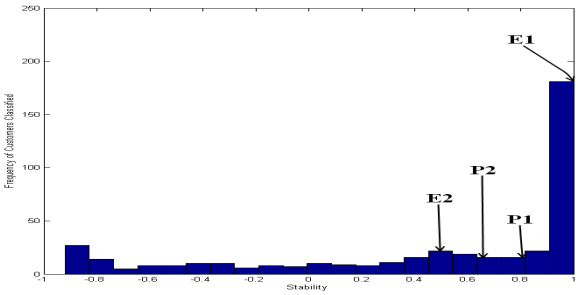
Adaptive Linear Regression Classifiers



Static Linear Regression Classifiers With Misclassified Customer Profiles Relabelled

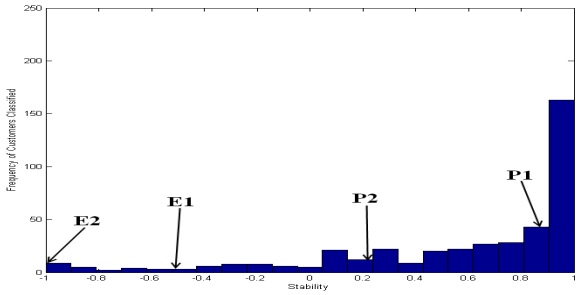


Adaptive SVM Classifiers

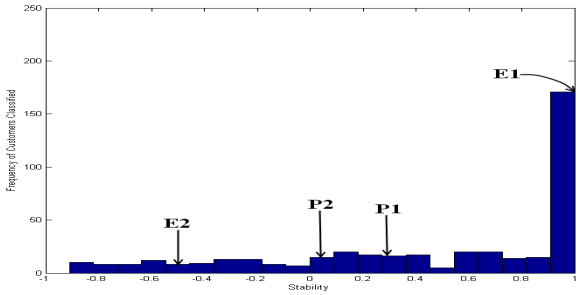


Static SVM Classifiers With Misclassified Customer Profiles Relabelled

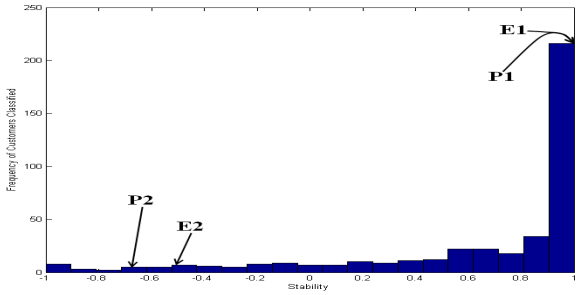
Figure J.2: Plots showing the comparative customer profiles classifications stability for the Decision Trees, Naive Bayes, Linear Regression and Support Vector Machines classifiers in the Three (3) Months Sliding Windows.



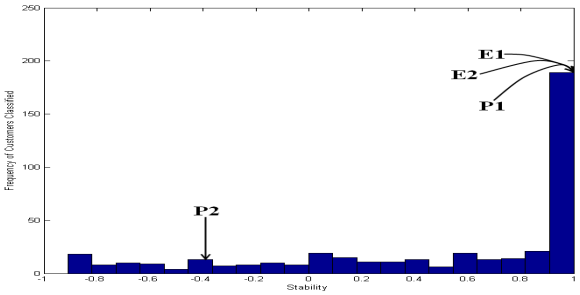
Adaptive Decision Tree Classifiers



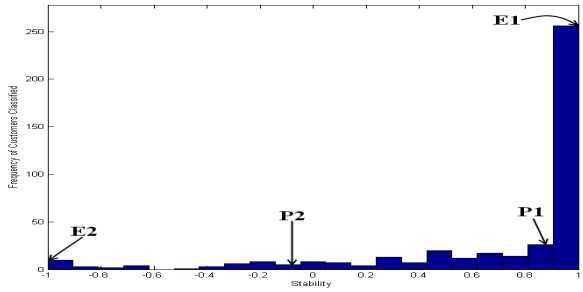
Static Decision Tree Classifiers With Misclassified Customer Profiles Relabelled



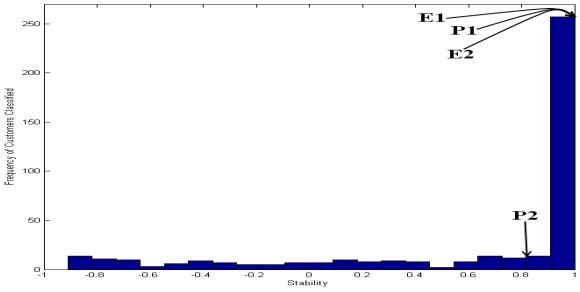
Adaptive Naive Bayes Classifiers



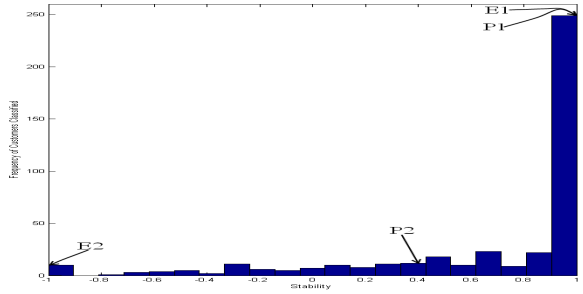
Static Naive Bayes Classifiers With Misclassified Customer Profiles Relabelled



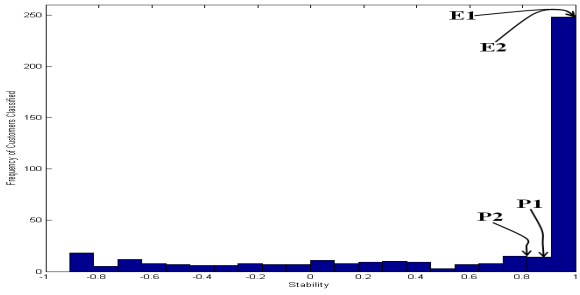
Adaptive Linear Regression Classifiers



Static Linear Regression Classifiers With Misclassified Customer Profiles Relabelled

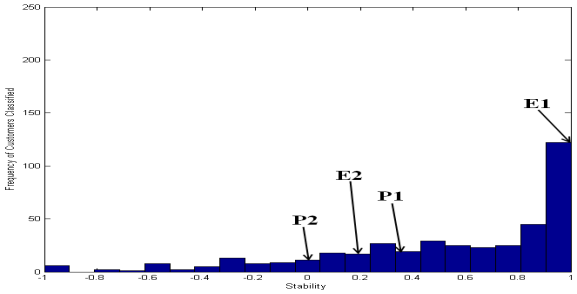


Adaptive SVM Classifiers

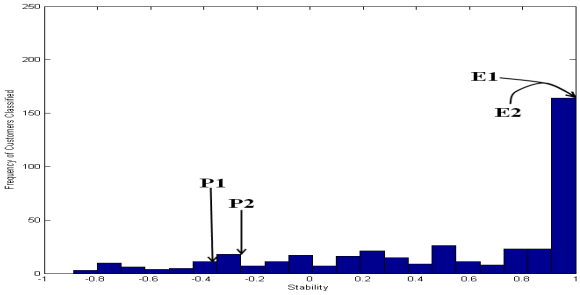


Static SVM Classifiers With Misclassified Customer Profiles Relabelled

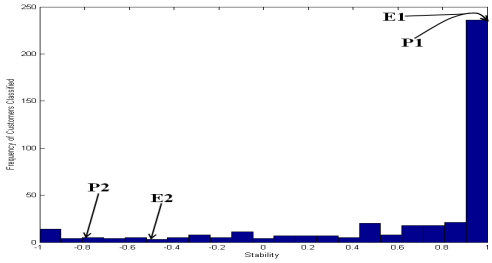
Figure J.3: Plots showing the comparative customer profiles classifications stability for the Decision Trees, Naive Bayes, Linear Regression and Support Vector Machines classifiers in the Three (3) Months Sliding Windows.



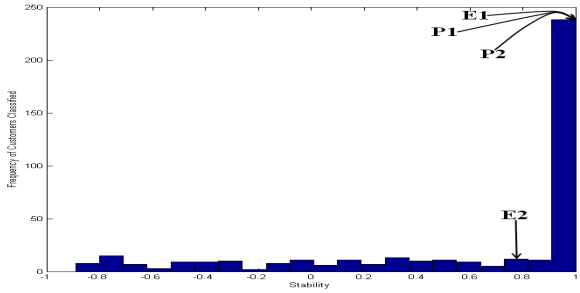
Adaptive Decision Tree Classifiers



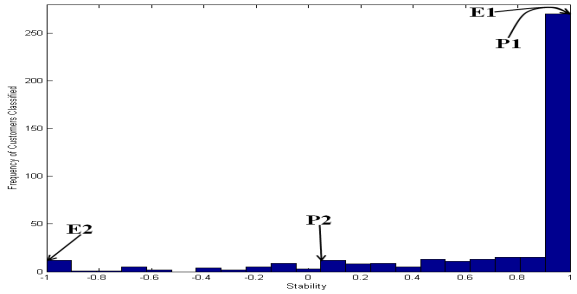
Static Decision Tree Classifiers With Misclassified Customer Profiles Relabelled



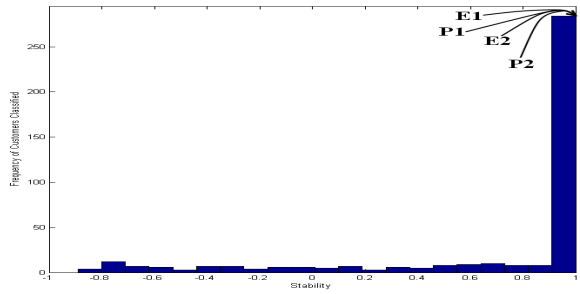
Adaptive Naive Bayes Classifiers



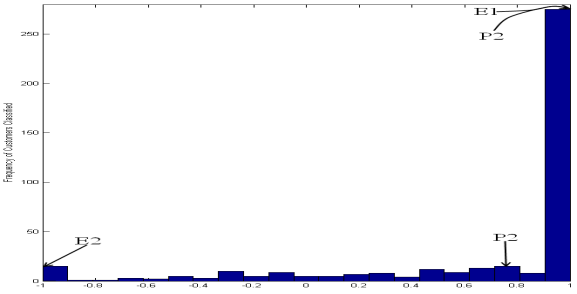
Static Naive Bayes Classifiers With Misclassified Customer Profiles Relabelled



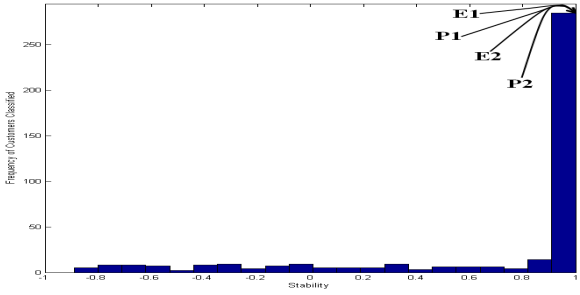
Adaptive Linear Regression Classifiers



Static Linear Regression Classifiers With Misclassified Customer Profiles Relabelled



Adaptive SVM Classifiers



Static SVM Classifiers With Misclassified Customer Profiles Relabelled

Figure J.4: Plots showing the comparative stability of customer profiles classifications for the Decision Trees, Naive Bayes, Linear Regression and Support Vector Machines classifiers in the Twelve (3) Months Sliding Windows.

Appendix K

Tables Illustrating the Changing Classifications of an Electrician and a PlumberHeater by Decision Trees (J48), Naive Bayes, and Support Vector Machines for the 3, 6, 9 and 12 Months Sliding Windows.

	Electrician (E)								PlumbHeater (P)							
Sliding Window Dataset Partitions	Class in 3 Months Window	Class in 6 Months Window	Class in 9 Months Window	Class in 12 months Window	Majority Vote Across Windows	Weighted Majority Vote Across Windows	Weighted Average Across Windows	Minority Vote Across Windows	Class in 3 Months Window	Class in 6 Months Window	Class in 9 Months Window	Class in 12 months Window	Majority Vote Across Windows	Weighted Majority Vote Across Windows	Weighted Average Across Windows	Minority Vote Across Windows
1	P				P	P	P	-	P				P	P	P	-
2	E				E	E	E	-	P				P	P	P	-
3	P				P	P	P	-	P				P	P	P	-
4	P	P			P	P	P	-	P	P			P	P	P	-
5	P	P			P	P	P	-	P	E			P	E	P	E
6	E	P			E	E	P	E	P	E			P	E	P	P
7	E	P	P		P	P	P	E	P	P	P		P	P	P	-
8	E	P	-		P	E	P	P	P	P	P		P	P	P	-
9	E	E	P		E	E	P	P	P	P	P		P	P	P	-
10	P	E	P	-	P	P	P	E	P	P	P	E	P	P	P	E
11	P	E	P	-	P	P	P	E	P	P	P	E	P	P	P	E
12	P	P	P	-	P	P	P	-	P	P	P	E	P	P	P	E
13	P	P	P	-	P	P	P	-	E	P	P	E	E	P	P	E
14	P	P	P	-	P	P	P	-	E	E	P	P	P	P	P	P
15	P	P	P	-	P	P	P	-	P	E	P	E	P	E	P	E
16	P	P	P	-	P	P	P	-	P	E	P	P	P	P	P	E
17	E	P	P	-	P	P	P	E	P	E	P	P	P	P	P	E
18	E	E	P	-	E	E	P	P	P	E	P	P	P	P	P	E
19	E	E	P	-	E	E	P	P	P	P	P	P	P	P	P	-
20	E	E	P	E	E	E	P	P	P	P	P	P	P	P	P	-
21	E	E	E	E	E	E	E	-	P	P	P	P	P	P	P	-
22	E	E	P	E	E	E	P	P	P	P	P	P	P	P	P	-
23	E	E	E	E	E	E	E	-	P	P	P	E	P	P	P	E
24	P	P	E	E	E	E	P	P	P	P	P	P	P	P	P	-
25	E	P	E	E	E	E	P	P	P	P	P	P	P	P	P	-
26	E	P	P	E	P	E	P	P	P	P	P	P	P	P	P	-
27	E	E	P	E	E	E	P	P	P	P	E	P	P	P	P	E
28	P	E	E	E	E	E	P	P	P	P	P	P	P	P	P	-
Majority Vote in Windows	E	P	P	E	P				P	P	P	P	P			
Weighted Majority Vote in Windows	E	E	E	E		P			P	P	P	P		P		
Weighted Average in Windows	E	P	P	E			P		P	P	P	P			P	
Minority in Windows	P	E	E	-				E	E	E	E	E				P
Stability	0.07	-0.12	-0.52	1	-0.07	0.07	-0.79	-0.38	0.86	0.44	0.91	0.37	0.93	0.79	1	-0.69

(a) Adaptive Decision Tree Classifiers

	Electrician (E)								PlumbHeater (P)							
Sliding Window Dataset Partitions	Class in 3 Months Window	Class in 6 Months Window	Class in 9 Months Window	Class in 12 months Window	Majority Vote Across Windows	Weighted Majority Vote Across Windows	Weighted Average Across Windows	Minority Vote Across Windows	Class in 3 Months Window	Class in 6 Months Window	Class in 9 Months Window	Class in 12 months Window	Majority Vote Across Windows	Weighted Majority Vote Across Windows	Weighted Average Across Windows	Minority Vote Across Windows
1	P*				P	P	P	-	P				P	P	P	-
2	E*				E	E	E	-	P				P	P	P	-
3	E				E	E	E	-	P				P	P	P	-
4	E	P*			E	P	P	E	P	P			P	P	P	-
5	P*	P			P	P	P	-	P	P			P	P	P	-
6	P	P			P	P	P	-	P	E*			E	E	P	P
7	P	P	P*		P	P	P	-	P	P*	P		P	P	P	-
8	E*	P	-		P	E	P	P	P	P	P		P	P	P	-
9	E	P	P		P	P	P	E	P	P	P		P	P	P	-
10	P*	P	P	-	P	P	P	-	E*	P	P	E*	P	P	P	P
11	E*	P	P	-	P	P	P	E	P*	E*	P	P*	P	P	P	E
12	E	E*	P	-	E	E	P	P	P	E	P	P	P	P	P	E
13	E	E	P	-	E	E	P	P	P	E	P	P	P	P	P	E
14	P*	E	P	-	P	P	P	E	P	P*	E*	P	P	P	P	E
15	P	E	P	-	P	P	P	E	P	P	E	P	P	P	P	E
16	P	E	P	-	P	P	P	E	P	P	E	P	P	P	P	E
17	E*	E	P	-	E	E	P	P	P	P	E	E*	E	E	P	P
18	E	P*	P	-	P	P	P	E	P	P	E	E	E	E	P	P
19	E	P	P	-	P	P	P	E	P	P	E	E	P	E	P	P
20	E	P	P	E	E	P	P	P	P	E*	E	P*	P	P	P	P
21	E	P	P	E	P	P	P	P	P	P*	E	P	P	P	P	E
22	E	E*	P	E	E	E	P	P	P	P	P*	P	P	P	P	-
23	E	E	P	E	E	E	P	P	P	P	P	E*	P	P	P	E
24	E	E	P	E	E	E	P	P	P	P	P	E	P	P	P	E
25	E	P*	P	E	E	P	P	P	P	P	P	P*	P	P	P	-
26	P*	E*	P	E	E	P	P	P	P	P	P	P	P	P	P	-
27	P	P*	P	E	P	P	P	E	P	P	P	P	P	P	P	-
28	P	P	P	E	P	P	P	E	P	P	P	P	P	P	P	-
Majority Vote in Windows	E	E	E	E	E				P	P	P	P	P			
Weighted Majority Vote in Windows	E	E	E	E		E			P	P	P	P		P		
Weighted Average in Windows	P	P	P	P			P		P	P	P	P			P	
Minority in Windows	P	P	P	P				P	E	E	-	-				E
Stability	0.36	0.44	0.9	1	0.14	0.36	0.86	0.05	0.93	0.52	0.82	-0.37	0.79	0.71	1	-0.20

(b) Static Decision Tree Classifiers With Misclassified Customer Profiles Relabelled (∗ Indicates point of relabeling)

Table K.1: Tables illustrating the comparative stability of the Electrician (E1) and Plumb-Heater(P1) customer profiles classifications for the Decision Tree Ensemble

	Electrician (E)								PlumbHeater (P)							
Sliding Window Dataset Partitions	Class in 3 Months Window	Class in 6 Months Window	Class in 9 Months Window	Class in 12 months Window	Majority Vote Across Windows	Weighted Majority Vote Across Windows	Weighted Average Across Windows	Minority Vote Across Windows	Class in 3 Months Window	Class in 6 Months Window	Class in 9 Months Window	Class in 12 months Window	Majority Vote Across Windows	Weighted Majority Vote Across Windows	Weighted Average Across Windows	Minority Vote Across Windows
1	E				E	E	E	-	P				P	P	P	-
2	E				E	E	E	-	E				E	E	E	-
3	E				E	E	E	-	P				P	P	P	-
4	P	E			E	E	P	E	E	P			P	P	P	E
5	P	E			P	E	P	P	P	P			P	P	P	-
6	E	E			E	E	E	-	P	P			P	P	P	-
7	E	E	E		E	E	E	-	P	P	P		P	P	P	-
8	E	E	-		E	E	E	-	P	P	P		P	P	P	-
9	E	E	E		E	E	E	-	P	E	P		P	P	P	E
10	E	E	E	-	E	E	E	-	P	E	P	P	P	P	P	E
11	E	E	E	-	E	E	E	-	P	P	P	P	P	P	P	-
12	E	E	E	-	E	E	E	-	P	P	P	P	P	P	P	-
13	E	E	E	-	E	E	E	-	P	P	P	P	P	P	P	-
14	P	E	E	-	E	E	P	P	P	E	P	P	P	P	P	E
15	E	E	E	-	E	E	E	-	P	P	P	P	P	P	P	-
16	E	E	E	-	E	E	E	-	P	P	P	P	P	P	P	-
17	E	E	E	-	E	E	E	-	P	P	P	P	P	P	P	-
18	E	E	E	-	E	E	E	-	P	P	P	P	P	P	P	-
19	E	E	E	-	E	E	E	-	P	P	P	P	P	P	P	-
20	E	E	E	E	E	E	E	-	P	P	P	P	P	P	P	-
21	E	E	E	E	E	E	E	-	P	P	P	P	P	P	P	-
22	E	E	E	E	E	E	E	-	P	P	P	P	P	P	P	-
23	E	E	E	E	E	E	E	-	P	P	P	P	P	P	P	-
24	E	E	E	E	E	E	E	-	P	P	P	P	P	P	P	-
25	E	E	E	E	E	E	E	-	P	P	P	P	P	P	P	-
26	E	E	E	E	E	E	E	-	P	P	P	P	P	P	P	-
27	E	E	E	E	E	E	E	-	P	P	P	P	P	P	P	-
28	E	E	E	E	E	E	E	-	P	P	P	P	P	P	P	-
Majority Vote in Windows	E	E	E	E	E				P	P	P	P	P			
Weighted Majority Vote in Windows	E	E	E	E		E			P	P	P	P		P		
Weighted Average in Windows	E	E	E	E			E		P	P	P	P			P	
Minority in Windows	P	-	-	-				E	E	E	-	-				-
Stability	0.79	1	1	1	0.93	1	0.79	-0.33	0.86	0.76	1	1	0.93	0.93	0.93	-1

(a) Adaptive Naive Bayes Classifiers

	Electrician (E)								PlumbHeater (P)							
Sliding Window Dataset Partitions	Class in 3 Months Window	Class in 6 Months Window	Class in 9 Months Window	Class in 12 months Window	Majority Vote Across Windows	Weighted Majority Vote Across Windows	Weighted Average Across Windows	Minority Vote Across Windows	Class in 3 Months Window	Class in 6 Months Window	Class in 9 Months Window	Class in 12 months Window	Majority Vote Across Windows	Weighted Majority Vote Across Windows	Weighted Average Across Windows	Minority Vote Across Windows
1	E				E	E	E	-	P				P	P	P	-
2	E				E	E	E	-	E*				E	E	E	-
3	E				E	E	E	-	P*				P	P	P	-
4	P*	E			E	E	P	E	E*	P			P	P	P	E
5	P	E			P	E	P	P	P*	P			P	P	P	-
6	P	E			E	E	P	P	P	P			P	P	P	-
7	E*	E	E		E	E	E	-	P	P	P		P	P	P	-
8	E	E	-		E	E	E	-	P	P	P		P	P	P	-
9	E	E	E		E	E	E	-	P	P	P		P	P	P	-
10	E	E	E	-	E	E	E	-	P	P	P	P	P	P	P	-
11	E	E	E	-	E	E	E	-	P	P	P	P	P	P	P	-
12	E	E	E	-	E	E	E	-	P	P	P	P	P	P	P	-
13	E	E	E	-	E	E	E	-	P	P	P	P	P	P	P	-
14	P*	E	E	-	E	E	P	P	P	P	P	P	P	P	P	-
15	E*	E	E	-	E	E	E	-	P	P	P	P	P	P	P	-
16	E	E	E	-	E	E	E	-	P	P	P	P	P	P	P	-
17	E	E	E	-	E	E	E	-	P	P	P	P	P	P	P	-
18	E	E	E	-	E	E	E	-	P	P	P	P	P	P	P	-
19	P*	E	E	-	E	E	P	P	P	P	P	P	P	P	P	-
20	P	E	E	E	E	E	P	P	P	P	P	P	P	P	P	-
21	P	E	E	E	E	E	P	P	P	P	P	P	P	P	P	-
22	P	E	E	E	E	E	P	P	P	P	P	P	P	P	P	-
23	E*	E	E	E	E	E	E	-	P	P	P	P	P	P	P	-
24	P*	E	E	E	E	E	P	P	P	P	P	P	P	P	P	-
25	E*	E	E	E	E	E	E	-	P	P	P	P	P	P	P	-
26	E	E	E	E	E	E	E	-	P	P	P	P	P	P	P	-
27	E	E	E	E	E	E	E	-	P	P	P	P	P	P	P	-
28	E	E	E	E	E	E	E	-	P	P	P	P	P	P	P	-
Majority Vote in Windows	E	E	E	E	E				P	P	P	P	P			
Weighted Majority Vote in Windows	E	E	E	E		E			P	P	P	P		P		
Weighted Average in Windows	P	P	P	P			P		P	P	P	P			P	
Minority in Windows	P	P	P	P				P	E	E	-	-				E
Stability	0.43	1	1	1	0.93	1	0.36	-0.78	0.71	1	1	1	0.93	0.93	0.93	-1.00

(b) Static Naive Bayes Classifiers With Misclassified Customer Profiles Relabelled (∗ Indicates point of relabeling)

Table K.2: Tables illustrating the comparative stability of the Electrician (E1) and PlumbHeater(P1) customer profiles classifications for the Naive Bayes Ensemble

	Electrician (E)								PlumbHeater (P)							
Sliding Window Dataset Partitions	Class in 3 Months Window	Class in 6 Months Window	Class in 9 Months Window	Class in 12 months Window	Majority Vote Across Windows	Weighted Majority Vote Across Windows	Weighted Average Across Windows	Minority Vote Across Windows	Class in 3 Months Window	Class in 6 Months Window	Class in 9 Months Window	Class in 12 months Window	Majority Vote Across Windows	Weighted Majority Vote Across Windows	Weighted Average Across Windows	Minority Vote Across Windows
1	E				E	E	E	-	P				P	P	P	-
2	E				E	E	E	-	P				P	P	P	-
3	E				E	E	E	-	P				P	P	P	-
4	E	E			E	E	E	-	E*	P			P	P	P	E
5	E	E			E	E	E	-	P*	P			P	P	P	-
6	E	E			E	E	E	-	P	P			P	P	P	-
7	E	E	E		E	E	E	-	P	P	P		P	P	P	-
8	E	E	0		E	E	E	-	P	P	P		P	P	P	-
9	E	E	E		E	E	E	-	P	E*	E*		E	E	P	P
10	E	E	E	0	E	E	E	-	E*	E	P*	P	P	P	P	E
11	E	E	E	0	E	E	E	-	E	P*	P	P	P	P	P	E
12	E	E	E	0	E	E	E	-	P*	P	P	P	P	P	P	-
13	E	E	E	0	E	E	E	-	P	P	P	P	P	P	P	-
14	E	E	E	0	E	E	E	-	P	P	P	P	P	P	P	-
15	E	E	E	0	E	E	E	-	P	P	P	P	P	P	P	-
16	E	E	E	0	E	E	E	-	P	P	P	P	P	P	P	-
17	E	E	E	0	E	E	E	-	P	P	P	P	P	P	P	-
18	E	E	E	0	E	E	E	-	P	P	P	P	P	P	P	-
19	E	E	E	0	E	E	E	-	P	P	P	P	P	P	P	-
20	E	E	E	E	E	E	E	-	P	P	P	P	P	P	P	-
21	E	E	E	E	E	E	E	-	P	P	P	P	P	P	P	-
22	E	E	E	E	E	E	E	-	P	P	P	P	P	P	P	-
23	E	E	E	E	E	E	E	-	P	P	P	P	P	P	P	-
24	E	E	E	E	E	E	E	-	P	P	P	P	P	P	P	-
25	E	E	E	E	E	E	E	-	P	P	P	P	P	P	P	-
26	E	E	E	E	E	E	E	-	P	P	P	P	P	P	P	-
27	E	E	E	E	E	E	E	-	P	P	P	P	P	P	P	-
28	E	E	E	E	E	E	E	-	P	P	P	P	P	P	P	-
Majority Vote in Windows	E	E	E	E	E				P	P	P	P	P			
Weighted Majority Vote in Windows	E	E	E	E		E			P	P	P	P		P		
Weighted Average in Windows	E	E	E	E			E		P	P	P	P			P	
Minority in Windows	-	-	-	-				-	E	E	E	-				P
Stability	1	1	1	1	1	1	1	0	0.71	0.84	0.82	1	0.93	0.93	1	-0.5

(a) Adaptive SVM Classifiers

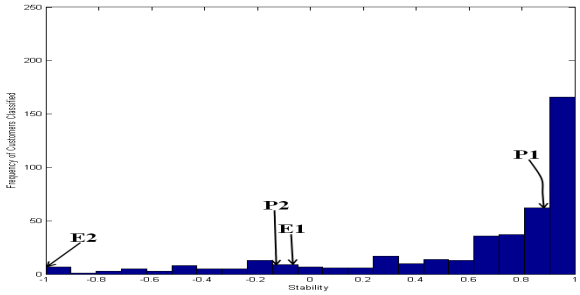
	Electrician (E)								PlumbHeater (P)							
Sliding Window Dataset Partitions	Class in 3 Months Window	Class in 6 Months Window	Class in 9 Months Window	Class in 12 months Window	Majority Vote Across Windows	Weighted Majority Vote Across Windows	Weighted Average Across Windows	Minority Vote Across Windows	Class in 3 Months Window	Class in 6 Months Window	Class in 9 Months Window	Class in 12 months Window	Majority Vote Across Windows	Weighted Majority Vote Across Windows	Weighted Average Across Windows	Minority Vote Across Windows
1	E				E	E	E	-	P				P	P	P	-
2	E				E	E	E	-	P				P	P	P	-
3	E				E	E	E	-	P				P	P	P	-
4	E	E			E	E	E	-	E	P			P	P	P	E
5	E	E			E	E	E	-	P	P			P	P	P	-
6	E	E			E	E	E	-	P	P			P	P	P	-
7	E	E	E		E	E	E	-	P	P	P		P	P	P	-
8	E	E	-		E	E	E	-	P	P	P		P	P	P	-
9	E	E	E		E	E	E	-	P	E	E		E	E	P	P
10	E	E	E	-	E	E	E	-	E	E	P	P	P	P	P	E
11	E	E	E	-	E	E	E	-	E	P	P	P	P	P	P	E
12	E	E	E	-	E	E	E	-	P	P	P	P	P	P	P	-
13	E	E	E	-	E	E	E	-	P	P	P	P	P	P	P	-
14	E	E	E	-	E	E	E	-	P	P	P	P	P	P	P	-
15	E	E	E	-	E	E	E	-	P	P	P	P	P	P	P	-
16	E	E	E	-	E	E	E	-	P	P	P	P	P	P	P	-
17	E	E	E	-	E	E	E	-	P	P	P	P	P	P	P	-
18	E	E	E	-	E	E	E	-	P	P	P	P	P	P	P	-
19	E	E	E	-	E	E	E	-	P	P	P	P	P	P	P	-
20	E	E	E	E	E	E	E	-	P	P	P	P	P	P	P	-
21	E	E	E	E	E	E	E	-	P	P	P	P	P	P	P	-
22	E	E	E	E	E	E	E	-	P	P	P	P	P	P	P	-
23	E	E	E	E	E	E	E	-	P	P	P	P	P	P	P	-
24	E	E	E	E	E	E	E	-	P	P	P	P	P	P	P	-
25	E	E	E	E	E	E	E	-	P	P	P	P	P	P	P	-
26	E	E	E	E	E	E	E	-	P	P	P	P	P	P	P	-
27	E	E	E	E	E	E	E	-	P	P	P	P	P	P	P	-
28	E	E	E	E	E	E	E	-	P	P	P	P	P	P	P	-
Majority Vote in Windows	E	E	E	E	E				P	P	P	P	P			
Weighted Majority Vote in Windows	E	E	E	E		E			P	P	P	P		P		
Weighted Average in Windows	E	E	E	E			E		P	P	P	P			P	
Minority in Windows	-	-	-	-				-	E	E	E	-				P
Stability	1	1	1	1	1	1	1	0	0.79	0.84	0.91	1	0.93	0.93	1	-0.5

(b) Static SVM Classifiers With Misclassified Customer Profiles Relabelled

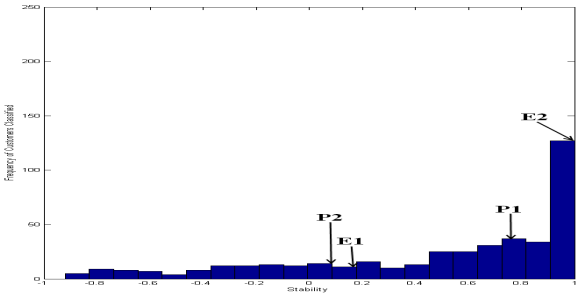
Table K.3: Tables illustrating the comparative stability of the Electrician (E1) and PlumbHeater(P1) customer profiles classifications for the Support Vector Machine Ensemble

Appendix L

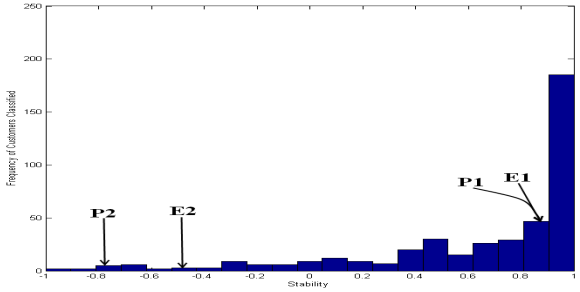
Figures Illustrating the Changing Stability of Two Electricians and Two PlumberHeaters obtained from using the majority, weighted majority, weighted average majority and minority voting combiners for Decision Trees (J48), Naive Bayes, and Support Vector Machines in the 3, 6, 9 and 12 Months Sliding Windows.



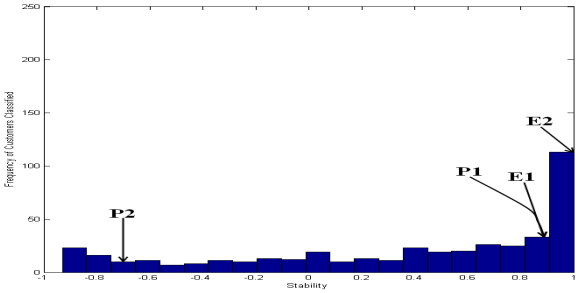
Decision Tree Adaptation Majority Voting Stability



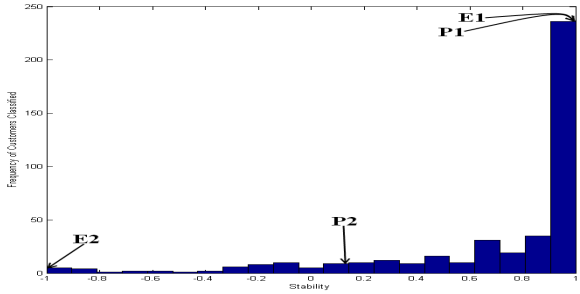
Decision Tree Relabeling Majority Voting Stability



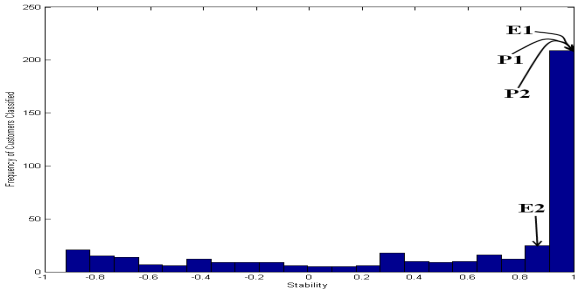
Naive Bayes Adaptation Majority Voting Stability



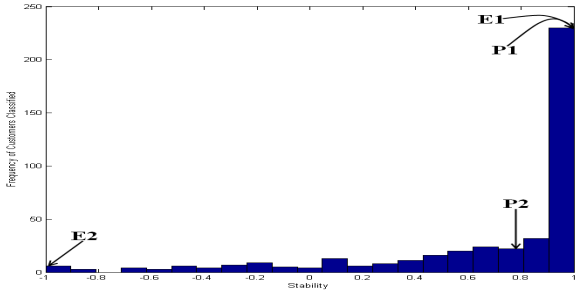
Naive Bayes Relabeling Majority Voting Stability



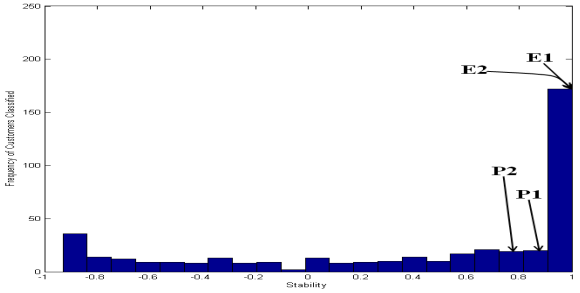
Linear Regression Adaptation Majority Voting Stability



Linear Regression Relabeling Majority Voting Stability

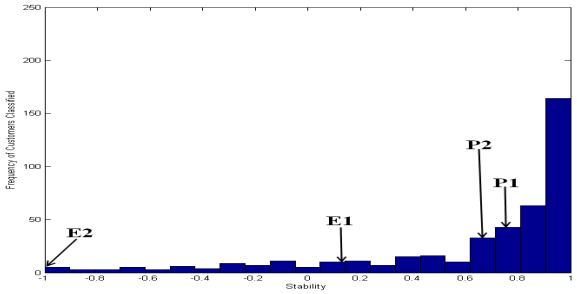


Support Vector Machine (SVM) Adaptation Majority Voting Stability

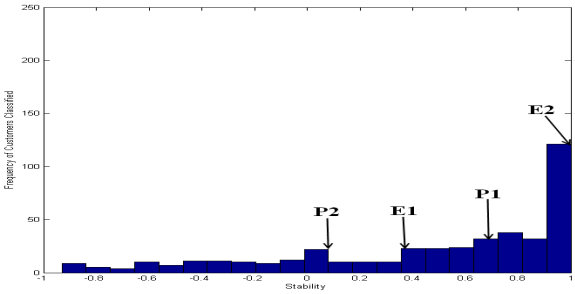


Support Vector Machine (SVM) Relabeling Majority Voting Stability

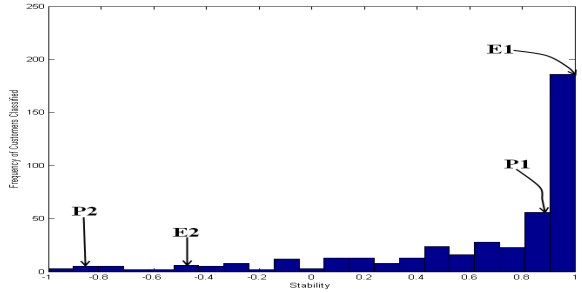
Figure L.1: Plots showing the stability of customer profiles classifications using the majority voting combiner for the Decision Trees, Naive Bayes, Linear Regression and Support Vector Machines Ensembles.



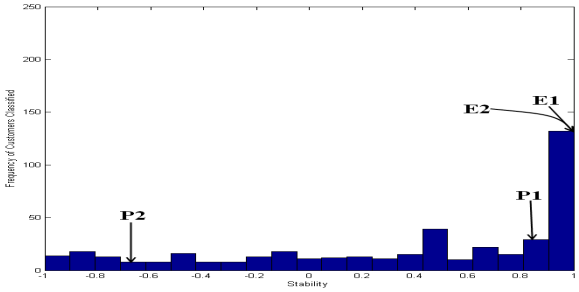
Decision Tree Adaptation Weighted Majority Voting Stability



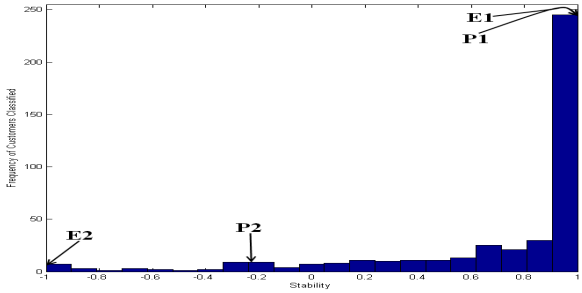
Decision Tree Relabeling Weighted Majority Voting Stability



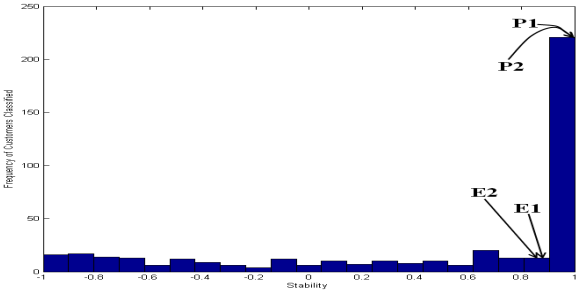
Naive Bayes Adaptation Weighted Majority Voting Stability



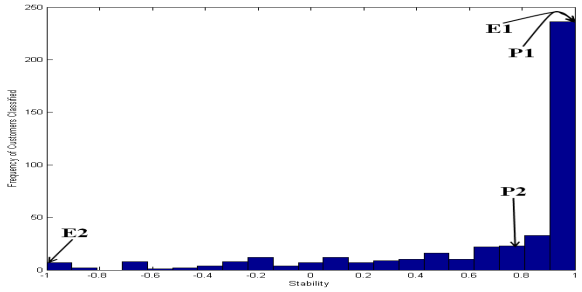
Naive Bayes Relabeling Weighted Majority Voting Stability



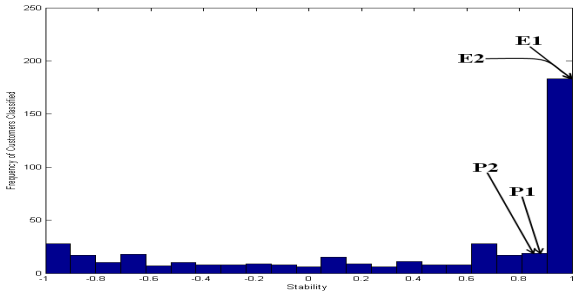
Linear Regression Adaptation Weighted Majority Voting Stability



Linear Regression Relabeling Weighted Majority Voting Stability

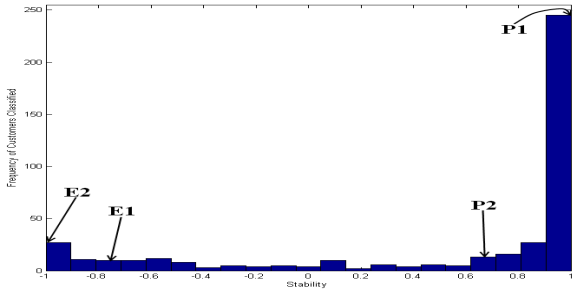


Support Vector Machine (SVM) Adaptation Weighted Majority Voting Stability

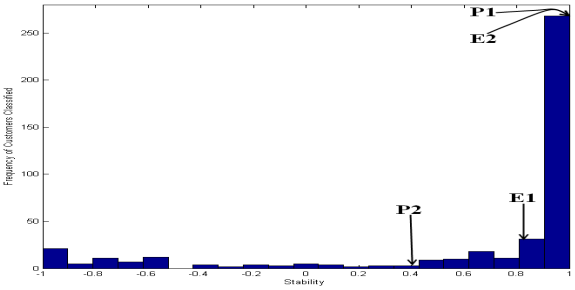


Support Vector Machine (SVM) Relabeling Weighted Majority Voting Stability

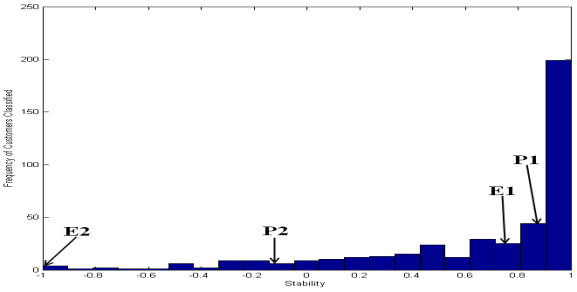
Figure L.2: Plots showing the stability of customer profiles classifications using the weighted majority voting combiner for the Decision Trees, Naive Bayes, Linear Regression and Support Vector Machines Ensembles.



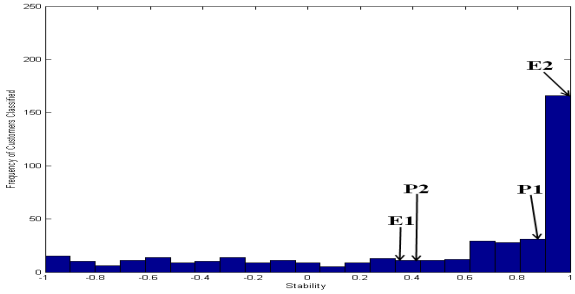
Decision Tree Adaptation Weighted Average Voting Stability



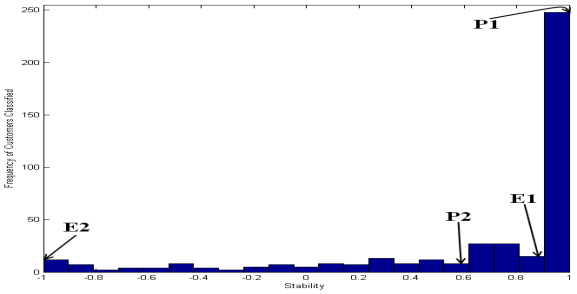
Decision Tree Relabeling Weighted Average Voting Stability



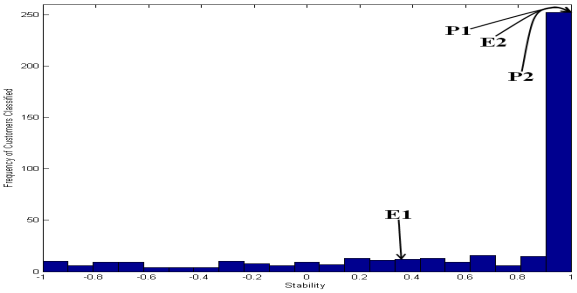
Naive Bayes



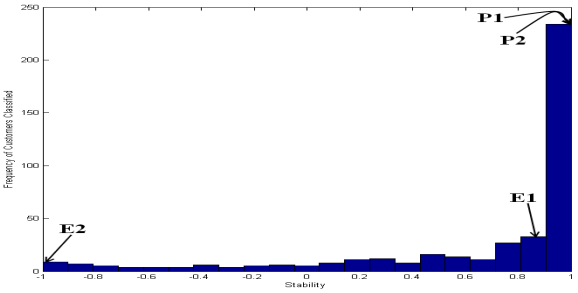
Naive Bayes Adaptation Weighted Average Voting Stability



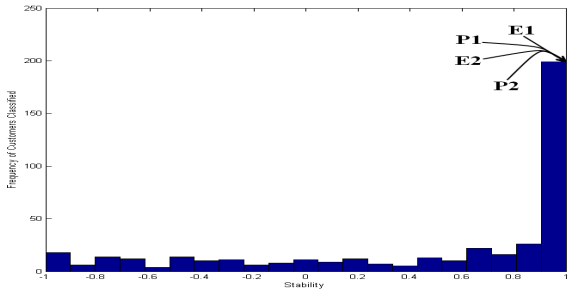
Linear Regression Adaptation Weighted Average Voting Stability



Linear Regression Relabeling Weighted Average Voting Stability

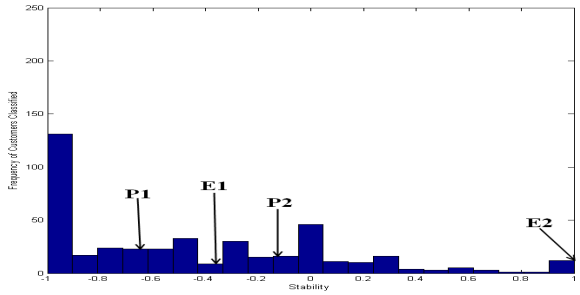


Support Vector Machine (SVM) Adaptation Weighted Average Voting Stability

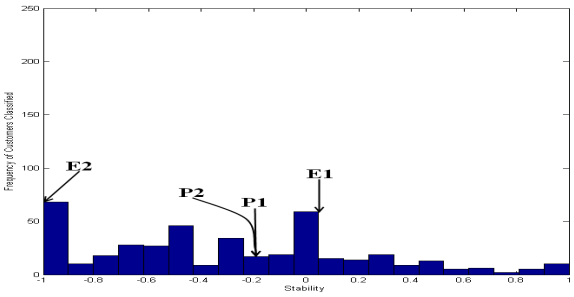


Support Vector Machine (SVM) Relabeling Weighted Average Voting Stability

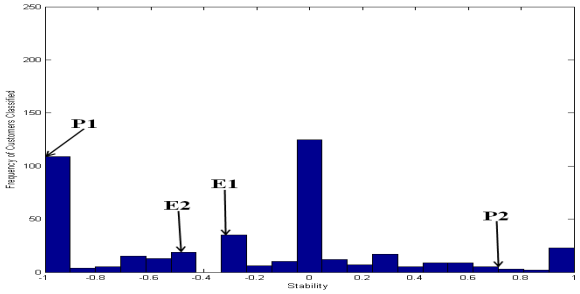
Figure L.3: Plots showing the stability of customer profiles classifications using the weighted average voting combiner for the Decision Trees, Naive Bayes, Linear Regression and Support Vector Machines Ensembles.



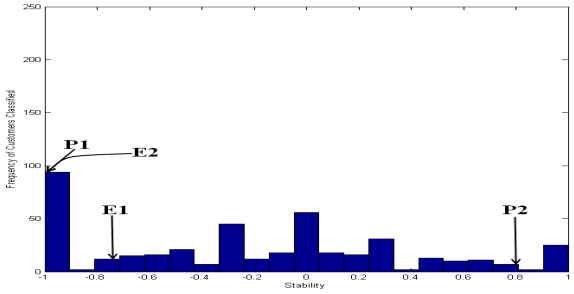
Decision Tree Adaptation Minority Voting Stability



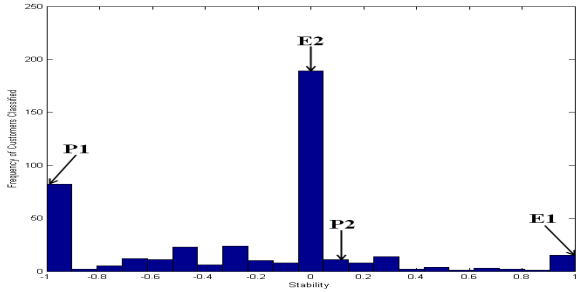
Decision Tree Relabeling Minority Voting Stability



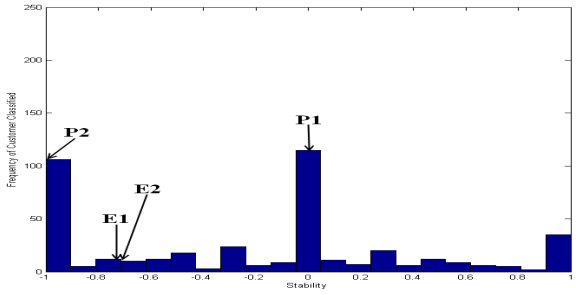
Naive Bayes Adaptation Minority Voting Stability



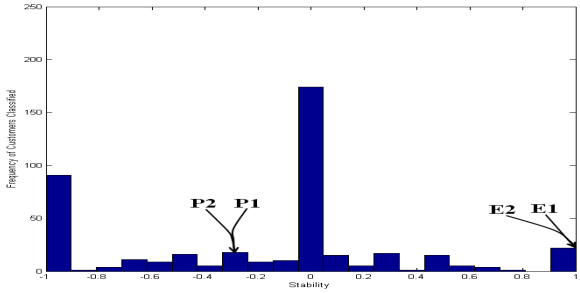
Naive Bayes Relabeling Minority Voting Stability



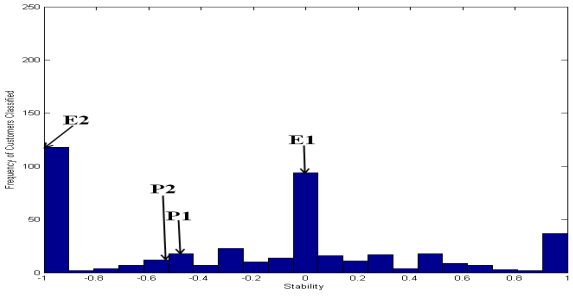
Linear Regression Adaptation Minority Voting Stability



Linear Regression Relabeling Minority Voting Stability



Support Vector Machine (SVM) Adaptation Minority Voting Stability



Support Vector Machine (SVM) Relabeling Minority Voting Stability

Figure L.4: Plots showing the stability of customer profiles classifications using the Minority Voting Combiner for the Decision Trees, Naive Bayes, Linear Regression and Support Vector Machines Ensembles.

References

P. Adriaans and D. Zantinge. *Data Mining*. Pearson Education, 1996.

R. C. Agarwal, F. G. Gustavson, and M. Zubair. A high performance algorithm using pre-processing for the sparse matrix-vector multiplication. In *Supercomputing '92: Proceedings of the 1992 ACM/IEEE conference on Supercomputing*, pages 32–41, Los Alamitos, CA, USA, 1992. IEEE Computer Society Press. ISBN 0-8186-2630-5.

C. Aggarwal, Z. Sun, and P. Yu. Fast algorithms for online generation of profile association rules. *Knowledge and Data Engineering, IEEE Transactions on*, 14(5):1017 – 1028, sep/oct 2002.

R. Agrawal and T. Imielinski. Mining association rules between sets of items in large databases. In *SIGMOD '93: Proceedings of the 1993 ACM SIGMOD international conference on Management of data*, pages 207–216, New York, NY, USA, 1993. ACM.

R. Agrawal and J. C. Shafer. Parallel mining of association rules. *IEEE Transactions on Knowledge and Data Engineering*, 8:962–969, 1996.

R. Agrawal and R. Srikant. Fast algorithms for mining association rules in large databases. In J. B. Bocca, M. Jarke, and C. Zaniolo, editors, *Proceedings of the 20th international Conference on Very Large Data Bases*, pages 487–499, San Francisco, CA, USA, September 1994. Morgan Kaufmann Publishers Inc.

S. Ahmed, F. Coenen, and P. Leng. A tree partitioning method for memory management in association rule mining. In *Data Warehousing and Knowledge Discovery, 6th International Conference*, 2004.

S. V. Allera and A. G. Horsburgh. Load profiling for energy trading and settlements in the uk electricity markets. In *Proceedings of DistribuTECH Europe DA/DSM Conference*, London, October 1998.

E. L. Allwein, R. E. Schapire, and Y. Singer. Reducing multiclass to binary: a unifying approach for margin classifiers. *J. Mach. Learn. Res.*, 1:113–141, September 2001. ISSN 1532-4435.

I. Ari, J. Li, J. Jain, and A. Kozlov. Management of data mining model lifecycle to support intelligent business services. Technical report, HP Software University Association Workshop, 2008a.

I. Ari, J. Li, A. Kozlov, and M. Dekhil. Data mining model management to support real-time business intelligence in service-oriented architectures. Technical report, HP Software University Association Workshop, 2008b.

D. S. Associates. *The New Direct Marketing: How to Implement A Profit-Driven Database Marketing Strategy*. McGraw-Hill, 3 edition, March 1999.

T. Bäck. *Evolutionary algorithms in theory and practice: evolution strategies, evolutionary programming, genetic algorithms*. Oxford University Press, Oxford, UK, 1996. ISBN 0-19-509971-0.

P. L. Bartlett, O. Bousquet, and S. Mendelson. Local rademacher complexities. In *The Annals of Statistics*, pages 44–58, 2002.

R. Battiti. Using mutual information for selecting features in supervised neural net learning. *Neural Networks, IEEE Transactions on*, 5(4):537 –550, jul 1994. ISSN 1045-9227.

M. Bazik and D. Feltes. Defining your customer profile - an essential tool. *Journal of Extension (JOE)*, 37(6), December 1999.

T. Bäck. Evolution strategies: an alternative evolutionary algorithm. In *Artificial Evolution*, pages 3–20. Springer-Verlag, 1995.

F. Bergadano, A. Giordana, and L. Saitta. Automated concept acquisition in noisy environments. *IEEE Trans. Pattern Anal. Mach. Intell.*, 10(4):555–578, 1988. ISSN 0162-8828. doi: http://dx.doi.org/10.1109/34.3917.

F. Bergadano, R. Gemello, A. Giordana, and L. Saitta. Ml-smart: A problem solver for learning from examples. *Fundamenta Informaticae*, 12:29–50, 1989.

M. J. A. Berry and G. S. Linoff. *Mastering Data Mining: The Art and Science of Customer Relationship Management*. John Wiley & Sons, January 2000.

M. J. A. Berry and G. S. Linoff. *Data Mining Techniques: For Marketing, Sales, and Customer Relationship Management*. John Wiley & Sons, April 2004.

J. C. Bezdek and L. I. Kuncheva. Nearest prototype classifier designs: An experimental study. *International Journal of Intelligent Systems*, 16(12):1445–1473, 2009.

J. Biethahn and V. Nissen. *Evolutionary Algorithms in Management Applications*. Springer-Verlag New York, Inc., Secaucus, NJ, USA, 1995. ISBN 3540603824.

A. Bifet and R. Gavaldà. Adaptive learning from evolving data streams. In *Proceedings of the 8th International Symposium on Intelligent Data Analysis: Advances in Intelligent Data Analysis VIII*, IDA '09, pages 249–260, Berlin, Heidelberg, 2009. Springer-Verlag. ISBN 978-3-642-03914-0.

C. M. Bishop. *Neural Networks for Pattern Recognition*. Oxford University Press, Inc., New York, NY, USA, 1995. ISBN 0198538642.

H. Blockeel and L. De Raedt. Top-down induction of first-order logical decision trees. *Artif. Intell.*, 101(1-2):285–297, 1998. ISSN 0004-3702. doi: http://dx.doi.org/10.1016/S0004-3702(98)00034-4.

H. Blockeel, M. Sebag, and M. E. Sebag. Scalability and efficiency in multi-relational data mining. *ACM SIGKDD Explorations Newsletter*, 5(1):17 – 30, July 2003.

A. Blumer, A. Ehrenfeucht, D. Haussler, and M. K. Warmuth. Occam's razor. *Information Processing Letters*, 24:377–380, April 1987. ISSN 0020-0190. doi: 10.1016/0020-0190(87)90114-1. URL http://dl.acm.org/citation.cfm?id=31168.31174.

A. Bordes, S. Ertekin, J. Weston, and L. Bottou. Fast kernel classifiers with online and active learning. *Journal of Machine Learning Research*, 6:1579–1619, 2005.

B. E. Boser, I. M. Guyon, and V. N. Vapnik. A training algorithm for optimal margin classifiers. In *Proceedings of the fifth annual workshop on Computational learning theory*, COLT '92, pages 144–152, New York, NY, USA, 1992. ACM.

M. Böttcher. Contrast and change mining. *Wiley Interdisc. Rew.: Data Mining and Knowledge Discovery*, 1(3):215–230, 2011.

M. Böttcher, F. Höppner, and M. Spiliopoulou. On exploiting the power of time in data mining. *SIGKDD Explor. Newsl.*, 10:3–11, December 2008. ISSN 1931-0145.

M. Böttcher, G. RuSS, D. Nauck, and R. Kruse. *Post-Mining of Association Rules: Techniques for Effective Knowledge Extraction*, chapter From Change Mining to Relevance Feedback: A Unified View on Assessing Rule Interestingness, pages 12–37. IGI Global, 2009.

M. Böttcher, M. Spott, D. Nauck, and R. Kruse. Mining changing customer segments in dynamic markets. *Expert Syst. Appl.*, 36:155–164, January 2009. ISSN 0957-4174.

M. Böttcher, D. Nauck, C. Borgelt, and R. Kruse. Temporal aspects in data mining. In *WCCI 2010 Plenary and Invited Lectures,*, pages 1–22. Institute of Electrical and Electronics Engineering, Inc., 2010.

R. J. Brachman and H. J. Levesque. *Knowledge Representation and Reasoning*. Morgan Kaufmann, 2004.

A. P. Bradley. The use of the area under the roc curve in the evaluation of machine learning algorithms. *Pattern Recogn.*, 30:1145–1159, July 1997. ISSN 0031-3203.

P. S. Bradley, U. M. Fayyad, and C. A. Reina. Scaling clustering algorithms to large databases. In *Proceedings of the 4th international Conf. on Knowledge Discovery and Data Mining (KDD98)*, pages 9–15, 1998.

M. Bramer. *Principles of Data Mining*. Springer, London, 2007.

S. Brin and L. Page. The anatomy of a large-scale hypertextual web search engine. In *Proceedings of WWW*, pages 107–117, Brisbane, Australia, 1998.

C. E. Brodley. Addressing the selective superiority problem: Automatic algorithm/model class selection. In *Proc 10th Machine Learning Conf.*, pages 17–24, 1993.

C. E. Brodley and M. A. Friedl. Identifying mislabeled training data. *Journal of Artificial Intelligence Research*, 11:131–167, 1999.

M. Budka and B. Gabrys. Density preserving sampling (dps) for error estimation and model selection. In *To appear In the proceedings of IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2010.

C. J. C. Burges. A tutorial on support vector machines for pattern recognition. *Data Min. Knowl. Discov.*, 2(2):121–167, 1998. ISSN 1384-5810. doi: http://dx.doi.org/10.1023/A:1009715923555.

M. M. Campos, P. J. Stengard, and B. L. Milenova. Data-centric automated data mining. In *ICMLA*, 2005.

G. Castillo and J. Gama. Bias management of bayesian networks classifiers. In *Proceedings of the 8th International Conference of Discovery Science*, volume 3735 of LNAI, pages 70–83. Springer Verlag, 2005.

J. Catlett. On changing continuous attributes into ordered discrete attributes. In *Proceedings of the European working session on learning on Machine learning*, pages 164–178, New York, NY, USA, 1991. Springer-Verlag New York, Inc. ISBN 0-387-53816-X.

S. Chalup and F. Maire. A study on hill climbing algorithms for neural network training. pages 2014–2021, 1999.

P. K. Chan, W. Fan, A. L. Prodromidis, and S. J. Stolfo. Distributed data mining in credit card fraud detection. *IEEE Intelligent Systems*, 14:67–84, 1999.

C.-C. Chang, Y.-C. Li, and J.-S. Lee. An efficient algorithm for incremental mining of association rules. In *Proceedings of the 15th International Workshop on Research Issues in Data Engineering: Stream Data Mining and Applications*, RIDE '05, pages 3–10, Washington, DC, USA, 2005. IEEE Computer Society. ISBN 0-7695-2390-0. doi: http://dx.doi.org/10.1109/RIDE.2005.6. URL http://dx.doi.org/10.1109/RIDE.2005.6.

N. Chawla. *Data Mining for Imbalanced Datasets: An Overview*, pages 853–867. Springer US, 2005.

V. Cherkassky and F. Mulier. *Learning from Data: Concepts, Theory, and Methods (Adaptive and Learning Systems for Signal Processing, Communications and Control Series)*. Wiley-Interscience, March 1998.

H. Chernoff. A measure of asymptotic efficiency for tests of a hypothesis based on the sum of observations. *The Annals of Mathematical Statistics*, 23:493–507, 1952.

D. W. Cheung, V. T. Ng, and B. W. Tam. Maintenance of discovered knowledge: A case in multi-level association rules. In *In Proc. of 2nd Intl. Conf. on Knowledge Discovery and Data Mining (KDD*, pages 307–310. AAAI Press, 1996.

D. W.-L. Cheung, S. D. Lee, and B. Kao. A general incremental technique for maintaining discovered association rules. In *Proceedings of the Fifth International Conference on Database Systems for Advanced Applications (DASFAA)*, pages 185–194. World Scientific Press, 1997. ISBN 981-02-3107-5. URL http://portal.acm.org/citation.cfm?id=646711.703155.

G. Chicco, R. Napoli, P. Postolache, M. Scutariu, and C. Toader. Electric energy customer characterisation for developing dedicated market strategies. In *Power Tech Proceedings, 2001 IEEE Porto*, page 6, 2001.

G. Chicco, R. Napoli, F. Piglione, P. Postolache, M. Scutariu, and C. Toader. Load pattern-based classification of electricity customers. *Power Systems, IEEE Transactions on*, 19(2):1232 – 1239, may 2004.

D. K. Y. Chiu, A. K. C. Wong, and K. C. C. Chan. Synthesis of statistical knowledge from time-dependent data. *IEEE Trans. Pattern Anal. Mach. Intell.*, 13:265–271, March 1991.

M. R. Chmielewski and J. W. Grzymala-Busse. Global discretization of continuous attributes as preprocessing for machine learning. In *Proceedings of the Third International Workshop on Rough Sets and Soft Computing*, pages 474–480, San Jose, CA, November 10-12 1995.

K. W. Church, P. Li, and T. J. Hastie. Conditional random sampling: A sketch-based sampling technique for sparse data. In *In NIPS*, pages 873–880, 2006.

E. Cox. *Fuzzy Modeling and Genetic Algorithms for Data Mining and Exploration*. Morgan Kaufmann, 2005.

K. Crammer and Y. Singer. On the learnability and design of output codes for multiclass problems. *Machine Learning*, 47:201–233, May 2002. ISSN 0885-6125.

CRISP-DM.org and N. Leaper. Crisp-dm 1.0, 2009. URL http://exde.wordpress.com/2009/03/13/a-visual-guide-to-crisp-dm-methodology/.

N. Cristianini and J. Shawe-Taylor. *An introduction to support Vector Machines: and other kernel-based learning methods*. Cambridge University Press, New York, NY, USA, 2000. ISBN 0-521-78019-5.

A. P. Dawid. Present position and potential developments: some personal views. statistical theory. the prequential approach (with discussion). *Statistical Analysis and Data Mining*, 95(4):277–305, 2008.

I. S. Dhillon, I. S. Dhillon, D. S. Modha, and D. S. Modha. Concept decompositions for large sparse text data using clustering. In *Machine Learning*, pages 143–175, 1999.

T. G. Dietterich and G. Bakiri. Solving multiclass learning problems via error-correcting output codes. *Journal of Artificial Intelligence Research*, 2:263–286, 1995.

A. J. Dobson and A. Barnett. *An Introduction to Generalized Linear Models*. Chapman & Hall/CRC Texts in Statistical Science. Chapman and Hall/CRC, third edition edition, 2008.

P. Domingos and G. Hulten. Mining high-speed data streams. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '00, pages 71–80, New York, NY, USA, 2000. ACM. ISBN 1-58113-233-6.

M. Dorigo and G. Di Caro. *The ant colony optimization meta-heuristic*. McGraw-Hill Ltd., UK, Maidenhead, UK, England, 1999. ISBN 0-07-709506-5.

A. Dries and U. Rückert. Adaptive concept drift detection. *Statistical Analysis Data Mining*, 2(5-6):311–327, 2009.

R. O. Duda, P. E. Hart, and D. G. Stork. *Pattern Classification (2nd Edition)*. Wiley-Interscience, 2000. ISBN 0471056693.

R. Duin, P. Juszczak, P. Paclik, E. Pekalska, D. de Ridder, D. Tax, and S. Verzakov. Prtools 4.1, a matlab toolbox for pattern recognition. Technical report, Delft University of Technology, 2007.

S. Džroski. *Relational data mining applications: an overview*, pages 339–360. Springer-Verlag New York, Inc., New York, NY, USA, 2000. ISBN 3-540-42289-7.

S. Dzeroski and N. Lavrac. *Relational Data Mining*. Springer, August 2001.

M. Ester, H.-P. Kriegel, and J. Sander. *Geographic data mining and knowledge discovery*, chapter Algorithms and Applications for Spatial Data Mining. CRC, 1 edition, October 2001.

B. S. Everitt, S. Landau, M. Leese, and D. Stahl. *Cluster Analysis*. Hodder Arnold, 5th edition edition, 2011.

U. M. Fayyad. *Advances in Knowledge Discovery and Data Mining*. MIT Press, March 1996.

F. Ferri, J. Albert, and E. Vidal. Considerations about sample-size sensitivity of a family of edited nearest-neighbor rules. *Systems, Man, and Cybernetics, Part B: Cybernetics, IEEE Transactions on*, 29(5):667 –672, oct 1999.

P. A. Flach. Knowledge representation for inductive learning. In A. Hunter and S. Parsons, editors, *Symbolic and Quantitative Approaches to Reasoning and Uncertainty (ECSQARU'99), volume of*, volume 1638 of *Lecture Notes in Artificial Intelligence*, pages 160–167. Springer-Verlag, July 1999.

S. Floyd and M. Warmuth. Sample compression, learnability, and the vapnik-chervonenkis dimension. In *Machine Learning*, pages 269–304, 1993.

D. P. Foster and R. A. Stine. Variable selection in data mining: Building predictive model. *Journal of the American Statistical Association*, 99:303–313, 2004.

E. Frank, Y. Wang, S. Inglis, G. Holmes, and I. H. Witten. Using model trees for classification. *Mach. Learn.*, 32:63–76, July 1998.

J. H. Friedman. Multivariate adaptive regression splines. *The Annals of Statistics*, 19(1): 1–67, 1991.

J. H. Friedman. Another approach to polychotomous classification. Technical report, Department of Statistics, Stanford University, 1996.

E. J. Friedman-Hill. Jess, the java expert system shell. Technical report, Sandia National Laboratories, October 1997.

J. Fürnkranz. Separate-and-conquer rule learning. *Artif. Intell. Rev.*, 13(1):3–54, Feb. 1999.

J. a. Gama, R. Sebastião, and P. P. Rodrigues. Issues in evaluation of stream learning algorithms. In *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '09, pages 329–338, New York, NY, USA, 2009. ACM. ISBN 978-1-60558-495-9.

V. Ganti, J. Gehrke, and R. Ramakrishnan. Mining very large databases. *Computer*, 32 (8):38–45, 1999.

R. Gemulla. *Sampling Algorithms for Evolving Datasets*. PhD thesis, Technische Universität Dresden, 2008.

R. Gemulla and W. Lehner. Sampling time-based sliding windows in bounded space. In *Proceedings of the 2008 ACM SIGMOD international conference on Management of data*, SIGMOD '08, pages 379–392, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-102-6.

D. Gerbec, S. Gasperic, I. Smon, and F. Gubina. Consumers' load profile determination based on different classification methods. In *Power Engineering Society General Meeting, 2003, IEEE*, volume 2, pages 990 – 995, july 2003.

F. Giannotti, C. Gozzi, and G. Manco. Clustering transactional data. In *Proceedings of the 6th European Conference on Principles of Data Mining and Knowledge Discovery*, PKDD '02, pages 175–187, London, UK, UK, 2002. Springer-Verlag. ISBN 3-540-44037-2.

D. E. Goldberg. *Genetic Algorithms in Search, Optimization and Machine Learning*. Addison-Wesley Longman Publishing Co., Inc., Boston, MA, USA, 1989. ISBN 0201157675.

K. Gouda and M. J. Zaki. Efficiently mining maximal frequent itemsets. In *ICDM '01: Proceedings of the 2001 IEEE International Conference on Data Mining*, pages 163–170, Washington, DC, USA, 2001. IEEE Computer Society. ISBN 0-7695-1119-8.

A. Gray and A. Moore. 'n-body' problems in statistical learning. In T. K. Leen, T. G. Dietterich, and V. Tresp, editors, *Advances in Neural Information Processing Systems 13*, pages 521–527. MIT Press, December 2001.

R. Grossman. Parallel methods for scaling data mining. In *Parallel Methods for Scaling Data Mining*. Oxford University Press, 2001.

R. Grossman and Y. Guo. Data mining tasks and methods: parallel methods for scaling data mining algorithms to large data sets. *Handbook of data mining and knowledge discovery*, pages 433–442, 2002.

R. Grossman, C. Kamath, and V. Kumar. Data mining for scientific and engineering applications. *Tutorial at SC2001*, November 2001.

S. Guha, R. Rastogi, and K. Shim. Rock: A robust clustering algorithm for categorical attributes. *Information Systems*, 25, 2000.

C. Günther, S. Rinderle, M. Reichert, and W. van der Aalst. Change mining in adaptive process management systems. *On the Move to Meaningful Internet Systems 2006: CoopIS, DOA, GADA, and ODBASE*, pages 309–326, 2006.

N. Gutierrez. Demystifying market basket analysis, October 2006. URL http://www.information-management.com/specialreports/20061031/1067598-1.html.

M. Hahsler. A model-based frequency constraint for mining associations from transaction data. *Data Mining and Knowledge Discovery*, 13:137–166, September 2006. ISSN 1384-5810.

M. Hall, E. Frank, G. Holmes, B. Pfahringer, P. Reutemann, and I. H. Witten. The weka data mining software: an update. *SIGKDD Explor. Newsl.*, 11:10–18, November 2009. ISSN 1931-0145.

E.-H. Han, G. Karypis, and V. Kumar. Scalable parallel data mining for association rules. In *SIGMOD '97: Proceedings of the 1997 ACM SIGMOD international conference on Management of data*, pages 277–288, New York, NY, USA, 1997. ACM. ISBN 0-89791-911-4. doi: http://doi.acm.org/10.1145/253260.253330.

J. Han and M. Kamber. *Data Mining, Concepts and Technique*. Morgan Kaufmann, 2nd edition, June 2006.

D. J. Hand. Data mining: Statistics and more? *The American Statistician*, 52(2):112–118, May 1998.

D. J. Hand, H. Mannila, and P. Smyth. *Principles of Data Mining*. The MIT Press, 2001.

T. Hastie and R. Tibshirani. Classification by pairwise coupling. In *Proceedings of the 1997 conference on Advances in neural information processing systems 10*, NIPS '97, pages 507–513, Cambridge, MA, USA, 1998. MIT Press. ISBN 0-262-10076-2.

T. Hastie, R. Tibshirani, and J. Friedman. *The Elements of Statistical Learning: Data Mining, Inference and Prediction*. Springer, 2nd edition, 2009.

G. Herschel. Magic quadrant for customer data-mining applications. Technical report, Gartner Inc., July 2008.

W. Hoeffding. Probability inequalities for sums of bounded random variables. *Journal of the American Statistical Association*, 58(301):13–30, 1963.

J. H. Holland. *Adaptation in Natural and Artificial Systems: An Introductory Analysis with Applications to Biology, Control and Artificial Intelligence*. MIT Press, Cambridge, MA, USA, 1992. ISBN 0262082136.

R. C. Holte. Very simple classification rules perform well on most commonly used datasets. *Machine Learning*, 11:63–90, 1993. ISSN 0885-6125.

C.-N. Hsu, H.-H. Chung, and H.-S. Huang. Mining skewed and sparse transaction data for personalized shopping recommendation. *Machine Learning*, 57:35–59, October 2004. ISSN 0885-6125.

C.-W. Hsu and C.-J. Lin. A comparison of methods for multiclass support vector machines. *Neural Networks, IEEE Transactions on*, 13(2):415 –425, mar 2002.

G. Hulten, L. Spencer, and P. Domingos. Mining time-changing data streams. In *Proceedings of the seventh ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '01, pages 97–106, New York, NY, USA, 2001. ACM. ISBN 1-58113-391-X.

A. K. Jain and R. C. Dubes. *Algorithms for clustering data*. Prentice Hall College Div, March 1988.

J. Jain, I. Ari, and J. Li. Designing dashboards for managing model lifecycle. In *Proceedings of the 2nd ACM Symposium on Computer Human Interaction for Management of Information Technology*, CHiMiT '08, pages 12:1–12:2, New York, NY, USA, 2008. ACM. ISBN 978-1-60558-355-6.

N. Japkowicz. Learning from imbalanced data sets: A comparison of various strategies. Technical report, AAAI Technical Report WS-00-05., 2000.

T. Jiang and A. Tuzhilin. Segmenting customers from population to individuals: Does 1-to-1 keep your customers forever? *IEEE Transactions on Knowledge and Data Engineering*, 18:1297–1311, October 2006.

P. Jin and Y. Zhu. Mining customer change model based on swarm intelligence. In *Proceedings of the 3rd International Conference on Intelligent Computing: Advanced Intelligent Computing Theories and Applications. With Aspects of Artificial Intelligence*, ICIC '07, pages 456–464, Berlin, Heidelberg, 2007. Springer-Verlag. ISBN 978-3-540-74201-2.

L. Kaufman and P. Rousseeuw. *Finding Groups in Data An Introduction to Cluster Analysis*. Wiley Interscience, New York, 1990.

L. Kaufman and P. J. Rousseeuw. *Finding Groups in Data: An Introduction to Cluster Analysis (Wiley Series in Probability and Statistics)*. Wiley-Interscience, 2005.

M. J. Kearns. *Computational Complexity of Machine Learning*. MIT Press, Cambridge, MA, USA, 1990. ISBN 0262111527.

M. G. Kelly, D. J. Hand, and N. M. Adams. The impact of changing populations on classifier performance. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '99, pages 367–371, New York, NY, USA, 1999. ACM. ISBN 1-58113-143-7.

R. Kerber. Chimerge: Discretization of numeric attributes. In *Proc. Ninth Int'l Conf. Artificial Intelligence*, pages 123–128, 1992.

R. Klinkenberg. Learning drifting concepts: Example selection vs. example weighting. *Intell. Data Anal.*, 8:281–300, August 2004. ISSN 1088-467X.

R. Klinkenberg and I. Renz. Adaptive information filtering: Learning in the presence of concept drifts. In *Workshop Notes of the ICML/AAAI-98 Workshop Learning for Text Categorization*, pages 33–40. AAAI Press, 1998.

G. J. Klir. *Uncertainty and Information: Foundations of Generalized Information Theory*. WileyBlackwell, December 2005.

A. J. Knobbe. Multi-relational data mining. In *Proceeding of the 2005 conference on Multi-Relational Data Mining*, pages 1–118, Amsterdam, The Netherlands, The Netherlands, 2005. IOS Press. ISBN 1-58603-661-0.

J. Kobielus. The forrester wave: Predictive analytics and data mining solutions, q1 2010. Technical report, Forrester Research, July 2008.

R. Kohavi and M. Sahami. Error-based and entropy-based discretization of continuous features. In *Proc. 13th Int'l Joint Conf. Artificial Intelligence*, pages 1022–1027, 1996.

S. B. Kotsiantis and et al. Data preprocessing for supervised learning. *International Journal of Computer Science*, 1(2):111–117, 2006.

B. Kovalerchuk and E. Vityaev. *Data Mining in Finance: Advances in Relational and Hybrid Methods (The Springer International Series in Engineering and Computer Science)*. Springer, March 2000.

S. Kramer, N. Lavrač, and P. Flach. *Relational Data Mining*, chapter Propositionalization approaches to relational data mining, pages 262–286. Springer-Verlag New York, Inc., New York, NY, USA, 2000.

N. Krasnogor. An unorthodox introduction to memetic algorithms. *SIGEVOlution*, 3(4): 6–15, 2008. doi: http://doi.acm.org/10.1145/1621943.1621945.

M.-A. Krogel, S. Rawles, F. Zelezny, P. A. Flach, N. Lavrac, and S. Wrobel. Comparative evaluation of approaches to propositionalization. In *In Proceedings of the 13th International Conference on Inductive Logic Programming*, pages 197–214. Springer-Verlag, 2003.

R. Kruse, C. Borgelt, and D. Nauck. Fuzzy data analysis: Challenges and perspectives. In *Proceedings of the 8th IEEE International Conference on Fuzzy Systems*, volume 3, pages 1211–1216, 1999.

R. Kruse, M. Steinbrecher, and C. Moewes. Temporal pattern mining. In *Signals and Electronic Systems (ICSES), 2010 International Conference on*, pages 3 –8, sept. 2010.

D. A. Kumar and V. Ravi. Predicting credit card customer churn in banks using data mining. *International Journal of Data Analysis Techniques and Strategies*, 1:4–28, August 2008.

L. Kuncheva, J. C. Bezdek, and R. Duin. Decision templates for multiple classifier fusion: an experimental comparison. *Pattern Recognition*, 34(2):299–314, feb 2001.

W. V. Laer and L. D. Raedt. *Relational Data Mining*, chapter How to Upgrade Propositional Learners to First Order Logic: A Case Study, pages 235–256. Springer-Verlag New York, Inc., New York, NY, USA, 2000.

K. Lang. Newsweeder: Learning to filter netnews. In *Proceedings of the 12th International Machine Learning Conference (ML95*, pages 331–339, 1995.

J. Langford. Tutorial on practical prediction theory for classification. *Journal of Machine Learning Research*, 6:273–306, march 2005.

E. R. Lara and R. Barandela. Scaling clustering algorithm for data with categorical attributes. In *ICCOMP'05: Proceedings of the 9th WSEAS International Conference on Computers*, pages 1–6, Stevens Point, Wisconsin, USA, 2005. ACM, World Scientific and Engineering Academy and Society (WSEAS).

N. Lavra and B. Zupan. *Data Mining and Knowledge Discovery Handbook*, chapter Data Mining in Medicine, pages 1107–1137. Springer, 2005.

N. Lavrac. Data mining in medicine: Selected techniques and applications. *Artificial Intelligence in Medicine*, 16(1):3–23, May 1999.

S. S. Lee. Noisy replication in skewed binary classification. *Comput. Stat. Data Anal.*, 34: 165–191, August 2000. ISSN 0167-9473.

E. L. Lehmann and H. J. M. D'Abrera. *Nonparametrics: statistical methods based on ranks*. Prentice-Hall, 1st ed. 1975. revised edition, 2006 edition, 2006.

R.-H. Li and G. G. Belford. Instability of decision tree classification algorithms. pages 570–575, 2002.

R.-P. Li and Z.-O. Wang. An entropy-based discretization method for classification rules with inconsistency checking. In *Proceedings of the International Conference on Machine Learning and Cybernetics, 2002.*, volume 1, pages 243 – 246, 2002.

H. Lieberman. Letizia: An agent that assists web browsing. In *Internatioal Joint Conference on Artificial Intelligence*, pages 924–929, 1995.

H. Liu, F. Hussain, C. L. Tan, and M. Dash. Discretization: An enabling technique. *Data Mining and Knowledge Discovery*, 6(4):393–423, Oct. 2002.

X. Liu and H. Wang. A discretization algorithm based on a heterogeneity criterion. *IEEE Transactions on Knowledge and Data Engineering*, 17(9):1166–1173, September 2005.

S. Madeira, A. Oliveira, and C. Conceição. A data mining approach to credit risk evaluation and behaviour scoring. In *Progress in Artificial Intelligence*, volume 2902 of *Lecture Notes in Computer Science*, pages 184–188. Springer Berlin / Heidelberg, 2003.

O. Maimon and L. Rokach, editors. *Data Mining and Knowledge Discovery Handbook*, volume 1. Springer, New York, NY, October 2005.

H. Mannila. Theoretical frameworks for data mining. *SIGKDD Explor. Newsl.*, 1:30–32, January 2000. ISSN 1931-0145. doi: http://doi.acm.org/10.1145/846183.846191. URL http://doi.acm.org/10.1145/846183.846191.

C. Mantas, J. M. Ruiz, and F. Rojas. A procedure for improving generalization in classification trees. *Neurocomputing*, 48(1):727–740, October 2002.

D. Margineantu, S. Bay, P. Chan, and T. Lane. Data mining methods for anomaly detection kdd-2005 workshop report. *ACM SIGKDD Explorations Newsletter*, 7(2): 132–136, 2005.

Z. Markov and D. T. Larose. *Data Mining the Web: Uncovering Patterns in Web Content, Structure, and Usage*. Wiley-Interscience, 2007.

D. A. McAllester. Pac-bayesian model averaging. In *In Proceedings of the Twelfth Annual Conference on Computational Learning Theory*, pages 164–170. ACM Press, 1999.

G. J. McLachlan. *Discriminant Analysis and Statistical Pattern Recognition*. Wiley-Interscience, 2004.

R. V. Meteren and M. V. Someren. Using content-based filtering for recommendation. In *Proceedings of MLnetECML2000 Workshop*, volume 4203/2006, 2000.

L. L. Minku, A. P. White, and X. Yao. The impact of diversity on online ensemble learning in the presence of concept drift. *IEEE Transactions on Knowledge and Data Engineering*, 22:730–742, 2010. ISSN 1041-4347.

D. Mladenic. Machine learning used by personal webwatcher. *Machine Learning & Applications*, pages 26–34, 1999.

B. Mobasher, H. Dai, T. Luo, and M. Nakagawa. Discovery and evaluation of aggregate usage profiles for web personalization. *Data Min. Knowl. Discov.*, 6(1):61–82, Jan. 2002. ISSN 1384-5810.

R. Mojena. Hierarchical grouping methods and stopping rules: an evaluation. *The Computer Journal*, 20(4):359–363, 1977. doi: 10.1093/comjnl/20.4.359. URL http://comjnl.oxfordjournals.org/cgi/content/abstract/20/4/359.

S. V. Nath. Champion-challenger based predictive model selection. In *Proceedings of IEEE conference of SoutheastCon (SoutheastCon 2007)*, 2007.

R. Nisbet, J. Elder, and G. Miner. *Handbook of statistical analysis and data mining applications*. Academic Press, 2009.

A. Nobile and A. T. Fearnside. Bayesian finite mixtures with an unknown number of components: The allocation sampler. *Statistics and Computing*, 17(2):147–162, June 2007. ISSN 0960-3174.

J. Olvera-López, J. Carrasco-Ochoa, and J. Martínez-Trinidad. A new fast prototype selection method based on clustering. *Pattern Analysis & Applications*, 13:131–141, 2010. ISSN 1433-7541.

P. Paulson and A. Tzanavari. Combining collaborative and content-based filtering using conceptual graphs. In *Words: Learning, Fusion, and Reasoning within a Formal Linguistic Representation Framework, LNAI 2873, SpringerŰVerlag Berlin Heidelberg 168Ű185 (2003*. Springer-Verlag, 2003.

M. J. Pazzani and D. Billsus. Learning and revising user profiles: The identification of interesting web sites. *Machine Learning*, 27(3):313–331, 1997.

R. Pechter. What's pmml and what's new in pmml 4.0? *SIGKDD Explor. Newsl.*, 11(1): 19–25, Nov. 2009. ISSN 1931-0145.

F. Pernkopf. Discriminative learning of bayesian network classifiers. In *Proceedings of the 25th conference on Proceedings of the 25th IASTED International Multi-Conference: artificial intelligence and applications*, pages 422–427, Anaheim, CA, USA, 2007. ACTA Press.

C. Phua, V. Lee, K. Smith-Miles, and R. Gayler. A comprehensive survey of data mining-based fraud detection research. *Artificial Intelligence Review*, March 2005.

R. Polikar. Ensemble based systems in decision making. *Circuits and Systems Magazine, IEEE*, 6(3):21 –45, quarter 2006.

F. Provost. Machine learning from imbalanced data sets 101 (extended abstract). Technical report, Technical Report WS-00-05, The AAAI Press, Menlo Park, California, 2000.

F. Provost, D. Jensen, and T. Oates. Efficient progressive sampling. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, KDD '99, pages 23–32, New York, NY, USA, 1999. ACM. ISBN 1-58113-143-7.

D. Pyle. *Data Preparation for Data Mining (The Morgan Kaufmann Series in Data Management Systems)*. Morgan Kaufmann, March 1999.

J. R. Quinlan. *Induction of Decision Trees*. 1986.

J. R. Quinlan. *C4.5: programs for machine learning*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 1993. ISBN 1-55860-238-0.

L. D. Raedt. Logical settings for concept-learning. *Artificial Intelligence*, 95(1):187–201, 1997.

V. C. Raykar. Computational tractability of machine learning algorithms for tall fat data. *Matrix*, page 218, 2005.

S. J. Russell and P. Norvig. *Artificial Intelligence: A Modern Approach*. Pearson Education, 2003. ISBN 0137903952.

D. Ruta and B. Gabrys. An overview of classifier fusion methods. *Computing and Information Systems*, 7(1):1–10, 2000.

G. Salton and M. J. McGill. *Introduction to Modern Information Retrieval*. McGraw-Hill, Inc., New York, NY, USA, 1986. ISBN 0070544840.

J. S. Sánchez, F. Pla, and F. J. Ferri. Prototype selection for the nearest neighbour rule through proximity graphs. *Pattern Recogn. Lett.*, 18(6):507–513, June 1997. ISSN 0167-8655.

C. Schaffer. Sparse data and the effect of overfitting avoidance in decision tree induction. In *In Proceedings of the Tenth National Conference on Artificial Intelligence, AAAI-92*, pages 147–152. MIT Press, 1992.

T. Schon, A. Eidehall, and F. Gustafsson. Lane departure detection for improved road geometry estimation. In *Intelligent Vehicles Symposium, 2006 IEEE*, pages 546–551, 2006.

J. C. Shafer, R. Agrawal, and M. Mehta. Sprint: A scalable parallel classifier for data mining. In T. M. Vijayaraman, A. P. Buchmann, C. Mohan, and N. L. Sarda, editors, *VLDB'96, Proceedings of 22th International Conference on Very Large Data Bases*, pages 544–555. Morgan Kaufmann, 1996.

S. Shekhar, P. Zhang, Y. Huang, and R. R. Vatsavai. *Data Mining: Next Generation Challenges and Future Directions*, chapter Trends in Spatial Data Mining, pages 357 – 381. AAAI/MIT Press, 2003.

C. Soares, Y. Peng, J. Meng, T. Washio, and Z.-H. Zhou. *Applications of Data Mining in E-Business Finance: Introduction*. IOS Press, Amsterdam, The Netherlands, The Netherlands, 2008. ISBN 978-1-58603-890-8.

H. S. Song, J. K. Kim, and S. H. Kim. Mining the change of customer behavior in an internet shopping mall. *Expert Systems with Applications*, 21(3):157–168, oct 2001.

B. Spillmann, M. Neuhaus, H. Bunke, E. Pękalska, and R. P. W. Duin. Transforming strings to vector spaces using prototype selection. In *Proceedings of the 2006 joint IAPR international conference on Structural, Syntactic, and Statistical Pattern Recognition*, SSPR'06/SPR'06, pages 287–296, Berlin, Heidelberg, 2006. Springer-Verlag. ISBN 3-540-37236-9, 978-3-540-37236-3.

A. Srinivasan. The aleph manual. Technical report, Computing Laboratory, Oxford University, 2000.

A. Srivastava, A. Srivastava, and V. Singh. An efficient, scalable, parallel classifier for data mining. Technical Report TR-97-010, Department of Computer Science, University of Minnesota, Minneapolis, http://www.cs.umn.edu/ kumar, 1997.

K. O. Stanley. Learning concept drift with a committee of decision trees. Technical Report AI-03-302, Department of Computer Sciences, University of Texas at Austin, USA., 2003.

A. Subramanian, S. Pramala, B. Rajalakshmi, and R. Rajaram. Improving decision tree performance by exception handling. *International Journal of Automation and Computing*, 7(3):372–380, Aug. 2010. ISSN 1476-8186.

S. Sumathi and S. Sivanandam. *Introduction to Data Mining and Its Applications*. Springer, 2006.

T. K. Sung, N. Chang, and G. Lee. Dynamics of modeling in data mining: interpretive approach to bankruptcy prediction. *Journal of Management Information Systems*, 16 (1):63–85, June 1999.

J. A. Swets. Measuring the accuracy of diagnostic systems. *Science*, 240(4857):1285–1293, June 1988.

C. Taylor, G. Nakhaeizadeh, and C. Lanquillon. Structural change and classification. In *Workshop notes of the ECML-97 workshop on dynamically changing domains: Theory revision and context dependence issues*, pages 67–78, Prague, Czech Republic, 1997.

P. S. M. Tsai, C.-C. Lee, and A. L. P. Chen. An efficient approach for incremental association rule mining. In *Proceedings of the Third Pacific-Asia Conference on Methodologies for Knowledge Discovery and Data Mining*, PAKDD '99, pages 74–83, London, UK, 1999. Springer-Verlag. ISBN 3-540-65866-1. URL http://portal.acm.org/citation.cfm?id=646417.759114.

A. Tsymbal. The problem of concept drift: Definitions and related work. Technical report, Department of Computer Science, Trinity College: Dublin, Ireland, 2004.

S. D. Unwin. Fuzzy set theoretic foundation for vagueness in uncertainty analysis. *Risk Analysis*, 6(1):27–34, 1986.

V. N. Vapnik. *The nature of statistical learning theory*. Springer-Verlag New York, Inc., New York, NY, USA, 1995. ISBN 0-387-94559-8.

V. N. Vapnik. *Statistical Learning Theory*. Wiley-Interscience, 1 edition, Sept. 1998. ISBN 0471030031.

M. Vazirgiannis, M. Halkidi, and D. Gunopulos. *Uncertainty Handling and Quality Assessment in Data Mining*. Springer, 2003.

I. Žliobaitė. Learning under concept drift: an overview. Technical report, Vilnius University, 2009.

J. Wang and G. Karypis. Summary: Efficiently summarizing transactions for clustering. In *Proceedings of the Fourth IEEE International Conference on Data Mining*, ICDM '04, pages 241–248, Washington, DC, USA, 2004. IEEE Computer Society. ISBN 0-7695-2142-8.

G. Widmer and M. Kubat. Learning in the presence of concept drift and hidden contexts. *Machine Learning*, 23(1):69–101, 1996.

D. L. Wilson. Asymptotic properties of nearest neighbor rules using edited data. *Systems, Man and Cybernetics, IEEE Transactions on*, 2(3):408 –421, july 1972.

D. R. Wilson and T. R. Martinez. Instance pruning techniques. In D. Fisher, editor, *Proc 14th International Conference on Machine Learning*, pages 403–411. Morgan Kaufmann, 1997.

D. M. Witten and R. Tibshirani. A framework for feature selection in clustering. *Journal of the American Statistical Association*, 105(490):713–726, 2010.

I. H. Witten and E. Frank. *Practical Machine Learning Tools and Techniques with Java Implementations*, volume 1 of *The Morgan Kaufmann Series in Data Management Systems*. Elsevier Science & Technology, 0ctober 1999.

S. Wrobel. *Relational Data Mining*, chapter Inductive logic programming for knowedge discovery in databases, pages 74 – 99. Springer-Verlag New York, Inc., New York, NY, USA, 2000.

H. Xiong, P.-N. Tan, and V. Kumar. Mining strong affinity association patterns in data sets with skewed support distribution. In *Proceedings of the Third IEEE International Conference on Data Mining*, pages 387–394, Washington, DC, USA, 2003. ISBN 0-7695-1978-4.

H. Yan, K. Chen, and L. Liu. Efficiently clustering transactional data with weighted coverage density. In *Proceedings of the 15th ACM international conference on Information and knowledge management*, CIKM '06, pages 367–376, New York, NY, USA, 2006. ACM. ISBN 1-59593-433-2.

H. Yan, K. Chen, L. Liu, and Z. Yi. Scale: a scalable framework for efficiently clustering transactional data. *Data Min. Knowl. Discov.*, 20(1):1–27, Jan. 2010. ISSN 1384-5810.

P. S. Yu. Data mining and personalization technologies. In *Proceedings of the Sixth International Conference on Database Systems for Advanced Applications*, DASFAA '99, pages 6–13, Washington, DC, USA, 1999. IEEE Computer Society. ISBN 0-7695-0084-6.

O. R. Zaïane, M. El-hajj, and P. Lu. Fast parallel association rule mining without candidacy generation. In *ICDM '01: Proceedings of the 2001 IEEE International Conference on Data Mining*, pages 665–668, Washington, DC, USA, 2001. IEEE Computer Society.

L. Zadeh. Fuzzy sets. *Information and Control*, 8(3):338–353, June 1965.

T. Zhang, R. Ramakrishnan, and M. Livny. Birch: An efficient data clustering method for very large databases. In H. V. Jagadish and I. S. Mumick, editors, *Proceedings*

of the 1996 ACM SIGMOD International Conference on Management of Data, pages 103–114, NY, New York., June 1996. ACM Press.

X. Zhu, X. Wu, and Q. Chen. Eliminating class noise in large datasets. In *In Proceeding of International Conference on Machine Learning (ICML2003)*, pages 920–927, 2003.