Classification of Linked Historical Data to Calculate Customer Lifetime Value

Dongyun Nie

B.Sc, M.Sc

A Dissertation submitted in fulfilment of the

requirements for the award of

Doctor of Philosophy (Ph.D.)

Supervisor: Professor Mark Roantree

Dublin City University

Faculty of Engineering and Computing, School of Computing

September 2020

Declaration

I hereby certify that this material, which I now submit for assessment on the programme of study leading to the award of Doctor of Philosophy is entirely my own work, and that I have exercised reasonable care to ensure that the work is original, and does not to the best of my knowledge breach any law of copyright, and has not been taken from the work of others save and to the extent that such work has been cited and acknowledged within the text of my work.

Signed: __________ (Candidate) ID No.: _________ Date: _________

Acknowledgements

First of all, I would like to thank my supervisor Prof. Mark Roantree for his support and guidance during my PhD life. Without his patience and inspiration, I would not have had the opportunity to present this dissertation.

I truly appreciate the funding support of Science Foundation Ireland (SFI) grant SFI/12/RC/2289 and SFI/12/RC/2289-P2 to the Insight Centre for Data Analytics. I would also like to thank the School of Computing for their support and our industry partner from the insurance sector.

I would also like to thank all of my colleagues who were always willing to help and assistant with all the challenges I faced during my research and life in general. Especially Sue and Michael for proof-reading my writing over the years.

I would also like to thank all my friends, especially those in DCU. The family of Li (Wei, the Family Rap, who give me a lift everyday and fed me during the weekends); Haiyan who will show up whenever I need her; Xiaozui Mo, who gives me all her time and love.

Indeed, I would like to thank my family: my parents, parents-in-law, grandparents-in-law, my siblings and in-laws for their love and support. I can never thank them enough.

Finally, I would like to thank Chenghao Yu, my dear husband, for his continued love.

Table of Contents

List of Figures

List of Tables

List of Abbreviations

Abbreviation	**Full Text**
CLV	Customer Lifetime Value
CRM	Customer Relationship Management
TF-IDF	term frequency by inverse document frequency
PRLA-CL	Parallel Record Linkage Algorithm - Complete Linkage
OWA	Ordered Weighted Averaging
MCDM	Multi-Criteria Decision Making
IOF	Inverse Occurrence Frequency
VE	Variable Entropy
SM	Simple Matching
RFM	Recency Frequency Monetary
RM	Repetitive Median
CF	Collaborative Filtering
MCM	Markov Chain Model
GAM	Generalised Additive Model
CCD	Central Customer Database
ETL	Extract Transform Load
GI	General Insurance
FTP	File Transfer Protocol
UI	User Interface
AHC	Agglomerative Hierarchical Clustering

TP	True Positive
TN	True Negative
FP	False Positive
FN	False Negative
id	Identifier
NBR	Number
DOB	Date Of Birth
FN	First Name
LN	Last Name
N/A	Not Applicable
PCA	Principal Component Analysis
PCs	Principle Components
ANN	Artificial Neural Network
SVM	Support Vector Machines
CART	Classification and Regression

Abstract

Dongyun Nie

Classification of Linked Historical Data to Calculate Customer Lifetime Value

Most organisations employ customer relationship management systems to provide a strategic advantage over their competitors. One aspect of this is applying a *customer lifetime value* to each client which effectively forms a fine-grained ranking of every customer in their database. This is used to focus marketing and sales budgets and in turn, generate a more optimised and targeted spend. The problem is that it requires a full customer history for every client and this rarely exists. In effect, there is a large gap between the available information in application databases and the types of datasets required to calculate customer lifetime values. This gap prevents any meaningful calculation of customer lifetime values. In this research, we present a methodology to close this gap, by using a record linkage methodology to create a holistic customer record for each client. At this point, the remaining gaps in data are filled by our imputation algorithms, a process which then facilitates the calculation of values for each customer. The final step, evaluating our methodology, is achieved using a clustering approach to classify customers so that the customer lifetime value scores can be validated against the clusters in which they reside.

Chapter 1

Introduction

In the opening chapter of this dissertation, we will present the background and motivation for the research tackled in this dissertation. We also outline the main research questions and highlight the main contributions of this work. In section §1.1, we describe what is meant by *Customer Lifetime Value* and the different approaches that researchers have taken when trying to compute these values. In section §1.2, we define our problem statement, articulate the research questions we must address and then, present our research contribution in section §1.3. Finally, in section §1.4, we summarise the chapter and provide a structure for the rest of the dissertation.

1.1 Customer Lifetime Value

Customer Relationship Management (CRM) is a well-established strategy to support companies in their management of customer relationships across 4 dimensions: Customer Identification (Customer Segmentation); Customer Attraction (Direct Marketing); Customer Retention (Loyalty Program); and Customer Development (Customer Lifetime Value) [79].

Customer *segmentation* is regarded as a natural process to help companies to classify customers and plan market investment strategies such as direct sales. As such, it has been widely adopted by industry planners [13, 46, 56, 106, 110, 125]. Moreover, it plays a critical role in the development of the company's position by combining product differentiation and marketing segmentation to provide resources, objectives and competences to the company.

Various customer segmentation methods have been developed [109], using clustering algorithms [99], different approaches to classification [58], and regression [50], where each approach was selected to meet a specific goal [2,70,77,120]. In general, some form of clustering has been the most popular method in segmentation projects [54,114,119]. While segmentation allows us to classify customers into broad groups, it falls a long way short of distinguishing customers by giving them all an individual score or ranking.

One of the major goals of CRM is to maximize the *Customer Lifetime Value* (CLV) for the purpose of supporting long term business investment [67]. *CLV* is a measure that focuses on predicting the net profit that can accrue from the future relationship with a customer [24]. This metric can be calculated by recording the behaviours of the customer over the longer term and thus, help to build a customized business strategy. It has been a popular research topic, addressed by researchers in different ways, for example, Formulaic CLV [7, 11, 92] and Probability Model CLV [30, 86, 104].

The concept of customer lifetime value has been well-accepted by both academic and industry in the 1990's [51]. One of the first cases announcing CLV with a worked example is presented in [105]. In early research on the topic [27], CLV was defined as a vital construct in designing and budgeting for customer acquisition programs. Later, in [32], it was viewed more as a *dollar* value of a customer relationship based on *projected* cash flows with that customer. More recently in [61], it was defined as a metric calculated using the accumulated cash flow a customer accrues during his or her lifetime. It is an important and useful concept in marketing not only for managing the customer retention and migration situation [87] but it also plays an important role in market segmentation and the allocation of marketing resources [37].

CLV scores can be calculated in different ways, for example, Formulaic CLV [7, 11, 92] and Probability Model CLV [30,86,104]. However, one issue with the the CLV concept is that it has become quite sophisticated and complex in its calculations since the 1990s with businesses preferring a straight CLV calculation. This calculation becomes a problem as CLV formulae require a list of parameters which are hard to provide or generate. Moreover, CLV research is generally theoretical in nature and lacks experimentation with real data.

In normal conditions, the performance of clustering algorithms depends on the selected variables. In order to deliver optimal segmentation, variables play a significant role. However, in most cases, the variables are not ready-to-go. It is a complex process to identify and prepare the most suitable variables before assigning them to the clustering methods. Those variables will undergo a data preparation process [42] to be data mining ready. Following this, another critical task is to present a recommended list of variables (Components) [35, 59, 116, 122] with the adapted normalization [75].

1.2 Problem Description and Research Questions

Given the highly theoretical nature and focus of CLV research, what are the avenues for extending research knowledge in this area? How can we address issues such as access to real-world data and the steps necessary to construct, transform and enrich datasets, to make them suitable for existing CLV formulae such as equation 1.1? These were two of the early challenges that presented themselves after a literature review.

$$\begin{aligned} CLV_f &= am - A + a * (m - R/r) * [r^n/(1 - r^n)] \\ with \quad r^n &= r/(1 + d) \end{aligned} \tag{1.1}$$

In equation 1.1, the parameters required to populate the formula taken from [11] are:

- a is the acquisition rate, given a specific level of acquisition cost (A).
- A is the acquisition cost per customer.
- d is the yearly discount rate.
- m is the net income of a transaction.

- R is the retention cost per customer per year.
- r is the yearly retention rate.

It was clear that any research effort would benefit greatly from a collaboration with a medium to large business partner and access to a real-world dataset. Apart from access to real world data, such a collaboration will provide a greater understanding as to which parameters are unavailable and how to develop a methodology for imputing missing variables from available ones.

The first step was to engage in an appropriate collaboration and we did so through an agreement with an Ireland Insurance Company, an international company, with a large office and IT department in Dublin. During initial discussions, it became clear why CLV customer scores were very difficult to calculate: the data required to populate formula 1.1 either did not exist or was very difficult to extract. We will present these reasons as we articulate the research questions we faced in this research.

Research Question 1: The Holistic Customer Record.
We discovered that a holistic (full history) record does not exist for the vast majority of customers. There are practical reasons for this as the insurance sector has different providers: the insurance companies, brokers and online sales, with each employing proprietary (and heterogeneous) mechanisms for data collection. In an ideal world, they would share a common customer ID (they do not) or would contain a set attributes (eg. Customer Name, DOB and mobile phone number) which makes data integration simple. In almost all cases, this did not exist as addresses were recorded differently or mobile phone numbers were not requested and even customer names were recorded differently. This leads us to our first research question: can we devise a methodology to construct a holistic customer record by integrating all of the parts of a customer's history?

Research Question 2: Customer Classification.
The next step in our research assumes a reasonable level of success in constructing complete customer records and is focused on automatic classification of customers.

This is a crucial mid-point stage in our research that helps to validate final CLV calculations. Given that customer numbers in most medium to large organisations will be high and our experiments had a total of 500,859 policies and 387,951 customers, it will not be possible to manually classify each customer record. Thus, an important mid-stage milestone in our research is to classify customers in broad terms. By this we mean a small set of categories (eg. 2 clusters={Good,Bad}, 3 clusters={Good, Average, Bad} or 4 clusters {Good, Above Average, Below Average, Bad} to which each customer must belong. In the event that full CLV calculations are not possible for all customers, this provides a fallback option for businesses. However, a robust validation for this step is crucial. The second research question can be articulated as: is there an approach to automatically classifying customers so that good customers are clustered together and not-so-good customers are clustered separately? A second part of this research task has the requirement that customer clusters must be validated.

Research Question 3: Data Imputation.

Our literature review will show that parameters to calculate customer lifetime value are rarely available: research is mainly on the theoretical aspect of CLV calculations. It is anticipated that at this point of our research, we have enough data to impute any missing variables. This may not be successful (accurate) for all customers but our goal is to target a high percentage of the overall customer database. It will require both the full customer history generated in step 1 and elements of the clustering process in step 2 to impute the missing variables. Thus, our third and final research question asks, given the dataset at our disposal, if we can devise a suitable algorithm to impute the final variables necessary to populate the most widely used CLV formulae?

Hypothesis.

We are now in a position to articulate a hypothesis for this research as follows: The established CLV formula may be applied to a real world dataset if holistic customer records can be constructed and classified, and an appropriate feature extraction process used to impute missing variables.

1.3 Main Contributions

The primary contributions arising from this research are associated with the three research questions which are addressed.

- The first contribution was to develop a record linkage methodology which was adapted for characteristics particular to the large insurance dataset. This work was presented in [81], with 12 variables used to generate the unified client records from the raw data provided by our industry partner.
- The next contribution was to develop an approach to automatically classify each of the 300,000+ customers in the database. This work, which included both feature extraction and feature selection methods, was presented in [80].
- Our third contribution was to impute missing data before finally generating CLV scores for each customer in the database. This work was presented in [101].

1.4 Summary

The goal of this chapter was to provide an overview of our research area and to motivate the key problems facing researchers in this area. We completed the chapter by articulating our research questions and hypothesis. It should be pointed out that the data included in our research are data records exported from relational databases and not *Linked Data* from the area of the semantic web and RDF triples. The *linked* historical term in our research comes from the linking of historical customer records in order to construct a unified customer record. We now provide an outline structure for this dissertation and describe the main goal of each chapter.

We begin in chapter 2 with a literature survey which will motivate our particular solution. In chapter 3, we will provide a high level methodology for our approach. Our three research questions are addressed in each of the next 3 chapters. In chapter 4, we present our *record linkage* method which creates the holistic customer record. In chapter 5, we present the classification method for customer segmentation. In chapter 6, present our method for imputing missing variables required for CLV calculation.

In chapter 7, we present an evaluation of our research together with insights and the main findings of our research. Finally, in chapter 8, we will present our summary and discuss topics for future research.

Chapter 2

Related Research

Our research proposes a method to generate CLV metrics for a customer base from half a million policies across various insurance types. However, since a single holistic record does not exist for each customer in the raw data, and a complete customer history is crucial to extracting a dataset suitable for machine learning and CLV prediction algorithms. There are three broad areas that must be examined in order to deliver a solution for CLV generation. For this reason, we split the literature review accordingly. We begin by examining methods for linking related customer data in section §2.1; approaches to segmenting customers are described in section §2.2; in section §2.3, we present existing CLV models with the approaches used by researchers to predict CLV variables; finally in section §2.4, we summarise our review and articulate outstanding research questions.

2.1 Matching Customer Records

The construction of a unified record is slightly different from matching and is often associated with two major challenges: the cost of pairwise record comparison and matching accuracy. The first challenge has major resource implications once a dataset goes beyond a certain size: as little as 100,000 records requires the construction (and continuous updating) of a large similarity matrix. With n=100,000 records and the number of matches calculated as $(n * n - n)/2$, this requires 4,999,950,000 comparisons. The second challenge seeks to ensure that records for the same customer are matched. A suitable method to creating the unified record, commonly known as *Record Linkage* [45], must balance these two factors.

The area of matching has become a hot topic because this could be applied to various types of data sources. Rahm in [89] developed an approach to integrate large amounts of source data like schemas, ontologies and entities. Their purpose was to provide an approach that could match data sources, not only for pairwise matching but also for holistic data integration of many data sources. For complex schema integration, they first used an intermediate result to merge schemas until all source schemas had been integrated. For entity integration, they first clustered data by semantic type and class, where only entities in one cluster were compared with each other. However, when clustering very large datasets, the execution time increases rapidly. This is a well known problem and in our work, we have the same issue. However, they have the benefit of using Linked Open Data, whereas insurance data does not have the same properties as Linked Data. Furthermore, our unified record must create a relationship graph (connected families and co-habitants) between every customer record. Thus, if we adopt their approach, a further layer of processing is still required. This type of matching which seeks to link the relationship between records is the *Record Linkage*.

The first task for record linkage must begin with identifying the suitable methodology and similarity measure. In [8], Bhattacharya and Getoor evaluated a series of entity resolution approaches to matching authors within references: an *attribute-based* entity resolution, which only considers attributes during matching; the *naive relational* entity resolution, which considered both attribute and related reference as an additional attribute using hyper-edge similarity (to label the best similarity match between references); and the *collective relational* entity resolution resolves the related references jointly.

In addition, multiple similarity measures were applied to calculate the distance between attributes. The Common Neighbours [16] method calculates the similarity between records by counting the number of common entities in both. This method could also be associated with frequency measuring in terms of weighting the variables.

The Jaccard coefficient [82] can calculate the similarity of records by measuring their common neighbours. In situations where one record may have many frequent neighbours, the Adamic/Adar similarity [1] can be applied. The Adamic/Adar similarity will predict the similarity between two records, but individually consider the significance of each element. This often associated with weight for the frequency represented by the TF-IDF (term frequency by inverse document frequency) [90]. Another technique is called, the Adar similarity with ambiguity estimate, using higher-order neighbourhoods, and negative constraints from relationships.

After presenting methods and associated similarity measures, they evaluated each method using their reference dataset. They proposed a clustering algorithm that used both attributes and relational information to determine the join while having an efficient implementation. They demonstrated their method using a real-world bibliographic dataset, their algorithm measures ambiguity (corresponding to the disambiguation aspect of resolution) and dispersion (corresponding to the identification aspect) to significantly outperformed other algorithms. Similar to us, their work provides a solid understanding of how similarity measures work with real-world data. However, the size of their experiment dataset is thousands of reference records while enterprise recordsets such as that used in our research, are far larger. As a result, the time required for calculating the similarity between pairwise records is $\mathcal{O}(n^2)$.

To address these types of issues, a blocking technique [73] could manage high volumes of data. If we wish to focus specifically on designing a matching and integration solution for large recordsets, it is necessary to efficiently check the similarity between every pair of customer records, which is further complicated by textual data. The technique for the quick similarity check between records, especially for text data, is called blocking [44]. With this method, each block contains n characters been called *n-gram* or *k-mer*. The block size is important, as the smaller the block size, the smaller the number of overall blocks. For example, if a block contains just 3 alphanumeric values, the potential number of blocks for a dataset is $(36)^3$ (26 possible letters plus the numbers 0-9). As a consequence, while blocking techniques are shown to work well in isolation, they can also be used in conjunction with other techniques as a potential for further improvement.

In [71], Mamun et al. used blocking techniques to match 3 million records inside 2 minutes with a PRLA-CL (Parallel Record Linkage Algorithm - Complete linkage) algorithm [57]. The authors begin with taking the datasets and assigning thresholds; then combining all datasets and sorting lexicographically [72]. The next step is to find the exact cluster by merging duplicate records, then generating the blocks using *k-mer*, and then connecting a pair of records where the distance is less than the threshold value. The records will also become vertices in a graph, then the component can be connected in the graph. Furthermore, split the connected components into one or multiple groups for the pairs of vertices with a group less than a threshold. Finally, bringing the duplicates back. The blocking method will quickly assign records to segments who shared a block into the same groups. Then the pairwise comparison will only be applied within the records in the same group.

The main issue is resolving records who share blocks with multiple other records and are thus, assigned to that group, as it means that the record is a *mismatch* for other potential matching records. The solution to this problem is to allow duplicate assignments: records can belong to multiple groupings. They then match using a minimum threshold distance between two clusters to remove duplicates. However, where the attributes and size of the clusters increase, the calculation times increase exponentially because the number of blocks will increase. If they did use a small n for block [53], the number of duplicates will increase because they will be easier to have a common *n-gram* compare, to match a longer block. Although they did (manually) incur mistakes for certain attributes randomly by insertion, deletion, and substitution operations. Our work is similar to this research but those manually inserted mistakes may not reflect real-world data. The difference is due to the complexity of the real-world dataset, a data cleaning process is necessarily used in our research. Moreover, the huge recordset they are utilising is obtained by duplicating existing records. In that particular case, the total number of blocks will not change significantly. In our work, we must match multiple relationships where, for a very low threshold, it creates very large segments. While in [53], authors employ a post-processing step to avoid merging all records into a single large cluster, it does not allow the detection of different types of relationships. We found it necessary to explore new functionality in our approach.

McCallum et al. in [73] present similar research to ours where they employ two steps to match references. Firstly, they used a method they term *Canopies*, which offered a quick and cheap text distance matrix to detect matches within a specific threshold and place them in subsets. The fast distance matrix is used to calculate the distance using an inverted index, which calculates the number of common words in a pairwise reference. A threshold is then applied to determine subsets and similar to our approach, subsets may overlap. They then use greedy agglomerate clustering to identify pairs of items inside Canopies. It is clear both of us applied a threshold to determine subset in the research, also subset may overlap with each other. However, the reference dataset is more precise than the real-world insurance dataset and by using the inverted index, may mismatch customers who, for example, had made a mistake in recording their name. Moreover, because of the size of our dataset, we required a more effective segmentation method to put objects into a smaller subset. Essentially they are limiting their approach to matching author names to detect the same author. Our matching is multi-dimensional, with similarity matrices across multiple attributes, and we are seeking to detect 3 forms of relationships and not merely the *author-author* link in this work.

Record Linkage Summary. Various approaches have been utilised, each approach satisfying a different set of requirements. Numerous works [10, 33, 48, 49, 62, 102] have examined the methods for managing *big* text values using clustering, where the common method is to use blocking techniques with *n-grams* or *k-mer* which convert strings to vectors. One applies *tf* or *tf-idf* to weight the vectors, then using similarities to measure the distance between vectors for clustering. This technique could prevent the problem of a customer record matching with a term of pairwise record matching required. While this saves from time complexity for comparing the pairwise records for $O(n^2)$, increasing by a distance measure for string matching [96] (often the Edit or Levenshtein distances), the time complexity is again $O(m^2)$ (m for the number of characters in a string) for each text attribute.

Most research has had limited access to the real-world datasets, where experiments use semantic datasets [85], reference datasets or text documents. The mismatch rate increases when data is present in heterogeneous ways or manual inserts are involved. In attempting to use these approaches, a cleaning process is required in order to prevent the common human mistakes. We are often faced with many mismatches as records for the same customer (or for family members) were placed in separate segments. However, string matching approaches as discussed in this literature are often inadequate where we are trying to determine if two entities (customers) are identical. The nature of string matching will give many false positives (for actual customers) and can miss - or rank much lower - two entities which may refer to the same customer. Consider a situation where the customer "Ryan" is misspelled as "Ryen". This is the same customer but may receive a lower score than "Ryan" and "O Ryan" which are different customers. These algorithms will require a level of adaptation for usage in entity resolution.

This highlights the types of issues that are involved in trying to extract and integrate (generate) the data necessary for CLV calculations. The main focus of previous research is *client-client* matching. Our real world problem requires more than this traditional relationship: our matching requirement must have efficient matching of client-client relationships, but also to link family members and customers who are domiciled with the target client.

2.2 Customer Segmentation

Marketing researchers have studied customer segmentation for decades. It is not only because this is a useful tool for customer management but it also provides guidance for business strategies. In [55], Kara and Kaynak investigate and conceptualise using segmentation theory to place customers into different marketing groups where they used traditional segmentation methods such as normative segmentation, niche marketing, micro marketing, database marketing, relationship marketing, and mass customisation. In this research, normative segmentation meant clustering customers using the average *within-group* similarity which computes the average similarity for all pairs from the same group. Their results demonstrated that finer segmentation (tailoring needs for the individual customer) were more successful than traditional segmentation. It increased customer retention and loyalty, with a higher competitive advantage for the company that had a flexibility and responsiveness structure. While this work developed a number of interesting hypotheses, it was not possible to *validate* these ideas through a robust set of experiments. While we have very similar goals to this research, we employ a large number of experiments, both to validate and *understand* our findings.

A conceptual model for customer ranking, based on principal components, was presented in [84]. This approach comprised multiple steps: Variable Definition - select the possible variables for experiments; Data Extraction - getting the data including variables for all customer records; Variable Scaling - this process will transform the variable into an interval of $[0, 1]$; Principle Component Analysis - transform the variables into a set of principle component; Defining and specifying the ranking functions to reduce the dimension of the dataset by selecting the principle components depending on a threshold for the variance explained result; and ultimately, sorting the customers by the calculated value of the ranking function. While this offers a comprehensive approach to customer segmentation, their research was focused entirely on the *theoretical* aspect of customer ranking. In our research, we employ a real-world dataset, develop a similar method, but crucially, we deliver a detailed evaluation using a large number of experiments.

Müllensiefen et al. ([77]) present two case studies using survey data collected from subjects with different socio-demographic characteristics. Their primary goal was to identify the contribution of variables to the customer segmentation process. In their experiments, their clustering algorithms delivered between 2 and 10 clusters with an analysis on how socio-demographic and personal variables performed in the different cluster configurations. Interestingly, their conclusions were that personality variables were more important for accurate market segmentation. Similarly, we also employ clustering algorithms to segment the customer dataset, and employ a full Extract-Transform-Load framework [111] where data is acquired from operational databases, integrated (customer record linkage), and transformed for machine learning. The difference is we had half a million records across multiples years as the original input dataset, whereas their variables are coming from the online survey with a smaller size. Their primary focus was the investigation of the different variables. In that respect, our focus differs in that we attempt to find the best overall *configuration* for customer segmentation. This requires a far more detailed evaluation than would be required for the examination of different dataset variables.

The researchers in [99] presented a classification selection method using five fuzzy criteria with aggregated Ordered Weighted Averaging (OWA) and a Multi-Criteria Decision Making (MCDM) system for customer segmentation. The OWA operator allows the implementation of the concept of fuzzy *majority* in the aggregation phase by means of a fuzzy linguistic quantifier [123]. The five fuzzy criteria are: the number of classes; balanced classes; coherent classification; dependency; and accuracy criterion of the predictive model. The aggregated OWA was comprised of three steps: re-order the input arguments in increasing order; determine the weights for the operator in a proper way; and then use OWA weights to aggregate the re-ordered arguments. They then applied identified classifications to an unsupervised clustering algorithm. They evaluated the clustering method by using 3 validation types: internal, external and relative. By testing a real marketing case based on a B2B environment, the result shows that a cluster size of 3 delivers the most suitable marketing strategy.

Their work focused on *classification* during customer segmentation and the results are evaluated using a clustering approach. Our method also uses a form of clustering for customer classification. However, classification selection is not the main driver in our research because our research is not about marketing scores for distinguished numbers [15]. Our validation is not only validate the *performance* of the clustering but also exploit the actual customer groups later in our research. Unlike [15], our dataset requires both feature extraction [41] and feature selection [21] [68] to extract the optimal variable set.

In [19], Cibulková and Sulc used hierarchical clustering methods (single-linkage, average-linkage and complete-linkage) on a travel dataset, in combination with popular distance measures [107] to make customer segmentation:

- ES (named by ESkin, Eleazar in [28]) is used to determine the similarity between two categories.
- IOF (Inverse Occurrence Frequency) uses the absolute frequencies of the observed categories to achieve a more precise similarity definition between two categories in the case of mismatches.
- LIN measure (named by LIN, Dekang in [66]) incorporates the relative frequencies of observed categories for similarity determination.
- VE (Variable Entropy) measures treat similarity between categories according to the *within-cluster* variability of this particular variable.
- SM (Simple Matching) The most well known coefficient measure.

They compared the three hierarchical clustering methods in combination with five distance measures with an optimal k (cluster) value determined by low within-cluster variability [94]. Their experiment reveals that the best customer segmentation configuration for three clusters is: using the complete-linkage hierarchical clustering and the IOF or LIN distance measure or the complete-linkage hierarchical clustering, in the combination with the ES measure. However, the dataset used in this research is customer travel data and the variables are always categorical or binary data. The size of the dataset is much smaller compared to our study. Besides, they are focused on similarity measure and validating only within clusters. In our case, we must first generate the dataset ready from existing data sources and then, validate the performance of customers across *all* clusters.

Customer typology is the name for approaches that intend to explore different consumer groups in order to focus marketing activities on consumer segments using various topological approaches [115]. According to [5], a 3-cluster was the solution for their particular customer typology. The author started by creating an online questionnaire to measure the popular customer characteristic in the literature: Extraversion, neuroticism, trust, attitude, perceived risk, shopping enjoyment, and willingness to buy. Those characteristic provide the input variable for clustering. They then identify the cluster centre before applying k-means where k=$\{3,4,5\}$, before finally, assigning the observed cases. The result proved that a 3-cluster solution can achieve a 97.7% of correct classification. In this research, a refined classification method by combining the personality dimension and culture determinants with online shopping behaviour is identified, where the segmentation will differentiate by county. Our research also uses customer personality variables and a clustering method to perform customer segmentation. Moreover, their study differs from ours as dataset is entirely different with different challenges and our validation must reflect that accordingly.

Recency Frequency Monetary (RFM) together with k-means clustering was used in [18] for customer categorisation. In this research, the online retail customer behaviours were converted into an RFM value. The RFM is presented by a three-dimension (R, F, M) number, where for each dimension, 1 is the best score and 5 the worst. Thus, the RFM value across three dimensions ranges from 111 to 555. They then segment customers by clustering the RFM scores. There are three comparative clustering methods used: k-means, Fuzzy c-means, and RM (Repetitive Median) k-means. From the performance of the three methods, k-means is the quickest because initially the effective medians for the data distribution are used for centroids calculation. The focus of their work is to compare the performance of the three methods: iteration, time taken, and average silhouette width (validation of consistency within a cluster). However, they didn't study the performance of the customers in each segment. There was no validation to test which method optimised the customer segmentation.

Rezaeinia and Rahmani in [93] presented a hybrid method for customer segmentation by using weighted RFM variables and Expectation Maximisation clustering algorithm with the closest k-neighbours. They used the popular precision and recall validation approach along with the F_1 score (also called F score, present the balance between precision and recall) to rank the cluster results in terms of ranking customers. Their output proved that this hybrid system outperformed the conventional recommender system: CF (collaborative filtering) method. Because of the nature of sales datasets, the customer recordset will be smaller than the transactions. Moreover, after the reduction process, the size rapidly increases. In this research, the number of clusters was set to 5 and the variables applied in the models is not R, F and M. We also used the validation measures of precision, recall and F_1 score in our research, the details of which are provided in 5. However, this research does not provide or discuss the variable and clusters selection process, which is crucial to understanding and making useful insights from their experiments.

Summary Customer segmentation has been widely used in the business sector because of its power in supporting businesses strategies. By examining the state of the art in its application to the business sector, it provided a list of methodologies and usable variables for understanding the customer segmentation process and corresponding problems. These variables have been shown to be a critical factor in determining the *accuracy* of the segmentation result. One of the popular approaches is clustering using customer purchase behaviour variables. Our approach is to experiment with a number of methods for segmenting customers using selected sets (different approaches to feature extraction) of variables in order to identify the most suitable method for our particular dataset. While all of the existing approaches focus on marketing, our research aims to construct a customer dataset suited to imputing missing variables by classifying customers into broad groups (good, bad, average) and validating these classifications.

2.3 CLV Prediction and Imputation

Today, CLV is an essential measure for marketing analysts, with the popular CLV calculating models introduced in [7]. The authors provide several cases with a detailed CLV calculation associated with various CLV variable conditions. The cases of sale occur in a different time period (annually, longer or shorter than one year); the cases of differentiating the retention rate assumption; and the cases of the different timing of cash flows. In each case, a formula is presented alongside an example predicting CLV using retention, acquisition, margin and discount. The work presents a basic insight of how to calculate CLV as parameters such as time period, cash flow and customer behaviour change. However, they have dropped the acquisition cost from their CLV models to distinguish their models from the prospective models (equation 1.1).

The annual sale model shown in equation 1.1 is more suited to our research because our data is renewal on a yearly basis. Compare to the research in [11], this research argued that equation 1.1 did not address the allocation/trade-off between spent on acquisition and retention so they adopted a non-linear programming approach to attempt to maximise the objects. We can use this equation because we would like to calculate and analyse the acquisition (acquisition spending/acquisition rate) and the retention (retention speeding/retention rate) separately, which has not been addressed in their research.

A survey analysing different CLV models has been provided in [38]. In this paper, Gupta et al. evaluated six CLV approaches:

- The first model type is the RFM model [31], a widely used model in business. The classification of the customer depends on the group number of each variable. For example, if there have 5 groups for each variable, there will be 125 different groups of customers. This model could efficiently group different customers. However, the drawback is that the *individual* CLV value of a customer could not be directly extracted.
- The probability model is one of the more popular CLV model types. In this model, customer behaviours will be transformed into probability distributions. The benchmark model here is the Pareto/NBD model [100]. The main focus of this type of model is to predict whether a customer is going to remain or not; the purchasing behaviour associated if they will stay. The model often requires two values from previous purchases: recency and frequency. Models can be derived and associated with other models. The MCM (Markov Chain

Model) has also been used as the probability model. The base of the probability model is formulae. Instead of the actual values, the variable often applied the customer behaviour characteristics probability. Because in the real world, it is not possible to have all the required variables as a value already. Moreover, it often required use of predicted variables to make the various predictions. Our research utilised the probability model approach.

- The Econometric model [112,118] shares the same philosophy as the probability model: to predict the CLV by estimating the acquisition, retention and the cross-selling (or margin). The authors have provided a list of related works on how to calculate these three variables. The effect of cross-selling on profit has been provided in Econometric model and here, it is differentiated with the probability model. However, the end result is substantially the same: predict the behaviour variables to enable the CLV prediction.

- The Persistence models [23, 121] focus on the behaviour of its components. Again, the components are acquisition, retention, and cross-selling. However, in the Persistence model, they are part of the dynamic system. It studies the movement of the behaviours where *movement* can be presented in a matrix. The MCM has often been present its variable in the matrix to make saleable predictions. The main difference for the Persistence model is that variables can be dynamic, which means the model can be adjusted over time. This is a benefit for datases that expect have big changes over the years.

- The number of Computer science models [34, 43] has increased over the years. The purpose of the models in this type is to emphasise the prediction capabilities. They have been used as a tool to solve issues like the explosion of variables. This model is popular for churn analysis because it can efficiently process a huge number of variables. Another advantage is that computer science models are easy to combine. As a support model for business intelligence, this model can be joined with the probability CLV model.

- The Diffusion/Growth model [47, 98] can make better long term predictions because this model can aggregate data and use diffusion or growth model to forecast the customer acquire. The base of this model is the formulae with a list of parameters, by forecasting the customer acquisition first, then feeding the value into another CLV forecast model to predict the customer value of a firm.

In research [38], detailed information about how models were developed and the results from other research has been provided. The logic behind the models are the same: use the parameters to estimate the CLV value. If the parameter is not available, use the model to forecast the parameters [14].

The goal of this research is similar: by summarising the existing works to point out the interesting future topics. This is very useful for researchers to start and develop a good understanding of this area to enable the identification of the potential solution. However, because they did not focus on real problem solving, this means that they have not tested the models with a real dataset. From the models mentioned in research [38], the CLV modelling is always based on a combination of acquisition, retention, and margin because there always has a based fundamental model, based on the original model [39]. In terms of improving results, multiple models can be combined. Thus, we use multiple models from the literature also. However, we will apply these models with a real world dataset and run a high number of experimental configurations to understand how the data behaves in different est conditions.

The most widely used probability approach is the Markov Chain Model (MCM). The MCM model can transform the CLV variables into probability values. This is a flexible model which could also work with the RFM model. In [88], Pfeifer and Carraway demonstrated how to use the MCM for CLV calculation. They begin with the state the probability of the customer will end purchase in each period. A list of probabilistic value will be entered into a $n \times n$ matrix, where n means the number of entries for the states. Then they constructed a matrix to summarised the cash flow. The row number of the cash flow is n. In the end, the two matrices will be assigned to a formula which also has the discount involved. To generate the firm value of the CLV by using this model, and the author tried to calculate the infinite horizon of the formula. The authors explained the MCM model step by step. Because this model is reliant on the recency value, to extend the ability of the model, each state could have multiple entries. They have mainly been placed at the same column but different rows in a matrix. The MCM model can also be modified into an RFM-based MCM, because the states can be defined by R, F, M values with some known integer upper bounds (4 or 5 as the most popular number). This model is very flexible because there is a various state you can use as an input. However, in order to use this model, a comprehensive catalogue of customer repurchase probabilities is required, which may not always be available. Moreover, the cash flow matrix which related to company expenditure is also challenging to measure.

This MCM (Markov Chain Model) has also been used in [26] in order to predict customer behaviours. The authors performed their experiments using data from the insurance industry. They used different models: a Relation-level Model and a Service-level Model to make predictions on customer retention and profit, and compared the performance between the models. However, the main concern with their research is their *retention* behaviours. They used the CLV results to segment by retention rate, but they did not go as far as to predict the actual CLV values.

One of the most popular variables in CLV is customer retention or, as the opposite value is called, *customer churn* [3]. However, predicting churn is not an easy task. Mostly, they are based on formulae, often associated with list of variables. The focus of churn analysis for insurance datasets has increased as of late. There are numerous examples of research which has applied several data mining methods for churn analysis.

In [76], Morik and Köpcke use *tf-idf* from information retrieval as a calculated attribute to analyse customer churn on insurance data and compared the accuracy of their method to other conventional classification methodologies. Unlike our work, their research had a higher degree of dimensional data available and were in a position to use cross-fold validation to address the class imbalance while we had to undersample to generate our churn dataset.

Günther et al. ([36]) analyse insurance data using logistic regression with a generalised additive model (GAM). Once again, this research uses data from the insurance sector but here, the researchers benefited from more fine-grained data, which provided a month-by-month view of the data. As a result, the authors did not provide a comparison between the GAM model and other classification algorithms.

In [12], Bolancé et al. present a means of predicting customer churn for motor insurance. They compare four methods used to calculate churn: decision trees, neural networks, logistic regression and support vector machines. In terms of accuracy across models, the authors' neural network approach had a similar accuracy to ours. However, the authors dataset focused specifically on motor insurance. Additionally, our evaluation compares different classification methods.

A one-class support vector machine which can be used to under-sample data was proposed by Sundarkumar et al. in [108]. The efficacy of the approach was evaluated using five classification methods on a motor insurance fraud dataset with a credit card churn dataset. The five methods employed were: decision trees, support vector machines, logistic regression, probabilistic neural networks and a group method for data handling. Similar to this research, we employ decision trees, support vector machines and neural networks for our evaluation. However, the authors' insurance dataset focused on fraud detection within motor insurance while our approach attempts to calculate customer churn across multiple insurance types.

Zhang et al. in [124] combine deep and shallow modelling to predict customer churn on insurance data. Similar to our approach, their evaluation compared the performance of their method with several classification algorithms using a similar set of metrics. However, the authors used a large dataset consisting solely of life insurance policies whereas our dataset contains several policy types. This difference is crucial as it results in different sets of dimensions, motivating the need for different methods to be applied.

In [95], Risselada et al. evaluate the staying power of various models of churn prediction for two domains: insurance and internet service providers. The insurance dataset comprises life insurance churn data and four predictive method were used: logit models and decision trees, both with and without bagging. The authors found the decision tree method with bagging to show the best performance. In our research, we employ a wider range of prediction algorithms to provide a greater comparison of churn prediction methods.

Based on the tournament in [78], Neslin et al. analysed the impact of methodological factors. By analysing the 44 entries from both academics and companies with techniques like logistic regression, decision tree, neural networks, discriminant analysis, cluster analysis, and Bayes. They found that prediction methods will affect the accuracy and the accuracy could be improved by developing the prediction methodology. This paper has made a clear statement of the importance of experiments and customising your own churn prediction model.

Another important element for CLV is *acquisition* which often interacts with retention. The relationship between retention, acquisition and customer equity was introduced in late 1990s in [11]. In this research, they described how to optimise acquisition spending, acquisition rate, retention spending and retention rate. They proposed an acquisition (acquisition spending/acquisition rate) and a retention (retention speeding/retention rate) relationship by graphing the behaviour of a budget increase and the curve of the rate. When spending increase, the corresponding rate will increase at the same time. However, there has a ceiling rate from the manager experience. In this case, they optimised the spending and the rate by calculating the peak from the curve. They also attempted to find a balance between finding and keeping customers. This research provided methodologies to control customer equity by managing acquisition and retention. However, it is based on historical data (it is first necessary to feed in historical data like manager opinion, expenditures, rates to training the model) and each variable cannot be computed independently. For example, to calculate acquisition rate it is necessary to have historical data to determine the steepness of the curve and ceiling rate of acquisition based on the manager's experience.

Pfeifer in [86] explore whether a firm should spend more money on customer retention if the cost to acquire a new customer is 5 times the cost of retaining an existing customer. They provided an in-depth study of computing acquisition and retention spending with examples where both the average model (average cost per customer) and marginal model (speculate that the number of customers may be zero) were considered. Their main goal was to determne the optimal *costs to acquire* and *costs to retain*. They created a model that includes growing margins and changing retention rates to verify their hypothesis. A challenge in modelling changing retention rates is to do so in a way that allows a meaningful analysis of the relationships between retention spending and the set of changing retention rates. However, a shortcoming in this work was that experiments required access to historical data for initialisation, meaning that customers without this data must be dropped. Thus, not all customers are evaluated by this study.

In [91], Reinartz et al. investigated the *balance* between retention and acquisition. Specifically, the impact of product-related costs, total retention costs, and acquisition cost effective on customer profitability was measured. The purpose of studying the several scenarios to analyse the implications for resource allocation with a large multinational B2B high-technology manufacturer data, is to provide the recommendation to the reader: how much to invest and how to invest. Unlike this research, our focus is to manage the costs to make accurate CLV predictions.

Summary. The measures margin, discount, retention, and acquisition are required for CLV prediction, especially retention and acquisition which are crucial. However, the relationship between these two variables is hard to interpret and manage. Most research captures the two variables inside a yearly spread, common to all customers. Nevertheless, the CLV value should be customised for each customer, in this case, the CLV variables are required in an individual-level.

Decision trees, support vector machines, neural networks and logistic regression models are more commonly used for predicting churn for insurance data. It is important to note that while all methods attempt to predict churn on insurance data, there is no homogeneous data model for the representation of data. Factors such as granularity, size, available features(and the types of each feature) have a significant impact when determining which model performs best.

In addition, calculating the acquisition at an individual-level, we must measure more factors because it is hard to measure the exact company expenditure on a customer. Especially the cost often applied for both retention and acquisition, for example, the advertisement costs. Because, there has no research providing the complete road-map, we need to begin with constructing the dataset and CLV variables and then, feed these variables into a proper CLV model.

2.4 Conclusions

The related research in this chapter covered research in record linkage [73], customer segmentation [115], churn and acquisition analysis, and CLV modelling [38]. Each research correlates with the research question posed in this dissertation. What we discovered is that research efforts have made an attempt to incorporate all of these topics in an effort to present a more holistic approach to Customer Lifetime Values for marketing analysts. Our work will seek to construct this holistic framework for CLV predictions, from start to end.

Predicting CLV is a complex process. It requires the transformation of data extracted from enterprises databases that may not be ready for CLV calculations. As this data is often unavailable or the transformation process too difficult, most CLV models are theory-based because they often had a fundamental model [11]. Thus, there remains a gap between the theoretical and the real world requirements.

It is important to note that, with the methods that attempt to predict churn and acquisition, there is no homogeneous data model for the representation of data. Whereas, in our research, we have to begin with unifying the features before any data mining prediction process. This is because the factors such as size, value, features, and the types of each feature have a significant impact on the predictions. In this case, we also extract and test some uncommon features from the unifying process.

Chapter 3

Research Methodology

The purpose of this chapter is to provide a high level description of our research methodology. Our research involved an industry partner with real-world data, presented to us as *disconnected* customer records. Using a series of processes, each with specific goals, we deliver customer lifetime values (CLVs) for each customer. In this type of research where assumptions are made about the data, it is important to have a detailed description of the data used in the various algorithms.

In this chapter, we begin by describing real-world problems and requirements in section §3.1. Then, we provide only an outline description of each process as our goal here is to look at the overall methodology, explain the role of each component and by tying these processes together, how we achieve our overall aim. In section §3.2, the first process, Record Linkage, is described briefly. The next three processes form part of the overall goal of customer classification and are described in section §3.3. In section §3.4, the final two processes, data imputation and CLV calculations are discussed. Finally, in section 3.5, we summarise the progress made in this chapter.

3.1 Dataset Description

To fully appreciate the problems faced in this research, some knowledge regarding the insurance sector and associated data is necessary. This section presents an overview of Insurance System technologies which comprises four parts: Current Architecture, Data Sources and Extraction, Data Staging Area and Central Customer Database (CCD), and Customer (thus, research) Requirements.

Our architecture of Insurance Systems was based on workshops and documents from the industry partner. In the Data Sources and Extraction section, we will examine the three main sources of data, extracted from source systems such as HLT, PAXUS and SOLAS. The Data Staging Area includes the Central Customer Database (CCD). For the CCD, we provide an illustration of this relational Oracle database. Database software utilised in this system are DB2 from IBM and Oracle. The datasets transformed between the systems are in CSV format.

3.1.1 Data Flow Architecture for Insurance Systems

This Architecture contains four parts, comprising Data Resource, Extract Transform Load (ETL), Data Stage and Data Storage. It can also be divided into two layers: the Real-time Data layer and the Reconciled Data layer. In the detailed flow diagram of the architecture shown in Figure 3.1, it only has two layers without derived data. The main reason for this is that the architecture does not contain an operating data warehouse, OLAP, Cube and Data Presentation Interface. Moreover, there are some quality issues in the reconciled data layer, and decision making provided by this system are not also accurate for end users. The details will be provided in §3.1.7.

3.1.2 Data Sources

In this section, we provide a description of the source information and how data is delivered to the internal server before the integration and data mining. The process involved in this section is located in the Real time Data layer because this process controls the real time data flow.

The three main sources involved are: Health, Life and General Insurance (GI). The customer and policy data extracted from the source system is transformed into CSV files, then loaded into the individual external server. Files from the external server are imported by transaction logic into an internal ETL Server, then data from distinct

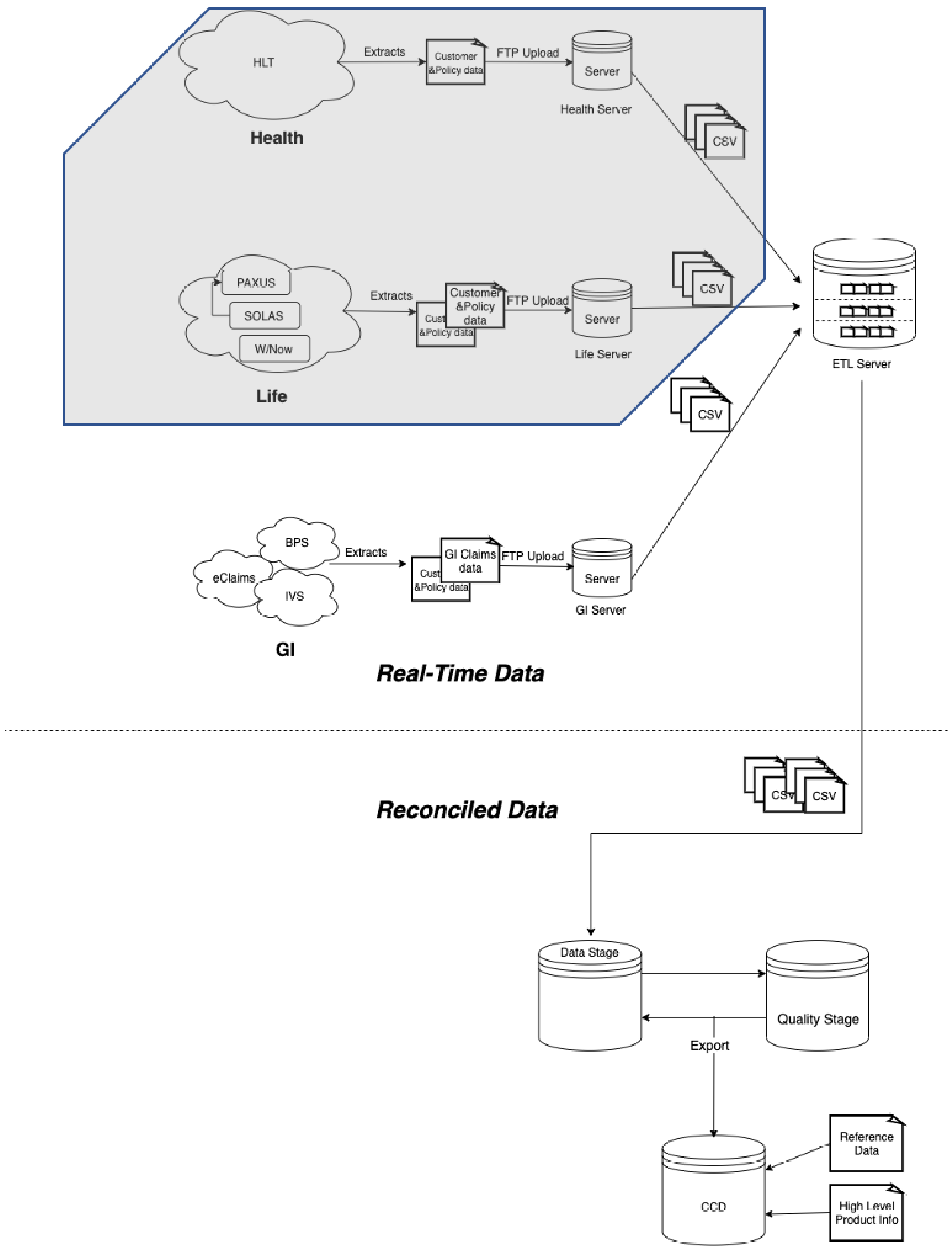


Figure 3.1: Insurance Architecture

servers is saved in separate units of this single internal ETL Server. Figure 3.2 illustrates what types of source system are used and how the Extraction process works. At this stage, no integration is involved, which means that if your policies are held in different sources, you will be seen as two separate customers.

3.1.3 Extraction Architecture

This process sees data transformed from external data sources into the Insurance Internal format. We now describe the source systems and initial extraction of CSVs, their movement to the individual servers before being imported into the Insurance Internal server. For individual sources, there are several types of insurance systems involved and they are shown in Figure 3.2 Layer 1.

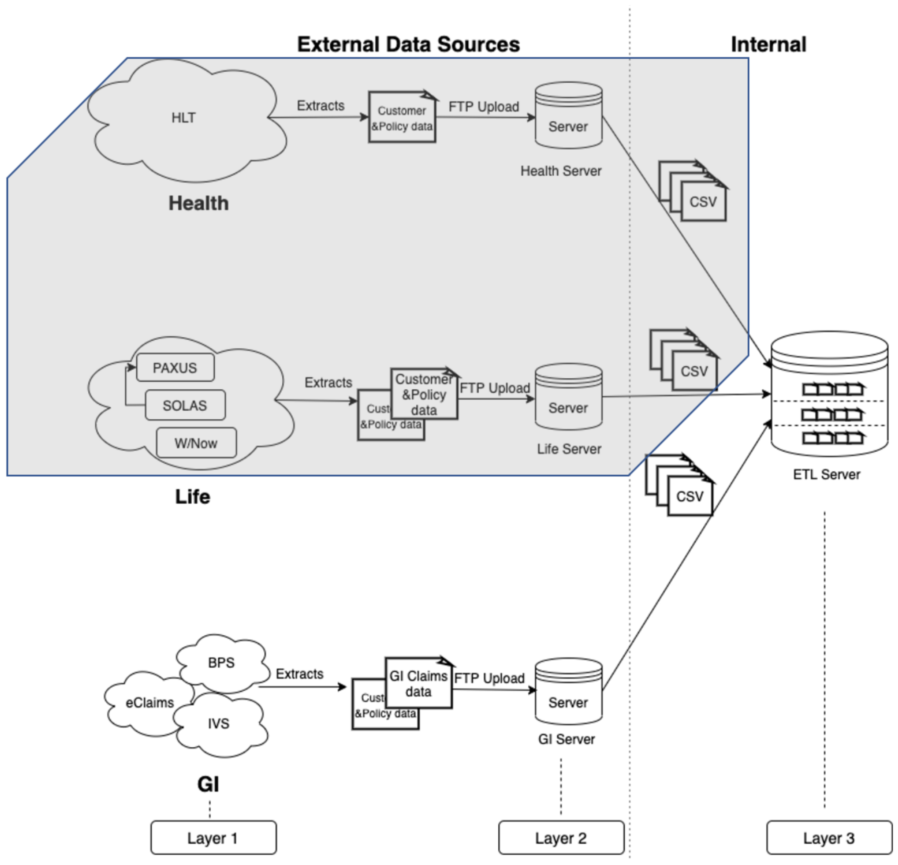


Figure 3.2: Data Sources and Extraction: Real time Data

The process is implemented as three layers. Layer 1 contains all source systems introduced earlier. Layer 2 has three individual external servers: Health Server, Life Server and GI Server. Layer 3 is a single internal ETL sever. Between each layer lies the Extraction Process, which is described now.

Extraction Process. Data is extracted from Layer 1 as CSV files and transformed in Layer 2. During the transformation, files will be uploaded to the corresponding data server at Layer 2 using an FTP (File Transfer Protocol) port. Data extracted from the Health and Life sources contain Customer and Policy data. Data extracted from the source GI includes not only Customer and Policy data but also GI Claims data. Layer 3, the single internal server has authority to store data that has been imported from the individual external server at Layer 2 in different units. Data delivered from Layer 2 to Layer 3 are transformed from the external data source into the format of internal data server.

There are two forms of refresh available to update records from the main source systems: Delta Refresh and Full Refresh. Full refresh updates all target records, even the records that have not been modified. Full refresh only occurs when requested by the user. Delta refresh updates records which have changed since the last upload, and is more commonly used in Insurance Systems.

3.1.4 Data Staging Area and CCD

At this point, data is captured in a single internal server but exported files from each unit are stored separately; they are not integrated. The purpose of the Data Staging Area is to clean and integrate data for the CCD. The Reconciled Data Layer in Figure-3.1 shows the structure of the Data Staging Area and CCD. There are two separate stages involved: *Data* Stage and *Quality* Stage. We can also treat these two stages as the IBM Staging Process. The Data Staging Area is located in the insurance companies internal servers, away from 3rd party software technologies.

Data Staging Area We now introduce the Data Stage ETL Server. This ETL model will retrieve files from different units of the internal server and convert data to an appropriate format that will be recognised by the Quality Stage. The Process between the Internal Server and the Data Stage is known as *Data Store.* Data Store use an IBM ETL tool to import data from the internal server and store all the data at the data stage. The main function for the Data Stage is as storage for the Quality Stage, preparing data for implementation in the next stage.

After data is transformed from Data Stage to Quality Stage, Data Quality can then be improved. Here, the major roles for the Quality Stage process is customer matching and standardisation of data. Data imported from the Data Stage will be cleaned and integrated at the Quality Stage. For enterprise initiatives, high quality data is required. The Quality Stage is intended to provide a data integration platform, reduce time and cost to implement data and maximise return on data quality. It is possible to construct consolidated customer, customer contention and create data structures for research, fraud detection and planning. After this Quality Stage, integrated data is exported into the CCD.

Central Customer Database. The Central Customer Database (CCD) not only contains an integrated view of the Insurance Company's data but is also periodically refreshed with changes from the source systems. The CCD is a relational database: it is not designed with a multidimensional schema found in traditional data warehouses. The database system used in CCD is Oracle and is used to store all updated records from the Data Stage and the Quality Stage. It contains functions which are supported by the Oracle Database with PL/SQL functions available to extract data.

3.1.5 Outstanding Business Requirements

There are several objectives or requirements for this system. This system should integrate all policies owned by an individual. This could be utilised by internal users, external users (brokers) and end customers from multiple platforms (e.g. IOS, Android, PC). This system also should be flexible and extendable, which would be suitable for future customer-related functionality. In addition, a novel system should be based on the existing assets such as from the Central Customer Database and existing web-services to back-office systems. This in turn could provide the solution for marketing sales. The Integrated system and High-Level Solution of the architecture is illustrated in Figure 3.3.

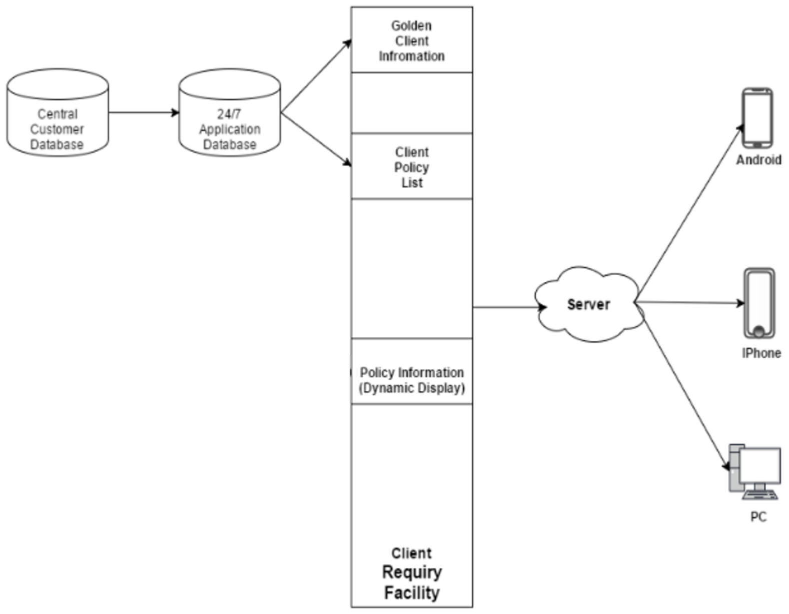

Central Customer Database
24/7 Application Database
Golden Client Infromation
Client Policy List
Policy Information (Dynamic Display)
Client Requiry Facility
Server
Android
IPhone
PC

Figure 3.3: High-Level Solution of Architecture

Most of these objects will be used to provide the functionality for the new architecture. The first problem is to enable enquiries on all insurance policies owned by an individual customer to be avilable in a single customer record. Secondly, we need to use Data Mining to improve the outcome of CCD, for example, how to rank customers to identify *good* clients which supports marketing decisions. Our proposed solution to address the problems is provided shortly in §3.1.7.

3.1.6 Dataset Description: Tables and Attributes.

Because various systems had different levels of data protection principles, it was not possible in practice, to have a unified Insurance System. Moreover, the GI sources already had all the problems discussed above because the GI itself had multiple source systems. Thus, we adopted the approach that if we can solve the problem for GI, the system can be scaled to the remaining systems. For this reason, we focused our attention on GI system, as this represented a microcosm of the overall problem. Thus, other sources are greyed in Figures 3.1 and 3.2. The rows of data we retrieved are from the data stage. The data stage will provide the tables and attributes that relevant in our research.

There are two major data entities (types of customer policies) involved in GI: Home and Motor. Our customer database contained 1 million records across six years, where 75% of policies are motor insurance and 25% are house insurance. There are two types of customer policies in the respect: renewal and new policies. Renewal is when a customer renewed an existing policy, while new policy means this is a new *customer* for the company. To purchase either type of policy, customers might call, go online or request an agent in store. The customer policy will be stored in a different database depending on purchase type. At the time that this research commenced, there was no method for identifying the same customer who had purchased multiple policies. In fact, customer information could be stored separately on individual databases. From the customer database, the 1 million policies are made by half a million unlinked individual clients. Thus, the first task is to link customer records to construct a unified customer record.

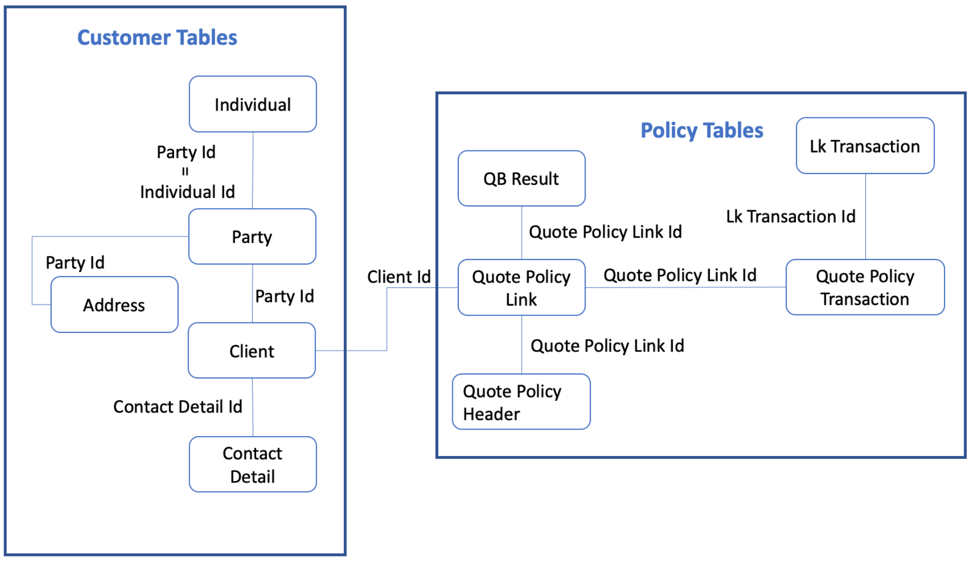


Figure 3.4: Table Relationships

There are two properties we identified early in this research: the attributes describing customer behaviours (customer-level) and the attributes for an individual policy (policy-level). The relationship between these two components is shown in 3.4.

The customer component comprised of 5 tables. The *Party* table stored the common attributes, and will connect customer detail tables (*address, individual, client*) together using *Party Id* the details of the attributes has shown in table 3.1. Using *Contact Details* for each customer, it was hoped to use *Contact Detail Id* to link the *Client*.

The table 3.2 shows all the Contact Details for the person. Moreover, the table 3.3 stored the details about an Individual. The table 3.4 listed the details of attributes about a Client. Table 3.5 contains Address related information for the clients and risks.

Each table we will shows the name of the (*Attribute*) associated with the *Description* of this attribute.

Table 3.1: Party

Name	**Descriptions**
Create By	The id of the user that created the record.
Create Dtime	The instance in time when the record was physically created.
Party Id	Unique identifier for the record.
Party Type Id	It can be used to classify a party.
Update By	The id of the user that updated the record.
Update Dtime	The instance in time when the record was last updated.

Table 3.2: Contact Detail Table

Name	**Descriptions**
Email Id	Email Id for the contact.
Fax Nbr	Fax number.
Home Phone Nbr	Home phone number.
Mobile Phone Nbr	Mobile phone number.
Work Extension Nbr	Extension number for the work phone.
Work Phone Nbr	Work phone number.

Table 3.3: Individual

Name	**Descriptions**
Date Of Birth	Date of birth of the individual.
First Name	Party First Name.
Gender Id	Male / Female
Individual Id	Unique identifier for the record.
Last Name	Party Last Name.
Title Id	Identifier for title.

Table 3.4: Client Table

Name	**Descriptions**
Confirm Gcflag	Confirm flag for Gcflag.
Contact Detail Id	Contact details (phones, email, etc) are stored on the Contact Detail Table.
Do Not Call Flag	If set, the customer do not want contact by phone.
Home Docpref Dtstp	The instance in time of Home document.
Home Online Doc	Status of Home online document.
Motor Docpref Dtstp	The instance in time of Motor document.
Motor Online Doc	Status of Motor online document.
Proposer Change Dtime	The instance in time when the proposer was last updated.
Client Id	Unique identifier for client.
Client Reference Nbr	System generated unique digit ID.
Client Type Id	Identifies the Type of the Customer.
Commercial Policy No	Commercial Policy Number.
Contact Branch Id	Branch code of the user that assisted the new customer with the quote process.
Dpa Consent Flag	Data protection act.
Duplicate Flag	Duplicate status of the customer.
Gcflag	General Marketing flag.
Link Present Flag	This field will describe the customer link.
Master Client Id	The master client record id of which this record is a duplicate.
Occupation Id	Occupation of the Customer.
Owner Subchannel Id	Client should have Owner SubChannel Id.
Ppp Eligibility Id	Possible values (Allow, Disallow). Always set to Allow for new customers.
Preferred Contact Id	Customer's preferred way of contact. Possible values are (Home Phone, Work Phone, Mobile number, Fax number, Email).
Product Consent Flag	Product marketing and specific promotions.
Status Id	Status of the customer.

Table 3.5: Address Table

Name	**Descriptions**
Address Id	Unique identifier for the record.
Address 1	Address line 1. The UI will capture the address information in this field.
Address 2	Address Line 2. The UI will capture the address information in this field.
Address 3	Address Line 3. The UI will capture the address information in this field.
Address 4	Address Line 4. The UI will capture the address information in this field.
Address Type Id	This field will determine the type of address stored.
Building Gname	Building Group Name for given address.
Building Name	Building Name for the given address.
Coordinate X	X co-ordinates in Geo Coding used for visualisation purposes.
Coordinate Y	Y co-ordinates in Geo Coding used for visualisation purposes.
Country Id	The country this address belongs to.
County Id	List of all supported counties in Ireland will be provided in the dropdown.
Geocode Id	Geodirectory unique reference identifier for the geocoded address.
Match Level	Match Level returned by Geo-coding.
Organisation	Geodirectory address element.
Post Code 1	Postal code for the address.
Post Code 2	Extended Postal code for the address.
Post Town	Post town for the given address.
Primary Locality	Primary Locality for the given address.
Primary Throughfare	Primary Throughfare for the given address.
Rating Code	Rating Code for the address will be captured.
Secondary Locality	Secondary Locality for the given address.
Secondary Throughfare	Secondary Street Number if any.
Sub Building Name	Name of Sub Building.
Townland	Townland for the given address.
Vanity Au Flag	A Flag to indicate if geocoded address has a corresponding Vanity Address which is different.

Table 3.6: Quote Policy Transaction Table

Name	Descriptions
Quote Policy Trans Id	Unique identifier for the record.
Quote Policy Link Id	Reference to the quote/policy record.
Lk Transaction Id	Identifies the type of transaction generated. eg. type of MTA - permanent or temporary.

Each policy comes from a customer quotation. When the quotation becomes an active policy, this information is specified in the *Quote Policy Link* table (table B.2 in appendix B). This table links the the table *Quote Policy Header* (table B.4 in appendix B) which has the details of a policy quotation, and the *QB Result in appendix B* table (table B.3 in appendix B) who presented the details of the steps in the premium calculation to show how the premium was sourced altogether using the attribute *Quote Policy Link Id.*

The *Quote Policy Transaction* table (table 3.6) stores all the type of modifications or amendments made to a Quote/Policy. It connected the policy transaction details (*Lk Transaction*) and the quotation table *Quote Policy Link* altogether, in this case, we will be able to know the transaction information about the policy. Due to the heterogeneity of the different sales policies, it is not possible to aggregate the policies to a customer level using *Client Id.*

3.1.7 Requirements for a New Methodology

In this section, we summarise the issues presented in the current system as a set of requirements for new data engineering practices to deliver an improved architecture.

The first requirement is to modify the current architecture to provide not only an integrated data architecture, but to use that central platform to provide a *record linkage* function. The main goal of this new approach is to deliver a *customer* centric view of data as opposed to the current *policy* centric view of data. Only through this new process can customers be truly evaluated.

Customer Lifetime Value has been widely used in marketing as a solution for decision marking. However, to calculate CLV, as well as to validate the results of CLV, it is necessary to combine customer records from multiple databases to construct a *richer* dataset. This is not only for customers who hold one or more policies but also to graph the relationships between customers: family members and other customers domiciled with the target customer.

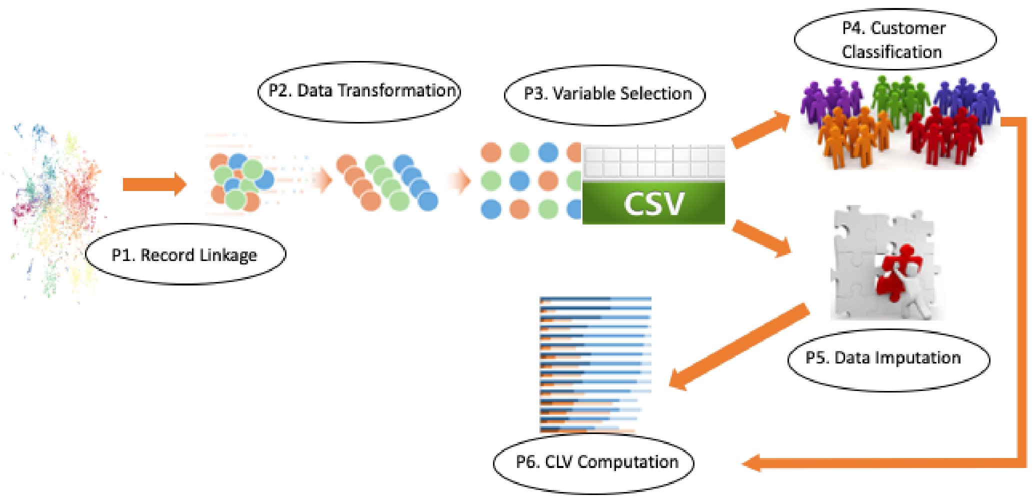


Figure 3.5: CLV Prediction Methodology

In this case, the we need a methodology to fulfil all requirements, and the processes are outlined as a system architecture shown in Figure 3.5. These six processes are divided into three components. The first component, tackled as a major part of this research is Data Integration (P1) which constructs the unified customer record ready for CLV prediction. The unified customer record can be transformed (P2) and selected (P3) as a historical dataset (P4), necessary to validate the CLV record or trained to predict CLV variables (P5) before the actual CLV prediction (P6) can take place.

The tables and attributes have been provided from the data stage will be imported into the MySQL database. Moreover, for all processes that involve a data flow, it has all interacted with MySQL database. Python is the main programming language to design the architecture with the package *mysql.connector* used to connect to the database. We now proceed to provide an overview of the major research topics covered in the remaining chapters with a more detailed overview of those not tackled in later chapters as they either provide only a small contribution or are outside the scope of our overall research.

3.2 Record Linkage (P1)

Data Integration is a crucial component in our research due to the problem of a high numbers of customers who cannot be matched as common identifiers do not exist across datasets and their identifying information is inexact or often, quite different (e.g. a change of address). In order to do that, we need to apply the Record Linkage [117] to fulfil this task. Record linkage is the task where algorithms are used to identify the individuals sharing the different datasets.

We must have in-depth knowledge of the datasets and methods that are necessary for our research. The Data Preparation (P1a) process including dataset understanding, and algorithm for identifying similarity measures. This process will be described in detail in chapter 4. In summary, the data integration methodology comprises 5 steps: pre-processing; segmenting the recordset; application of the matching algorithm; using a ruleset to improve matching results; and validation, with only an overview of each step provided now.

Pre-processing. The dataset we use in this research is a real-world dataset. In order to maintain high accuracy with good performance, before matching can commence we need a level of pre-processing for the dataset. Firstly, all characters are converted to lowercase English to eliminate the dissimilarity due to case sensitivity, and a normalisation of Irish/English characters. Secondly, all non-alpha-numeric characters are removed. Finally, the 4-attribute address is concatenated but the most abstract level of granularity (normally information after county) is removed.

An address pre-processing step take as an example shown in table 3.7, while some other attributes has to go though step 1 and step 2 also.

After pre-processing, address 1 and 2 will be assigned as: *1oconnellstreet* and address

3 will become *1oconnellst.*

Table 3.7: Sample Addresses

Address	Address 1	Address 2	Address 3	Address 4
1	1	O'Connell Street	Dublin 1	Ireland
2	1 O'Connell Street, Dublin 1	Dublin		
3	1 O Connell St	Dublin	Ireland	

Dataset Segmentation. The test dataset containing approximately 200,000 records will require approximately 20 billion pairwise comparison operations for a single evaluation, using only a single attribute, as the number of operations are $= (n*n-n)/2$. For this reason, the first task is to segment the recordset with the goal of minimising the possibility of a customer having records in separate segments, as those records will never be matched. Moreover, the technique will support our experiment for multi-relationship matches, not just the same *person* which is referred to as *Client-Client* matching using our approach. In addition, a separate requirement is to link family members and non-family member co-habitants.

Clustering Client Records. We adopt a clustering approach based on Agglomerative Hierarchical Clustering (AHC) [22], where a similarity matrix is computed to represent the distance between each pair of records. For us, the process stops after a deliberately low threshold for distance (dissimilarity) has been reached. Our first point of difference with traditional AHC lies in our construction of the similarity matrix. We do not construct a single 2-dimensional matrix but instead compute a multidimensional matrix which enables us to examine distance measures across different variables. We chose this method due to poor results obtained when using a single aggregated distance measure across all variables. This represents another significant part of this research with a detailed discussion presented in chapter 4.

Application of Rules. While using a multidimensional similarity matrix allows for a more fine grained comparison of distance between client records, the application of all dimensions was not suited in all matching requirements. Furthermore, we required a facility to apply different thresholds across the dimensions. There are three types of matches required in our research: client matches (records for the same client); family matches (family members for a client); and domiciled (where non family members reside at the same address).

Because all other processes are reliant on the research in P1, it is necessary to have a validation process (P1b) for P1 itself. The validation for this component begins with identifying the similarity measures, null punishment scenario, efficient segment measure, and best parameter and configuration settings for the types of relationship links.

3.3 Customer Classification (P2, P3, P4)

Our second research goal is a method to classify customers into broad categories, eg. good, average, bad. The primary goal for this work is to construct a dataset for CLV validation using historical raw data. In addition, this component comprises five sub-processes, from data acquisition to validation. The first process was addressed at component - Record Linkage (P1). The other 4 processes in this component employed to classify customers as follows: data transformation (P2), variable selection (P3), clustering (P4a) and validation (P4b).

3.3.1 P2: Data Transformation and P3: Variable Selection

This is a common and crucial process for data mining. The initial data instances are at the yearly *policy* level. After P1 has completed, the recordset is annotated with relationship links. The purpose of the transformation process is to create a dataset with a *single* instance per customer and with variables that are appropriate for machine learning algorithms of the variables selected. In other words, we require a form of variable selection from the dataset generated by the P1 (linkage) process.

Variable selection includes analysis of variables common to various research projects ([5,18,64,77,114]) but also used the opportunity to include and test a number of less commonly used variables. We score and rank ([40]) all defined variables so that it is possible to discern which variables positively contribute to the performance, from the those that do not.

Transformation and Variable Selection are not only important for P4 but are crucial for P5 and P6. Those two processes are preparation process for those processes required machine learning methodology.

3.3.2 P4: Clustering Customers

In this section, we present an outline of the strategy to evaluate customer clusters obtained from the different experiment configurations. For the purpose of comparison, we applied multiple clustering methods and normalisation algorithms along with the validation method for each experiment. We implemented two validation methods (P4b): objective validation and subjective validation. Both methods are evaluated by the standard evaluation metrics, which used to compare the different data mining models. Those metrics are used across multiple components in our research and as they are standard and common to may research projects, we present their description here.

Evaluation Metrics. The measures TP, TN, FP and FN are used to refer to *True Positive*, *True Negative*, *False Positive* and *False Negative* measures respectively. These are widely used when measuring $Accuracy$, $Precision$, $Recall$, $Specificity$ and F_1 score. The standard accuracy (percentage of correct classifications) is insufficient in evaluating a classifier but can be used to provide a useful baseline. The precision, recall and specificity metrics provide better detail as to the actual performance of a classifier within classes. All model (experiment) configurations are validated using all 5 metrics, which are defined in equations 3.1 to 3.5.

Equation 3.1 represents $Accuracy$, the percentage of correctly identified classes. As it is calculated as the total number of accurate (true) predictions ($TP + TN$), divided by the overall number of predictions ($FP+FN+TP+TN$), the result will always be in the range 0 *to* 1. This metric provides an overview of how well a model performs.

$$Accuracy = \frac{TP + TN}{FP + FN + TP + TN} \tag{3.1}$$

Precision is a well known method which determines the number of predicted positive instances divided by the total number of positive predictions $(TP + FP)$ as shown in Equation 3.2. It has the effect of identifying how accurately positive classes were predicted.

$$Precision = \frac{TP}{TP + FP} \tag{3.2}$$

The *Recall* measure shown in Equation 3.3 is often used in conjunction with Precision, and calculates the number of *predicted* positive instances divided by the total number of *actual* positive classes. If this value is low, it can indicate a high number of *false negatives* within a classifier.

$$Recall = \frac{TP}{TP + FN} \tag{3.3}$$

In Equation 3.4, $Specificity$ is defined as the number of negative instances detected, divided by the total number of negative predictions. This total for all true negatives $(TN + FN)$ is computed. Specificity indicates the level of accuracy of the classifier when predicting only negative classes.

$$Specificity = \frac{TN}{TN + FP} \tag{3.4}$$

The F-Measure $(F - Score)$ is a metric used to evaluate the performance of a classification algorithm using both *precision* and *recall* scores [17]. In Equation 3.5, the formula is defined as the harmonic mean between precision and recall. This provides a standard method of conveying the performance of a classification algorithm and thus, gives the benefit of comparing our future predictions and results with other research efforts in different domains or using different predictive models.

$$F - Score = 2\frac{precision * recall}{precision + recall} \tag{3.5}$$

3.4 Data Imputation and CLV Calculations (P5, P6)

While the well known Customer Lifetime Value models were introduced in chapters 1 and chapter 2, most CLV prediction algorithms are *theory* based models often associated with retention and acquisition. As a result, it is difficult to make concrete interpretations when trying to determine a model's suitability for real-world datasets. As was seen in equation 1.1 back in chapter 1, the CLV variables are customised at a customer level. However, in real world enterprise datasets, many of these data properties are hard to capture. In this case, some form of Variable Imputation is required to obtain an individual-level dataset: where parameters are per-customer and not broad global values. As the insurance policy is updated annually, to calculate the CLV, we must have all variables at a per-customer level.

3.4.1 P5: Data Imputation

Data Imputation is necessary to generate *customer retention* and *acquisition* values before applying CLV calculations. Our efforts up to this point ensure that the remaining attributes required for CLV calculations are available.

Customer retention. Based on the nature of policy renewals, the *current* customer retention value can be calculated on a year-to-year basis: a single value for all customers in a year. However, the CLV is a customer based calculation and thus, this value is imputed per-customer. We employed a combination of linear regression and decision trees to evaluate the baseline for customer retention. We also adopted dataset into the machine learning method neural network to predict a more accurate value for customer retention. Our approach and results are presented in chapter 6.

Customer acquisition. The customer acquisition value imputed answers how likely it is for a customer to purchase a new policy. This measure is more effective using the current customer dataset: customers who already hold a policy. The variable directly affects CLV calculations because the spend for this customer increases. Once again, the prediction of individual-level acquisition values used machine learning algorithms described in detail in chapter 6.

Once customer retention and acquisition are known, we can proceed to step WP6.

3.4.2 P6: CLV Prediction

To deliver accurate CLV predictions, it is necessary to identify and experiment with different CLV models. CLV modelling includes identifying the most suitable CLV formula for insurance data depending on our end-user requirement, using Machine Learning algorithms to predict CLV elements and using CLV to support business assessment and prediction.

After we identify the most suitable CLV formula for this customer dataset, inputs are necessary to populate the formula. A multivariate data mining model will be used to impute each missing element. Moreover, we will also create data marts to store CLV elements and generate CLV rules from created data marts. CLV calculations can only take place when we have all values we need. From the CLV modelling work packages, CLV elements will be transformed into machine readable queries, extracted from the database. Each element is combined across multiple dimensions from different tables, therefore using data marts to store data will be more efficient. The assessments and predictions will be generated from the business perspective.

There are two types of CLV prediction methods used in the experiments presented later in this dissertation: a Formula Based Approach, and a Probability Based Approach. The primary focus of this process is to compare the two approaches for CLV calculation. We begin by presenting the Formula Based Approach, which is the industry standard to provide a baseline set of results. Next, we present the probabilistic approach, which required a modification to the standard approach as it was necessary to *impute* some of the input variables. Finally, we validate (P6b) the computed CLV values using the dataset we created during process P4. The details are provided in chapter 6.

3.5 Summary

The purpose of this chapter was to provide a high level overview of our system architecture and the processes involved in constructing CLV. This helps to provide the context for our research and allows us to present the most critical components in depth, over the next few chapters. Because each of the components is reliant on the previous component, it is necessary to have the validation process within each component to make a better understanding of the CLV constructing journal. In the following chapter, we provide a specification of our integration model and present a detailed description of how CLV is predicted from the unified client record.

Chapter 4

Multi-Relationship Record Linkage

In the previous chapter, we presented our methodology for generating Customer Lifetime Values for a company's client base. This cannot start until organisations have a proper customer history for each of their clients but for many organisations, this does not exist. As explained in chapter 1, there are many reasons for this and it requires a process known as record linkage to piece together a customer's complete history. The chapter focuses on the detailed steps of record linkage and how we validate the creation of this dataset that underpins the rest of our work. The first step, the pre-processing of data is presented in section §4.1 followed by how we segment data in section §4.2. In section §4.4, we describe the clustering process which performs a *baseline* linkage of customers which is then fine-tuned using a rulebase in section §4.3 before we describe the overlapped methodology in section §4.5. Finally, we present a brief summary in section §4.6.

4.1 Data Preparation

The first step is to generate similarity measurements to identify records that belong to the same user, an action which requires a degree of pre-processing. In the section, we develop an approach to the problem in three phrases: attribute representation, similarity calculation and result evaluation. The attribute representation phase details which attributes are chosen and why. Similarity metrics are individually generated for numeric and string types. Both accuracy and performance are taken into consideration in the evaluation phrase.

4.1.1 Calculating Distance Measures

In the initial customer matching experiment, the similarity matrix is constructed to store the *distance* between each customer where a short distance value indicates that it is likely to be the same customer. In statistics and related fields, a similarity measure or similarity function is a real-valued function that quantifies the similarity between two objects. The size of the similarity matrix is a square of the total number of active customers in our current database. In reality, the similarity matrix is a symmetric matrix with a diagonal line equal to 0 as shown in figure 4.1, representing 2 equal halves. This reduces computation time as only one half of the matrix must be constructed. Thus, the time to complete our matrix computation is $O(\frac{n(n-1)}{2})$, where n is the overall number of customers.

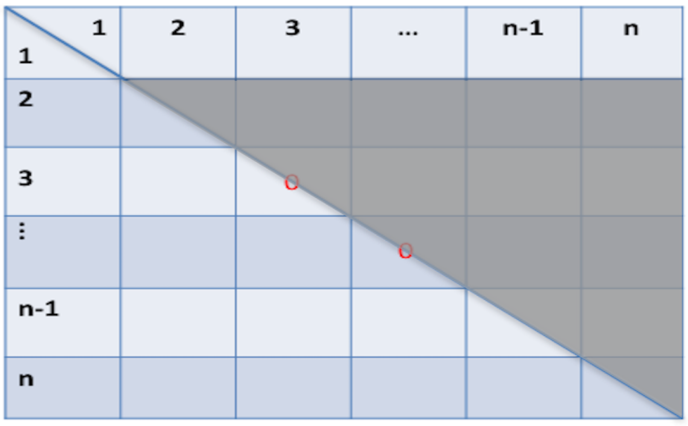


Figure 4.1: Similarity Matrix

The distance between every two customers will be generated during this step. For each dataset chosen, the customer record will have a series of attributes: $Attr_1$, $Attr_2$, ..., $Attr_n$. For all customers, there are two types of data involved: numeric data or string. To calculate numeric attributes, we use Euclidean Distance(D_e) [83]:

In Cartesian geometry, where $p = (p1, p2, \ldots, pn)$ and $q = (q1, q2, \ldots, qn)$ are two points in Euclidean n-space, then the distance $d(p, q)$ from p to q, or from q to p is given by the Pythagorean formula in equation 4.1.

$$\begin{aligned} \mathrm{d}(\mathbf{p}, \mathbf{q}) = \mathrm{d}(\mathbf{q}, \mathbf{p}) &= \sqrt{(q_1 - p_1)^2 + (q_2 - p_2)^2 + \cdots + (q_n - p_n)^2} \\ &= \sqrt{\sum_{i=1}^{n} (q_i - p_i)^2} \end{aligned} \tag{4.1}$$

To calculate these distance between 2 strings a and b, we use the Levenshtein distance $(lev(a, b))$ [60]. This measure of distance between two words, is the minimum number of single-character edits (i.e. insertions, deletions or substitutions) required to change one word into the other. Mathematically, the Levenshtein distance between two strings a and b (of absolute distance $|a|$ and $|b|$ respectively) is given by $\mathrm{lev}_{a,b}(|a|, |b|)$ as shown in formula 4.2, where $1_{(a_i \neq b_j)}$ means the indicator function is equal to 0 when $a_i = b_j$ and equal to 1 otherwise. The value of $\mathrm{lev}_{a,b}(i, j)$ is the distance between the first i characters of a and the first j characters of b.

$$\mathrm{lev}_{a,b}(i,j) = \begin{cases} \max(i,j) & \text{if} \min(i,j) = 0, \\ \min \begin{cases} \mathrm{lev}_{a,b}(i-1,j) + 1 \\ \mathrm{lev}_{a,b}(i,j-1) + 1 \\ \mathrm{lev}_{a,b}(i-1,j-1) + 1_{(a_i \neq b_j)} \end{cases} & \text{otherwise.} \end{cases} \tag{4.2}$$

The time complexity for calculating Levenshtein distance is $O(ij)$; where i and j are the lengths of the two strings to be compared. The distance between two customers is the sum of the individual attributes as shown in equation 4.3, where n + m (= A_{total}) are the cardinalities of numeric and string attributes respectively. A_{total} is the total number of attributes in this dataset; n is the total number of numeric attributes and m is for string attributes.

$$D_{total} = D_{e1} + D_{e2} + D_{en} + \ldots + D_{l1} + D_{l2} + \ldots + D_{lm} \tag{4.3}$$

A database is used to store the unique identifier (id) for customer x and y, together with the distance between individual attributes. The columns created in this result table are: x, y, D_{Attri_1}, D_{Attri_2}, $\ldots$, D_{Attri_n}. The benefit of this type of storage model is that we can now calculate distance by different combinations of attributes and thus, try different parameters for record linkage. The scipy (0.13.0b1) library was used to provide distance measures.

4.1.2 Dataset Description

Matching records for the same individual is not an easy task because different people may share the same information like common names and date of birth. Moreover, for the same person, their information may differ as he or she changes address. As the size of the recordset increases, this becomes increasingly likely. In addition, a real world dataset will generally have a lot of quality issues. If any field is not compulsory, people are more likely to ignore it and this is the primary reason for missing values in our data.

Five dataset configurations were used for record linkage in order to identify a suitable combination of attributes as well as measuring the distance matrix generation time for different types of data. DS2 to DS5 are subsets of DS1. We analyse the result and use it to make updates in our algorithm. The overview information for each dataset are listed below with a full description provided in Appendix A.1.

- **DS1: Customer information dataset.** We extracted and combined all customer information from the customer database, a total of 70 features.
- **DS2: FTI Attributes Dataset.** A common set of 23 attributes that industry use for this type of customer analysis.
- **DS3: Personal Detail Dataset.** We take 12 attributes, all of which are personal details.

- **DS4: Numeric Attributes Dataset.** A selection of 18 *numeric* attributes.
- **DS5: String Attributes Dataset.** A selection of 18 *string* attributes.

4.1.3 Dataset Selection

The time complexity to calculate the distance between every two customers is $O(n^2)$ where n is the total number of records. Due to finite capabilities of workstations and RAM, we divided the whole dataset into multiple segments. Execution times are shown in table 4.1, where: Rank shows datasets in terms of accuracy; Dataset identifies the data configuration used; No.Attributes, the attribute count; Attribute Type is the attribute mix; Block Time is the time to execute a single block of 10,000 records (randomly selected); and Exp. Time is estimated number of days it will take to run the overall 195,849 rows.

Note that comparison is pair-wised and thus, each block will include nearly 50 million comparisons. *Exp Time* is estimated and depends on whether or not we used parallel computing (with package *multiprocessing*) with 6 processes. To optimise the capability of our workstation, we divided the 195,849 records into 20 segments where each segment has 10,000 rows. In total, it is necessary to compute 190 full blocks and 20 half blocks to compute the half matrix required.

This experiment excludes the result for dataset DS1, because even with the small subset tested, there are too many NULL values inside this data. This is crucial as NULL values on the same attribute from different records would be regarded as equal with a distance value of 0. As a consequence, a pair of records with a lot of missing information, could be labelled as being the same customer. Shortly, we will explain how this problem was addressed.

Table 4.1: Execution Time

Rank	Dataset	No.Attributes	Attribute Type	Block Time	Exp. Time
1	DS3	12	String	25h16m23s	35
2	DS2	23	String&Num	30h28m31s	42
3	DS5	18	String	28h36m24s	39
4	DS4	18	Numeric	51m20s	1

As would be expected, the higher the dimensionality of the dataset, the more time that is required to compute the similarity matrix.

Some observations can me made even at this early point in our study of record linkage. Dataset DS3 has the best accuracy because all features are customers' details. For those customers' detail data, there are fewer null values. For those records who are not same customer but were recognised by our algorithm to be from the same user, it is probable that the records belong to different members of one family or that friends are living together. They might share information like mobile number, email, address etc.

The DS4 and DS5 datasets, both with 18 attributes, were used to compare the performance of numeric data and string data. From the result, the time cost for calculating string data is 30 times than numeric data.

The Rank column shows the dataset from the most to least accurate depending on which gives the greater number of matches when using the overall distance between pairwise records. We eventually settled on a distance threshold $T_{dist} = 300$ to validate the top 300 similar customer for the purpose of measuring the top 3% of the examined block. From the block we examined, dataset DS3 matched most records. It was necessary to choose a suitable subset to start, then generating and improving on this subset, before moving to the full size of the dataset. We decided to segment by region based on discussions with our industry partner. This was the approach used by commercial companies contracted to perform record linkage operations for other parts of the insurance organisation. Our candidate dataset contained 30,281 records, with all customers living in Dublin. The attributes set used was DS2 it contains all attribute from the top ranked DS3 dataset. In this way, we used a lot more attributes while also comprising the top ranked data subset.

4.2 Clustering Customers

The main issue with clustering, common to all research projects with large datasets, is that the computation time for the similarity matrix is very high. Our approach to reducing the computational cost was to divide the customer records into multiple segments and calculate a *multidimensional* similarity matrix. In effect, we compute n similarity matrices for each of the n attributes in our dataset. As will be shown later in this chapter, this provides us with a greater degree of flexibility when trying to optimise record linkage.

Table 4.2: County Attribute and Candidate Clusters

County	Description	NBR_of_Record	Province	Cluster
1206	Dublin	30281	Leinster	C1
1207	Dublin 1	20	Leinster	C2
1208	Dublin 2	35	Leinster	C2
1209	Dublin 3	500	Leinster	C2
1210	Dublin 4	382	Leinster	C2
1211	Dublin 5	765	Leinster	C2
1212	Dublin 6	415	Leinster	C2
1213	Dublin 6W	513	Leinster	C2
1214	Dublin 7	386	Leinster	C2
1215	Dublin 8	240	Leinster	C2
1216	Dublin 9	784	Leinster	C2
1217	Dublin 10	72	Leinster	C2
1218	Dublin 11	467	Leinster	C2
1219	Dublin 12	632	Leinster	C2
1220	Dublin 13	859	Leinster	C2
1221	Dublin 14	755	Leinster	C2
1222	Dublin 15	1404	Leinster	C2
1223	Dublin 16	1089	Leinster	C2
1224	Dublin 17	114	Leinster	C2
1225	Dublin 18	918	Leinster	C2
1227	Dublin 20	200	Leinster	C2
1229	Dublin 22	439	Leinster	C2
1231	Dublin 24	1026	Leinster	C2
1239	Longford	1513	Leinster	C3
1240	Louth	5610	Leinster	C3
1242	Meath	8426	Leinster	C3
1249	Westmeath	2677	Leinster	C3
1234	Kildare	8930	Leinster	C4
1236	Laois	2626	Leinster	C4
1243	Offaly	2311	Leinster	C4
1251	Wicklow	5388	Leinster	C4
1201	Carlow	2555	Leinster	C5
1235	Kilkenny	6331	Leinster	C5
1250	Wexford	8172	Leinster	C5
1204	Cork	29549	Munster	C6
1203	Clare	6919	Munster	C7
1233	Kerry	8125	Munster	C7
1238	Limerick	7798	Munster	C7
1247	Tipperary	4453	Munster	C7
1248	Waterford	6579	Munster	C7
1202	Cavan	3301	Ulster	C8
1205	Donegal	5139	Ulster	C8
1246	Monaghan	2895	Ulster	C8
1237	Leitrim	1208	Connacht	C9
1241	Mayo	5046	Connacht	C9
1244	Roscommon	3210	Connacht	C9
1245	Sligo	2047	Connacht	C9
1232	Galway	12745	Connacht	C10

It was decided to run trial experiments for record linkage using the most recent year's data. This year dataset comprised 195,849 records and 10 clusters were formed using the *County* identifier (*County*), with the largest county undergoing a further sub-division. Ideally each cluster will have a similar number of objects, but in fact, the number of records for individual counties are different. For this reason, we chose to merge some clusters by size and geographical position. Table 4.2 presents the details of each county, which includes the description, the number of records (NBR_of_Record), and which province and cluster it belongs to.

4.2.1 Match by Record

We ran a series of experiments to identify the best entity resolution method. From data identification, we detected that the value difference between attributes can be extensive. Moreover, as was discovered in section §4.1.3, Null values are negatively impacting on matching because by default, the distance between two Nulls is 0 which means the records with more missing data will outperform others. It was necessary to prevent this by *punishing* Null values. Furthermore, to identify the most suitable entity resolution approach, we incorporated a series of scenarios to calculate the distance between pairwise customers. These experiments were labelled using a process (Experiment a.b) shown in definition 4.1.

Definition 4.1. The different experiment are defined as $\mathcal{V}a.b$ using a Null value replacement method **a**, where:

$$\mathbf{a} = \begin{cases} 1, & \text{baseline experiment - without replacement} \\ 2, & \text{replacing null values by average} \\ 3, & \text{replacing null values by maximum} \end{cases}$$

The set of attribute values for **b** are shown as:

$$\mathbf{b} = \begin{cases} 0, & \text{Sum all attributes} \\ 1, & \text{Sum all attributes but not } CountryID \\ 2, & \text{Sum all attributes but not } CountryID \text{ and } OccupationID \\ 3, & \text{Sum all attributes but not } OccupationID \\ 4, & \text{Sum only 12 personal detail attributes} \end{cases}$$

The result for all experiments are presented in tables 4.3 to 4.5. In each table, the number of pairwise customers matched using a specific distance (Threshold) is shown. For example, in table 4.3, there are 29 paired customers with a distance of less than 10 for experiment 1.0 (v1.0). For customers with small distances (for example: >10), they are likely to be the same customer. To be precise, table 4.4 and table 4.5 now show matched customers where Null values were replaced by the average distance and replaced by the maximum distance respectively.

Table 4.3: Baseline Experiment

Threshold	v1.0	v1.1	v1.2	v1.3	v1.4
<10	29	29	34	27	140
11-20	94	96	171	135	433
21-30	223	233	497	379	1421
31-40	523	543	1716	1342	20271
41-50	3054	3316	24821	18683	351884
51-60	32503	37421	385288	281223	3494720

Table 4.4: Replacing Null with Average Distance

Threshold	v2.0	v2.1	v2.2
<10	0	1	1
11-20	1	26	30
21-30	2	101	160
31-40	7	282	553
41-50	11	469	1089
51-60	20	801	5531
61-70	127	8106	120919
71-80	1936	150469	2636589

Table 4.5: Replacing Null with Maximum Distance

Threshold	v3.0	v3.1	v3.2
<50	0	10	13
51-60	3	70	102
61-70	3	163	309
71-80	4	318	687
81-90	4	629	3181
91-100	21	4604	65891

Having analysed the results of all experiments, it is clear than when the distance threshold increases, more records are matched. From the result shown in experiment 1.b (b={0,1,2,3,4}), deleting $Occupations$ has more of an effect than deleting $CountyID$. The distance between two customers increased after the replacement of Null values: replacing by maximum had a greater effect than replacing by average. However, it is hard to identify the *influence* of each attribute as opposed to simply matching by D_{total}. This required a more fine-grained solution so that the effects of each attribute on the distance score, could be better understood.

Table 4.6: Match Accuracy

Experiment	Top 300	Top 1000
v1.0	83%	66%
v1.1	84%	66%
v2.0	10%	0%
v2.1	90%	82%
v3.0	6%	2%
v3.1	89%	78%

In table 4.6, we show the accuracy of matches for Top 300 and Top 1000 most similar customers for each experiment. It is clear that replacing Null values with distance average achieves the highest accuracy across all experiments. For this reason, replacement by average distance was chosen for subsequent experiments where better customer matching was sought.

4.2.2 Matching Client by Clustering

To match clients, we adapted an Agglomerative Hierarchical Clustering (AHC) [22] method, where a similarity matrix was again used to represent the distance between each pair of records. Given that customers should be in very small clusters, the process stops when a deliberately low threshold for distance (dissimilarity) has been reached. Our first point of difference with traditional AHC lies in our construction of the similarity matrix. We do not construct a single 2-dimensional matrix but instead compute a multidimensional matrix which enables us to examine distance measures across different variables. We chose this method due to poor results obtained when using a single aggregated distance measure across all variables. There are nine dimensions in our current similarity matrix as presented in table 4.7, with each dimension (matrix) given a specific label comprising $SM\_$ and the name of the attribute. This reference to similarity dimensions (or matrices) is also used in the rules presented in §4.3.1. The *SM_BirthDate* dimension captures the distance between Dates of Birth; $SM\_FirstName$ and $SM\_LastName$ for the first and last names; $SM\_Address$ for the distance between address strings; SM_Email, SM_Mobile, $SM\_HomePhone$, $SM\_WorkPhone$ and SM_Fax for the distance between each type of contact details. There are three types of matches required in our research: *Client* matches (records for the same client); *Family* matches (family members for a client); and *CoHab* (where non-family members at the same address). The supported dimensions for each type of match with the label: Required (Y);Not Required (N); Optional (O).

Table 4.7: Similarity Matrix usage in Relationship Matching

Similarity Matrix	Client	Family	CoHab
$SM\_BirthDate$	Y	N	N
$SM\_FirstName$	Y	N	N
$SM\_LastName$	Y	Y	N
$SM\_Address$	O	O	Y
SM_Email	O	O	N
SM_Mobile	O	O	N
$SM\_HomePhone$	O	O	N
$SM\_WorkPhone$	O	O	N
SM_Fax	O	O	N

4.3 Ruleset Based Matching

As part of our understanding about the effect different attributes had on the matching process, it became clear that a rule-based approach could could automate an improved matching algorithm. In this section, we describe a set of basic rules that were developed to fine-tune matching where a series of experiments helped us to understand the threshold levels to apply in each case. The end goal is a unified customer record containing three different relationships: records for the same client; records of family members; and those of cohabitants (domicile).

The workflow and various concepts are shown in figure 4.2. We begin by constructing the multi-dimensional similarity matrix by applying a similarity measure to each attribute. We then apply the rules which set distance thresholds to cluster according to the different relationships. Finally, we merge the related records into unified client records.

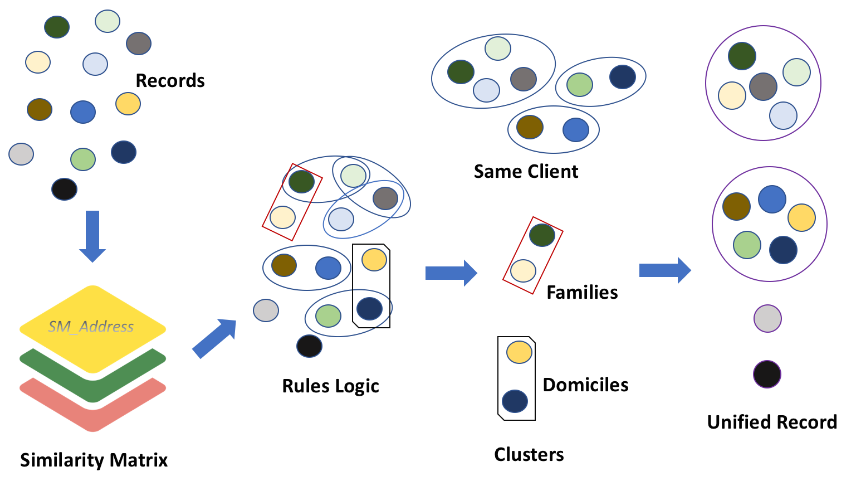


Figure 4.2: Matching Using Modified AHC

4.3.1 Client Rules

In this section, we provide definitions for the *client* rules used in the matching process. Each rule works with one or more *attribute* distance measures and a specified threshold.

Rule 4.1. *Client-Client_ Rule*
$[DOB\_Check]$ *and*
$([Full\_Name\_Check]^{*}$ *or* $)$ *and*
$[Contact\_Detail\_Check]$

The *Client-Client_ Rule* presented in Rule 4.1, *must* have 3 separate clauses, each separated by a logical *and* operator. All conditions must evaluate to true if records are to be clustered (matched). The condition $([Full\_Name\_Check]^{*}$ *or* $)$ will contain one or more clauses for $Full\_Name\_Check$, separated by a logical *or* operator.

Rule 4.2. *DOB_ Check*
$SM\_BirthDate[i, j] \leq T_{DOB}$

In Rule 4.2, the DOB_Check clause is specified as a Boolean statement. In this case, the similarity for records i and j are tested using the $SM\_BirthDate$ distance measures against a specified threshold value T_{DOB}.

Rule 4.3. *Full_ Name_ Check*
$[FirstName\_Check]$ *and*
$[LastName\_Check]$

In Rule 4.3, the $Full\_Name\_Check$ clause is a Boolean statement with two conditions $FirstName\_Check$ and $LastName\_Check$, separated by a logical *and* operator.

Rule 4.4. *FirstName_ Check*
$SM\_FirstName[i, j] \leq T_{FName}$

In Rule 4.4, the $FirstName\_Check$ clause is specified as a Boolean statement. In this case, the similarity for records i and j are tested using the $SM\_FirstName$

distance measures against a specified threshold value T_{FName}.

Rule 4.5. *LastName_ Check*
$SM\_LastName[i,j] \leq T_{LName}$

In Rule 4.5, the $LastName\_Check$ clause is specified as a Boolean statement. In this case, the similarity for records i and j are tested using the $SM\_LastName$ distance measures against a specified threshold value T_{LName}. Here, two similarity matrices $SM\_FirstName$ and $SM\_LastName$ along with their specified threshold values T_{FName} and T_{LName} in $Full\_Name\_Check$ clauses must be tested together.

The threshold application may be tested in multiple ways, the sum of T_{FName} and T_{LName} together is matched against the required threshold. This provides a great deal of flexibility, with an example shown in Example 4.1. Here, if the threshold set for $Full\_Name\_Check$ is 2, different combinations can be specified.

Example 4.1.

(SM_ FirstName[i,j] $\leq$ 0 and SM_ LastName[i,j] $\leq$ 2) or (SM_ FirstName[i,j] $\leq$ 1 and SM_ LastName[i,j] $\leq$ 1) or SM_ FirstName[i,j] $\leq$ 2 and SM_ LastName[i,j] $\leq$ 0)

Rule 4.6. *Contact_ Details_ Check*
$[Address\_Check]$ *or*
$[Contact\_Check]$

In Rule 4.6, the $Contact\_Details\_Check$ clause is a Boolean statement with two clauses $Address\_Check$ and $Contact\_Check$ separated by a logical *or* operator.

Rule 4.7. *Address_ Check*
$SM\_Address[i,j] \leq T_{AD}$

In Rule 4.7, the $Address\_Check$ clause is specified as a Boolean statement. In this case, the similarity for records i and j are tested using the $SM\_Address$ distance measures against a specified threshold value T_{AD}.

Rule 4.8. *Contact_ Check*
$SM\_Email[i,j] \leq T_{EM}$ *or*
$SM\_Mobile[i,j] \leq T_{MO}$ *or*
$SM\_HomePhone[i,j] \leq T_{HP}$ *or*
$SM\_WorkPhone[i,j] \leq T_{WP}$ *or*
$SM\_Fax[i,j] \leq T_{Fax}$

The $Contact\_Check$ rule checks the similarity for records i and j against a list of contact similarity metrics: SM_Email; SM_Mobile; $SM\_HomePhone$; $SM\_WorkPhone$; and SM_Fax. Each similarity matrix had its assigned threshold T_{EM} for SM_Email; T_{MO} for SM_Mobile; T_{HP} for $SM\_HomePhone$; T_{WP} for $SM\_WorkPhone$; and T_{Fax} for SM_Fax. An example of the *Client-Client_ Rule* setting the threshold for each clause to 3 is shown in Example 4.2:

Example 4.2.

(SM_ BirthDate[i,j] $\leq$ 3) and
((SM_ FirstName[i,j]$\leq$ 0 and SM_ LastName[i,j]$\leq$ 3) or
(SM_ FirstName[i,j]$\leq$ 1 and SM_ LastName[i,j]$\leq$ 2) or
(SM_ FirstName[i,j]$\leq$ 2 and SM_ LastName[i,j]$\leq$ 1) or
(SM_ FirstName[i,j]$\leq$3 and SM_ LastName[i,j]$\leq$ 0)) and
(SM_ Address[i,j] $\leq$ 3 or SM_ Email[i,j] $\leq$ 3 or
SM_ Mobile[i,j] $\leq$ 3 or SM_ HomePhone[i,j] $\leq$3 or
SM_ WorkPhone[i,j] $\leq$ 3 or SM_ Fax[i,j] $\leq$3)

4.3.2 Client-Family Rule

The purpose of the *client-family* rule is to detect where family members are also clients of the company. Once again, this rule has a specific *attribute* distance measures and a specified threshold.

Rule 4.9. *Client-Family_ Rule*
$[LastName\_Check]$ *and*
$[Contact\_Detail\_Check]$

The $Client - Family\_Rule$ which must contain two separate clauses, each separated by a logical *and* operator. Shown in Rule 4.9, it reuses a combination of Rule 4.5 and Rule 4.8 to specify each clause. An example of the $Family\_Rule$ setting the threshold for each clause to 3, is shown in Example 4.3:

Example 4.3.

(SM_ LastName[i,j]$\leq$ 3) and
(SM_ Address[i,j] $\leq$ 3 or SM_ Email[i,j] $\leq$ 3 or
SM_ Mobile[i,j] $\leq$ 3 or SM_ HomePhone[i,j] $\leq$3 or
SM_ WorkPhone[i,j] $\leq$ 3 or SM_ Fax[i,j] $\leq$3)

4.3.3 Client-Domicile Rules

The final rule seeks to determine if clients are domiciled with other customers of the company.

Rule 4.10. *Client-Domicile_ Rule*
$[Address\_Check]$

In Rule 4.10, the $Client - Domicile\_Rule$ rule tests for cohabiting clients by reusing Rule 4.7 to set the predicate. An example of the $Client - Domicile\_Rule$ assigning the threshold to 3 is shown in Example 4.4:

Example 4.4.

SM_ Address[i,j] $\leq$ 3

4.4 Validating Record Linkage

There are two major data entities (types of customer policies) involved: General Insurance and Life. In combining these, we had 195,849 individual client records from a year for our experiments. However, during the experiment we found there were 1,452 records duplicated (all attributes values are identical), and when removed, the number of records was 194,396. There has no standard record linkage validation on a specified real-world dataset. In most cases, manual verification is required and was necessary in our research.

4.4.1 Matching Results

Table 4.8: Rules Assisted Matching over Similarity Matrix Dimensions

Cluster	Records	CC0	CC1	CC2	CC3	CF0	CF1	CF2	CD0
C1	30281	1231	299	56	178	2650	851	1620	3814
C2	12015	149	36	9	48	731	362	519	912
C3	18226	635	143	42	87	2579	1027	1666	8691
C4	19255	622	179	38	80	2397	1308	1532	5784
C5	17058	882	166	44	97	3021	1706	1616	9684
C6	29549	1233	368	94	267	4492	6488	4020	11569
C7	33874	1402	355	85	264	5416	5236	4896	17690
C8	11335	478	117	39	52	1613	477	569	4493
C9	11511	297	80	31	45	1558	476	612	5365
C10	12745	382	88	20	77	1945	761	1066	10713
Accuracy		99.99%	98.25%	91.27%	17.66%	64.69%	26.53%	0.6%	2.6%

In table 4.8, we present the set of matching results which incorporate the rules presented in the previous section. The column *Cluster* lists the segments (clusters) generated using the segmentation method described earlier in §4.2. *Records* shows the total number of records in listed segment. Columns $CC0, CC1, CC2$ and $CC3$ present the result of matching by the $Client - Client\_Rule$ with thresholds 0, 1, 2 and 3 respectively. Columns $CF0, CF1$ and $CF2$, shows the result of the $Client-Family\_Rule$, again with threshold values of 0, 1 and 2 respectively, for both clauses in this rule. The final column $CD0$, is the result for $Client-Domicile\_Rule$ with a threshold 0. However, the accuracy for domiciled customers at 2.6% is very low, even though the threshold setting is zero.

For all rules, a larger threshold will always incorporate the matches found by the smaller threshold. However, for comparison reasons, we present the different threshold values to highlight the increased number of matches. For example, CC1 provides the count of matches gained above those matched for CC0. What we learned from these experiments was that setting the threshold $T = 2$, $Client - Client\_Rule$ (CC0 to CC2) achieved 99% accuracy and $Client - Family\_Rule$ (CF0 to CF1) achieved grater than 49% of accuracy. Conversely, as the threshold value is relaxed, the accuracy drops rapidly. To avoid high numbers of inaccurate records, the threshold limits for the $Client - Client\_Rule$ is 3 and for the $Client - Family\_Rule$ is 2.

For the $Client - Client\_Rule$ matching achieved good levels of accuracy. However, segmenting by county had the disadvantage of failing to match the customers who changed their address. Nevertheless, for both the $Client - Family\_Rule$ and $Client - Domicile\_Rule$, our insight as to why matches have a low accuracy level, is because a significant number of people enter addresses at different levels of detail, and this causes sufficient *confusion*, that they are labelled as false positives. This is mainly caused where customers do not enter a house number (everybody in that street lives at the same address) or address is at too high a level of granularity (district) which has the same effect. In some cases, the system regards the client's neighbours as the client themselves because of the same names and date of birth. To address both of these issues, the final iteration in our record linkage algorithm incorporates *overlapped blocking* when matching text fields. This is presented in the following section.

4.5 Overlapped Customer Linkage

Data quality issues with address attributes caused the decrease in accuracy of our matching algorithm. Thus, we sought to improve our algorithm by concatenated the address lines before the county level and using a form of *blocking* (described below) when comparing these customer attributes.

Data segmented by county had the downside of failing to match those customers that changed their address or had family relations in a different county. Thus, a desirable segmentation strategy would minimise the possibility of a customer having records in separate segments, as those records will never be matched. At the same time, the size of each segment should be small enough for efficient execution times. The most commonly used segmentation methods are clustering with vectoring attributes [6].

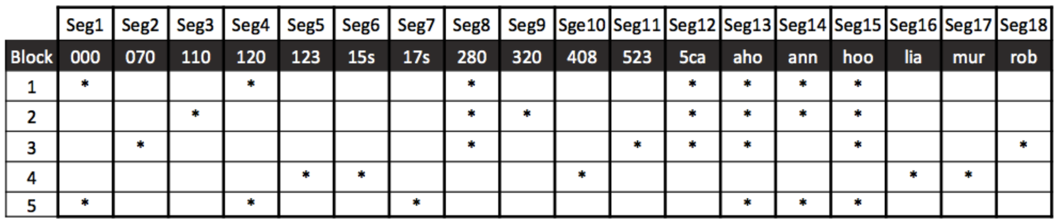
4.5.1 Optimisation by Overlapping Block

Similar to other approaches, we seek to match two different records for the same client. However, we must also identify family members as a parent or spouse may buy a policy for their child or partner. It is not unusual for this type of relationship (family member) to have a higher matching score than for two records the same client. Our approach also matches (non family member) co-habitants. In our research, we developed a hybrid segmentation method which reduced the matching (search) space between records.

Table 4.9: Synthetic Sample Records

Record	*BirthDate*	*FirstName*	*LastName*	*Address*	*Email*	*Mobile*	*HomePhone*	*WorkPhone*	*Fax*
1	12091990	anna	hood	5capelst	ahood21gmailcom	0876720000	013333280	null	null
2	11051964	ann	hood	5capelst	ahood21gmailcom	0860802320	013333280	null	null
3	07041993	robert	hood	5capelst	ahood21gmailcom	0897034523	013333280	null	null
4	12301992	liam	murphy	15silloguerdballymun	liam2murphygmailcom	0867723408	null	null	null
5	12071990	anna	hood	17sillogueroadballymun	annahood1gmailcom	353876720000	null	null	null

While attempting record linkage for a large dataset, most approaches (e.g. [29, 33]) to segmentation adopt a clustering approach that employs *blocking* and a form of vectorization for fast processing of the large pairwise matching required in their similarity matrix. Blocking involves the selection of a block (always small e.g. 3 chars) of consecutive characters which are used for distance matching. This can be illustrated using table 4.9 which contains 5 sample records after our pre-processing step. Customer records 1 and 5 refer to the same client where a mistake was made for dimension *BirthDate*. Customer records 1, 2 and 3 are family members with shared Contact (Dimension 5 to 9) information. Additionally, customer 4 lives with customer 5. Figure 4.3 allocates the sample records from table 4.9 into their respective segments (one of 18 possible segments) based on the block that represents each segment. Our *overlapping* approach is different to other approaches: if that block is found in any attribute in the same record, it is placed into that segment. Thus, a record can appear in more than one segment, e.g. Record #1 is placed into segments 1, 4, 8, 12, 13, 14 and 15.



	Seg1	Seg2	Seg3	Seg4	Seg5	Seg6	Seg7	Seg8	Seg9	Sge10	Seg11	Seg12	Seg13	Seg14	Seg15	Seg16	Seg17	Seg18
Block	000	070	110	120	123	15s	17s	280	320	408	523	5ca	aho	ann	hoo	lia	mur	rob
1	*			*				*				*	*	*	*			
2			*					*	*			*	*	*	*			
3		*						*			*	*	*		*			*
4					*	*				*						*	*	
5	*			*			*						*	*	*			

Figure 4.3: Segmentation by Blocking using Table 4.9

This was necessary as, in early tests using the DBSCAN clustering method [42], up to 30% of records for the same clients were in separate segments, meaning they could never be matched. On the other end of the scale, setting the distance to 13, all records were placed in the same cluster, meaning the number of matching operations was too large to compute.

In [29], the authors employed prefix blocking for attributes $FirstName$ and $LastName$ and other approaches included blocking for $Address$, $BirthDate$ and $Email$. This meant taking a block of n-characters from the *start* of each string for comparison purposes. For contact attributes, we employ suffix blocking. This meant taking a block of characters from the *end* of each string for attributes ($Mobile$, $HomePhone$, $WorkPhone$ and Fax). This had the advantage of avoiding issues with country and area codes where they may or may not exist. Our approach to blocking (prefix or suffix) is consistent across all experiments in §4.5.2. After segmentation, the matching algorithm presented in §4.3.1 is employed to deliver record linkage across the entire dataset. In our evaluation, we used the $scikit-learn$(0.18.1) library in experiment development.

4.5.2 Evaluating the Final Algorithm

Having arrived at a final version of our record linkage algorithm, it was necessary to ensure that it met our strict criteria for evaluation. We use 2 tables to present our results: table 4.10 presents the configuration details for each of 6 experiments while table 4.11 presents the total matches and accuracy for different thresholds across all 6 experiments.

The first column in table 4.10, *Exp*, is the label for the 3 sets of experiments, each with different configurations for the block length (*Block Length*). Columns 2-5 show the length assigned to the attributes: the second column represents the length for attribute *BirthDate* (*DOB*); *FN* and *LN* columns for $FirstName$ and $LastName$; the Contact column represents all the contact details including $Address$, $Email$, $Mobile$, $HomePhone$, $WorkPhone$ and Fax. A value between 3 and 6 indicates the block length for strings and `n/a` indicates that this attribute was not used in the experiment. The *Dims* column lists the number of similarity dimensions used in the matching process. *Records* refers to the size of the recordset involved in that experiment with the total number of segments created listed in the *Segment* column. The total number of records compared for a single dimension of the similarity matrix are shown in *Comparison* and finally, the number of records in the largest segment is shown in *Max*.

Table 4.10: Experiment Configurations and Matching Requirements

	Block Length								
Exp	DOB	FN	LN	Contact	Dims	Records	Segment	Comparison	Max
1.1	3	3	3	6	9	1,168,406	311,150	843,109,791	17,292
2.1	n/a	n/a	3	5	7	808,396	137,177	185,526,138	8,026
2.2	n/a	n/a	3	6	7	778,666	271,215	137,298,018	4,469
3.1	n/a	n/a	n/a	4	6	583,776	42,016	186,793,296	13,090
3.2	n/a	n/a	n/a	5	6	583,924	141,549	94,824,563	8,026
3.3	n/a	n/a	n/a	6	6	584,290	268,451	46,904,079	4,455

The experimental goal was to achieve the maximum number of matches while identifying any limitations caused by threshold settings for each rule, in other words, on measuring matching *accuracy*. The results in table 4.10 show that decreasing the length during blocking will decrease the number of segments created but an increase in segment size will see an increase in the number of comparisons required. Experiment running time is dependent on the number of comparisons in each experiment.

For all 6 experimental configurations, as with those presented earlier in table 4.8, we validated the results for each rule. Presented in table 4.11 are: 4 client matching experiments (labelled with rules CC0, CC1, CC2 and CC3), 3 client-family experiments (CF0, CF1 and CF2) and 1 experiment for co-habitants (CD0). However,

Table 4.11: Results of Experiments by Threshold

	Exp1.1		Exp2.1		**Exp2.2**		Exp3.1		Exp3.2		Exp3.3	
Rules	Match	Accuracy	Match	Acc %	**Match**	**Acc %**	Match	Acc %	Match	Acc %	Match	Acc %
CC0	9609	99.95	9609	99.95	**9609**	**99.95**	9609	99.95	9609	99.95	9609	99.95
CC1	10146	99.70	10144	99.72	**10144**	**99.72**	10116	99.73	10113	99.73	10109	99.73
CC2	10434	98.73	10422	98.84	**10418**	**98.88**	10373	98.92	10354	99.04	10339	99.10
CC3	14649	72.08	13688	77.14	**13493**	**78.26**	12379	84.86	12054	87.09	11776	89.06
CF0	30330	72.14	30330	72.14	**30330**	**72.14**	30330	72.14	30330	72.14	30330	72.14
CF1	36057	64.12	35877	64.44	**35856**	**64.48**	32924	68.59	32692	68.99	32533	69.24
CF2	58754	39.52	57330	40.50	**56695**	**40.96**	40311	56.24	38775	58.39	37656	60.04
CD0	13270	15.59	13270	15.59	**13270**	**15.59**	13270	15.59	13270	15.59	13270	15.59
Total	86673	41.36	84288	42.53	**83458**	**42.95**	65960	53.43	64099	54.93	62702	56.08

unlike table 4.8, in order to determine the optimal thresholds, we validated the result in an aggregated fashion (for example, CC1 will include all matches of CC0).

The final row *Total* represents the total number of matches for all matching experiments (sum of CC3, CF2, and CD0) for the appropriate *Exp.* The Accuracy (*Acc %*) for the total is the true accurate matches across all experiments divided by the *Total.*

- As expected, applying a very low threshold (distance value) will result in very high accuracy. Increasing the threshold will match more records but will, as a result, reduce the accuracy. In general terms, the number of matches increases, row by row, within each matching category.

- For all blocking experiments (1.1 to 3.3), where the threshold is set to 0 (CC0, CF0 and CD0), identical records are matched and thus, the same level of accuracy is achieved.

- Setting the threshold to zero (CC0, CF0 and CD0) will override (has too much effect on) all experimental configurations: neither blocking algorithms nor matrix usage has any effect.

- If we look across the experiments, when the matching criteria is more strict (reduction in attribute comparisons), matches decrease, with the accuracy improving. For Client-Client matching with distance threshold of 2, Exp1.1 detects 10,434 matches with an accuracy of 98.7%. However, with a similar accuracy of 99%, Exp3.3 loses 95 records (10,339). This appears to indicate a strong case for using contact details only.

- Overall, Exp2.2 was chosen as best because it included all the accurate matches and is efficient while constructing the similarity matrix. The number of true matches can be calculated by multiplying the number of matches (*Match*) by the accuracy percentage (*Acc %*). The total of true matches in Exp2.2 is 35848 ($Total \times Acc\%$). Across three matching rules: 10559 ($CC3 \times Acc\%$) accurate matches identified by the *Client-Client_ Rule (C-C)*; there are 23220 ($CF2 \times Acc\%$) true matches identified from *Client-Family_ Rule (C-F)* and 2069 ($CD0 \times Acc\%$) from *Client-Domicile_ Rule (C-D)*.

- By employing this final experimental configuration (updated segment method and addresses), and comparing with the result in table 4.8, the algorithm now matches 750 more using $Client-Client\_Rule$, and 1143 for $Client-Family\_Rule$. In addition, the accuracy for each rule with associated threshold has increased.

The result for the unified records is shown in table 4.12. Columns 2-4 represent the 3 types of matches: the *C-C* match, *C-F* match and *C-D* match. *Y* indicates if there are one or more matches for that match type and *N* for no relationship in this type. *Records* shows the number of records for that combination. In brief, there are 8 combinations and we can highlight some findings from the data regarding all the combinations. *Combination 1* for clients who are single policy holders; *Combination 2* to *Combination 4* are clients who have multiple policies for themselves or one for themselves and one or more policies for families or co-habitants; *Combination 5* to *Combination 7* are the clients involved in two types of relationships; finally, *Combination 8* are clients who had all three types of relationship.

In total there are 162,929 unified client records for a validation dataset of 194,396. Additionally, 30% of clients satisfied at least one of the relationship types.

4.5.3 Analysis and Findings

From table 4.11, Exps 1.1, 2.1 and 2.2 performed best in terms of detecting most matches. The total figure, calculating by adding the best performing threshold experiments (CC3, CF2 and CD0) ranges between 83,458 and 86,673 although

Table 4.12: Unified Client Records

Combination	C-C	C-F	C-D	Records
1	N	N	N	137,114
2	Y	N	N	6,780
3	N	Y	N	14,174
4	N	N	Y	1,383
5	Y	Y	N	2,936
6	Y	N	Y	335
7	N	Y	Y	148
8	Y	Y	Y	59

accuracy drops when detecting high numbers of matches. Of these, Exp2.2 is the most efficient due to the far lower number of comparisons required (see table 4.10). This is to be expected as the blocking length increases and number of attributes reduced. Note that the overall accuracy is affected by the low accuracy for co-habitants.

It is useful to note the numbers of *dimensions* used for matching (as opposed to segmenting) when discussing these results. In Client-Client matching, 4 dimensions are used; in Client-Family matching, 2 are used and for matching co-habitants only 1 dimension is used. Thus, the quality of matching will inevitably decrease as we discuss the different types of matches.

Our related research highlights the many approaches to record linkage and it is no surprise that, using a combination of these techniques, the *Client-Client_Rule* performance has the best accuracy across matches. The 0.05% (5) false matches that occurred in CC0, were as a result of the poor data quality for the *address* attribute. When providing address information, only 49% of clients provided the address detailed to door number and thus, all clients on the same street would normally be matched. The same quality issue for *Address* will result in false hits across all types of matches even where the distance threshold is set to 0.

If we look at experiments using all 9 attributes (and dimensional matrices) and the distance threshold is set to 1 (CC1) for all, then matches for all experiments increased by around 500 with a slight drop in accuracy to 99.7%. In effect, this means an extra 23 false matches (Exp2.2) and these were found to be as a result of the client misspelling data somewhere in the 9 attributes. Again looking at Exp2.2, the reason 512 more records were matched between CC0 and CC1 is mainly due to misspelling names. In general, for Client-Client matches, there is a dramatic drop in accuracy when the distance threshold is increased from 2 to 3. As reported in the previous section summary, experiment 2.2 with threshold set to 2 (CC2) is the best configuration for *Client-Client* matching.

The *Client-Family_Rule* is generally not part of record linkage research. As expected, in sparse datasets (datasets with low numbers of client-client matches), the system detected more Client-Family matches. Interestingly, the optimum distance threshold is different. While we still have a significant change with threshold settings 2 and 3, there is enough deterioration in results between 1 and 2 to select a threshold setting of 1 (CF1). However, in Exp1.1, there were 30,330 matches detected with an accuracy of 72.14%. By increasing the distance threshold to 1, while this detects an extra 5,727 records, only 1,239 were accurate resulting in a drop in overall accuracy to 64.12%.

The *Client-Domicile_Rule* did not perform well either on accuracy nor on the number of true matches. The accuracy for co-habitants is very low even though the threshold was set to 0. The poor quality of *Address* is problematic for this match type, because $SM\_Address$ is the only similarity matrix used in this rule. Our fuzzy matching (threshold greater than 0) can handle abbreviations like 'rd' for 'road' and 'st' for 'saint' in the *Client-Client_Rule* and *Client-Family_Rule* only because those rules required a higher dimensionality (used additional similarity metrics). In summary, while the number of false hits is high, it succeeded in providing a new dimension to the relationship graph for our industry partner.

4.6 Summary

In this chapter, we used real world customer datasets from the insurance sector with the goal of uniting client records by: connecting all records (various policy data) for the same client; connecting clients to family members (where both have policies);

and connecting clients with co-habitants (where the co-habitant is also a client).

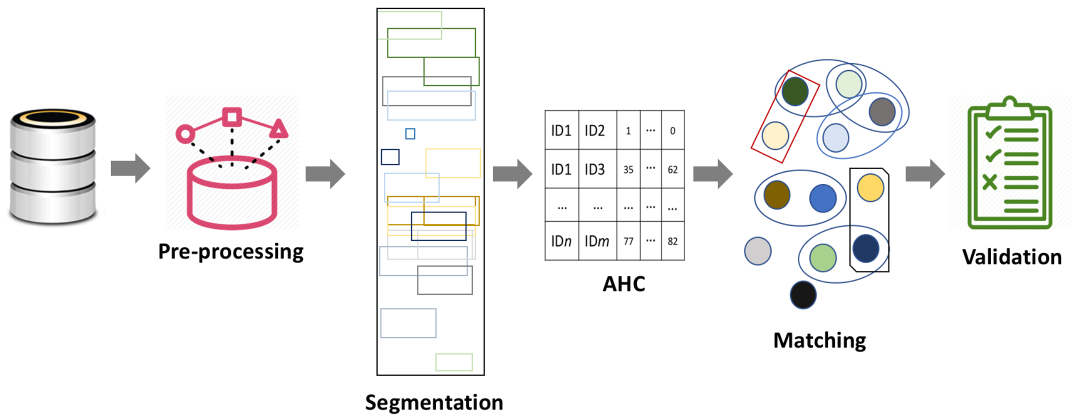


Figure 4.4: Pre-processing, Segment, Comparing and Matching Client Records

The approach described in this chapter comprised 5 processes: pre-processing; segmenting the recordset; application of the matching algorithm; using a ruleset to improve matching results; and validation, with an overview of our system architecture shown in figure 4.4. Using this system, we reduced the matching complexity in a manner that kept *matching* records in the same segment. Additionally, we had very positive results when comparing client-to-client data; *quite positive* results when matching clients with family members. As data is never clean, this is a significant task, even for only relatively large datasets. However, the goal of this chapter, the unified customer record, is crucial: without this we will not have the datasets necessary to continue with clustering records and imputing missing data.

Chapter 5

Customer Classification

In the previous chapter, we presented a methodology to link parts of a customer's history in order to create a complete customer record. That initial phase of our research must subsequently be followed by a mechanism to classify customers into categories according to their *value* to the organisation. Furthermore, this process must be automated due to the large numbers of customers involved. In this chapter, we present this method for classification of customers into broad categories. As this process is fully automated, a robust validation mechanism is provided as part of the overall process, in order to both expose how we validate this work, and to fully understand the results we have obtained.

The first step, described in section §5.1, requires taking the unified record constructed in chapter 4 through a transformation process so that it is more suited to it to a data mining operations and predictive algorithms. This process is generally regarded as *feature extraction.* In section §5.2, this dataset is then subjected to different *feature selection* methods to determine which set of features provides the most accurate results. The cluster process used to classify customers is presented in section §5.3 with the validation methodology presented in section §5.4. In section §5.5, we provide a detailed discussion of the results of our classification process together with our own insights as to implications of our results. Finally, section §5.6 presents our conclusions.

5.1 Data Transformation

In this section, we describe our data model transformation process which as stated above, is a form of feature extraction. After the record linkage process described in the previous chapter, the *customer* recordset is now annotated with links while data instances remain at the *policy* level. The dataset reduced from 500,859 records to 387,951 unified customer records, meaning 112,908 policies were matched to existing customers. The purpose of the transformation process is to create a dataset with a single instance per customer and with variables that are appropriate for machine learning algorithms. In the literature, we found that a number of variables were common across research projects ([5, 18, 64, 77, 114]): Age, County (Address), Gender, and Margin. Given our access to raw data over a 6-year period, we used the opportunity to include and test a number of less commonly used variables: AdjAmt, AdjCT, CCNbr, CFNbr, Disct, Gap, Maxh, and YNbr which explained in table 5.1.

Table 5.1: Classification Variable Descriptions

Variable	**Description**
AdjAmt	Cost associated with adjustments required by the customer.
AdjCT	Number of adjustments made by the customer.
Age	Customer Age.
CCNbr	Number of policies a customer holds.
CFNbr	Number of policies held by a customer's family member.
County	County code part of the customer's address.
Disc	Discount rate the customer receives, based upon each policy.
Gap	Indicates if the customer is a returning customer.
Gender	Customer Gender.
Margin	Net income generated by the customer.
Maxh	Maximum number of policies a customer held for any calendar year.
YNbr	Total number of years a customer held at least one policy.

In table 5.1, we present the set of variables generated from the transformation process. While some of the variables require no further description, there are some semantics associated with others, that benefit from closer inspection.

- **AdjAmt:** This variable represents the cost (in monetary terms) associated with adjustments a customer made during their tenure (AdjCT), e.g. a policy modification due to a change of car. This measure is highly correlated to customer retention in the insurance industry ([103]): policy modifications are more likely to affect the premium (policy price). Where customers leave after a policy modification, AdjCT and the AdjAmt are very important.

- **CCNbr:** This variable is cumulative and highly correlated to *Margin*.
- **CFNbr:** The number of policies held by a customer's family members. It is assumed that this measure has some impact on the customer's spending or retention behaviour.
- **Disct:** The discount rate the customer receives, based upon each policy. The discount is a rate calculated by $1 - Value_p/Value_b$, where: $Value_b$ is a basic calculation on a policy's worth; and $Value_p$ represents the *actual* price they were charged. For each policy and year, a discount rate is applied. The Discount present in the dataset is an *average* rate according to whether a policy is held for multiple years or a client had multiple policies. This is a crucial variable in determining the retention and acquisition figures for a customer.
- **Margin:** The *net income* generated (purchased) by the customer. Margin is the total policy spend over the 6 years. This is very common variable for segmentation and is the *Monetary* variable in RFM model.
- **Maxh:** The maximum number of policies a customer held for any calendar year. This is a measure to capture the customer buying behaviour in the short term (a single year) and differs from *CCNbr*, which captures the entire customer record.
- **YNbr:** The total number of years a customer held at least one policy. The value measures customer loyalty for the company, which is a very important variable for CRM. This variable is the same as *Frequency* in the RFM validation model.

When dealing with new datasets, it is good practise to perform a descriptive summary to obtain an overall picture of the data. Table 5.2 presents this summary for each of the 12 variables, where: row **mean** presents the average of that variable; **std**

Table 5.2: Descriptive Summary for Variables.

	AdjAmt	AdjCT	Age	CCNbr	CFNbr	County	Disct	Gap	Gender	Margin	Maxh	YNbr
mean	72.06	1.61	52.13	1.29	0.05	N/A	0.59	0.08	N/A	1336.41	1.16	2.57
std	300.08	2.34	16.47	0.65	0.23	N/A	0.01	0.39	N/A	1109.13	0.41	1.61
min	0	0	18	1	0	N/A	-4.44	0	N/A	33	1	1
25%	0	0	39	1	0	N/A	0.52	0	N/A	586	1	1
50%	0	1	51	1	0	N/A	0.63	0	N/A	996	1	2
75%	34	2	64	1	0	N/A	0.69	0	N/A	1756	1	3
max	33562	111	117	33	3	N/A	0.85	4	N/A	22696	22	6

the standard deviation; **min** the minimum value; and **max** the maximum values for that variable. In addition, **25%**, **50%** and **75%** represent the 25th, 50th, and 75th percentiles respectively. Finally, as the dataset reflects a period of 6 years, the maximum value of YNbr is 6. The variables County and Gender are categorical and as such do not have aggregate statistics such as mean and standard deviation (std). This explains the presence of `N/A` (not applicable) values in table 5.2.

From this analysis, we can see that all variables are numeric but with very different ranges of values. The difference (min to max) in Margin and AdjAmt is very high in comparison to all others. For some variables, the percentiles show little or no increase: CCNbr, CFNbr, Gap, Maxh variables change only at the top 25th percentile (max). Given the broad range of variables, it is crucial that one variable does not overly influence algorithms and thus, the dataset was normalised.

5.2 Variable Selection

The quality of the clusters that emerge from machine learning algorithms is greatly influenced by the set of variables considered for input. Therefore, it is a crucial step to identify the set of input variables that maximise the performance. In order to determine which variables should be part of the feature set, it is necessary to score and rank ([40]) all defined variables so that it is possible to discern which variables positively contribute to the performance, and those that do not. As part of this step, we adopted two popular approaches: correlation coefficient and principal component.

5.2.1 Correlation Coefficient Analysis

The Correlation Coefficient as defined in [63], quantitatively describes the degree to which two variables are (corr-)related. The correlation coefficient formula used in our approach is presented in Equation 5.1, where: CC_{xy} represents the correlation value between the (pair of) variables x and y; s_x and s_y are the sample standard deviations, and s_{xy} is the sample covariance.

$$CC_{xy} = \frac{s_{xy}}{s_x s_y} \tag{5.1}$$

Table 5.3 shows the correlation values between all variables presented in the previous section and in table 5.1, where a correlation value ranges between 0 and 1. The higher the value for the correlation between two variables, the more the values for such variables are related. For instance, the two most correlated variables are CCNbr and Maxh, because their correlation value, 0.79, is higher than all others; conversely, CCNbr has no relation at all with AdjAmt, since the correlation value is 0. Table 5.3 also has an additional column for All Correlation Coefficient (**ACC**): the values in this column represent a variable's correlation with *all other* variables. It provides a score for each variable by summing its correlation value to provide an overall ranking for variables.

Table 5.3: Correlation Coefficient Results

	AdjAmt	AdjCT	Age	CCNbr	CFNbr	County	Disc	Gap	Gender	Margin	Maxh	YNbr	ACC
AdjAmt	1												0.99
AdjCT	**0.47**	1											2.04
Age	0.04	0.12	1										0.71
CCNbr	0.00	0.26	0.04	1									2.52
CFNbr	0.00	0.01	0.03	0.25	1								0.39
County	0.00	0.01	0.01	0.01	0.04	1							0.43
Disc	0.04	0.18	0.22	0.04	0.02	0.07	1						0.82
Gap	0.03	0.07	0.00	**0.33**	0.01	0.00	0.03	1					0.67
Gender	0.00	0.00	0.02	0.00	0.00	0.21	0.01	0.00	1				0.27
Margin	0.23	**0.39**	0.02	**0.45**	0.00	0.03	0.05	0.06	0.00	1			2.4
Maxh	0.07	0.19	0.08	**0.79**	0.02	0.01	0.09	0.05	0.00	**0.45**	1		1.98
YNbr	0.10	**0.33**	0.12	**0.34**	0.01	0.04	0.06	0.09	0.00	**0.71**	0.25	1	2.05

It can be said that variables exhibiting high correlation values carry the same information and are thus, redundant. Data from redundant variables do not contribute to improve the quality of the results of the clustering algorithms and can therefore, be ignored, i.e. removed. The correlation between two variables can be defined as high when exceeding the 0.3 mark threshold, as per [4]. Correlation values exceeding the set threshold are highlighted in table 5.3, which in turn identifies highly correlated variables in associated columns and rows. Given two variables exhibiting a high correlation value, one of them can be removed. Algorithm 1 identifies all variables that do *not* contribute to the clustering performance, returning only significant variables.

The first step in the algorithm is to identify all variables exhibiting high correlation with any other, ln. 4, and, for each variable, to count with how many times its correlation value with another variable exceeds the set threshold ($T = 0.3$), ln. 5. The second step is to isolate the variables with the highest correlation count, after having found such max value, ln. 11 and ln. 15. Finally, the algorithm identifies the variables to retain, and update the variable selection in $\mathcal{V}_{\mathcal{S}}$, ln. 19. Briefly, the latter step implements the following logic: given two variables with same highest number of (high) correlation occurrences with other variables, eliminate the one with higher ACC. The algorithm repeats until there is at least a pair of variables that exhibit high correlation, that is exceeding the threshold T, ln. 3.

When applied to the variables and correlation values in Table 5.3, the selection algorithm identifies the following unnecessary variables: Margin, AdjCT, and CCNbr.

5.2.2 Principal Component Analysis

Principal Component Analysis (PCA) is predominantly used when we have a large set of variables. PCA [25] automatically reduces the variable set while retaining most of the variation in the dataset. In brief, PCA will take the the full set of variables and generates a set of *dimensions*, constructed from this original variable set. Thus, the goal of this process is to identify the best set of dimensions.

$$\begin{aligned} YY^T u_k &= \lambda_k u_k, \\ Y^T Y v_k &= \lambda_k v_k, \\ v_k &= Y^T u_k / \lambda_k^{1/2} \end{aligned} \tag{5.2}$$

Algorithm 1: Variables selection.

```
Input  : Set $\mathcal{V}_{\mathcal{A}}$ of all variables , threshold $T = 0.3$, correlation function $\mathcal{F}_{\mathcal{CC}}$,
         correlation ratio function $\mathcal{F}_{\mathcal{ACC}}$
Output : Set $\mathcal{V}_{\mathcal{S}}$ of selected variables.
1  $\mathcal{V}_{\mathcal{S}} \leftarrow \mathcal{V}_{\mathcal{A}}$
   /* Until a pair of high correlated variables exists                    */
2  while $\exists$ $x,y \in \mathcal{V}_{\mathcal{S}}$ such that $\mathcal{F}_{\mathcal{CC}}$ $(x,y) > T$ do
       /* For each high correlated variable, count high correlations
          with other variables                                            */
3      $\mathcal{V}_{\mathcal{HC}} \leftarrow \emptyset$
4      . foreach $v,\ w \in \mathcal{V}_{\mathcal{S}}$ do
5          if $\mathcal{F}_{\mathcal{CC}}$ $(v,w) > T$ then
6              if $\exists \langle\ v,c\ \rangle \in \mathcal{V}_{\mathcal{HC}}$ then
7                  $\mathcal{V}_{\mathcal{HC}} \leftarrow \mathcal{V}_{\mathcal{HC}} \setminus \{\ \langle\ v,\ c\ \rangle\ \}$
8                  $\mathcal{V}_{\mathcal{HC}} \leftarrow \mathcal{V}_{\mathcal{HC}} \cup \{\ \langle\ v,\ c+1\ \rangle\ \}$
9              else
10                 $\mathcal{V}_{\mathcal{HC}} \leftarrow \{\ \langle\ v,1\ \rangle\ \}$
       /* Find highest number of occurrences                              */
11     $o^{max} \leftarrow 0$
12     . foreach $\langle\ v,c\ \rangle \in \mathcal{V}_{\mathcal{HC}}$ do
13         if $c > o^{max}$ then
14             $o^{max} \leftarrow c$
       /* Find variables with highest occurrences                         */
15     $\mathcal{V}_{\mathcal{MO}} \leftarrow \emptyset$
16     . foreach $\langle\ v,c\ \rangle \in \mathcal{V}_{\mathcal{HC}}$ do
17         if $c = o^{max}$ then
18             $\mathcal{V}_{\mathcal{MO}} \leftarrow \{\ v\ \}$
       /* Among correlated variables with highest occurrences, retain
          those with lower ACC                                            */
19     $\mathcal{V}_{\mathcal{R}} \leftarrow \emptyset$
20     . foreach $v,\ w \in \mathcal{V}_{\mathcal{MO}}$ do
21         if $\mathcal{F}_{\mathcal{CC}}$ $(v,w) > T$ then
22             if $\mathcal{F}_{\mathcal{ACC}}$ $(v) > \mathcal{F}_{\mathcal{ACC}}$ $(w)$ then
23                 $\mathcal{V}_{\mathcal{R}} \leftarrow \mathcal{V}_{\mathcal{R}} \cup \{\ w\ \}$
       /* Refine set of selected variables                                */
24     $\mathcal{V}_{\mathcal{S}} \leftarrow \mathcal{V}_{\mathcal{S}} \setminus (\mathcal{V}_{\mathcal{MO}} \setminus \mathcal{V}_{\mathcal{R}})$
25 return $\mathcal{V}_{\mathcal{S}}$
```

We begin with a brief overview of PCA calculations: $X = (x_1, ..., x_n)$ represents the original data matrix; $Y = (y_1, .., y_n)$ where $y_i = x_i - \overline{x}$, and $\overline{x}$ is the mean of X. The principle direction u_k and principle component v_k, are the eigenvectors satisfying equation 5.2. Finally, YY^T is the covariance matrix computed using equation 5.3:

$$YY^T = \sum_i (x_i - \overline{x})(x_i - \overline{x})^T \tag{5.3}$$

The Singular Value Decomposition [113] for Y is presented in equation 5.4. The variables in v_k are the projected values of the data points on the principle direction u_k. The eigenvector presented in figure 5.1 shows for each PC, the request partial of each variable.

$$Y = \sum_k \lambda_k u_k v_k^T \tag{5.4}$$

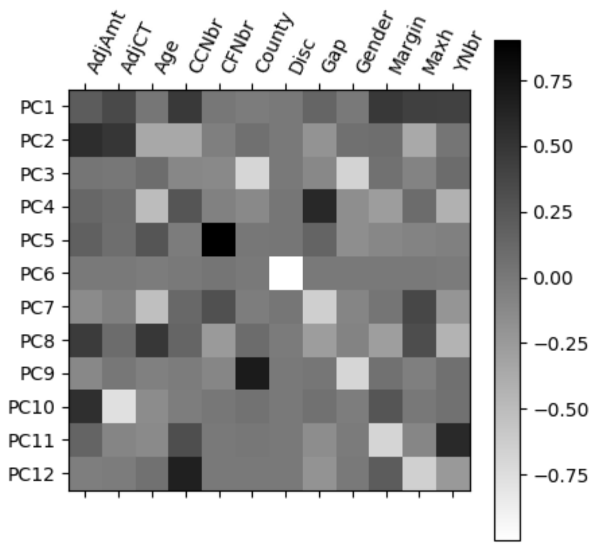


Figure 5.1: Highlighting Variance by Different Principle Components

As principle components (PCs) each have a different Variance Ratio, we investigated the importance of components by the Variance Ratio of each variable, shown in figure 5.2. The x axis contains the list of components and the y axis shows the percentage of explained variance. From figure 5.2, each bar shows the Variance Ratio for the displayed component. From the graph, we it is clear that PC1 is the most important component with a variance ratio of 23.6%. For those components like PC12 (1.26%), it can be removed because they will not have a high impact during predictions.

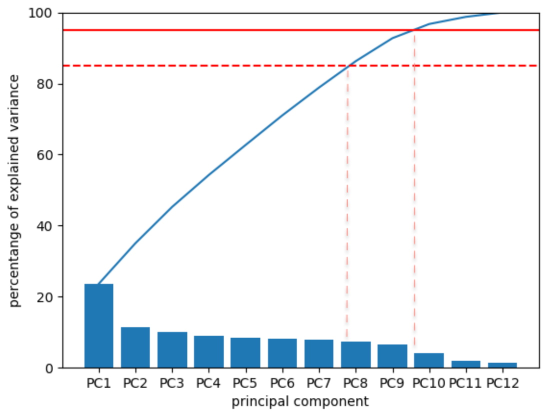
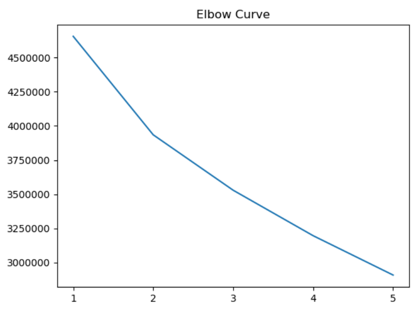


Figure 5.2: Explained Variance by Different Principle Components

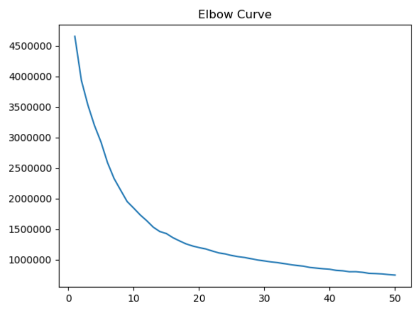
The curve also shows that by summing PCs, the Variance Ratio will increase until 100%. When applying the most popular Components with Variance Ratio 95% and 85%, we can see for the 95%, 10 components are selected, for 85%, the count is 8. These components were selected for the clustering algorithm in the next section and will form part of the discussion in section §5.5.

5.3 Clustering Customers

It was decided to run the simple and popular k-means clustering algorithm to try to determine if good customers could be placed inside the same cluster. The goal was to try to find the optimum number of clusters for customer segmentation. From [99], the number of classes distinguished is usually taken to be between three and five. In this case, and for both methods, we ran experiments with the number of clusters between 2 and 5 inclusive. As part of these experiments, We applied the internal error functions, Elbow Curve ([9]) and Silhouette Score ([97]), to test the k values for clustering methods.



(a) Elbow Curve for k = 5



(b) Elbow Curve for k = 50

Figure 5.3: Elbow Curve Test

The Elbow curve in figure 5.3(b) shows the optimal k for k = 1 to 50 configurations and appears to indicate an optimal value around 14. It is often the case in clustering algorithms that a high number of clusters records the best (lowest) error function score. In practice however, we can never have 14 distinct classifications for customers, especially where we are simply looking for broad categories such as *Good, Bad, Average*. Thus, if we examine figure 5.3(b) which uses a more fine-grained scale, it is clear that 5 is a good value for k but perhaps k=2 also shows an indication that the error value is starting to flatten.

Table 5.4: The Silhouette Score for 5 Random Tests.

k	Test-Run 1	Test-Run 2	Test-Run 3	Test-Run 4	Test-Run 5
2	0.45	0.48	0.46	0.48	0.47
3	0.31	0.30	0.33	0.31	0.28
4	0.33	0.38	0.36	0.31	0.35
5	0.33	0.33	0.36	0.34	0.30

The result for the Silhouette Score for k values between 2 to 5 is shown in table 5.4. Because of the computing requirements for tests using half a million records, not all data can be captured into the memory when calculating the Silhouette Score. Thus, we ran five tests, randomly selecting 30,000 records for each test. From the results we can see that for all test runs, experiments where k=2 outperform all others. While this provides a useful input into a more robust evaluation, it does not preclude the need for that more detailed validation and this is presented in the following section.

5.4 Validating Clusters

Similar to the previous chapter where we developed a process that required a level of validation, the clustering step presented in this chapter also requires a level of validation. As a result, we developed a strategy to evaluate the customer clusters obtained from a wide range of experimental configurations. Specifically, we implemented two validation methods: Query-based Validation (VQ), and Recency-Frequency-Monetary validation (RFM). RFM validation [18] is a well known approach, while the Query-based Validation is a customised approach resulting from discussions with our industry partners.

5.4.1 RFM Validation

RFM ([18]) validation is a very popular model for classifying customer value. The RFM method is based on the Recency, Frequency, and Monetary value of a customer, defined as follows:

- **R**ecency (R): How recent was their last policy purchase?
- **F**requency (F): How often do they purchase a policy?
- **M**onetary Value (M): How much do they spend?

In order to calculate the score of a customer, these metrics must be associated with values. Values for each metric can range from 1 to 4, where 1 is the best score and 4 the worst. The function defining the values for R is described in definition 5.1.

Definition 5.1. Given a customer **c**, the year **d** of the last purchase or renewal of a policy, and the year D of the most recent policy in the dataset, then the score $\mathcal{R}$ for recency is:

$$\mathcal{R}(c,d,D) = \begin{cases} 1, & \text{if } d = D \\ 2, & \text{if } d = D-1 \\ 3, & \text{if } d = D-2 \\ 4, & \text{if } d <= D-3 \end{cases}$$

For example, the most recent policy was purchased (renewed) in 2015, therefore the value for R are: 1 for policies from 2015, 2 for policies from 2014, 3, for policies from 2013, and 4 for all others. F and M are described by variables YNbr and Margin. Values for F and M are defined by the percentile brackets the values for YNbr and Margin in table 5.3 fall into, as per definition 5.2.

Definition 5.2. Given a customer **c**, and **v** the value from table 5.2, then score $\mathcal{F}$ for frequency and the score $\mathcal{M}$ for monetary value are defined as, respectively:

$$\mathcal{F}(c,v), \mathcal{M}(c,v) = \begin{cases} 1, & \text{if } v \text{ is equal or higher than the } 75^{th} \text{ percentile} \\ 2, & \text{if } v \text{ is between the } 75^{th} \text{ and the } 50^{th} \text{ percentile} \\ 3, & \text{if } v \text{ is between the } 50^{th} \text{ and the } 25^{th} \text{ percentile} \\ 4, & \text{Otherwise} \end{cases}$$

In simple terms, recency is lower if the policy purchase or renewal is more recent: frequency and monetary values are lower if the attributes are in the highest percentile. Ultimately, for the RFM method we define a *good* customer as one having a score of 1 in *all* metrics.

5.4.2 Query-Based Validation

For the query-based validation VQ, we defined five different criteria or queries. Queries are mutually independent, each measuring a different aspect of the customer (compared to the cluster). Each query is defined on a single variable from section 5.1 with the general definition of a query presented in definition 5.3.

Definition 5.3. Given a customer **c**, the function $\mathcal{A}_v$ defining the value of a set of customers given a variable **v**, and a cluster $\mathcal{C}$ of customers, then a query-validation $\mathcal{VQ}$ is defined as follows:

$$\mathcal{VQ}(c,v,C) = \mathcal{A}_v(c) > Average(\mathcal{A}_v(C)) \tag{5.5}$$

where $\boldsymbol{v} \in \boldsymbol{V}$, and with $V = \{Margin, CCNber, CFNber, Maxh, YNbr\}$

Table 5.5: Query validation legend for all variables considered.

Name	Query
VQ1	$\mathcal{VQ}(c, Margin) = \mathcal{A}_v(c) > Average(\mathcal{A}_{Margin}(C))$
VQ2	$\mathcal{VQ}(c, CCNber) = \mathcal{A}_v(c) > Average(\mathcal{A}_{CCNber}(C))$
VQ3	$\mathcal{VQ}(c, CFNber) = \mathcal{A}_v(c) > Average(\mathcal{A}_{CFNber}(C))$
VQ4	$\mathcal{VQ}(c, Maxh) = \mathcal{A}_v(c) > Average(\mathcal{A}_{Maxh}(C))$
VQ5	$\mathcal{VQ}(c, YNber) = \mathcal{A}_v(c) > Average(\mathcal{A}_{YNber}(C))$

The query validation in definition 5.3 generates one query for each variable in V. Each query defines whether a customer is *good* or not. A good customer is defined as one having above average values. The variables considered for these queries are *Margin, CCNber, CFNber, Maxh,* and *YNbr*, and were selected because:

- Margin, and YNbr represent the customer margin and the number of years a customer stayed with the company, and are two metrics widely used in the literature, e.g. [56, 104];
- CCNber is related to Margin and YNbr, and reflects the number of policies a customer holds;
- CFNber reflects that family member policies contribute to the customer;
- Maxh is the maximum number of policies the customer had during a year.

Table 5.5 shows the list of query validation obtained by instantiating definition 5.3 to the above variables.

5.5 Analysing Results

Before examining the results of this evaluation, it is useful to revisit the goals of the clustering process. The primary goal is to determine if customers could be grouped according to their *value*. Preferably, those groups would be as small as 2 or 3 clusters as the second goal was to identify the *good* customers. Using table 5.6, the evaluation has 8 distinct experiments but within each experiment, every clustering algorithm has values for k from 1 to 5, meaning 5 experimental runs within each experiment. The result was 15 cluster datasets generated during *every* experiment as can be seen in table 5.8. As 120 clusters (8x15) represents too much information for discussion, we selected the best performing experiments and in tables 5.7 and 5.8, present the clusters and validation results for discussion. Experiments were developed in python using the libraries $scikit-learn$(0.21.1), $pyclust$(0.2.0).

Table 5.6: Experimental Configurations

Experiment	**Algorithm**	**Normalisation**	**Variable Set**
Exp1	k-means	Min-Max	All
Exp2	k-medoids	Min-Max	All
Exp3	k-means	Categorised	All
Exp4	k-medoids	Categorised	All
Exp5	k-means	Min-Max	CC
Exp6	k-medoids	Min-Max	CC
Exp7	k-means	Standard	PCA_{95}
Exp8	k-means	Standard	PCA_{85}

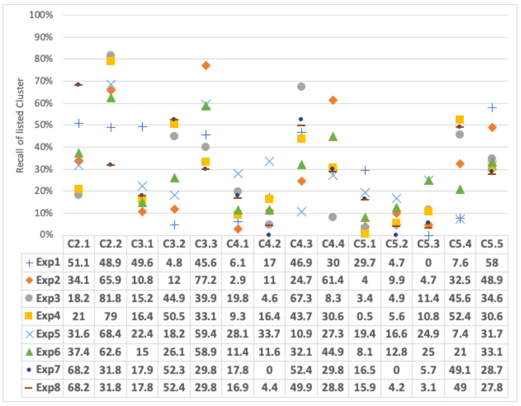
In section 5.5.1, we discuss our use of recall and precision to identify the best performing experiment and in section §5.5.2, we highlight the best performing set of clusters within that experiment. In the final part of this section, we present a detailed discussion on the impact of these results.

5.5.1 Recall and Precision Result.

We now present an overview of the results for experiments 1-8, using evaluation metrics VQ1 to VQ5 and RFM together with recall and precision to identify the best performing experimental configuration.

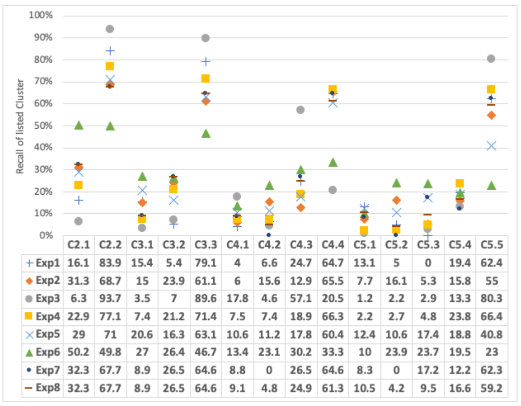
Assume that TC represents the total number of (customer) objects classified as *good* in the entire dataset; TG the number of objects in a cluster; and CG the number of customers classified as good in that cluster. Both validations are equally important: *Recall* represents the fraction of *good* customers in the cluster, divided by the overall number of good customers $\frac{CG}{TC}$; and *Precision* is the fraction of *good* customers in a cluster $\frac{CG}{TG}$.

Figure 5.4 presents the results for recall and figure 5.5 presents the result for precision. In each figure, we have 6 graphs. In each graph, a validation (VQ1 to VQ5 and RFM) result for all experiments is presented for comparison. The results in the graph are the percentage of either recall or precision in a specified cluster, with the results kept to 1 decimal place.



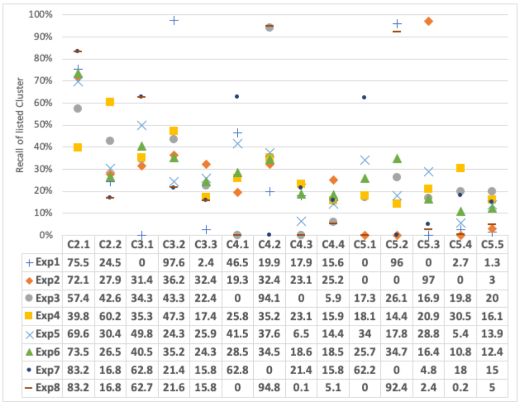
	C2.1	C2.2	C3.1	C3.2	C3.3	C4.1	C4.2	C4.3	C4.4	C5.1	C5.2	C5.3	C5.4	C5.5
+ Exp1	51.1	48.9	49.6	4.8	45.6	6.1	17	46.9	30	29.7	4.7	0	7.6	58
◆ Exp2	34.1	65.9	10.8	12	77.2	2.9	11	24.7	61.4	4	9.9	4.7	32.5	48.9
● Exp3	18.2	81.8	15.2	44.9	39.9	19.8	4.6	67.3	8.3	3.4	4.9	11.4	45.6	34.6
■ Exp4	21	79	16.4	50.5	33.1	9.3	16.4	43.7	30.6	0.5	5.6	10.8	52.4	30.6
× Exp5	31.6	68.4	22.4	18.2	59.4	28.1	33.7	10.9	27.3	19.4	16.6	24.9	7.4	31.7
▲ Exp6	37.4	62.6	15	26.1	58.9	11.4	11.6	32.1	44.9	8.1	12.8	25	21	33.1
• Exp7	68.2	31.8	17.9	52.3	29.8	17.8	0	52.4	29.8	16.5	0	5.7	49.1	28.7
— Exp8	68.2	31.8	17.8	52.4	29.8	16.9	4.4	49.9	28.8	15.9	4.2	3.1	49	27.8

(a) Recall VQ1 for all experiment



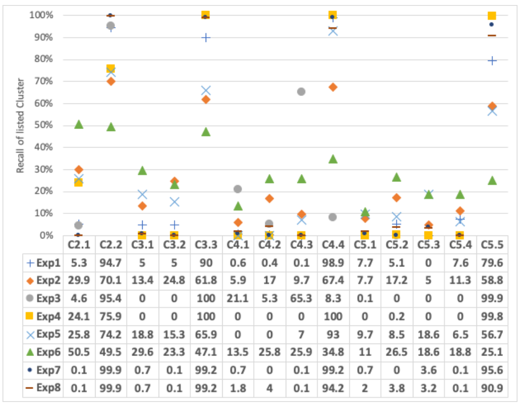
	C2.1	C2.2	C3.1	C3.2	C3.3	C4.1	C4.2	C4.3	C4.4	C5.1	C5.2	C5.3	C5.4	C5.5
+ Exp1	16.1	83.9	15.4	5.4	79.1	4	6.6	24.7	64.7	13.1	5	0	19.4	62.4
◆ Exp2	31.3	68.7	15	23.9	61.1	6	15.6	12.9	65.5	7.7	16.1	5.3	15.8	55
● Exp3	6.3	93.7	3.5	7	89.6	17.8	4.6	57.1	20.5	1.2	2.2	2.9	13.3	80.3
■ Exp4	22.9	77.1	7.4	21.2	71.4	7.5	7.4	18.9	66.3	2.2	2.7	4.8	23.8	66.4
× Exp5	29	71	20.6	16.3	63.1	10.6	11.2	17.8	60.4	12.4	10.6	17.4	18.8	40.8
▲ Exp6	50.2	49.8	27	26.4	46.7	13.4	23.1	30.2	33.3	10	23.9	23.7	19.5	23
• Exp7	32.3	67.7	8.9	26.5	64.6	8.8	0	26.5	64.6	8.3	0	17.2	12.2	62.3
— Exp8	32.3	67.7	8.9	26.5	64.6	9.1	4.8	24.9	61.3	10.5	4.2	9.5	16.6	59.2

(b) Recall VQ2 for all experiment



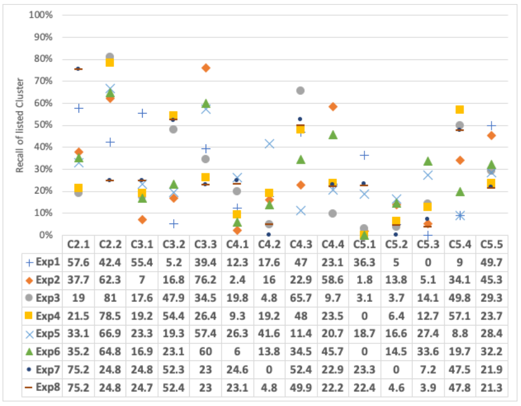
	C2.1	C2.2	C3.1	C3.2	C3.3	C4.1	C4.2	C4.3	C4.4	C5.1	C5.2	C5.3	C5.4	C5.5
+ Exp1	75.5	24.5	0	97.6	2.4	46.5	19.9	17.9	15.6	0	96	0	2.7	1.3
◆ Exp2	72.1	27.9	31.4	36.2	32.4	19.3	32.4	23.1	25.2	0	0	97	0	3
● Exp3	57.4	42.6	34.3	43.3	22.4	0	94.1	0	5.9	17.3	26.1	16.9	19.8	20
■ Exp4	39.8	60.2	35.3	47.3	17.4	25.8	35.2	23.1	15.9	18.1	14.4	20.9	30.5	16.1
× Exp5	69.6	30.4	49.8	24.3	25.9	41.5	37.6	6.5	14.4	34	17.8	28.8	5.4	13.9
▲ Exp6	73.5	26.5	40.5	35.2	24.3	28.5	34.5	18.6	18.5	25.7	34.7	16.4	10.8	12.4
• Exp7	83.2	16.8	62.8	21.4	15.8	62.8	0	21.4	15.8	62.2	0	4.8	18	15
— Exp8	83.2	16.8	62.7	21.6	15.8	0	94.8	0.1	5.1	0	92.4	2.4	0.2	5

(c) Recall VQ3 for all experiment



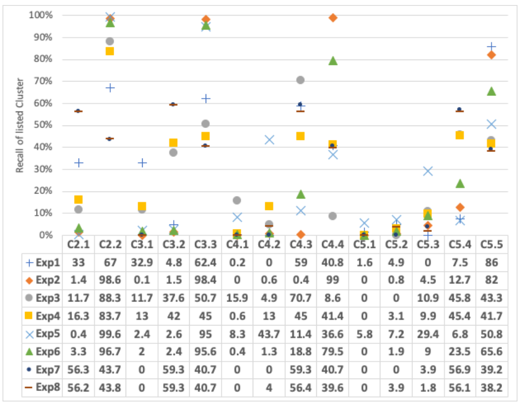
	C2.1	C2.2	C3.1	C3.2	C3.3	C4.1	C4.2	C4.3	C4.4	C5.1	C5.2	C5.3	C5.4	C5.5
+ Exp1	5.3	94.7	5	5	90	0.6	0.4	0.1	98.9	7.7	5.1	0	7.6	79.6
◆ Exp2	29.9	70.1	13.4	24.8	61.8	5.9	17	9.7	67.4	7.7	17.2	5	11.3	58.8
● Exp3	4.6	95.4	0	0	100	21.1	5.3	65.3	8.3	0.1	0	0	0	99.9
■ Exp4	24.1	75.9	0	0	100	0	0	0	100	0	0.2	0	0	99.8
× Exp5	25.8	74.2	18.8	15.3	65.9	0	0	7	93	9.7	8.5	18.6	6.5	56.7
▲ Exp6	50.5	49.5	29.6	23.3	47.1	13.5	25.8	25.9	34.8	11	26.5	18.6	18.8	25.1
• Exp7	0.1	99.9	0.7	0.1	99.2	0.7	0	0.1	99.2	0.7	0	3.6	0.1	95.6
— Exp8	0.1	99.9	0.7	0.1	99.2	1.8	4	0.1	94.2	2	3.8	3.2	0.1	90.9

(d) Recall VQ4 for all experiment



	C2.1	C2.2	C3.1	C3.2	C3.3	C4.1	C4.2	C4.3	C4.4	C5.1	C5.2	C5.3	C5.4	C5.5
+ Exp1	57.6	42.4	55.4	5.2	39.4	12.3	17.6	47	23.1	36.3	5	0	9	49.7
◆ Exp2	37.7	62.3	7	16.8	76.2	2.4	16	22.9	58.6	1.8	13.8	5.1	34.1	45.3
● Exp3	19	81	17.6	47.9	34.5	19.8	4.8	65.7	9.7	3.1	3.7	14.1	49.8	29.3
■ Exp4	21.5	78.5	19.2	54.4	26.4	9.3	19.2	48	23.5	0	6.4	12.7	57.1	23.7
× Exp5	33.1	66.9	23.3	19.3	57.4	26.3	41.6	11.4	20.7	18.7	16.6	27.4	8.8	28.4
▲ Exp6	35.2	64.8	16.9	23.1	60	6	13.8	34.5	45.7	0	14.5	33.6	19.7	32.2
• Exp7	75.2	24.8	24.8	52.3	23	24.6	0	52.4	22.9	23.3	0	7.2	47.5	21.9
— Exp8	75.2	24.8	24.7	52.4	23	23.1	4.8	49.9	22.2	22.4	4.6	3.9	47.8	21.3

(e) Recall VQ5 for all experiment



	C2.1	C2.2	C3.1	C3.2	C3.3	C4.1	C4.2	C4.3	C4.4	C5.1	C5.2	C5.3	C5.4	C5.5
+ Exp1	33	67	32.9	4.8	62.4	0.2	0	59	40.8	1.6	4.9	0	7.5	86
◆ Exp2	1.4	98.6	0.1	1.5	98.4	0	0.6	0.4	99	0	0.8	4.5	12.7	82
● Exp3	11.7	88.3	11.7	37.6	50.7	15.9	4.9	70.7	8.6	0	0	10.9	45.8	43.3
■ Exp4	16.3	83.7	13	42	45	0.6	13	45	41.4	0	3.1	9.9	45.4	41.7
× Exp5	0.4	99.6	2.4	2.6	95	8.3	43.7	11.4	36.6	5.8	7.2	29.4	6.8	50.8
▲ Exp6	3.3	96.7	2	2.4	95.6	0.4	1.3	18.8	79.5	0	1.9	9	23.5	65.6
• Exp7	56.3	43.7	0	59.3	40.7	0	0	59.3	40.7	0	0	3.9	56.9	39.2
— Exp8	56.2	43.8	0	59.3	40.7	0	4	56.4	39.6	0	3.9	1.8	56.1	38.2

(f) Recall RFM for all experiment

Figure 5.4: Recall Results (All Experiments)

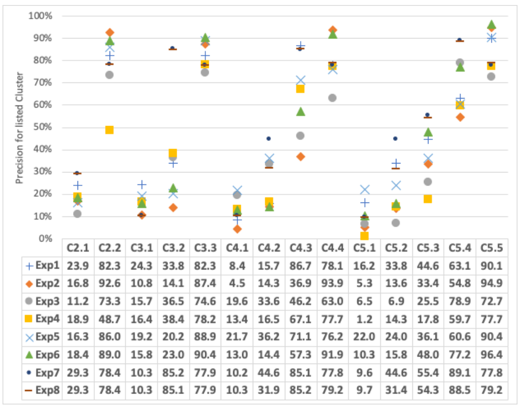
Within each experiment, clusters are named as Ckn, where k is the input parameter for the clustering algorithm and n the number of the cluster within that experiment, eg.

C3.2 indicates the 2nd cluster in a 3-cluster (k=3) experiment. For each k={2,3,4,5}, the recall result for the aggregate will always be 100% because recall calculates the fraction of *good* from the overall total of *good* customers. From the result for recall in figure 5.4, we can see that:

1. Experiment 1 (Exp1) ranks as the best configuration for VQ3 which has a match rate of 97.6%.
2. Exp2 ranked as best for RFM at cluster C4.4 (99.0%). In addition, this experiment also performed the second best for VQ3 at C5.2 with 97.0%.
3. The cluster C2.2 in Exp3 has outperformed others in VQ1 (81.8%), VQ2 (93.7%), VQ5 (81%), and also VQ4 (100%) at C3.3. Moreover, the cluster C3.3 also got the second best for VQ2 (89.6%).
4. In Exp4, the cluster C3.3 received the best score for VQ4 (100%), second best for VQ1 (79.0%) and RFM (98.4%). Additionally, this experiment also got second best in VQ4 at C4.4 (99.0%), and VQ5 at C2.2 (78.5%).
5. In summary, if we apply a score of 2 for the top performing cluster, and score of 1 for the second best: Exp3 ranks highest with 4 top ranked matches and 1 second with a score of 9; Exp4 is next best with a total score of 6; and Exp2 was next with a score of 3.

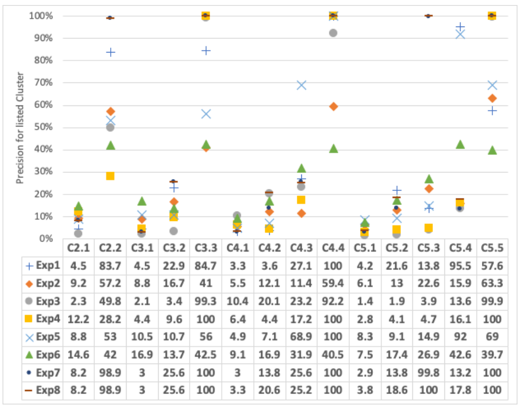
From the precision results, we would like to see a very high value for a single cluster with all remaining clusters in the same k-experiment scoring low. This suggests that *good* customers are assigned in the single high scoring cluster. From figure 5.5, we can see:

1. The customers for cluster C4.4 in Exp1 are all (100%) *good* customers using VQ2 and clusters C3.2 and C5.2 are all (100%) *good* customers using VQ3.
2. The objects in cluster C5.3 in Exp2 are all (100%) *good* customers using VQ3. The cluster C5.5, has the highest value 94.9% for VQ1. This experiment also has the 2nd highest rate at 93.9% for VQ1.
3. The cluster C4.2 in Exp3 are all (100%) *good* customers using VQ3. Then,



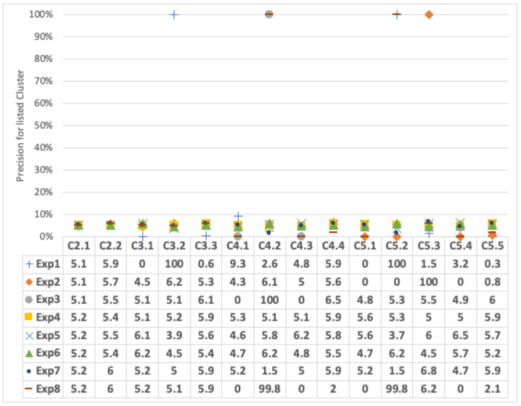
	C2.1	C2.2	C3.1	C3.2	C3.3	C4.1	C4.2	C4.3	C4.4	C5.1	C5.2	C5.3	C5.4	C5.5
+ Exp1	23.9	82.3	24.3	33.8	82.3	8.4	15.7	86.7	78.1	16.2	33.8	44.6	63.1	90.1
◆ Exp2	16.8	92.6	10.8	14.1	87.4	4.5	14.3	36.9	93.9	5.3	13.6	33.4	54.8	94.9
● Exp3	11.2	73.3	15.7	36.5	74.6	19.6	33.6	46.2	63.0	6.5	6.9	25.5	78.9	72.7
■ Exp4	18.9	48.7	16.4	38.4	78.2	13.4	16.5	67.1	77.7	1.2	14.3	17.8	59.7	77.7
× Exp5	16.3	86.0	19.2	20.2	88.9	21.7	36.2	71.1	76.2	22.0	24.0	36.1	60.6	90.4
▲ Exp6	18.4	89.0	15.8	23.0	90.4	13.0	14.4	57.3	91.9	10.3	15.8	48.0	77.2	96.4
• Exp7	29.3	78.4	10.3	85.2	77.9	10.2	44.6	85.1	77.8	9.6	44.6	55.4	89.1	77.8
– Exp8	29.3	78.4	10.3	85.1	77.9	10.3	31.9	85.2	79.2	9.7	31.4	54.3	88.5	79.2

(a) Precision of VQ1 for all experiment



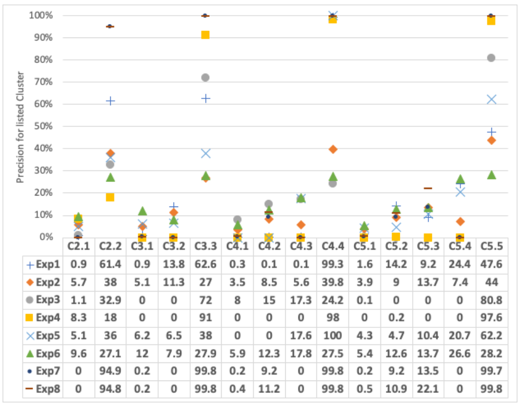
	C2.1	C2.2	C3.1	C3.2	C3.3	C4.1	C4.2	C4.3	C4.4	C5.1	C5.2	C5.3	C5.4	C5.5
+ Exp1	4.5	83.7	4.5	22.9	84.7	3.3	3.6	27.1	100	4.2	21.6	13.8	95.5	57.6
◆ Exp2	9.2	57.2	8.8	16.7	41	5.5	12.1	11.4	59.4	6.1	13	22.6	15.9	63.3
● Exp3	2.3	49.8	2.1	3.4	99.3	10.4	20.1	23.2	92.2	1.4	1.9	3.9	13.6	99.9
■ Exp4	12.2	28.2	4.4	9.6	100	6.4	4.4	17.2	100	2.8	4.1	4.7	16.1	100
× Exp5	8.8	53	10.5	10.7	56	4.9	7.1	68.9	100	8.3	9.1	14.9	92	69
▲ Exp6	14.6	42	16.9	13.7	42.5	9.1	16.9	31.9	40.5	7.5	17.4	26.9	42.6	39.7
• Exp7	8.2	98.9	3	25.6	100	3	13.8	25.6	100	2.9	13.8	99.8	13.2	100
– Exp8	8.2	98.9	3	25.6	100	3.3	20.6	25.2	100	3.8	18.6	100	17.8	100

(b) Precision of VQ2 for all experiment



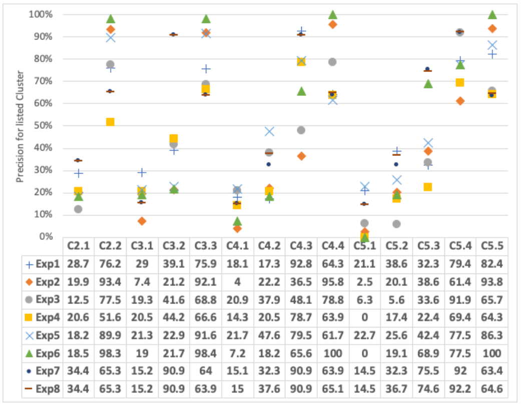
	C2.1	C2.2	C3.1	C3.2	C3.3	C4.1	C4.2	C4.3	C4.4	C5.1	C5.2	C5.3	C5.4	C5.5
+ Exp1	5.1	5.9	0	100	0.6	9.3	2.6	4.8	5.9	0	100	1.5	3.2	0.3
◆ Exp2	5.1	5.7	4.5	6.2	5.3	4.3	6.1	5	5.6	0	0	100	0	0.8
● Exp3	5.1	5.5	5.1	5.1	6.1	0	100	0	6.5	4.8	5.3	5.5	4.9	6
■ Exp4	5.2	5.4	5.1	5.2	5.9	5.3	5.1	5.1	5.9	5.6	5.3	5	5	5.9
× Exp5	5.2	5.5	6.1	3.9	5.6	4.6	5.8	6.2	5.8	5.6	3.7	6	6.5	5.7
▲ Exp6	5.2	5.4	6.2	4.5	5.4	4.7	6.2	4.8	5.5	4.7	6.2	4.5	5.7	5.2
• Exp7	5.2	6	5.2	5	5.9	5.2	1.5	5	5.9	5.2	1.5	6.8	4.7	5.9
– Exp8	5.2	6	5.2	5.1	5.9	0	99.8	0	2	0	99.8	6.2	0	2.1

(c) Precision of VQ3 for all experiment



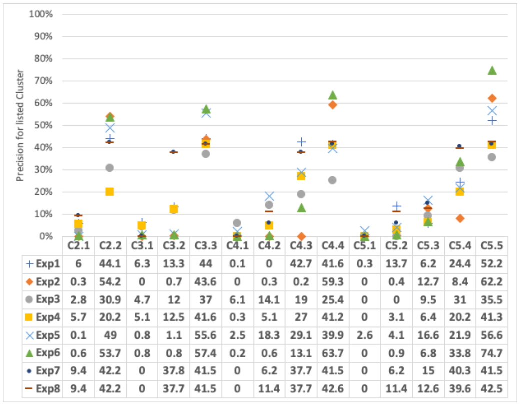
	C2.1	C2.2	C3.1	C3.2	C3.3	C4.1	C4.2	C4.3	C4.4	C5.1	C5.2	C5.3	C5.4	C5.5
+ Exp1	0.9	61.4	0.9	13.8	62.6	0.3	0.1	0.1	99.3	1.6	14.2	9.2	24.4	47.6
◆ Exp2	5.7	38	5.1	11.3	27	3.5	8.5	5.6	39.8	3.9	9	13.7	7.4	44
● Exp3	1.1	32.9	0	0	72	8	15	17.3	24.2	0.1	0	0	0	80.8
■ Exp4	8.3	18	0	0	91	0	0	0	98	0	0.2	0	0	97.6
× Exp5	5.1	36	6.2	6.5	38	0	0	17.6	100	4.3	4.7	10.4	20.7	62.2
▲ Exp6	9.6	27.1	12	7.9	27.9	5.9	12.3	17.8	27.5	5.4	12.6	13.7	26.6	28.2
• Exp7	0	94.9	0.2	0	99.8	0.2	9.2	0	99.8	0.2	9.2	13.5	0	99.7
– Exp8	0	94.8	0.2	0	99.8	0.4	11.2	0	99.8	0.5	10.9	22.1	0	99.8

(d) Precision of VQ4 for all experiment



	C2.1	C2.2	C3.1	C3.2	C3.3	C4.1	C4.2	C4.3	C4.4	C5.1	C5.2	C5.3	C5.4	C5.5
+ Exp1	28.7	76.2	29	39.1	75.9	18.1	17.3	92.8	64.3	21.1	38.6	32.3	79.4	82.4
◆ Exp2	19.9	93.4	7.4	21.2	92.1	4	22.2	36.5	95.8	2.5	20.1	38.6	61.4	93.8
● Exp3	12.5	77.5	19.3	41.6	68.8	20.9	37.9	48.1	78.8	6.3	5.6	33.6	91.9	65.7
■ Exp4	20.6	51.6	20.5	44.2	66.6	14.3	20.5	78.7	63.9	0	17.4	22.4	69.4	64.3
× Exp5	18.2	89.9	21.3	22.9	91.6	21.7	47.6	79.5	61.7	22.7	25.6	42.4	77.5	86.3
▲ Exp6	18.5	98.3	19	21.7	98.4	7.2	18.2	65.6	100	0	19.1	68.9	77.5	100
• Exp7	34.4	65.3	15.2	90.9	64	15.1	32.3	90.9	63.9	14.5	32.3	75.5	92	63.4
– Exp8	34.4	65.3	15.2	90.9	63.9	15	37.6	90.9	65.1	14.5	36.7	74.6	92.2	64.6

(e) Precision of VQ5 for all experiment



	C2.1	C2.2	C3.1	C3.2	C3.3	C4.1	C4.2	C4.3	C4.4	C5.1	C5.2	C5.3	C5.4	C5.5
+ Exp1	6	44.1	6.3	13.3	44	0.1	0	42.7	41.6	0.3	13.7	6.2	24.4	52.2
◆ Exp2	0.3	54.2	0	0.7	43.6	0	0.3	0.2	59.3	0	0.4	12.7	8.4	62.2
● Exp3	2.8	30.9	4.7	12	37	6.1	14.1	19	25.4	0	0	9.5	31	35.5
■ Exp4	5.7	20.2	5.1	12.5	41.6	0.3	5.1	27	41.2	0	3.1	6.4	20.2	41.3
× Exp5	0.1	49	0.8	1.1	55.6	2.5	18.3	29.1	39.9	2.6	4.1	16.6	21.9	56.6
▲ Exp6	0.6	53.7	0.8	0.8	57.4	0.2	0.6	13.1	63.7	0	0.9	6.8	33.8	74.7
• Exp7	9.4	42.2	0	37.8	41.5	0	6.2	37.7	41.5	0	6.2	15	40.3	41.5
– Exp8	9.4	42.2	0	37.7	41.5	0	11.4	37.7	42.6	0	11.4	12.6	39.6	42.5

(f) Precision RFM for all experiment

Figure 5.5: Precision Results (All Experiment)

cluster C5.5 in the same experiment has the 2nd highest rate at 99.9% for VQ2.

4. In Exp4, the customers in cluster C3.3, C4.4 and C5.5 are all (100%) *good* customers using VQ2.

5. Cluster C4.4 in Exp5 contains (100%) *good* customers for VQ2 and VQ4.
6. For Exp6, clusters C4.4 and C5.5 are all (100%) *good* customers using VQ5. Cluster C5.5 also got the highest rate (74.7%) for RFM. Moreover, the 2nd highest rate for VQ5 (C3.3 with 98.4%) and RFM (C4.4 with 63.7%) is also in this cluster.
7. Clusters C3.3, C4.4 and C5.5 are all (100%) *good* customers using VQ2 which applied to Exp7 and Exp8 also. Moreover, Exp8 had cluster C5.3 using VQ2 identified 100% *good* customers also. Exp7 scored 2nd best for VQ3 (C4.2) and Exp8 scored the 2nd best for VQ4 at cluster C4.4.
8. In summary, using the same scoring approach as recall, Exp6 and Exp8 rank as the highest, as both have a score of 6 for precision, Exp3 had a score of 3.

In terms of evaluating the performance for the recall and the precision, we calculated the value called the **F-Score**, which is a measure of how desirable a cluster of customer is: the higher the F-Score for a cluster of customers, the better the customers in that cluster are, formally presented in definition 5.4.

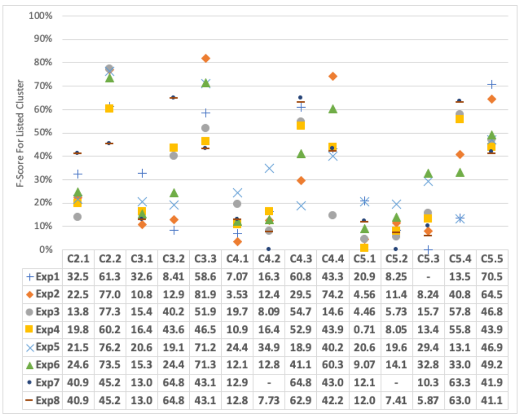
Definition 5.4. Given a cluster C, and a validation rule VR$_i$, and being $recall_i$, and $precision_i$ the recall and precision for C and VR$_i$, respectively, then F-Score is:

$$\text{F-Score}(C) = 2 * \frac{recall_i * precision_i}{recall_i + precision_i}$$

where VR$_i$ $\in$ VRS, with VRS = {VQ1, VQ2, VQ3, VQ4, VQ5, RFM }

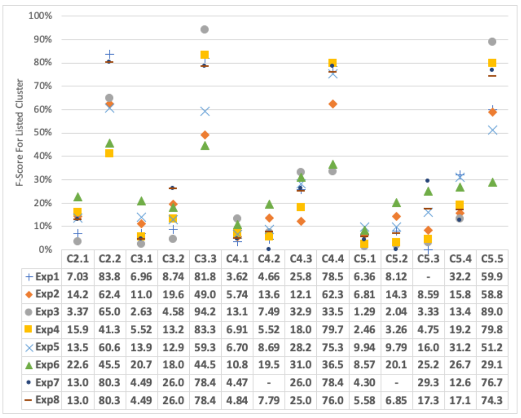
If we consider the count results for both recall and precision:

- The details of the combined result of recall and precision (the F-score) are presented in figure 5.6. The presence of symbol - in the graph means the F-score equals 0 for this cluster in the listed experiment. This was caused by one or both recall and precision equal to 0. The pattern of F-score is similar to the Precision result.
- As for a individual cluster, the cluster C4.4 performs the best as it has appeared



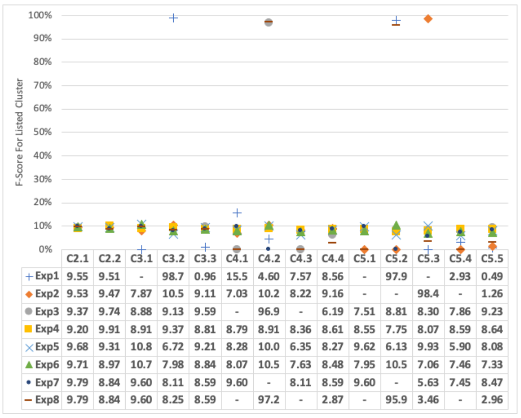
	C2.1	C2.2	C3.1	C3.2	C3.3	C4.1	C4.2	C4.3	C4.4	C5.1	C5.2	C5.3	C5.4	C5.5
+ Exp1	32.5	61.3	32.6	8.41	58.6	7.07	16.3	60.8	43.3	20.9	8.25	-	13.5	70.5
◆ Exp2	22.5	77.0	10.8	12.9	81.9	3.53	12.4	29.5	74.2	4.56	11.4	8.24	40.8	64.5
● Exp3	13.8	77.3	15.4	40.2	51.9	19.7	8.09	54.7	14.6	4.46	5.73	15.7	57.8	46.8
■ Exp4	19.8	60.2	16.4	43.6	46.5	10.9	16.4	52.9	43.9	0.71	8.05	13.4	55.8	43.9
× Exp5	21.5	76.2	20.6	19.1	71.2	24.4	34.9	18.9	40.2	20.6	19.6	29.4	13.1	46.9
▲ Exp6	24.6	73.5	15.3	24.4	71.3	12.1	12.8	41.1	60.3	9.07	14.1	32.8	33.0	49.2
• Exp7	40.9	45.2	13.0	64.8	43.1	12.9	-	64.8	43.0	12.1	-	10.3	63.3	41.9
− Exp8	40.9	45.2	13.0	64.8	43.1	12.8	7.73	62.9	42.2	12.0	7.41	5.87	63.0	41.1

(a) F-Score of VQ1 for all experiment



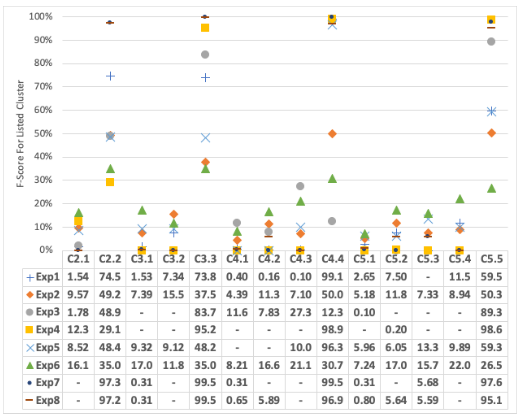
	C2.1	C2.2	C3.1	C3.2	C3.3	C4.1	C4.2	C4.3	C4.4	C5.1	C5.2	C5.3	C5.4	C5.5
+ Exp1	7.03	83.8	6.96	8.74	81.8	3.62	4.66	25.8	78.5	6.36	8.12	-	32.2	59.9
◆ Exp2	14.2	62.4	11.0	19.6	49.0	5.74	13.6	12.1	62.3	6.81	14.3	8.59	15.8	58.8
● Exp3	3.37	65.0	2.63	4.58	94.2	13.1	7.49	32.9	33.5	1.29	2.04	3.33	13.4	89.0
■ Exp4	15.9	41.3	5.52	13.2	83.3	6.91	5.52	18.0	79.7	2.46	3.26	4.75	19.2	79.8
× Exp5	13.5	60.6	13.9	12.9	59.3	6.70	8.69	28.2	75.3	9.94	9.79	16.0	31.2	51.2
▲ Exp6	22.6	45.5	20.7	18.0	44.5	10.8	19.5	31.0	36.5	8.57	20.1	25.2	26.7	29.1
• Exp7	13.0	80.3	4.49	26.0	78.4	4.47	-	26.0	78.4	4.30	-	29.3	12.6	76.7
− Exp8	13.0	80.3	4.49	26.0	78.4	4.84	7.79	25.0	76.0	5.58	6.85	17.3	17.1	74.3

(b) F-Score of VQ2 for all experiment



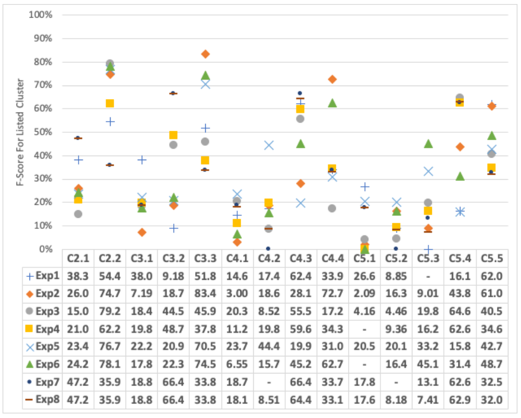
	C2.1	C2.2	C3.1	C3.2	C3.3	C4.1	C4.2	C4.3	C4.4	C5.1	C5.2	C5.3	C5.4	C5.5
+ Exp1	9.55	9.51	-	98.7	0.96	15.5	4.60	7.57	8.56	-	97.9	-	2.93	0.49
◆ Exp2	9.53	9.47	7.87	10.5	9.11	7.03	10.2	8.22	9.16	-	-	98.4	-	1.26
● Exp3	9.37	9.74	8.88	9.13	9.59	-	96.9	-	6.19	7.51	8.81	8.30	7.86	9.23
■ Exp4	9.20	9.91	8.91	9.37	8.81	8.79	8.91	8.36	8.61	8.55	7.75	8.07	8.59	8.64
× Exp5	9.68	9.31	10.8	6.72	9.21	8.28	10.0	6.35	8.27	9.62	6.13	9.93	5.90	8.08
▲ Exp6	9.71	8.97	10.7	7.98	8.84	8.07	10.5	7.63	8.48	7.95	10.5	7.06	7.46	7.33
• Exp7	9.79	8.84	9.60	8.11	8.59	9.60	-	8.11	8.59	9.60	-	5.63	7.45	8.47
− Exp8	9.79	8.84	9.60	8.25	8.59	-	97.2	-	2.87	-	95.9	3.46	-	2.96

(c) F-Score of VQ3 for all experiment



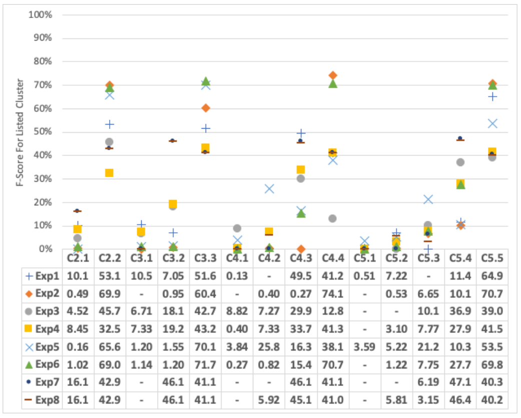
	C2.1	C2.2	C3.1	C3.2	C3.3	C4.1	C4.2	C4.3	C4.4	C5.1	C5.2	C5.3	C5.4	C5.5
+ Exp1	1.54	74.5	1.53	7.34	73.8	0.40	0.16	0.10	99.1	2.65	7.50	-	11.5	59.5
◆ Exp2	9.57	49.2	7.39	15.5	37.5	4.39	11.3	7.10	50.0	5.18	11.8	7.33	8.94	50.3
● Exp3	1.78	48.9	-	-	83.7	11.6	7.83	27.3	12.3	0.10	-	-	-	89.3
■ Exp4	12.3	29.1	-	-	95.2	-	-	-	98.9	-	0.20	-	-	98.6
× Exp5	8.52	48.4	9.32	9.12	48.2	-	-	10.0	96.3	5.96	6.05	13.3	9.89	59.3
▲ Exp6	16.1	35.0	17.0	11.8	35.0	8.21	16.6	21.1	30.7	7.24	17.0	15.7	22.0	26.5
• Exp7	-	97.3	0.31	-	99.5	0.31	-	-	99.5	0.31	-	5.68	-	97.6
− Exp8	-	97.2	0.31	-	99.5	0.65	5.89	-	96.9	0.80	5.64	5.59	-	95.1

(d) F-Score of VQ4 for all experiment



	C2.1	C2.2	C3.1	C3.2	C3.3	C4.1	C4.2	C4.3	C4.4	C5.1	C5.2	C5.3	C5.4	C5.5
+ Exp1	38.3	54.4	38.0	9.18	51.8	14.6	17.4	62.4	33.9	26.6	8.85	-	16.1	62.0
◆ Exp2	26.0	74.7	7.19	18.7	83.4	3.00	18.6	28.1	72.7	2.09	16.3	9.01	43.8	61.0
● Exp3	15.0	79.2	18.4	44.5	45.9	20.3	8.52	55.5	17.2	4.16	4.46	19.8	64.6	40.5
■ Exp4	21.0	62.2	19.8	48.7	37.8	11.2	19.8	59.6	34.3	-	9.36	16.2	62.6	34.6
× Exp5	23.4	76.7	22.2	20.9	70.5	23.7	44.4	19.9	31.0	20.5	20.1	33.2	15.8	42.7
▲ Exp6	24.2	78.1	17.8	22.3	74.5	6.55	15.7	45.2	62.7	-	16.4	45.1	31.4	48.7
• Exp7	47.2	35.9	18.8	66.4	33.8	18.7	-	66.4	33.7	17.8	-	13.1	62.6	32.5
− Exp8	47.2	35.9	18.8	66.4	33.8	18.1	8.51	64.4	33.1	17.6	8.18	7.41	62.9	32.0

(e) F-Score of VQ5 for all experiment



	C2.1	C2.2	C3.1	C3.2	C3.3	C4.1	C4.2	C4.3	C4.4	C5.1	C5.2	C5.3	C5.4	C5.5
+ Exp1	10.1	53.1	10.5	7.05	51.6	0.13	-	49.5	41.2	0.51	7.22	-	11.4	64.9
◆ Exp2	0.49	69.9	-	0.95	60.4	-	0.40	0.27	74.1	-	0.53	6.65	10.1	70.7
● Exp3	4.52	45.7	6.71	18.1	42.7	8.82	7.27	29.9	12.8	-	-	10.1	36.9	39.0
■ Exp4	8.45	32.5	7.33	19.2	43.2	0.40	7.33	33.7	41.3	-	3.10	7.77	27.9	41.5
× Exp5	0.16	65.6	1.20	1.55	70.1	3.84	25.8	16.3	38.1	3.59	5.22	21.2	10.3	53.5
▲ Exp6	1.02	69.0	1.14	1.20	71.7	0.27	0.82	15.4	70.7	-	1.22	7.75	27.7	69.8
• Exp7	16.1	42.9	-	46.1	41.1	-	-	46.1	41.1	-	-	6.19	47.1	40.3
− Exp8	16.1	42.9	-	46.1	41.1	-	5.92	45.1	41.0	-	5.81	3.15	46.4	40.2

(f) F-Score of RFM for all experiment

Figure 5.6: F-Score Results (All Experiment)

12 times (8 in top, 4 in second rank). The cluster C3.3 as the second best cluster for 9 times. Cluster C2.2 only appears 4 times - even less than C5.5. This proved that using a error function in a clustering method, you will not

have an optimal k in terms of accuracy.

- Experiment 3 performs best as it has the highest number of top (5) and second ranked (2) clusters, with a summed score of 12 using our simple metric. Thus, experiment 3 is used for our discussion in the following section.

5.5.2 Understanding the Best Performing Configuration

Tables 5.7 and 5.8 present the result of the best experiment, as selected by our recall and precision results, with a configuration (from table 5.6) of {k-means, Categorised, All}. The goal of this part of the evaluation is to determine which k-cluster configuration for Experiment 3 out-performs others by best classifying *good* customers into a single cluster.

In table 5.7, the **Clus** column labels the cluster for Exp3 denoting k={1,2,3,4,5}. Columns **VQ1**, **VQ2**, **VQ3**, **VQ4**, **VQ5**, and **RFM** present both the **Recall** and **Precision** scores for every cluster and validation method. The final column is the **AF-Score**, which is the mean of F-score values for all 6 validation rules VQs, and RFM.

Table 5.7: Recall and Precision of Experiment 3

	VQ1		VQ2		VQ3		VQ4		VQ5		RFM		
Clus	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	Recall	Precision	AF-Score
C1	100.0%	36.5%	100.0%	21.7%	100.0%	5.3%	100.0%	14.1%	100.0%	39.0%	100.0%	14.3%	34.2%
C2.1	18.2%	11.2%	6.3%	2.3%	57.4%	5.1%	4.6%	1.1%	19.0%	12.5%	11.7%	2.8%	8.0%
C2.2	81.8%	73.3%	93.7%	49.8%	42.6%	5.5%	95.4%	32.9%	81.0%	77.5%	88.3%	30.9%	51.0%
C3.1	15.2%	15.7%	3.5%	2.1%	34.3%	5.1%	0.0%	0.0%	17.6%	19.3%	11.7%	4.7%	8.7%
C3.2	44.9%	36.5%	7.0%	3.4%	43.3%	5.1%	0.0%	0.0%	47.9%	41.6%	37.6%	12.0%	19.4%
C3.3	39.9%	74.6%	89.6%	99.3%	22.4%	6.1%	100.0%	72.0%	34.5%	68.8%	50.7%	37.0%	**54.7%**
C4.1	19.8%	19.6%	17.8%	10.4%	0.0%	0.0%	21.1%	8.0%	19.8%	20.9%	15.9%	6.1%	12.3%
C4.2	4.6%	33.6%	4.6%	20.1%	94.1%	100.0%	5.3%	15.0%	4.8%	37.9%	4.9%	14.1%	22.7%
C4.3	67.3%	46.2%	57.1%	23.2%	0.0%	0.0%	65.3%	17.3%	65.7%	48.1%	70.7%	19.0%	33.4%
C4.4	8.3%	63.0%	20.5%	92.2%	5.9%	6.5%	8.3%	24.2%	9.7%	78.8%	8.6%	25.4%	16.1%
C5.1	3.4%	6.5%	1.2%	1.4%	17.3%	4.8%	0.1%	0.1%	3.1%	6.3%	0.0%	0.0%	2.9%
C5.2	4.9%	6.9%	2.2%	1.9%	26.1%	5.3%	0.0%	0.0%	3.7%	5.6%	0.0%	0.0%	3.5%
C5.3	11.4%	25.5%	2.9%	3.9%	16.9%	5.5%	0.0%	0.0%	14.1%	33.6%	10.9%	9.5%	9.6%
C5.4	45.6%	78.9%	13.3%	13.6%	19.8%	4.9%	0.0%	0.0%	49.8%	91.9%	45.8%	31.0%	30.1%
C5.5	34.6%	72.7%	80.3%	99.9%	20.0%	6.0%	99.9%	80.8%	29.3%	65.7%	43.3%	35.5%	52.4%

We now highlight some of the more interesting observations from the results presented in table 5.7:

- In general, C3.3 outperforms others using **AF-Score** values. However, the performance of C2.1 and C5.5 is close to the best performance. For those clusters who performed well in AF-Score, most of the time, both recall and precision for a validation method are higher than other clusters in this set of k.

- For cluster C4.4, the precision value is not low. However, because the number of objects in this cluster is lower than TG, this decreases the recall value, which has the effect of reducing the AF-Score.
- When the value of k increases, good customers are spread across multiple clusters. For $k = 2$, over 80% of good customers lie in cluster C2.2. For k=3, many were assigned to cluster C3.2. For $k = 4$, most good customers were placed into cluster C4.3 and for k=5, a number were assigned to C5.4.
- CFNbr (having a family member) represents an interesting outlier. For k=4, C4.2 clustered more than 94% of customers who had a family member who also had a policy. For other k experiments, customer with CFNbr values are spread into different clusters.

We provide a more detailed analysis in table 5.8. We can see that the **Entities** column lists the number of objects assigned to this cluster. The columns **Margin**, **CCNbr**, **CFNbr**, **Maxh**, and **YNbr** show the average of the listed variables in a certain cluster. The columns **VQ1**, **VQ2**, **VQ3**, **VQ4**, **VQ5**, and **RFM** present the number of matches in the cluster by the validation methods. The higher the number, the more *good* customers are found in this cluster. There is a high volume of matches in **Union**. The percentage of the match is listed in the Percentage (**Perc.**) column.

The **Union** is listed for the objects that fulfil any 4 validations in {VQ1 to VQ5 and RFM}.

In general, for the total number of entities of 387,951, and from table 5.8, there are:

- 141,783 (VQ1) of clients with a margin greater than 1336.41 euro.
- 84,071 (VQ2) indicates customers (CCNber) with more than 1 policy.
- 20,468 (VQ3) indicates customers with a family member (CFNber) who also holds a policy;
- 54,629 (VQ4) for customers who have held more than 1 policy during their

Table 5.8: Experiment 3 Results Across *k* Configurations

Clu.	Entities	Margin	CCNbr	CFNbr	Maxh	YNbr	VQ1	VQ2	VQ3	VQ4	VQ5	RFM	Union	Perc.
C1	387951	1336.4	1.29	0.05	1.16	2.57	141783	84071	20468	54626	151336	55439	50080	100%
C2.1	229702	790.78	1.02	0.05	1.01	1.73	25841	5276	11743	2521	28764	6476	353	0.7%
C2.2	158249	2129.59	1.68	0.06	1.37	3.79	115942	78795	8725	52105	122572	48963	49727	**99.3%**
C3.1	137876	862.06	1.02	0.05	1	1.92	21598	2932	7011	0	26574	6476	805	1.6%
C3.2	174236	1287.14	1.04	0.05	1	2.6	63593	5844	8868	0	72547	20865	2742	5.5%
C3.3	75839	2314.45	2.37	0.06	1.8	3.67	56592	75295	4589	54626	52215	28098	46533	**92.9%**
C4.1	143427	947.93	1.12	0	1.09	2	28054	14952	0	11545	29934	8799	4754	9.5%
C4.2	19253	1286.91	1.28	1	1.17	2.54	6477	3876	19253	2884	7291	2707	3994	8.0%
C4.3	206577	1572.61	1.32	0	1.19	2.88	95480	48006	0	35670	99389	39180	31156	62.2%
C4.4	18694	1767.86	2.35	0.07	1.27	3.57	11772	17237	1215	4527	14722	4753	10176	20.3%
C5.1	73950	657.33	1.01	0.05	1	1.36	4810	1016	3532	52	4665	0	79	0.2%
C5.2	101169	733.39	1.02	0.05	1	1.55	7005	1872	5337	0	5635	0	0	0.0%
C5.3	63259	1088.99	1.04	0.05	1	2.55	16143	2479	3462	0	21264	6023	1171	2.3%
C5.4	81990	2090.08	1.15	0.05	1	4.14	64723	11183	4051	0	75387	25416	9046	18.1%
C5.5	67583	2302.2	2.4	0.06	1.89	3.54	49102	67521	4086	54574	44385	24000	39784	79.4%

tenure.

- 151,336 (VQ5) for customers who stayed more than 2 years.
- In total, there are 55,439 *good* clients for RFM method, where R=F=M=1;
- There are 50,080 customers who fulfil at least 4 validations (Union).
- In this experiment, for any k={2,3,4,5}, clusters C2.2, C3.3, and C5.5 contained more than 80% of *good* customers (**Perc**) with a maximum of 41% assigned to cluster C2.2 (158249/387951=41%). Cluster C3.3 is noteworthy with only 20% of instances and 93% of matches.
- When focusing on clusters of good customers, C2.2, C3.3, and C5.5, the average of most variables is highest. The customers in these clusters appear to stay longer, buy more policies, and pay more money to the company. In summary, our evaluation highlights 3 candidate clusters.

Finally, using our methodology and this dataset, we recommend k=3 for customer segmentation because the *good* customer are nicely positioned in cluster C3.3. In the main, this cluster contains the good customers and *only* the good customers.

5.5.3 Analysis

The first question to address is if we successfully auto-classified insurance customer datasets. A positive response to this question is supported by the results of our evaluation. Our method successfully classifies customer datasets with an accuracy of

90%, as shown by the results in tables 5.7 and 5.8. It is important to point out that while we can accurately classify good customers, if an organisation requires more granular groupings (as k increases), this is more difficult. However, our experiments have shown that by examining the average value for Margin, CCNbr, CFNbr, Maxh, and YNbr in each cluster (table 5.8), an algorithm can label the group of clusters with scores of 99.3%. This is a finding that we did not anticipate prior to performing our experiments.

While trying to determine optimal clusters, there was an interesting finding during the segmentation process. When k (the number of segments) increases, it becomes easier for us to label the type of customers which reside in each cluster. For example, for $k = 3$ in table 5.8, customers in clusters C3.1 and C3.2 are mainly *below average.* The difference between the average value in these clusters is not as great when compared to cluster C3.3. However, for $k = 5$, the class labelling is more acute: C5.5 is the cluster for good customers; C5.1 and C5.2 contain bad customers; and clusters C5.3 and C5.4 are for the mid ranking customers. It is important to note that as the value of k increases the potential for over-fitting also increases.

Variable selection is an important process for prediction algorithms and we employed two of the more popular methods. Analysing the results from this evaluation provides strong evidence that configurations using *all* variables outperform all other variable combinations (Exp3 performs best while Exp2 is second best) when ranking the best clusters. For both recall and precision, the performance for experiments in VQ1 is quite similar to VQ5 because these two variables are highly correlated. For the same reason, the graphs for VQ2 and VQ4 are very similar.

However, the datasets generated using Correlation Coefficient (CC) and PCA should not be discarded completely. While both CC and PCA variable sets perform badly in recall experiments, they perform well in Precision experiments: the CC variable set outperforms others, having the highest (or second highest) accuracy at VQ2, VQ4, VQ5 and RFM and the PCA variable set performs well in VQ2, VQ3, VQ4. This is because CC and PCA variables are very accurate at identifying good customers. The issue is that the size of the good customer cluster is small. However, this does provide an interesting insight for follow-on experiments that use these variable selection techniques.

5.6 Summary

In this chapter, we presented an important component to our work: the ability to categorise customers in board terms, or into coarse grained clusters. Our validation indicates that this automated process has high levels of accuracy. The work was crucial in providing a validation dataset for the later calculation of customer lifetime values: those with high CLV scores should all be located inside the *good customer cluster*. In addition, the process of variable (feature) extraction and selection delivered the dataset that can subsequently be used for imputation of missing variables for final CLV calculations. Our work has now progressed to a point where we can focus on the more fine-grained classification of customers that sees a customer lifetime value score given to every customer, which is the ultimate goal of our research.

Chapter 6

Data Preparation and Experimental Methodology

While our research thus far has provided us with a unified history for each customer and a methodology for segmenting customers according to their value to the organisation, we need to impute the missing variables using the dataset that we have generated so far. What is now required is a final transformation of the data to suit predictive models and to develop an experimental methodology which can optimise the model selection and parameters, to achieve the best set of CLV metrics for each customer. The details of the predictive models we use in our experiments are presented in section §6.1. In section §6.2, a discussion of data transformation, cleaning and encoding is presented. In section §6.3, the experimental methodology used to impute individual-level retention and acquisition is introduced, with a validation plan described in section §6.4. Finally, a summary of this chapter is given in section §6.5.

6.1 CLV Prediction Algorithms

There are two approaches that are applied to CLV calculation and in this section, we present details of the variables used in both models. We begin with presenting the Formula Based approach which is the industry standard. In the following section, we present the probabilistic approach which required a modification to the standard approach normally used, as it was necessary to impute some of the input variables.

6.1.1 Formula Based Approach

We began this element of our research by deploying the well known CLV formula model [7], which is presented in equation 6.1.

$$\begin{gathered} CLV_f = am - A + a * (m - R/r) * [r^n/(1 - r^n)] \\ with \quad r^n = r/(1 + d) \end{gathered} \tag{6.1}$$

where:

- a is the acquisition rate, given a specific level of acquisition costs (A);
- A is the acquisition cost per customer;
- d is the yearly discount rate;
- m is the net income of a transaction;
- R is the retention cost per customer per year;
- r is the yearly retention rate.

In this model, not only has the retention effect been modelled but also the acquisition impact. This model also provides a way to calculate the year of customer stay. A simpler derivation formula to calculate the year of stay is presented in equation 6.2.

$$n = \log_r(r/(1 + d)) \tag{6.2}$$

However, for most yearly renewal cases, the concern of the infinity time horizon (the customer staying) will focus more on the retention [7,37].

Here, a derivate model from equation 6.1 which only concerned the retention impact has shown in equation 6.3.

$$CLV_r = m + (m - R/r) * [r/(1 + d - r)] \tag{6.3}$$

It is necessary to have a model in comparison to evaluate the impact of company expenditure. In this case, we will have model with and wit A simplified model that been presented in [37] has predicted the CLV in an infinite time horizon for the retention rate constant over time. The model without costs and acquisition impact is shown in equation 6.4.

$$CLV_s = m * [r/(1 + d - r)] \tag{6.4}$$

6.1.2 Probability Based Approach

For the probabilistic approach, the Markov Chain Model (MCM) is popular for CLV predictions. In [88], the authors used a MCM approach to determine the CLV with a step by step calculation approach. Matrix $\mathbf{P}$ is the one-step transition matrix representing the probabilities of moving from one period to the next.

Definition 6.1. One step transition matrix

$$\mathbf{P} = \begin{bmatrix} p_1 & 1-p_1 & 0 & ... & 0 \\ p_2 & 0 & 1-p_2 & ... & 0 \\ ... & .... & ... & ... & ... \\ p_n & 0 & 0 & ... & 1-p_n \\ 0 & 0 & 0 & ... & 1 \end{bmatrix}$$

The element p_1 is the probability that a customer would remain after the first year (recency $= 1$). We can use $\mathbf{T}$ to present the cash flows received by the firm in any future period, where the total period is n in Definition 6.2. In $\mathbf{T}$, M presents the margin received by the company when $recency = 1$ at period n. The value for E is the calculated value of these 2 marketing expenditures. Thus, the relationship between the E, R, A is $E = R + A$.

Definition 6.2. Cash Flow Vector

$$\mathbf{T} = \begin{bmatrix} M-E \\ -E \\ -E \\ ... \\ 0 \end{bmatrix}$$

The MCM model to calculate CLV for a customer is presented in equation 6.5. Variables CLV_{mcm}, $\mathbf{P}$ and $\mathbf{T}$ are explained above and the remaining variable d, represents the discount rate. After using equation 6.1, the recency (n) of a customer can then be calculated using equation 6.2.

$$CLV_{mcm} = \sum_{n=1}^{N} [(1+d)^{-1}\mathbf{P}]^n \mathbf{T} \tag{6.5}$$

Combining the customised transaction matrix with cash flow vector, the MCM model to calculate CLV is shown in equation 6.6.

$$CLV_{mcm} = \sum_{n=1}^{N} [(1+d)^{-1} \begin{bmatrix} r_1 & 1-r_1 & 0 & ... & 0 \\ r_2 & 0 & 1-r_2 & ... & 0 \\ ... & .... & ... & ... & ... \\ r_n & 0 & 0 & ... & 1-r_n \\ 0 & 0 & 0 & ... & 1 \end{bmatrix}]^n \begin{bmatrix} M-E \\ -E \\ ... \\ -E \\ 0 \end{bmatrix} \tag{6.6}$$

A derivate computation simpler MCM model has presented in equation 6.7. This model can resolve the problem of the customer relationship value gets small when n increasing. In the equation 6.7, $\mathbf{I}$ is the identity matrix.

$$CLV_{mcmi} = \{\mathbf{I} - (1+d)^{-1}\mathbf{P}]^{-1}\}\mathbf{T} \tag{6.7}$$

6.1.3 CLV variables for Insurance Dataset

In this section, we will introduce and discuss the CLV variables used in our insurance dataset: margin (m), discount (d), acquisition rate (a), retention rate (r), acquisition cost (A), and retention cost (R). The overall company expenditure (E) is a sum of A and R.

Margin (m) and Discount (d).

Margin (m) is the policy premium after tax while the discount (d) is the percentage of price reduction for a policy. Both m and d can be extracted directly from the policy record. The remaining variables A, R, a and r, are connected with expenditure.

Expenditure: Acquisition (A) and Retention (R) costs.

The Retention (R) and Acquisition (A) variables are the costs to retain an existing customer and the cost of an existing customer to buy a new policy. This is related to many channels like Advertising (TV & Digital); Direct Marketing such as customer emails, SMS, mail and outbound calling; Sponsorship Activation and Social media; as well as research which supports both new business and retention in terms of delivering insights.

In this research, we examined the historical company expenditure (R, A) over 6 *Years* for different policy types (*Product*) for each customer. These results are shown in table 6.1, where `N/A` highlights the variable missing from the company database. These values require generating using simple formulae before CLV calculations are possible.

Table 6.1: Company Expenditure with Missing Values

Years	Product	Retention Cost (R)	Acquisition Cost(A)
Year 1	1	N/A	N/A
Year 1	2	N/A	N/A
Year 2	1	N/A	0.02
Year 2	2	N/A	13.11
Year 3	1	N/A	0.89
Year 3	2	N/A	205.44
Year 4	1	1.27	3.19
Year 4	2	1.27	69.97
Year 5	1	5.58	13.87
Year 5	2	5.58	57.37
Year 6	1	8.31	15.29
Year 6	2	8.31	45.62

As can be seen in table 6.1, the R variable could not be determined on a by-product basis and naturally, this continued after missing variables were computed (see table 6.2). However, the investment of existing customer has increased over the years, which does provide different values for each year. In this case, we must calculate the increasing percentage over the years and take an average as the rate of increase, to support the generating of missing values. The A for product 1 has a similar behaviour to R, and we applied the same method to calculate R for product type product 1.

Table 6.2: Company Expenditure for Experiments

Years	Product	Retention Cost (R)	Acquisition Cost(A)	Expenditure (E)
Year 1	1	0.05	0.01	0.06
Year 1	2	0.05	57.65	57.7
Year 2	1	0.15	0.02	0.17
Year 2	2	0.15	13.11	13.26
Year 3	1	0.43	0.89	1.32
Year 3	2	0.43	205.44	205.87
Year 4	1	1.27	3.19	4.46
Year 4	2	1.27	69.97	71.24
Year 5	1	5.58	13.87	19.45
Year 5	2	5.58	57.37	62.95
Year 6	1	8.31	15.29	23.6
Year 6	2	8.31	45.62	53.93

The A variable for product 2 has a significant difference over the years. In this case, to prevent over-fitting, we removed the highest and lowest value for Acquisition cost in product 2, so that the missing value is then the average of the remaining values. After calculation, the company expenditure is applied for different years and policies as shown in table 6.2.

We now provide some examples to add detail to the values in table 6.2. To generate the R value for year 3 for products 1 and 2, the average increase rate taken from the database is calculated using equation 6.8 as 2.94. These values are taken directly from yearly retention costs in table 6.1. To calculate the missing R value for both products in Year 3 is 0.43, as shown in equation 6.9. For Year 2, equation 6.10 calculates this value as 0.15. The same method has applied to generate the R for year 1, and A value for product 1.

$$AvgInc(Y3, P1) = \frac{\frac{5.58}{1.27} + \frac{8.31}{5.58}}{2} = 2.94 \tag{6.8}$$

$$CalcR(AvgInc, Y3) = \frac{1.27}{2.94} = 0.43 \tag{6.9}$$

$$CalcR(AvgInc, Y2) = \frac{0.43}{2.94} = 0.15 \tag{6.10}$$

It is also useful to present an example which provides detail on how we generated the A values. For the A for product 2, to prevent the over-fitting, we first removed the highest value 205.44, and lowest value 13.11 from the Acquisition Cost column for product 2. We then averaged the acquisition costs for product 2 for the remaining years to get a final value for the missing A variable. Using equation 6.11, the product 2 value for year 1 was calculated as 57.65.

$$CalcA(Y1, P2) = \frac{69.97 + 57.37 + 45.62}{3} = 57.65 \tag{6.11}$$

While there are three dimensions to these variables *(customer by product by year)*, the methods we applied to generate missing values in table 6.1 were unable to deliver personalised customer values, but were at the level of product and year. In other words, customers must share the R and A values. This was not a limitation of our approach but rather a limitation caused by available data. Our assumption is that this lack of available data is widespread and the major reason for the highly theoretical approach to CLV research.

Retention (r) and Acquisition (a) rates.

For insurance datasets, data changes (policy renewal) annually. The retention variable (r) is defined as how likely a customer will renew their policy next year. The acquisition rate (a) refers to how likely a customer would buy a policy for the first time. As any insurance dataset contains only existing customers, the acquisition will capture the possibility of an *existing* customer purchasing a *new* policy. With respect to forecasting a customer value at the level of the individual, we must *impute* the two remaining variables r and a for each customer. As this is the final remaining contribution of this research, we provide a detailed description over the next three sections of this chapter.

6.2 Data Preprocessing

An ETL (Extract, Transform, Load) pipeline comprises a series of components extracting data from input sources, transforming the data to match the system's data model, and loading into a data mart (data cube) for reporting and analysis. Our approach, shown in figure 3.5 in chapter 3, is a specialised form of ETL, due to the specific requirements of the task (customer lifetime value) and the nature of the data. In particular, this work began with a dataset that was policy-level and not customer-level. In effect, it was not suited to analysis by customer. Thus, the first step was record linkage where, upon acquisition, data was pivoted to be customer-level, where a customer record contained 1 or more policies. This work was discussed in chapter 4 and while it provided a more holistic customer record, the data was not suited to the imputation algorithms to effectively impute the missing CLV variables.

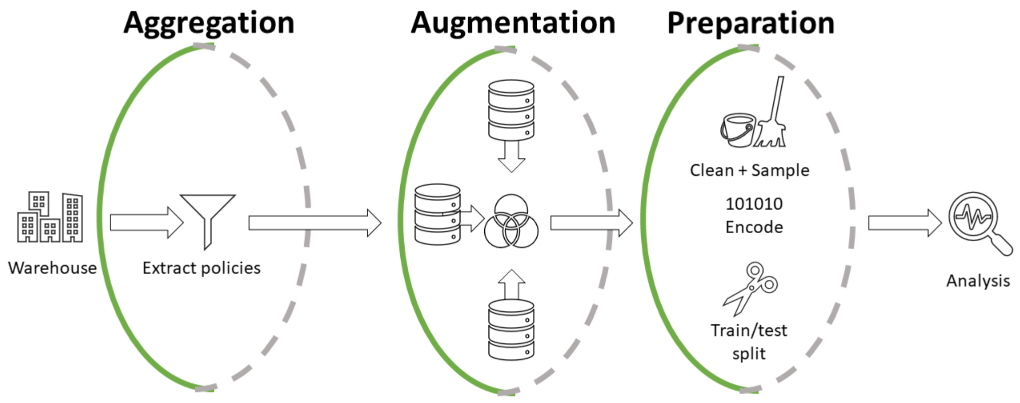


Figure 6.1: ETL Pipeline Architecture

In this chapter, we focus on imputing components of data for CLV computation. The large data imports are combined with other data sources within the warehouse to provide a dataset suitable for predicting *customer retention* and *customer acquisition*. There are three processes involved in the transformation of a dataset suitable for imputation: *Aggregation*, *Augmentation* and *Preparation*, with the process outlined in figure 6.1. The first step, *Aggregation*, constructs the initial *per-policy* and *per-customer* view which provides information on policy renewals and acquisition based on the dataset from chapter 5. The second step, *Augmentation*, adds features to this dataset such as customer information and pricing. These two processes can be equated to the *E* and *T* processes within a standard *ETL* architecture. The final process, *Preparation*, provides a final transformation of the dataset to make it ready for machine learning algorithms.

The details of the initial variables before the aggregation with details is shown in table 6.3. *Name* is the name of the feature; *Description* briefly describes the feature and *Type* indicates if *Categorical* or *Continuous*. For our evaluation, the dimensions representing unique identifiers (*pid* and *cid*) were not used.

Aggregation.

The goal of the first step is to construct a policy centric view containing those policies

Table 6.3: Initial Variable Details

Name	Description	Type
pid	The policy identifier	Categorical
cid	The policy holder identifier	Categorical
premium	The actual policy price customer will pay	Continuous
adjustment	Adjustments a customer mad	Continuous
adjust_amount	Associated value with the adjustments a customer mad	Continuous
year_from	Policy started year	Categorical
year_to	Policy ended year	Categorical
fid	The family identifier	Categorical
gender	The gender of the policy holder	Categorical
county	The county of the policy holder	Categorical
age	Customer age	Categorical

that may or may not been renewed. This involves a RollUp operation on the `detail` view to create an aggregated view containing the policy identifier (`policy_id`), the number of years for which the policy is held (`years_held`) and whether or not the policy was renewed (`renewed`). The variable table 5.1 we presented in chapter 5 is the unified way to construct a customer-centric dataset using the variables in table 6.3. In aggregation process, all the records linked by Record Linkage will be merged.

Augmentation.

The next step is `augmentation` where views within the warehouse are integrated with the policy-centric aggregation. In total, six additional views are integrated: policy prices, family policy holders, latest renewal premium, insurance type, location, and payment method. **Policy Prices** include the average premium and the standard deviation for premium, which can indicate the amount of variation in year-on-year premium prices. **Family Policy Holders** is the number of family members per customer who also hold policies with the company. **Latest Renewal Premium** is the latest premium for a given policy. **Insurance Type** is the type of insurance, which has four possible values: `Private Motor`, `Commercial Motor`, `Home` and `Travel`. **Location** is the county the customer resides in. **Payment Method** indicates if the premium is paid either in full or monthly. Finally, **Gender** relates to the gender of the policy holder. The result is a dataset with fourteen dimensions including a class label of *Renewed* or *Not Renewed* as seen in table 6.4 where: same to table 6.3, `Name`, `Description`, `Type` are presented. For our evaluation, the dimensions representing unique identifiers (`pid` and `cid`) were not used.

6.3 Imputation Methodology

One issue with machine learning algorithms is that different approaches to a data driven problem will have different results and no single machine learning algorithm works best with all datasets. Thus, we now present an approach to use a range of statistical methods in order to determine the best imputation results. We begin with presenting the set of algorithms used to impute the retention value (churn), and then applied the algorithm which works the best to impute both retention and acquisition variables.

6.3.1 Algorithm Selection

The process of determining customer churn in any domain is generally a classification problem with two classes: *Renewed* and *Not Renewed*. The classification methods we employed were: Bernoulli Naive Bayes; Multinomial Naive Bayes; two different support vector machine (SVM) configurations; two decision tree configurations; and a set of artificial neural network (ANN) configurations.

Support Vector Machines. Support Vector Machines [20] are used regularly in classification. For our experiments we used a single linear SVM to provide a baseline to our other methods.

Naive Bayes. Two experimental configurations using Naive Bayes were employed, one using a Bernoulli model and the second using a multinomial model which has been shown to have increased performance on binary data under certain conditions [74]. For both models, 100 different alpha values were used on each, ranging from 0.0 to 1 in degrees of 0.01.

Decision Trees. Two decision trees using the CART (Classification and Regression) algorithm [69] were employed, the first using *entropy* as the splitting measure and the second using Gini impurity. For both approaches, a decision tree was created for each level of depth until the maximum depth was reached and at each depth, test data was used to obtain the accuracy of the tree.

Artificial Neural Networks. Artificial Neural Networks have seen extensive use in predicting customer churn due to their ability to model interactions between features that may otherwise go unnoticed. For our experiments, 20 different configurations of ANNs were constructed with various configurations of hyperparameters. The details of these configuration and the results will be presented in chapter 7.

6.3.2 Data Preparation

Table 6.4 lists the set of policy-centric variables selected after the aggregation and augmentation steps to deliver a dataset ready for retention prediction. At this point, we have all of the necessary data to begin imputation but not all of the data is in a format suitable for the chosen learning models. Thus, A final data preparation step is still required before we can commence experiments.

Table 6.4: Post-Integration Dataset Features

Name	**Description**	**Type**
pid	The policy identifier	Categorical
cid	The policy holder identifier	Categorical
years	The number of years the policy was held for	Continuous
avg_total	The average premium since first purchase	Continuous
std_total	The standard deviation of the premium	Continuous
family	The number of family members of the policy holder who also hold policies	Continuous
renew_p	The current renewal premium price	Continuous
total_p	The current premium price	Continuous
pol_type	The policy type	Categorical
pay_type	The payment type, either Partial (pays monthly) or Full (payment in full on purchase)	Categorical
gender	The gender of the policy holder	Categorical
county	The county of the policy holder	Categorical
province	The province of the policy holder	Categorical
class	Value: Renewed or Not Renewed	Categorical

There are four steps in the preparation phase: *cleaning*, *sampling*, *encoding* and *splitting*. The dataset used for imputation contained 500,859 unique policies, with various policies active across years 1 to 6. Customers left and joined policies at any year. In our first experiment to evaluate the imputation algorithms, we would only predict if the policy was renewed for year 6 or not. The details of numbers of policy renewed and the new policy started in each year shown in table 6.5. Because year 1 is the starting year, there was no historical data for this initial year. In this case, both Renewed and New Business variables are presented as a single value.

The purpose of the *cleaning phase* was to remove data which is missing attributes that are crucial to the machine learning algorithms to function properly. After a cleaning process, 56,993 records were removed, with a dataset with 443,866 instances remaining. The precise

Table 6.5: Yearly Policy Status

	Renewed	New Business
Year 1	187575	
Year 2	131035	82606
Year 3	104106	42198
Year 4	105189	71800
Year 5	130300	59714
Year 6	143247	56966

makeup of this dataset were 300,621 policies not renewed and the remaining 143,245 policies renewed.

The purpose of the *sampling phase* was to ensure an even distribution to maximise the accuracy of our predictive models. Determining whether or not a policy is renewed, in effect, a classification problem. The class labels for each policy are *Renewed* or *Not Renewed*. As is common with real world data, our dataset has a class imbalance where 300,621 records are labelled *"Not Renewed"* and the remaining 143,245 records are labelled *"Renewed"*. This class imbalance can greatly affect classification results ([52]) and three methods are generally employed to resolve this problem: undersampling, oversampling and synthetic sampling.

As we have a large number of records for the minority *Renewed* class, *undersampling* was the method selected to address this issue. Using this method, 143,245 records with the class label *Not Renewed* were randomly chosen so that both classes had the same cardinality. The downside to this approach is that some of the *Not Renewed* data could increase the effectiveness of our analysis which is addressed in our discuss in chapter 7.

After undersampling, the dataset comprised 286,490 records, with an equal distribution of the *renewed* and *not renewed* classes, with 143,245 instances. The *encoding step* transforms categorical dimensions so they are ready for machine learning algorithms. The dimensions encoded were *insurance_type*, *payment_method* and *county*.

Table 6.6: Training / Testing Data Split

	Renewed	Not Renewed
Train	114,495	114,697
Test	28,750	28,548

The final *splitting step* divides the data into training and testing sets using the 80/20 rule. The final distribution is shown in table 6.6, as an almost identical split. The evaluation metrics *Accuracy*, *Precision*, *Recall*, *Specificity* and *F* score will assigned to validate the result.

6.4 CLV Experimental Evaluation Plan

We now present the list of experimental configurations in table 6.7 to try and determine the best set of CLV calculations for customers. Here, the experiment *Exp.* identifies each experiment with different variable configurations. The extraction and generation process for variables m, d, A and R was described earlier in section §6.1.3. The retention cost (R) and acquisition cost (A) for a specific year, and product are extracted from table 6.2. In section §6.3, we described how the individual-level acquisition rate (a) and retention rate (r) are acquired. The best-performing algorithm for churn analysis will be applied to predict customer retention (r) and acquisition (a). Because there is a specific requirement for variables assigned to each model, we customise the prediction method to get the required variable. For the MCM model, the value of customer retention probability is required on a yearly basis.

Table 6.7: CLV Experiment Plan

Exp.	Model	m	d	a	A	r	R	E	Start	Period
1	Equation 6.1	*	*	*	*	*	*		Year 1	$p \in \{1,2,3,4,5,6\}$
2	Equation 6.3	*	*	*		*			Year 1	$p \in \{1,2,3,4,5,6\}$
3	Equation 6.4	*	*			*			Year 1	$p \in \{1,2,3,4,5,6\}$
4	Equation 6.5	*	*			*		*	Year 1	$p \in \{1,2,3,4,5,6\}$
5	Equation 6.7	*	*			*		*	Year 1	$p \in \{1,2,3,4,5,6\}$

An "*" in the column means that variable was used for the experiment.

The column *Model* indicates which methodology from section 6.1 was deployed for this experiment. The column *Start* shows the commencement year of the records used and for these experiments, always began at the first year. The reasoning for this is that for a *long term* prediction, we must start from the beginning, which is year 1, Which involves 187,575 policies made by 166,570 customers.

The *Period* expresses the duration the deployed experiment will predict, where Period p will have a set ranged from value 1 to 6. The value for p is decided by the equation 6.2, a dynamic value depending on the inputs. This is a critical variable for CLV calculation because, intuitively if a customer is staying longer, their CLV will be higher. More importantly, unlike all previous variables so far, this variable is calculated at the individual. Thus, the allowable parameters for the *Start* and *Period* combination for each individual customer are $\{(1,1),(1,2),(1,3),(1,4),(1,5),(1,6)\}$.

For a genuine comparison, a decision was made to run the 5 models with 3 sets of configurations. The first set is to calculate without all the work we did in this research, which means without record linkage and the CLV variable imputation. In this experiment, we will still have m and d at an individual level; A and R break down to the product level; and a and r are at a yearly scale, but not an individual level. In the second experimental configuration, we apply our record linkage work but not the CLV imputation. At the final set, we apply all of the data linkage, transformation and preparation work carried out in this research.

We also evaluate the difference between long term prediction and the impact of the dataset itself. After evaluating the models in table 6.7, the best model is selected to test the performance with different starting years and period, where: $CLV_{(s,p)} \in \{(year1 \cup year2, n), (year1 \cup year2 \cup year3, n), (year1 \cup year2 \cup year3 \cup year4 \cup year5, 1), (year1, 1)\}$.

For existing customers, we impute the acquisition for those customers as a percentage of how likely they are going to purchase a new policy. For the experiment which involves the input from multiple years, the value of the variable will be the average across those years.

6.4.1 CLV Validation

We use a cross-validation methodology to evaluate the CLV results. We use the objective validation method, which is validated using the margin variable. The second validation approach is the subjective validation method which using the pre-classified data we constructed in chapter 5.

Validate using Margin.

In many CLV definitions, CLV been regarded as "*The dollar value of a customer relationship based on the present value of the projected future cash flows from the relationship*" - from [32]; and "*a metric used in many industries, is based on the accumulated cash flow a customer accrues during his or her lifetime*" - from [61]. In this case, the *margin* a company can get from a customer is the most important measure. Based on this, we choose to use the margin variable as our first validation approach.

There are three scenarios to validating using the margin for a long term CLV:

- The first uses the overall $Margin$ across multiple years, as shown in equation 6.12. The total margin is the sum of the margin (m) customer paid each time (t), where the number of years, N=6.

$$Margin = \sum_{t=1}^{N} m \tag{6.12}$$

- For comparison reasons, we would like to see the CLV prediction capability to predict a *recent* year. In our second validation approach, the time variable t=2 and the years, N=1
- To prevent over-fitting the value we used for training the prediction model, we will remove the year we used for training, meaning where $n \neq n_{train}$. In most cases, $n_{train} = year1$.

For all models validated using margin, we validate how the CLV model predicts for the best 20% and the worst 20% of customers. The first set of the experiments for the 5 models will only validate using this method.

Validate using Pre-Classified Dataset.

We constructed a predefined customer dataset where the unified customer was assigned into one of three groups: {good, bad, average}. In our identified customer groups, the good customer does not only have a high margin but also purchases more frequently and is loyal (stays longer) to the company. Using this classified dataset, we validate the customer in these three groups, treating each group separately.

6.5 Summary

Using the research presented in chapters 4 and 5, we have established a platform by which we can compute CLV metrics for all customers. This work linked customer policies together to provide a *per-client* history and then used a clustering algorithm to segment customers into appropriate groupings. However, before we could proceed to imputing the missing variables necessary for the CLV calculation formula, a final transformation of the dataset was required together with a robust methodology for determining the best predictive model and thus, CLV metrics for each customer. In the next chapter, we present our imputation methodology, experiments and results together with a discussion of our insights and findings.

Chapter 7

Experiments, Findings and Discussion

In this chapter we present the results for identifying the imputation method using churn dataset in long term in section §7.1 and churn and acquisition analysis in section §7.1.3. The experiment with using the result from retention and acquisition as input to calculate CLV in two different models will be present in section §7.2. A detailed analysis with suggestions to the person who is the manager will be presented in section §7.3.

7.1 Long-term Churn Evaluation

We begin this section with an overview of the 4 different algorithms in isolation, reporting on their relative performances. We then take a comparative view across all algorithms, using different configurations for the more complex models.

7.1.1 Outline Results

The four machine learning models used in our evaluation were Support Vector Machines, Naive Bayes models, Decision Trees and Artificial Neural Networks. The experiments were developed using the $scikit-learn$(0.21.0) and $tensorflow$(1.15.0) libraries. As will be shown, results varied across each model.

Support Vector Machines.

Unsurprisingly the Linear SVMs show a weak performance with an accuracy of 0.754 and an F score of 0.76. However, this model was always intended as a baseline for our evaluation of other models and as such, provides a value to our research.

Naive Bayes.

For both Naive model types, 100 different alpha values were used, ranging from 0.0 to 1 in degrees of 0.01. Interestingly, these changes had no effect on the accuracy score across model configurations.

Decision Trees.

For the algorithm which incorporated entropy, the best performing tree had a depth of 11 with an accuracy of 88.82%. For the Gini-tree, the best performing depth was also 11 and with a very similar accuracy of 88.72%.

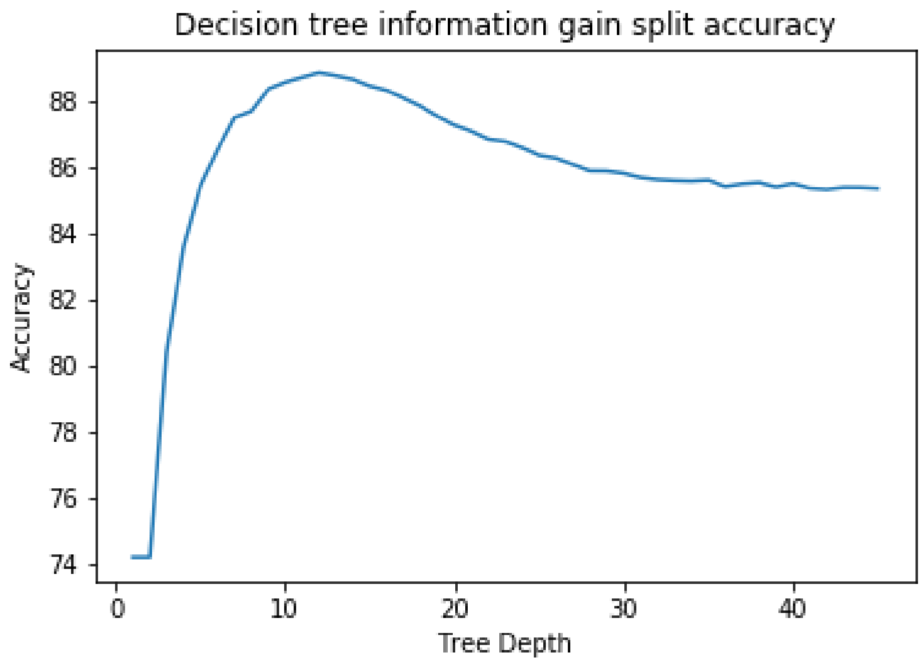


Figure 7.1: Accuracy of Entropy-Split tree *vs* Tree Depth

Accuracy values when compared to the level of depth for both models, can be seen in figure 7.1 for the *Entropy-Split* and 7.2 for the *Gini-Split*. Both models were tested with a depth value from 0 to 45. We observed accuracy increasing before the depth reached a value of 11. While the accuracy dropped just after this point, it was encouraging to see the model become stable, remaining at 85%. The difference in starting performance is based

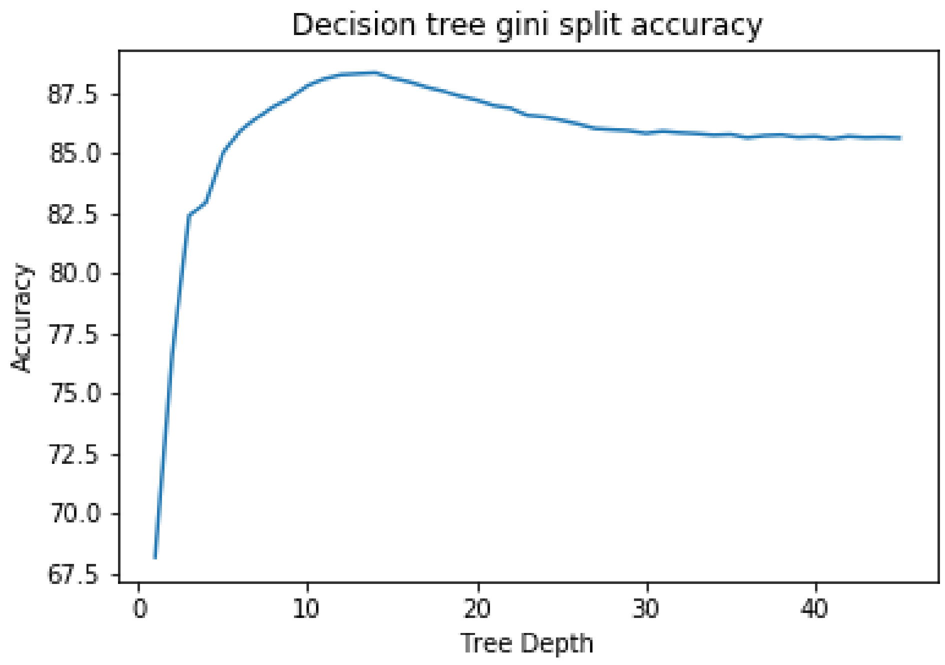


Figure 7.2: Accuracy of Gini-Split tree *vs* Tree Depth

on the algorithm used to split the data, Entropy split seeks to use information, or lack of information to correctly identify the next attribute to split on when constructing the tree whereas Gini calculates the probability of a mis-classification and uses that as the splitting metric. As a result, the starting rate for the gini-split tree is below 70%, while this rate for the entropy-split tree is above 74%.

For shallow trees we can see that Entropy outperforms the Gini split. The decrease in accuracy is a product of both the data itself and the Gini algorithm. If a tree is shallow, the entire dataset may not be examined in enough detail to construct an accurate tree i.e. for shallow trees, only a fraction of all potential attributes have been examined which may not be enough to correctly classify any unseen data. The details of evaluation the result for decision tress will be provided in Table 7.2.

Artificial Neural Networks.

An artificial Neural Network (ANN) uses interconnected artificial neurons (nodes) to simulate the functioning of a human brain to process information. It is composed of layers, nodes and synapses. Each layer contains a number of nodes, and nodes are connected between layers through the use of synapses. The first layer is the input layer which is the layer that receives external data. The final result is produced at the output layer. In between the input and output layer are zero or more hidden layers (containing hidden nodes). The layers in the network are only connected to the immediately preceding and immediately following layers. A graph of a simple neural network with 2 hidden layers is shown in figure-7.3.

When we train a neural network the training dataset may be iterated multiple times. An epoch refers to one cycle through the full training dataset. Usually, neural network takes more than one epoch, however a high number of epochs may result in over-fitting.

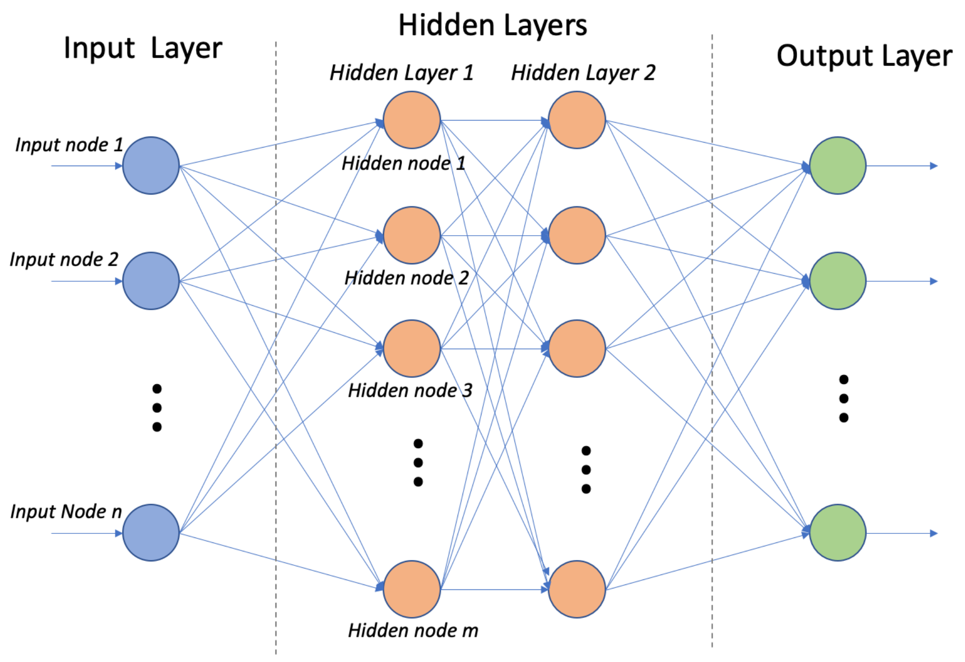


Figure 7.3: Artificial Neural Network Example

The results of the 20 ANN configurations can be seen in Table 7.1, where *id* is the experimental id; *epoch* is the number of epochs (a range between 10 to 100); *hlayer* is the number of hidden layers (1 or 2); *hnode* is the number of hidden nodes with a number between 5 to 66; *drate* used in regularisation, is the random dropped percentage of instance from the data to avoid over-fitting; *tr_ ac* is the accuracy of the training data; *tr_ l* is the loss of the training data; *te_ ac* is the accuracy on the test data; and finally, *te_ l* is the loss on the test data. For all configurations, a maximum dropout rate of 0.05 was used. The table has sorted by test accuracy from the highest to the lowest.

The results of the top 5 performing configurations are bolded in table 7.1. The top 5 were selected by rank accuracy for the experiment in test result. For those top experiments, most of them used 2 hidden layers; all of them had the hidden node number greater than 30 with

Table 7.1: Configuration of the ANNs

id	epoch	hlayer	hnode	drate	tr_ac	tr_l	te_ac	te_l
5	**50**	**1**	**31**	**0.02**	**0.892**	**0.251**	**0.889**	**0.259**
17	**37**	**2**	**66**	**0.02**	**0.892**	**0.252**	**0.887**	**0.266**
18	**37**	**2**	**36**	**0.02**	**0.891**	**0.257**	**0.886**	**0.263**
15	**37**	**2**	**56**	**0.02**	**0.889**	**0.26**	**0.885**	**0.268**
12	**50**	**2**	**56**	**0.02**	**0.893**	**0.25**	**0.884**	**0.265**
9	50	2	47	0.01	0.892	0.255	0.883	0.273
13	100	2	56	0.02	0.895	0.241	0.883	0.289
14	25	2	56	0.02	0.888	0.259	0.883	0.267
1	10	1	31	n/a	0.882	0.275	0.882	0.278
6	100	1	31	0.02	0.895	0.243	0.882	0.283
10	50	1	16	0.02	0.889	0.256	0.881	0.285
11	50	2	32	0.01	0.889	0.256	0.881	0.285
4	10	1	5	n/a	0.88	0.28	0.879	0.288
7	75	1	31	0.05	0.893	0.249	0.879	0.299
19	50	2	48	0.02	0.892	0.249	0.876	0.283
16	44	2	56	0.02	0.892	0.25	0.875	0.293
2	10	2	47	n/a	0.883	0.273	0.873	0.293
3	50	1	31	n/a	0.892	0.249	0.872	0.299
8	75	1	31	0.01	0.894	0.244	0.862	0.318
20	50	2	60	0.02	0.892	0.249	0.857	0.334

0.02 dropout rate, with epochs between 37 and 50. Of course, the issue with neural networks is that is is difficult to interpret *how* we got the actual results and thus, we focus only on the results themselves.

In general, the training accuracy is higher than the test accuracy and those two values are correlated. This means if the training accuracy for the labelled experiment is high, the accuracy for the test dataset will also be high. Nevertheless, this would have exceptions. The experiment 6 and 13 had the highest accuracy on training, however, the training model did not perform well for testing dataset.

From table 7.1, experiment 5 is the best performing with an accuracy of 0.888 on the test dataset. This model consisted of one hidden layer with 31 hidden nodes with a dropout rate of 0.02%. There were other models with increased training accuracy *tr_ ac* but are not shown as they have a lower *te_ ac* than 88.6% which is generally an indication of over fitting.

Table 7.2: Comparison of classification methods

method	**acc**	**pre**	**rec**	**spe**	F	**tp**	**tn**	**fp**	**fn**
NB - Bernoulli	0.776	0.911	0.717	0.879	0.802	25999	18477	2551	10271
NB - Multinomal	0.691	0.651	0.706	0.679	0.678	18594	21023	9956	7725
DT - Entropy	0.887	**0.938**	0.851	**0.931**	0.892	**26766**	**24048**	1784	4700
DT - Gini	0.887	**0.938**	0.851	**0.931**	0.892	**26767**	**24055**	1783	4693
SVM - Linear	0.754	0.779	0.742	0.769	0.760	22234	20997	6316	7751
ANN - 5	**0.889**	0.920	**0.867**	0.914	**0.893**	26449	**24478**	2301	4070
ANN - 12	0.884	0.930	0.852	0.922	0.889	26730	23917	2020	4631
ANN - 15	0.885	0.924	0.858	0.917	0.890	26578	24133	2172	4415
ANN - 17	0.887	0.937	0.853	0.929	0.893	**26934**	23893	1816	4655
ANN - 18	0.886	0.919	0.863	0.912	0.890	26412	24370	2338	4178

7.1.2 Churn Analysis

Table 7.2 provides a comparison across all experimental model and parameter configurations. *Method* is the classification algorithm and configuration used; *acc* is the overall accuracy; *err* is the overall error; *pre* is the precision; *rec* is recall; *spe* is specificity; and F is the F score. Overall, most models performed well with 7 of the experiments achieving an accuracy > 0.88, with 6 of those having an F score > 0.89. Interestingly, the difference between the two decision tree methods (Entropy & Gini split) was so small (> 5 decimal places) that they effectively performed the same. The worst performing method was the multinomial Naive Bayes with an accuracy of 0.69 and an F score of 0.678.

In terms of accuracy, the best performing model was *ANN-5* with an accuracy of 0.889. This configuration also achieved the highest F score with 0.893. On the other hand, both decision tree methods have higher precision (0.938 *vs* 0.920) and specificity (0.931 *vs* 0.914) scores. However, *ANN-5* had a higher recall rate (0.867 *vs* 0.851). Between these three high performing configurations, *ANN-5* had the highest number of true negatives (24,487) while both decision tree methods had a higher number of true positives (26,766 and 26,767). By examining the *NB-Bernoulli* model results, there is a clear requirement for more in-depth statistics than accuracy alone. This method has an overall accuracy of 0.776 but there is a difference between the measures for recall and specificity (0.717 and 0.879 respectively), indicating that this method is better at predicting negative classifications over positive ones.

If we examine the ANN configurations, while *ANN-5* has the highest overall accuracy and F score, other methods show a higher value for specificity (the highest being *ANN-17* with 0.929). However, *ANN-5* shows the highest value for recall out of all neural network configurations indicating that it performs best when predicting positives classes. A high recall value necessitates a low *false negative* rate. Out of all methods employed *ANN-5* has the lowest rate of *fn*, with 4,070 records being classified incorrectly.

The question of which model and configuration to use depends ultimately on the classification most important to businesses. From our findings, a decision tree classifier is recommended for organisations that wish to obtain the highest number of correct *negative* classes, while *ANN-5* should be used if the goal is to correctly identify the highest number of *positive* classes. As the ultimate goal of our research is to impute retention and acquisition rate for its use in CLV calculations, the *DT-Gini* performance the best for this. However a decision tree does not produce a probability, only the predicted class, as such *ANN-5* is ultimately selected for CLV calculations.

7.1.3 Retention and Acquisition Predictions

From section 7.1, ANN-5 was selected as the best performing algorithm for long term customer retention prediction. We will also applied this method to predict the *year on year* retention to construct the Matrix **P** and also the acquisition rate (a) in the long term. For the retention prediction for long term, we achieved an accuracy of 0.889. However, for yearly predictions, it has less training data and thus, the yearly retention rate accuracy is $\sim 75\%$. Accuracy for the prediction of acquisition steadily worsens as only $\sim 2\%$ of customers buy new policies in the coming years and thus, acquisition rate accuracy was calculated at $\sim 65\%$.

There is another important variable which is crucial for determining the CLV value: the number of years (n) the customer remains with the company. Because the n value could be greater than 1, it is necessary to predict for a year-on-year dataset. In chapter 6, this n value was determined using equation 6.2. Our first step in CLV prediction is to validate the accuracy of the n value. The correlation coefficients between n and the customer's actual stay is > 0.88. This means that when the predicted value for n is high, the customer actually stayed longer with an accuracy rate of 0.88.

For the first step, we will need to clean the records that are not suitable for CLV models. Moreover, data used for training may not be suitable for analyses as the classes will be pre-learned by the model, ultimately outputting predictions of probabilities 0 and 1 for previously encountered records. Such records need to be removed from the dataset prior to CLV analysis.

To prevent the over-fitting, it was necessary to remove all associated records where this customer had a retention rate < 0.1 or > 0.99 for a policy, or an acquisition rate < 0.1 or > 0.99. as a result, 86,553 out of 166,570 individual customer records are removed, meaning

80,017 customers with 84,924 associated policies were applied for CLV calculations. The CLV validation dataset provided in chapter 5 was segmented into 3 categories: {good, bad, and average}. It is necessary to verify that the deleted records will not affect the categorisation of customer segments. Table 7.3 lists the records in each customer category before and after the deletion process for the purposes of comparison.

Table 7.3: Dataset Comparison

	Customers in Year 1	After Cleaning
Good	49,199	24,825
Average	8,217	3,588
Bad	109,154	51,604

The records remaining in each group remained at the same size in percentage terms, as the full recordset. For example, there are now 29.5% (49199/166570 $\approx$ 29.5%) good customers in year 1, while previously, there were 31% (24825/80017 $\approx$ 31.0%) of them in the good customer category. Because the percentage of each group remains the same after the cleaning process, we are now able to assign the 80,017 customer records with 84,924 associated policies to make long term CLV predictions using the 5 predictive models.

7.2 CLV Evaluation

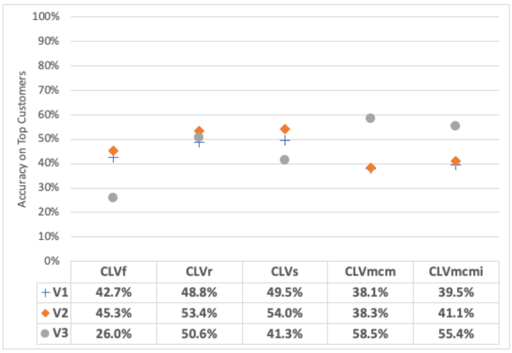
In this section, we validate the experiments set up earlier in section §6.4. For our evaluation, we examine our 5 CLV models against the datasets generated from each stage of our work. There are three datasets:

- **V1** - This is the baseline dataset with no record linkage performed or data imputation.
- **V2** - This is the dataset after record linkage has occurred but *prior* to imputation. For this dataset the data is grouped yearly meaning the variables a and r are identical for customers within the same year.
- **V3** - This is the final dataset with record linkage and the imputation of the acquisition and retention variables.

All three datasets are compared and then are validated by $Margin$ for the 6 years within the dataset. We further validate the models in Dataset 3 ($V3$) with different $Margin$ combination and a *pre-classified customer dataset.* Finally, we validate using an average for variables across *multiple* years to compare customer behaviour.

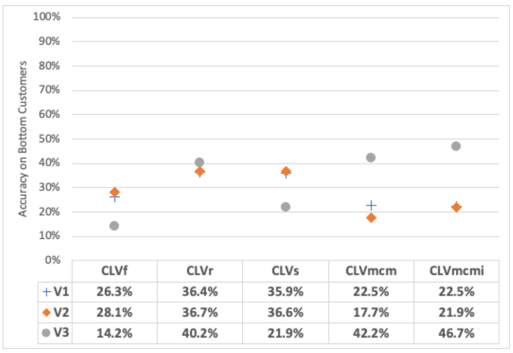
7.2.1 Validate using *Margin*

We used the *Top* and *Bottom* 20% of customer records (16,000), ranked by margin, to validate the results. The top 20% customer are mainly the most profitable customers while the bottom 20% is the opposite in terms of customer value. The margin across multiple years is the *aggregation* for an individual policy. The baseline will provide a good view of the behaviour of multiple policyholders. The details of the result for each model across the top and bottom 20% of the three datasets is been provided in figure-7.4 and figure-7.5.



	CLVf	CLVr	CLVs	CLVmcm	CLVmcmi
+ V1	42.7%	48.8%	49.5%	38.1%	39.5%
◆ V2	45.3%	53.4%	54.0%	38.3%	41.1%
● V3	26.0%	50.6%	41.3%	58.5%	55.4%

Figure 7.4: Accuracy for Top Customers Across 3 Versions



	CLVf	CLVr	CLVs	CLVmcm	CLVmcmi
+ V1	26.3%	36.4%	35.9%	22.5%	22.5%
◆ V2	28.1%	36.7%	36.6%	17.7%	21.9%
● V3	14.2%	40.2%	21.9%	42.2%	46.7%

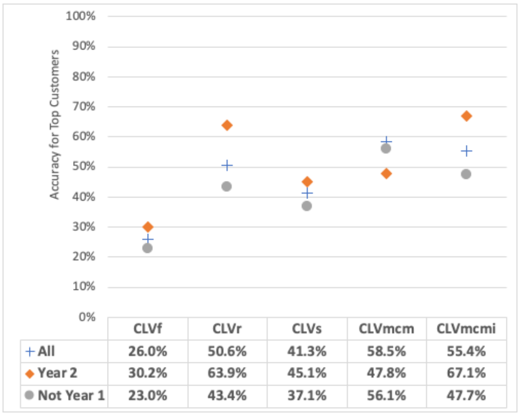
Figure 7.5: Accuracy for Bottom Customers Across 3 Versions

Figure 7.4 highlights the accuracy for all CLV methods for the most profitable (*Top*) 20% of customers and figure 7.5 presents the least profitable (bottom) for all three datasets. From both top and bottom accuracy graphs, on average, the accuracy rate increased after each step for CLV preparation (i.e. V1 to V3). The Record Linkage process improved the accuracy with an average rate of 2.7% using the top customers matched (from V1 to V2); the average rate of 4.3% increased from V1 to V3. However, the accuracy differs in models depending on input variables, so when selecting the CLV prediction models, one must be careful as it depends on requirements.

In general, the formula-based CLV calculation models outperforms the probability-based model, MCM, in V1 and V2. The performance of each type of model had a similar curve. It is interesting to note that the model which used imputations had reduced the accuracy. However, it did rapidly improved with probability-based models. The yearly-individual-level CLV variables associated with MCM models outperformed the other two versions. It has an accuracy of 58.5% for profitable customers and 46.7% for non-profitable customers.

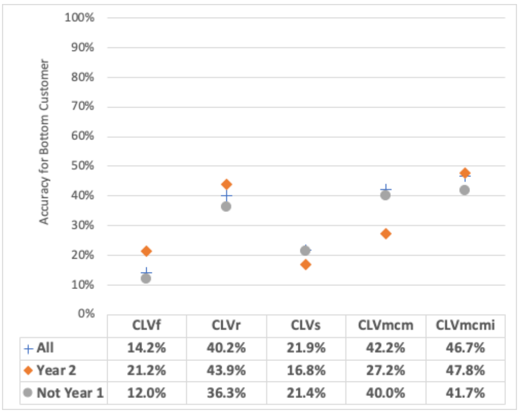
The formula-based models: CLV_f and CLV_s showed decreased accuracy after the imputation process. Moreover, the model CLV_f had the poorest performance in comparison to other model configurations. However, for the model CLV_r, the accuracy remained largely the same across experiments. Moreover, unlike the other two ormula-based experiments, its accuracy increased using V3 to identify non-profitable customers.

We evaluated the effects of margin across years with the details were provided in §6.4.1 in chapter 6. Figures 7.6 and 7.7 show the 3 different margin scenarios applied to validate the 5 CLV models. Similar to the baseline validation, we also validated using the top and the bottom 20% of customers in ranking by margin across the years. Here, *All* means using total margin across 6 years for validation; *Year 2* uses margin in only year 2 for validation. The last column *Not Year 1* in the graph means using all years other than the margin for year 1.



	CLVf	CLVr	CLVs	CLVmcm	CLVmcmi
+ All	26.0%	50.6%	41.3%	58.5%	55.4%
◆ Year 2	30.2%	63.9%	45.1%	47.8%	67.1%
● Not Year 1	23.0%	43.4%	37.1%	56.1%	47.7%

Figure 7.6: Accuracy for Top 20 percent of customers



	CLVf	CLVr	CLVs	CLVmcm	CLVmcmi
+ All	14.2%	40.2%	21.9%	42.2%	46.7%
◆ Year 2	21.2%	43.9%	16.8%	27.2%	47.8%
● Not Year 1	12.0%	36.3%	21.4%	40.0%	41.7%

Figure 7.7: Accuracy for Bottom 20 percent of customers

In general, the shape of both graphs are quite similar. Models which performed well in predicting profitable customers will also have a good performance when predicting non-profitable ones. However, the accuracy rate for the bottom 20% is lower than the rate for the top 20%. The experiment which validated using only year 2 margin had a reasonable result in accuracy for profitable customers with a rate of 67.1%. The the accuracy for the bottom customers is 47.8%.

By checking the margin average in a different configuration, year 2 performed best on predicting profitable customers with an average rate of 50.8%. However, the *All* margin performances were slightly better when predicting non-profitable customers with an average rate of 33.0% across multiple models.

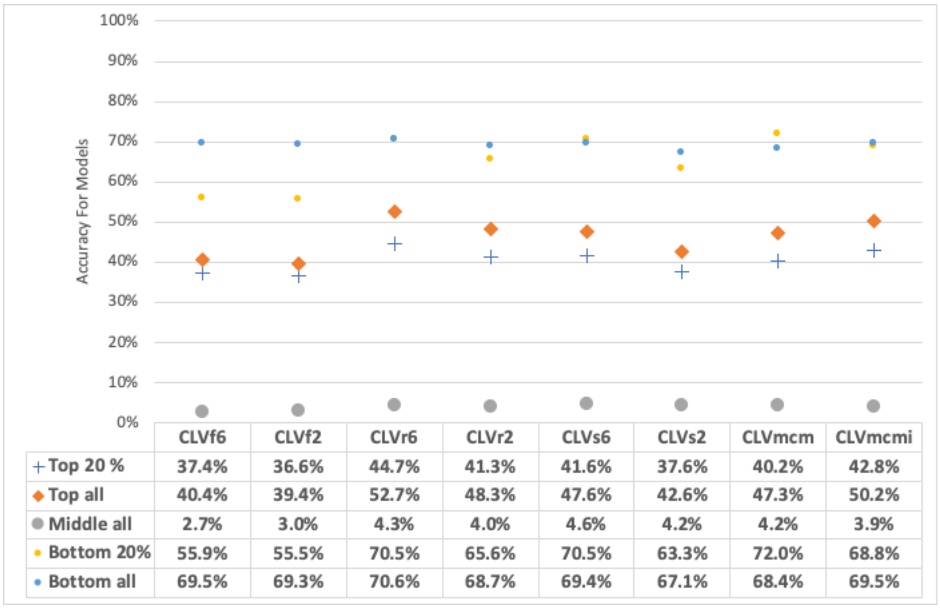
The formula-based (CLV_r) and MCM models in the infinity horizon (CLV_{mcmi}), both had good prediction rates when identifying profitable customers in the coming year. Moreover, the CLV_r performance is the best in formula-based model and the CLV_{mcmi} is the best in a probability model. Both of these models rely heavily on the retention rate. In contrast, the CLV_f model did not perform well for any validation sets.

For the validation method requiring a margin, our research suggests including the initial year utilised for training because, by the definition of CLV, this first year contributes to CLV metrics also. Additionally, it is a crucial value for all CLV prediction models.

7.2.2 Validating Using *Pre-defined Customer Groups*

In this section, we report on validating the individual CLV prediction model using the pre-defined customer groups.

To validate CLV prediction results over the longer term, we use the validation dataset shown in chapter 5, where the customers were segmented into 3 groups by the k-means clustering method. The three groups naturally formed three categories as *Good* (Top), *Bad* (Bottom), *Average* (Middle) customers. We segment the prediction results and measure their accuracy for each category. Principally, high CLV customers should be classed as good customers. However, in addition to examining the accuracy overall, we would like to see the performance of the top 20% of good and bad customers specifically. In this case, we use 5 validation rules to validate the accuracy of the 3 customer categories and the top and bottom 20% of the customers. The validated results for the prediction models are presented in figure 7.8.



	CLVf6	CLVf2	CLVr6	CLVr2	CLVs6	CLVs2	CLVmcm	CLVmcmi
+ Top 20 %	37.4%	36.6%	44.7%	41.3%	41.6%	37.6%	40.2%	42.8%
◆ Top all	40.4%	39.4%	52.7%	48.3%	47.6%	42.6%	47.3%	50.2%
● Middle all	2.7%	3.0%	4.3%	4.0%	4.6%	4.2%	4.2%	3.9%
• Bottom 20%	55.9%	55.5%	70.5%	65.6%	70.5%	63.3%	72.0%	68.8%
• Bottom all	69.5%	69.3%	70.6%	68.7%	69.4%	67.1%	68.4%	69.5%

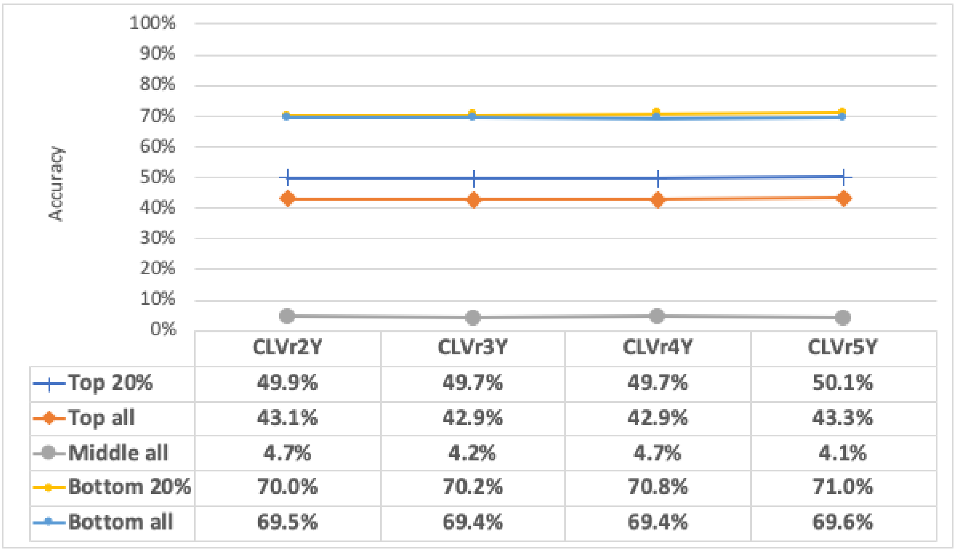
Figure 7.8: Validation Result Using Pre-Classified Dataset

The yearly retention rate (r) is imputed for this experiment and we compare the difference between using a long term retention rate and a recent retention rate on CLV predictions. This is only applicable for formula-based models. In this case, for each formula-based model, we input two different r values: year 2 and year 6 (as retention over the longer term) into the formula-based model. Ultimately, we will have 3 more experiments depending on the assignment of r values. For models that used the retention rate in year 2, the model will have a post-fix of 2, and for the long term will have a value of 6.

Overall, the performance across models was quite similar, having similar accuracy for each validation rule. Specially, these results refer to the two accuracy measures for good customers and the matches on the bottom 20% of bad customers. For example, if one rule had a high value in a model, the other two rules will show a similar performance.

The middle group was hardest to assess with the worst accuracy of 3.9% across multiple models. On a more positive note, the accuracy for bad customers had an average matching rate of 69.1% across the models. The highest performing method for good customers is CLV_{r6} at 52.7% accuracy. The best method for bad customers being CLV_{mcm} (72%), for the bottom 20% of customers. In general, the accuracy for bad customers is better than good customers while the accuracy for customers in the middle group performing worst.

The CLV model CLV_r shows slightly better results across the 5 validation methods in terms of predictions based on different categories of customers. For a further study, we use evaluation methods with different *Period* and *Start* year, applied to this model. The result of the chosen experiment described in §6.4 are shown in figure-7.9.



	CLVr2Y	CLVr3Y	CLVr4Y	CLVr5Y
Top 20%	49.9%	49.7%	49.7%	50.1%
Top all	43.1%	42.9%	42.9%	43.3%
Middle all	4.7%	4.2%	4.7%	4.1%
Bottom 20%	70.0%	70.2%	70.8%	71.0%
Bottom all	69.5%	69.4%	69.4%	69.6%

Figure 7.9: Predict CLV using Average Value Across Different Years

The performance of the same model using different years do not show much difference in terms of the accuracy of results. Their accuracy rate on the top 20% of good customers increased compared to predictions using only one year. However, the accuracy using all good customers reduced. For the other 3 validation rules: Middle, Bottom 20% and Bottom all, shows a similar performance to the experiment using only the 1-year training set. Calculating using the all 5 years to predict CLV for year 6 had slightly better accuracy. Again, in this set of experiments, it was shown that the highest accuracy is obtained from the highest and lowest performing customers.

7.2.3 CLV Prediction Analysis

The validation result presented in §7.2.1 shows that our work has significantly contributed to CLV predictions, with increasing accuracy at each stage. Less than 6% of customer are multiple policy holders so in our analysis, the increase in accuracy between V1 and V2 seems reasonable. It is interesting that the *bad* customer is easier to predict in comparison to the 2 other groups. The accuracy for bad customers was 30% higher than the average across the models. However, it has the opposite performance in predicting good customers.

The MCM and CLV_r model had a increase after the imputation because it relied heavily on a accurate retention rate. The MCM required inputs from the retention matrix $\mathbf{P}$ and if there was a requirement to reuse those models, it needs a good retention prediction method at the individual customer level. The CLV_f model performed worst due to the fact that this model did not place much emphasis on acquisition. However, our acquisition result is not as good as the retention result: that is the reason for reduced accuracy.

For validating CLV metrics, we suggest using our pre-classified customer groups instead of the purely focusing on margin. The reason for this is that our segmentation identified good customers with more dimensions, not only on margin but also concerning the loyalty such as staying longer, brought more frequently etc.

The undersampling process decreased our accuracy on year to year retention prediction and the acquisition. This was especially so for acquisition, where the dataset imbalance problem is worse than the year to year retention prediction. Moreover, by training the acquisition prediction model, it seems we are missing the key variables at the same time. This means that changing of the configuration of the acquisition training model, has not had a positive effect on the prediction accuracy.

As a result of this research, we can now predict Customer Lifetime Value for Irish Insurance

Datasets in the long term. The variables for CLV prediction are common to both insurance and a number of other domains. As a result, our work is transferable and has the potential for further exploitation. Moreover, from the results presented in our evaluation, those selected CLV models appear sufficiently robust to predict customer values.

7.3 Final Discussion

We have 21% of customers who are multiple policyholders and overall, 387,951 customers made 36% of the profit. This figure does not fulfil the 20-80 rule because the policy appeared to get more customers more likely to stay. The policy price for a new customer can be several times higher than a long staying customer. Moreover, by the result we presented in chapter 5, the multiple policy holders are loyal to the company and retaining them can nearly double the profit. However, this cannot be achieved without the Record Linkage process presented in this dissertation.

After record linkage, we provided an approach for managers to segment customers depending on the business requirements with subjective and objective validation methods. This segmentation research provided an auto-classified customer dataset, usable for later CLV validation. In addition, customer segmentation has already been a very useful topic for the industry to manage customer behaviours and provide support for business decisions. Our experiments have shown that using a key variables Margin, CCNbr, CFNbr, Maxh, and YNbr with AdjAmt, AdjCT, Age, County, Disc, Gap and Gender in each cluster (table 5.8), an algorithm can label the group of clusters with an accuracy of 99.3%. Our segmentation work in chapter 5 has a clear step by step process for classifying customers from the beginning. Also, if you want to construct your own variables, a detailed ETL process was also provided in terms to cover the acquisition and transformation of variables.

In figure 7.8, we demonstrated the ability to identify good customers with an accuracy of 52.7% (CLV_r) in the longer term, and for identifying low ranking customers, the rate increased to 72% (CLV_{mcm}). The CLV was gradually improved through the processes developed in this research. If organisations have accurate, year by year, individual-level variables, the MCM model is suitable for this dataset. It is more flexible and could present the result in a step-by-step fashion for a better understanding as to how final results are reached. If the manager only requires a simple calculation, the CLV_r model is easy to implement with acceptable accuracy. If all CLV variables are available and accurate, one could experiment with the popular CLV_f model. In our research, the models (CLV_r, CLV_{mcmi}) without the acquisition variables (a, A) have shown to outperform others. We were able to identify the problem of undersampling during the acquisition imputation, and this point is discussed as part of future work in chapter 8.

Chapter 8

Conclusions

This final chapter has 2 parts: in section §8.1, a summary of the work completed for this dissertation is presented and in section §8.2, we discuss some areas for future research projects that were outside the scope of this work and could provide new areas of research.

8.1 Dissertation Overview

In the opening chapter of this dissertation, we introduced the concept of customer lifetime value, its benefits to market analysts and the issues that prevent its computation for each customer, with the result that it remains an open area of research. Our hypothesis included 3 research questions, regarding the possibility of generating a holistic customer record; the development of a strategy to auto-classify customers according to their value to the company; and finally, if the dataset and suitable predictive models exist to impute the missing CLV variables. Our literature review covered research in record linkage, customer segmentation, customer churn, acquisition analysis, and CLV modelling. The main issue for CLV research was the gap between the theoretical and the real world requirements and we believed that by collaborating with an industry partner, we could seek to bridge that gap.

Our first contribution, record linkage comprised 5 processes: pre-processing; segmenting the recordset; application of the matching algorithm; using a ruleset to improve matching results; and validation. Without the unified customer record, it we could not have had the datasets necessary to begin customer segmentation, a first step to computing CLV metrics for customers. Having successfully completed this first research challenge and validated a positive set of results, the next step was to attempt to classify an untrained customer dataset.

We then presented our segmentation process to categorise customers into coarse grained clusters which was shown to have high levels of accuracy. This step was crucial in providing a validation dataset for the later calculation of customer lifetime values: those with high CLV scores should all be located inside the *good customer cluster*. In addition, the process of variable (feature) extraction and selection delivered the dataset that formed the basis for imputing missing variables that are necessary for CLV calculations. However, before we could proceed to imputing the missing variables necessary for the CLV calculation formula, a final transformation of the dataset was required together with a robust methodology for determining the best predictive model for the missing acquisition and retention variables.

The previous chapter showed that using our predictive models, the subsequent CLV calculations can identify good customers with an accuracy of 52.5%, and for identifying bad customers, the rate increased to 72%. In addition, accuracy gradually improved through the processes developed in this research. If organisations have accurate, year by year, individual-level variables, the MCM model is suitable for this dataset. Our research showed that both $CLVr$ and CLV_{mcmi} models, without using acquisition variables (a, A) performed best.

8.2 Ideas for Future Research

In the final section section, we will explore possible future work for our research.

8.2.1 Data Availability and Wider Applicability

Our research used a real-world dataset to solve real-world problems. However, the availability of data and publication of results can be a major problem in this research. Most work on CLV utilises data provided by an industry partner. This data is private to the industry and hampers the ability of other researchers to compare and review different CLV systems. The decision to make such a dataset publicly available rests with the industry partner. However, these datasets may be examined to construct a synthetic dataset which can be made widely available to the research community allowing greater collaboration and cooperation within the CLV domain.

The capability of any analysis project is constrained by the available data. For our work we were provided with a limited feature set, most features are common across many domains. The addition of more features, for example, features related to social-demographic; Psychographic (interest, lifestyle, personality, etc.) [65] could provide new insights into predicting CLV, customer segmentation and ultimately improve our prediction results.

The processes involved in our research could be applicable to other domains, for example, healthcare, telecoms, etc. The methodology can be customised during the requirement. For example, customise the threshold on record linkage in order to balance the number of matches and the accuracy. Our work used an insurance dataset provided by our industry partner to predict the CLV for each customer. However, our model could be considered domain-specific, using features unique to insurance data (e.g. premium renewal price). A different data source from another domain would provide a new set of features and behaviours which could provide the ability to examine the modifications to be made to our methodology, to ensure genericity and to test the accuracy of our models across differing domains.

8.2.2 Improving Record Linkage

While we have presented a methodology for record linkage in this work, there remains a number of research areas which could further improve record linkage results.

- A key component in record linkage is the string-matching algorithm used. Due to the size of the dataset and computing resources available, a trade-off must be made between accuracy of matches and computation time. For our work, we utilised a blocking approach but other approaches such as a sliding window or an NLP system could be utilised instead. These methods may increase computation time. In addition, they may also improve the string matching accuracy and ultimately improve our record linkage results.
- At present, our work classifies three types of customer relationship: client-client, client-family and client-domiciled. However, the concept of a familial unit and relationship is not limited to these three groupings (e.g. civil-partnership, customers who are married but kept their surnames etc.). Determining other relationship types requires more in-depth analysis and a more robust series of rules, ultimately increasing the time taken to classify customer relationships.
- As with any real-word dataset, there are data quality issues and specifically, the textual representation of customer addresses. The address data is user-generated but a process utilising a suitable ontology and address-matching matrix could be used to mitigate these factors and improve record linkage results.

8.2.3 Research on Customer Acquisition

With our research, the same methodology was used to predict acquisition and retention where 6% of records had the class `acquired`. We made the decision to undersample to manage this issue. Using this method, a significant amount of the dataset cannot be used for training which may have an impact on our overall classification results. A different approach could use other methods for dealing with class imbalances to see if these methods can improve the accuracy for our classifiers. Possible methods are:

- Using random oversampling to balance the dataset for acquisition. This method involved "doubling-up" instance of the minor class. The benefit of this approach is that the model continues to be trained on real world data. However, the repeating of previously seen records within the training data may lead to model overfitting.
- Generate synthetic records for acquisition to address the class imbalance. This requires a significant investment in analysing the properties of records marked as acquisition and developing a system which can generate synthetic records.
- Examine other classification methods. In our work, we examined 4 different classification methods using a number of different configurations. However this list is not exhaustive and there remains other classification algorithms (e.g. KNN) and logistic methods which may improve the accuracy for acquisition.

8.2.4 Impact of Relationships on CLV

Inter-customer relationships can provide useful indicators to a company. They can be attributed to reduced churn, higher acquisition and ultimately increased sales. However, little work has been done to examine the impact relationships have on the overall CLV figure for an individual customer. With our relationships defined and classified on a per-customer level, a potential area of research would be to examine the effect such relationships have on the overall CLV metric for an individual customer. This work could provide key indicators to industry partners on where to focus their acquisition efforts.

Bibliography

[1] Lada A Adamic and Eytan Adar. Friends and neighbors on the web. *Social networks*, 25(3):211–230, 2003.

[2] Ali Ahani, Mehrbakhsh Nilashi, Othman Ibrahim, Louis Sanzogni, and Scott Weaven. Market segmentation and travel choice prediction in spa hotels through tripadvisor's online reviews. *International Journal of Hospitality Management*, 80:52–77, 2019.

[3] Amal M Almana, Mehmet Sabih Aksoy, and Rasheed Alzahrani. A survey on data mining techniques in customer churn analysis for telecom industry. *International Journal of Engineering Research and Applications*, 45:165–171, 2014.

[4] AG Asuero, A Sayago, and AG González. The correlation coefficient: An overview. *Critical Reviews in Analytical Chemistry*, 36:41–59, 2006.

[5] Stuart J Barnes, Hans H Bauer, Marcus M Neumann, and Frank Huber. Segmenting cyberspace: a customer typology for the internet. *European journal of marketing*, 41(1/2):71–93, 2007.

[6] Rohan Baxter, Peter Christen, Tim Churches, et al. A comparison of fast blocking methods for record linkage. In *ACM SIGKDD*, volume 3, pages 25–27. Citeseer, 2003.

[7] Paul D Berger and Nada I Nasr. Customer lifetime value: Marketing models and applications. *Journal of interactive marketing*, 12(1):17–30, 1998.

[8] Indrajit Bhattacharya and Lise Getoor. Collective entity resolution in relational data. *ACM Transactions on Knowledge Discovery from Data (TKDD)*, 1(1):5, 2007.

[9] Purnima Bholowalia and Arvind Kumar. Ebk-means: A clustering technique based on elbow method and k-means in wsn. *International Journal of Computer Applications*, 105(9), 2014.

[10] Mikhail Bilenko, Beena Kamath, and Raymond J Mooney. Adaptive blocking: Learning to scale up record linkage. In *Data Mining, 2006. ICDM'06. Sixth International Conference on*, pages 87–96. IEEE, 2006.

[11] Robert C Blattberg and John Deighton. Manage marketing by the customer equity test. *Harvard business review*, 74(4):136, 1996.

[12] Catalina Bolancé, Montserrat Guillen, and Alemar E Padilla-Barreto. Predicting probability of customer churn in insurance. In *International Conference on Modeling and Simulation in Engineering, Economics and Management*, pages 82–91. Springer, 2016.

[13] Luis Callarisa, Javier Sánchez García, John Cardiff, and Alexandra Roshchina. Harnessing social media platforms to measure customer-based hotel brand equity. *Tourism Management Perspectives*, 4:73–79, 2012.

[14] Li-Juan Cao and Francis Eng Hock Tay. Support vector machine with adaptive parameters in financial time series forecasting. *IEEE Transactions on neural networks*, 14(6):1506–1518, 2003.

[15] Mònica Casabayó Bonàs. *Shopping behaviour forecasts: Experiments based on a fuzzy learning technique in the Spanish food retailing industry.* PhD thesis, University of Edinburgh, 2005.

[16] Lin Chen, Richi Nayak, and Yue Xu. A common neighbour based two-way collaborative recommendation method. In *Proceedings of the 27th Annual ACM Symposium on Applied Computing*, pages 214–215, 2012.

[17] Nancy Chinchor. MUC-4 evaluation metrics. In *Fourth Message Uunderstanding Conference (MUC-4): Proceedings of a Conference Held in McLean, Virginia, June 16-18, 1992*, 1992.

[18] A Joy Christy, A Umamakeswari, L Priyatharsini, and A Neyaa. Rfm ranking–an effective approach to customer segmentation. *Journal of King Saud University-Computer and Information Sciences*, 2018.

[19] Jana Cibulková and Zdenek Sulc. A case study of customer segmentation with the use of hierarchical cluster analysis of categorical data. 2018.

[20] Corinna Cortes and Vladimir Vapnik. Support-vector networks. *Machine learning*, 20(3):273–297, 1995.

[21] Manoranjan Dash and Huan Liu. Feature selection for classification. *Intelligent data analysis*, 1(3):131–156, 1997.

[22] William HE Day and Herbert Edelsbrunner. Efficient algorithms for agglomerative hierarchical clustering methods. *Journal of classification*, 1(1):7–24, 1984.

[23] Marnik G Dekimpe and Dominique M Hanssens. Persistence modeling for assessing marketing strategy performance. 2004.

[24] C Anthony Di Benedetto and Kyung Hoon Kim. Customer equity and value management of global brands: Bridging theory and practice from financial and marketing perspectives: Introduction to a journal of business research special section. *Journal of Business Research*, 69(9):3721–3724, 2016.

[25] Chris Ding and Xiaofeng He. K-means clustering via principal component analysis. In *Proceedings of the twenty-first international conference on Machine learning*, page 29. ACM, 2004.

[26] Bas Donkers, Peter C Verhoef, and Martijn G de Jong. Modeling clv: A test of competing models in the insurance industry. *Quantitative Marketing and Economics*, 5(2):163–190, 2007.

[27] F. Robert Dwyer. Customer lifetime valuation to support marketing decision making. *Journal of Direct Marketing*, 3(4):8 – 15, 1989.

[28] Eleazar Eskin, Andrew Arnold, Michael Prerau, Leonid Portnoy, and Sal Stolfo. A geometric framework for unsupervised anomaly detection. In *Applications of data mining in computer security*, pages 77–101. Springer, 2002.

[29] Brian Etienne, Michelle Cheatham, and Pawel Grzebala. An analysis of blocking methods for private record linkage. In *2016 AAAI Fall Symposium Series*, 2016.

[30] Peter Fader, Bruce Hardie, and Paul D Berger. Customer-base analysis with discrete-time transaction data. 2004.

[31] Peter S Fader, Bruce GS Hardie, and Ka Lok Lee. Rfm and clv: Using iso-value curves for customer base analysis. *Journal of marketing research*, 42(4):415–430, 2005.

[32] Paul W Farris, Neil Bendle, Phillip Pfeifer, and David Reibstein. *Marketing metrics: The definitive guide to measuring marketing performance.* Pearson Education, 2010.

[33] John Ferguson, Ailish Hannigan, and Austin Stack. A new computationally efficient algorithm for record linkage with field dependency and missing data imputation. *International journal of medical informatics*, 109:70–75, 2018.

[34] Jerome H Friedman. Recent advances in predictive (machine) learning. *Journal of classification*, 23(2):175–197, 2006.

[35] Daniel Granato, Jânio S Santos, Graziela B Escher, Bruno L Ferreira, and Rubén M Maggio. Use of principal component analysis (pca) and hierarchical cluster analysis (hca) for multivariate association between bioactive compounds and functional properties in foods: A critical perspective. *Trends in Food Science & Technology*, 72:83–90, 2018.

[36] Clara-Cecilie Günther, Ingunn Fride Tvete, Kjersti Aas, Geir Inge Sandnes, and Ørnulf Borgan. Modelling and predicting customer churn from an insurance company. *Scandinavian Actuarial Journal*, 2014(1):58–71, 2014.

[37] Sunil Gupta, Dominique Hanssens, Bruce Hardie, Wiliam Kahn, V. Kumar, Nathaniel Lin, Nalini Ravishanker, and S. Sriram. Modeling customer lifetime value. *Journal of Service Research*, 9(2):139–155, 2006.

[38] Sunil Gupta, Dominique Hanssens, Bruce Hardie, Wiliam Kahn, V Kumar, Nathaniel Lin, Nalini Ravishanker, and S Sriram. Modeling customer lifetime value. *Journal of service research*, 9(2):139–155, 2006.

[39] Sunil Gupta, Donald R Lehmann, and Jennifer Ames Stuart. Valuing customers. *Journal of marketing research*, 41(1):7–18, 2004.

[40] Isabelle Guyon and André Elisseeff. An introduction to variable and feature selection. *Journal of machine learning research*, 3(Mar):1157–1182, 2003.

[41] Isabelle Guyon, Steve Gunn, Masoud Nikravesh, and Lofti A Zadeh. *Feature extraction: foundations and applications*, volume 207. Springer, 2008.

[42] Jiawei Han, Jian Pei, and Micheline Kamber. *Data mining: concepts and techniques*. Elsevier, 2011.

[43] Trevor J Hastie and Robert J Tibshirani. *Generalized additive models*, volume 43. CRC press, 1990.

[44] Mauricio A Hernández and Salvatore J Stolfo. The merge/purge problem for large databases. *ACM Sigmod Record*, 24(2):127–138, 1995.

[45] Thomas N Herzog, Fritz J Scheuren, and William E Winkler. *Data quality and record linkage techniques*. Springer Science & Business Media, 2007.

[46] Abdulkadir Hiziroglu and Serkan Sengul. Investigating two customer lifetime value models from segmentation perspective. *Procedia-Social and Behavioral Sciences*, 62:766–774, 2012.

[47] John E Hogan, Katherine N Lemon, and Barak Libai. What is the true value of a lost customer? *Journal of Service Research*, 5(3):196–208, 2003.

[48] Andreas Hotho, Steffen Staab, and Gerd Stumme. Ontologies improve text document clustering. In *Data Mining, 2003. ICDM 2003. Third IEEE International Conference on*, pages 541–544. IEEE, 2003.

[49] Zhexue Huang. Extensions to the k-means algorithm for clustering large data sets with categorical values. *Data mining and knowledge discovery*, 2(3):283–304, 1998.

[50] Hyunseok Hwang, Taesoo Jung, and Euiho Suh. An ltv model and customer segmentation based on customer value: a case study on the wireless telecommunication industry. *Expert systems with applications*, 26(2):181–188, 2004.

[51] Dipak Jain and Siddhartha S Singh. Customer lifetime value research in marketing: A review and future directions. *Journal of interactive marketing*, 16(2):34–46, 2002.

[52] Nathalie Japkowicz and Shaju Stephen. The class imbalance problem: A systematic study. *Intelligent data analysis*, 6(5):429–449, 2002.

[53] Liang Jin, Chen Li, and Sharad Mehrotra. Efficient record linkage in large data sets. In *Database Systems for Advanced Applications, 2003.(DASFAA 2003). Proceedings. Eighth International Conference on*, pages 137–146. IEEE, 2003.

[54] Bonnie JK Simpson and Scott K. Radford. Situational variables and sustainability in multi-attribute decision-making. *European Journal of Marketing*, 48(5/6):1046–1069, 2014.

[55] Ali Kara and Erdener Kaynak. Markets of a single customer: exploiting conceptual developments in market segmentation. *European journal of marketing*, 31(11/12):873–895, 1997.

[56] Mahboubeh Khajvand, Kiyana Zolfaghar, Sarah Ashoori, and Somayeh Alizadeh. Estimating customer lifetime value based on rfm analysis of customer purchase behavior: Case study. *Procedia Computer Science*, 3:57 – 63, 2011. World Conference on Information Technology.

[57] Hung-sik Kim and Dongwon Lee. Parallel linkage. In *Proceedings of the sixteenth ACM conference on Conference on information and knowledge management*, pages 283–292, 2007.

[58] Su-Yeon Kim, Tae-Soo Jung, Eui-Ho Suh, and Hyun-Seok Hwang. Customer segmentation and strategy development based on customer lifetime value: A case study. *Expert systems with applications*, 31(1):101–107, 2006.

[59] Jacquelynne R King and Donald A Jackson. Variable selection in large environmental data sets using principal components analysis. *Environmetrics*, 10(1):67–77, 1999.

[60] Joseph B Kruskal. An overview of sequence comparison: Time warps, string edits, and macromolecules. *SIAM review*, 25(2):201–237, 1983.

[61] V Kumar and Anita Pansari. National culture, economy, and customer lifetime value: Assessing the relative impact of the drivers of customer lifetime value for a global retailer. *Journal of International Marketing*, 24(1):1–21, 2016.

[62] Bjornar Larsen and Chinatsu Aone. Fast and effective text mining using linear-time document clustering. In *Proceedings of the fifth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 16–22. ACM, 1999.

[63] I Lawrence and Kuei Lin. A concordance correlation coefficient to evaluate reproducibility. *Biometrics*, pages 255–268, 1989.

[64] Paul H Lee and LH Philip. Distance-based tree models for ranking data. *Computational Statistics & Data Analysis*, 54(6):1672–1682, 2010.

[65] Chin-Feng Lin. Segmenting customer brand preference: demographic or psychographic. *Journal of Product & Brand Management*, 2002.

[66] Dekang Lin et al. An information-theoretic definition of similarity. In *Icml*, volume 98, pages 296–304, 1998.

[67] Raymond Ling and David C Yen. Customer relationship management: An analysis framework and implementation strategies. *Journal of computer information systems*, 41(3):82–97, 2001.

[68] Suzanne Little, Sara Colantonio, Ovidio Salvetti, Petra Perner, et al. Evaluation of feature subset selection, feature weighting, and prototype selection for biomedical applications. *Journal of Software Engineering and Applications*, 3(01):39, 2010.

[69] Wei-Yin Loh. Classification and regression trees. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(1):14–23, 2011.

[70] Zhang Lu, Wang Peiyi, Chen Ping, Li Xianglong, Zhang Baoqun, and Ma Longfei. Customer segmentation algorithm based on data mining for electric vehicles. In *2019 IEEE 4th International Conference on Cloud Computing and Big Data Analysis (ICCCBDA)*, pages 77–83. IEEE, 2019.

[71] Abdullah-Al Mamun, Robert Aseltine, and Sanguthevar Rajasekaran. Efficient record linkage algorithms using complete linkage clustering. *PloS one*, 11(4):e0154446, 2016.

[72] JE Martinez-Legaz. Lexicographical order, inequality systems and optimization. In *System Modelling and Optimization*, pages 203–212. Springer, 1984.

[73] Andrew McCallum, Kamal Nigam, and Lyle H Ungar. Efficient clustering of high-dimensional data sets with application to reference matching. In *Proceedings of the sixth ACM SIGKDD international conference on Knowledge discovery and data mining*, pages 169–178. ACM, 2000.

[74] Vangelis Metsis, Ion Androutsopoulos, and Georgios Paliouras. Spam filtering with naive bayes-which naive bayes? In *CEAS*, volume 17, pages 28–69. Mountain View, CA, 2006.

[75] Glenn W Milligan and Martha C Cooper. A study of standardization of variables in cluster analysis. *Journal of classification*, 5(2):181–204, 1988.

[76] Katharina Morik and Hanna Köpcke. Analysing customer churn in insurance data–a case study. In *European conference on principles of data mining and knowledge discovery*, pages 325–336. Springer, 2004.

[77] Daniel Müllensiefen, Christian Hennig, and Hedie Howells. Using clustering of rankings to explain brand preferences with personality and socio-demographic variables. *Journal of Applied Statistics*, 45(6):1009–1029, 2018.

[78] Scott A Neslin, Sunil Gupta, Wagner Kamakura, Junxiang Lu, and Charlotte H Mason. Defection detection: Measuring and understanding the predictive accuracy of customer churn models. *Journal of marketing research*, 43(2):204–211, 2006.

[79] E.W.T. Ngai, Li Xiu, and D.C.K. Chau. Application of data mining techniques in customer relationship management: A literature review and classification. *Expert Systems with Applications*, 36(2, Part 2):2592 – 2602, 2009.

[80] Dongyun Nie, Paolo Cappellari, and Mark Roantree. A methodology for classification and validation of customer datasets. *To appear in Journal of Business and Industrial Marketing*, 2020.

[81] Dongyun Nie and Mark Roantree. Detecting multi-relationship links in sparse datasets. In *21st International Conference on Enterprise Information Systems*, volume 1, pages 137–145. SCITEPRESS, 2019.

[82] Suphakit Niwattanakul, Jatsada Singthongchai, Ekkachai Naenudorn, and Supachanun Wanapu. Using of jaccard coefficient for keywords similarity. In *Proceedings of the international multiconference of engineers and computer scientists*, volume 1, pages 380–384, 2013.

[83] Jon Orwant, Jarkko Hietaniemi, and John Macdonald. *Mastering algorithms with Perl.* " O'Reilly Media, Inc.", 1999.

[84] Uladzimir Parkhimenka, Mikhail Tatur, and Olga Khandogina. Unsupervised ranking of clients: machine learning approach to define a" good customer". 2017.

[85] Joan Peckham and Fred Maryanski. Semantic data models. *ACM Computing Surveys (CSUR)*, 20(3):153–189, 1988.

[86] Phillip E Pfeifer. The optimal ratio of acquisition and retention costs. *Journal of Targeting, Measurement and Analysis for Marketing*, 13(2):179–188, 2005.

[87] Phillip E. Pfeifer and Robert L. Carraway. Modeling customer relationships as Markov chains. *Journal of Interactive Marketing*, 14(2):43–55, 2000.

[88] Phillip E Pfeifer and Robert L Carraway. Modeling customer relationships as markov chains. *Journal of interactive marketing*, 14(2):43–55, 2000.

[89] Erhard Rahm. The case for holistic data integration. In *East European Conference on Advances in Databases and Information Systems*, pages 11–27. Springer, 2016.

[90] Juan Ramos et al. Using tf-idf to determine word relevance in document queries. In *Proceedings of the first instructional conference on machine learning*, volume 242, pages 133–142. Piscataway, NJ, 2003.

[91] Werner Reinartz, Jacquelyn S Thomas, and Viswanathan Kumar. Balancing acquisition and retention resources to maximize customer profitability. *Journal of marketing*, 69(1):63–79, 2005.

[92] Werner J Reinartz and Vijay Kumar. On the profitability of long-life customers in a noncontractual setting: An empirical investigation and implications for marketing. *Journal of marketing*, 64(4):17–35, 2000.

[93] Seyed Mahdi Rezaeinia and Rouhollah Rahmani. Recommender system based on customer segmentation (rscs). *Kybernetes*, 2016.

[94] Hana Rezankova, Tomas Loster, and Dusan Husek. Evaluation of categorical data clustering. In *Advances in Intelligent Web Mastering–3*, pages 173–182. Springer, 2011.

[95] Hans Risselada, Peter C Verhoef, and Tammo HA Bijmolt. Staying power of churn prediction models. *Journal of Interactive Marketing*, 24(3):198–208, 2010.

[96] Eric Sven Ristad and Peter N Yianilos. Learning string-edit distance. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 20(5):522–532, 1998.

[97] Peter J Rousseeuw. Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65, 1987.

[98] Roland T Rust, Tim Ambler, Gregory S Carpenter, Vl Kumar, and Rajendra K Srivastava. Measuring marketing productivity: Current knowledge and future directions. *Journal of marketing*, 68(4):76–89, 2004.

[99] Germán Sánchez-Hernández, Francisco Chiclana, Núria Agell, and Juan Carlos Aguado. Ranking and selection of unsupervised learning marketing segmentation. *Knowledge-based systems*, 44:20–33, 2013.

[100] David C Schmittlein, Donald G Morrison, and Richard Colombo. Counting your customers: Who-are they and what will they do next? *Management science*, 33(1):1–24, 1987.

[101] Michael Scriney, Dongyun Nie, and Mark Roantree. Predicting customer churn for insurance data. In *Big Data Analytics and Knowledge Discovery - 22nd International Conference, DaWaK2020, Proceedings*, volume 12393. Springer, 2020.

[102] Julian Sedding and Dimitar Kazakov. Wordnet-based text document clustering. In *proceedings of the 3rd workshop on robust methods in analysis of natural language data*, pages 104–113. Association for Computational Linguistics, 2004.

[103] Kate A Smith, Robert J Willis, and Malcolm Brooks. An analysis of customer retention and insurance claim patterns using data mining: A case study. *Journal of the operational research society*, 51(5):532–541, 2000.

[104] Babak Sohrabi and Amir Khanlari. Customer lifetime value (clv) measurement based on rfm model. *Iranian Accounting & Auditing Review*, 14(47):7–20, 2007.

[105] Merlin Stone and R Shaw. Database marketing. *Aldershot, Gower*, 1988.

[106] Joanna Strycharz, Guda van Noort, Natali Helberger, and Edith Smit. Contrasting perspectives–practitioner's viewpoint on personalised marketing communication. *European Journal of Marketing*, 53(4):635–660, 2019.

[107] Zdeněk Šulc and Hana Řezanková. Comparison of similarity measures for categorical data in hierarchical clustering. *Journal of Classification*, 36(1):58–72, 2019.

[108] G Ganesh Sundarkumar, Vadlamani Ravi, and V Siddeshwar. One-class support vector machine based undersampling: Application to churn prediction and insurance fraud detection. In *2015 IEEE International Conference on Computational Intelligence and Computing Research (ICCIC)*, pages 1–7. IEEE, 2015.

[109] Daniel R Swanson Sr. Systems, methods and apparatus for self directed individual customer segmentation and customer rewards, March 31 2011. US Patent App. 12/567,029.

[110] Thorsten Teichert, Edlira Shehu, and Iwan von Wartburg. Customer segmentation revisited: The case of the airline industry. *Transportation Research Part A: Policy and Practice*, 42(1):227–242, 2008.

[111] Panos Vassiliadis. A survey of extract–transform–load technology. *International Journal of Data Warehousing and Mining (IJDWM)*, 5(3):1–27, 2009.

[112] Peter C Verhoef, Philip Hans Franses, and Janny C Hoekstra. The impact of satisfaction and payment equity on cross-buying: A dynamic model for a multi-service provider. *Journal of Retailing*, 77(3):359–378, 2001.

[113] Michael E Wall, Andreas Rechtsteiner, and Luis M Rocha. Singular value decomposition and principal component analysis. In *A practical approach to microarray data analysis*, pages 91–109. Springer, 2003.

[114] Gianfranco Walsh, Louise M Hassan, Edward Shiu, J Craig Andrews, and Gerard Hastings. Segmentation in social marketing: Insights from the european union's multi-country, antismoking campaign. *European Journal of Marketing*, 44(7/8):1140–1164, 2010.

[115] Michel Wedel and Wagner A Kamakura. *Market Segmentation: Conceptual and Methodological Foundations*. Springer Science & Business Media, 2000.

[116] Frank Westad, Margrethe Hersletha, Per Lea, and Harald Martens. Variable selection in pca in sensory descriptive and consumer data. *Food Quality and Preference*, 14(5-6):463–472, 2003.

[117] William E Winkler. Matching and record linkage. *Business survey methods*, 1:355–384, 1995.

[118] Dick R Wittink. Econometric models for marketing decisions. *Journal of Marketing Research*, 42(1):1–3, 2005.

[119] Roung-Shiunn Wu and Po-Hsuan Chou. Customer segmentation of multiple category data in e-commerce using a soft-clustering approach. *Electronic Commerce Research and Applications*, 10(3):331–341, 2011.

[120] Mirella Yani-de Soriano, Paul HP Hanel, Rosario Vazquez-Carrasco, Jesus Cambra-Fierro, Alan Wilson, and Edgar Centeno. Investigating the role of customers' perceptions of employee effort and justice in service recovery: A cross-cultural perspective. *European Journal of Marketing*, 53(4):708–732, 2019.

[121] Shijin Yoo and Dominique M Hanssens. Modeling the sales and customer equity effects of the marketing mix. *UCLA Anderson School of Management*, 2:1–42, 2005.

[122] Yong-Huan Yun, Hong-Dong Li, Bai-Chuan Deng, and Dong-Sheng Cao. An overview of variable selection methods in multivariate analysis of near-infrared spectra. *TrAC Trends in Analytical Chemistry*, 2019.

[123] Lotfi A Zadeh. A computational approach to fuzzy quantifiers in natural languages. In *Computational linguistics*, pages 149–184. Elsevier, 1983.

[124] Rong Zhang, Weiping Li, Wei Tan, and Tong Mo. Deep and shallow model for insurance churn prediction service. In *2017 IEEE International Conference on Services Computing (SCC)*, pages 346–353. IEEE, 2017.

[125] Judy Zolkiewski, Peter Turnbull, Sabrina Helm, Ludger Rolfes, and Bernd Günter. Suppliers' willingness to end unprofitable customer relationships. *European Journal of Marketing*, 2006.

Appendices

Appendix A

Record Linkage Datasets Details

Table A.1: Record Linkage Attributes Detail

Attribute	**Data Type**	**DS1**	**DS2**	**DS3**	**DS4**	**DS5**	**Total**	**Percentage**
Address Id	Number	*			*		0	0%
Address 1	Varchar	*	*	*			339	0%
Address 2	Varchar	*	*	*		*	0	0%
Address 3	Varchar	*	*	*		*	78743	40%
Address 4	Varchar	*	*	*		*	111418	57%
Address Type Id	Number	*			*		0	0%
Date Of Birth	Varchar	*	*	*			18	0%
Building Gname	Varchar	*				*	195634	100%
Building Name	Varchar	*				*	186057	95%
Client Id	Number	*			*		0	0%
Client Reference Nbr	Number	*	*		*		0	0%
Client Type Id	Number	*			*		0	0%
Commercial Policy No	Varchar	*				*	49311	25%
Confirm Gcflag	Varchar	*				*	97014	50%
Contact Branch Id	Number	*			*		1	0%
Contact Detail Id	Number	*			*		0	0%
Coordinate X	Number	*					94388	48%
Coordinate Y	Number	*					94388	48%
Country Id	Number	*	*		*		173533	89%

Attribute	Data Type	DS1	DS2	DS3	DS4	DS5	Total	Percentage
County Id	Number	*	*		*		0	0%
Create By	Varchar	*				*	0	0%
Create Dtime	Time Stamp	*	*				0	0%
Do Not Call Flag	Varchar	*					53315	27%
Dpa Consent Flag	Varchar	*	*				0	0%
Duplicate Flag	Varchar	*					36457	19%
Email Id	Varchar	*	*	*			40391	21%
Fax Nbr	Varchar	*	*	*			195096	100%
First Name	Varchar	*	*	*			18	0%
Gender Id	Number	*	*		*		18	0%
Gcflag	Varchar	*	*				0	0%
Geocode Id	Number	*					94616	48%
Home Docpref Dtstp	Time Stamp	*					144311	74%
Home Online Doc	Varchar	*				*	153509	78%
Home Phone Nbr	Varchar	*	*	*			62617	32%
Individual Id	Number	*			*		18	0%
Last Name	Varchar	*	*	*		*	18	0%
Link Present Flag	Varchar	*					36469	19%
Master Client Id	Number	*					0	0%
Match Level	Varchar	*					94637	48%
Mobile Phone Nbr	Varchar	*	*	*			66393	34%
Motor Docpref Dtstp	Varchar	*					42519	22%
Motor Online Doc	Varchar	*				*	48551	25%
Occupation Id	Number	*	*				12	0%
Organisation	Varchar	*					195801	100%
Owner Subchannel Id	Number	*					145407	74%
Party Id	Number	*			*		0	0%
Party Type Id	Number	*			*		0	0%
Post Code 1	Varchar	*					195849	100%
Post Code 2	Varchar	*					195849	100%
Post Town	Varchar	*				*	183683	94%
Ppp Eligibility Id	Number	*			*		0	0%

Attribute	Data Type	DS1	DS2	DS3	DS4	DS5	Total	Percentage
Preferred Contact Id	Number	*			*		0	0%
Primary Locality	Varchar	*				*	140165	72%
Primary Throughfare	Varchar	*				*	141283	72%
Product Consent Flag	Varchar	*	*				0	0%
Proposer Change Dtime	Time Stamp	*					89430	46%
Rating Code	Varchar	*					96260	49%
Secondary Locality	Varchar	*				*	162700	83%
Secondary Throughfare	Varchar	*				*	188133	96%
Status Id	Number	*			*		0	0%
Sub Building Name	Varchar	*				*	195743	100%
Title Id	Number	*	*		*		18	0%
Townland	Varchar	*				*	189255	97%
Update By	Varchar	*					10782	6%
Update Dtime	Time Stamp	*	*				10782	6%
Vanity Au Flag	Varchar	*					48426	25%
Work Extension Nbr	Varchar	*					195833	100%
Work Phone Nbr	Varchar	*	*	*			191076	98%

Note: The "*" in all DS column means the listed attribute has applied for assigned Data Set.

The Total listed the total number of missing values in this attribute.

The Percentage shows the percentage of missing values.

Table A.2: Replace Null Value

Type	**NbrOfNulls**	**avg_exact**	**avg_int**	**max_exact**	**max_exact+1**
Title Id	1	1.3642	1	1531	1532
First Name	1	5.8142	6	23	24
Last Name	1	6.5449	7	24	25
Mobile Phone Nbr	11651	9.2451	9	15	16
Home Phone Nbr	8884	8.3486	8	15	16
Work Phone Nbr	29624	0.4483	1	15	16
Email Id	4367	17.2349	17	45	46
Fax Nbr	30186	0.0786	1	15	16
Date Of Birth	1	4.5708	5	8	9
Gcflag	0	0.4841	1	1	2
Product Consent Flag	0	0.4612	1	1	2
Dpa Consent Flag	0	0	0	0	1
Gender Id	1	0.6044	1	1602	1603
Occupation Id	1	275.8785	276	42120	42121
Address1	10	14.9197	15	53	54
Address2	0	11.9046	12	50	51
Address3	7500	8.6898	9	48	49
Address4	13027	3.7001	4	20	21
County Id	0	0	0	0	1
Country Id	28115	1715.1179	1715	12912	12913
Update Dtime	1518	3.7723	4	8	9
Create Dtime	0	3.9098	4	6	7

Appendix B

Policy Datasets Details

Table B.1: Lk Transaction Table

Name	**Descriptions**
Lk Transaction Id	Unique identifier for the record.
Transaction Type	This field denotes the type of transaction.
Transaction Description	This field explains the Transaction Type.
BPS Transaction Type	This field identifies the equivalent transaction type on BPS system.
Effective From Date	The date and time from which the transaction is effective.
Effective To Date	The date and time up to which the transaction is effective.
Update DTime	The instance in time when the record was last updated.
Update By	The id of the user that updated the record.

Table B.2: Quote Policy Link Table

Name	**Descriptions**
Quote Policy Link Id	Reference to the quote/policy record.
Product Master Id	A unique sequence number specifying the type of product.
Client Id	Unique identifier for client.
Sales Node Id	The sales node in the organisation hierarchy that sold this policy.
Quote Reference Nbr	A digit unique reference number for a quote which will be retained on all versions.
Quote Version Nbr	Identifies the version of the quote.
WEB Quote Reference Nbr	QRN of a quote that was generated by HDWeb.
Policy Nbr	The Policy Number is set on the record when a quote is fulfilled and a policy created.
Policy Version Nbr	The version number of the policy.
Pol Fulfilment DTime	This is the date and time on which the Complete quote was fulfilled and the policy created.
Status Id	Values for Quote and policy status.
HDWeb Status Id	The status of a Quote/Policy on HDWeb system.
Quote Expiry DTime	Expiry date will be set to 30 days from the day quote was marked as Complete.
Branch Team Id	When a quote is first created, Create_By and Branch Team Identifier (BTI) will mark the owner and his/her branch.
Quote Owner	When a quote is first created, Create_By and Branch Team Identifier (BTI) will mark the owner and his/her branch. When quote is rated, Quote_Owner and BTI will be set to the user who rated the quote and his/her branch.
PPP Confirmed Flag	This flag will be used for the PPP control report.
Quote Originator Id	This field identifies the originator of the quote.
Create DTime	The instance in time when the record was physically created.
Update DTime	The instance in time when the record was last updated.
Update By	The id of the user that last updated the record.
Create By	The id of the user that created the record.

Table B.3: QB Result Table

Name	**Descriptions**
QB Result Identifier	A unique Identifier
Quote Policy Link Id	Reference to the quote/policy record.
QB Result Type Id	Identifies the type of Qb Result.
Code	Code value of the selected Code for this quote breakdown item
Reason	Code value of the selected Reason for this quote breakdown item
Description	A textual description of the quote breakdown item
Percentage	The percentage by which this step changed the premium
Amount	The value of the change in premium due to this step in the calculation
Running Total	The accumulation of the amounts calculated in this and the previous components
Rounding Code	Code value of the selected Rounding Code for this quote breakdown item
Vehicle Ref Num	The permanent reference number of the vehicle, included in the policy, to which the quote applies
Review Date	The date when the parameters (e.g. Pct) for this component were last reviewed/updated
Units	A Textual description of the units which apply to the amount of this component (e.g. £, or Points).
Cover Code	A code value that identifies which cover section the endorsement applies and also as a qualifier for the item PRN property if set
Display Code	Optional Benefits and Policy Extras code for Display Purpose.

Table B.4: Quote Policy Header Table

Name	Descriptions
Quote Policy Link Id	Reference to the quote/policy record.
Policy Owner Org Id	This is the organisation id of he branch that sold the 1st version of the policy.
Policy Year Nbr	During quotation, Policy Year set to 0. When converted to a policy, the year is set to 1 and subsequently on each renewal the year is incremented.
Incept DTime	The Date the policy comes or came into effect.
Policy Effective From Date	Policy start date and time for either new or Renewal.
Policy Effective To Date	Policy end date and time for either new or renew.
Pol Cancellation DTime	Cancellation Date of Policy
Policy Lapse DTime	Lapse Date of Policy
Policy Reinstate DTime	The instance in time when the policy is reinstated.
Pol Suspension DTime	The instance in time when the policy is suspended.
Last Renewal DTime	The Date the policy last renewed - may be null on quote or in first year of policy.
Next Renewal DTime	The date the policy will next be renewed.
Policy Reason Code Id	A code specifying the reason code for the Transaction.
Policy Cancellation Desc	The description of policy cancellation.
Rate Date	Date the Quote/Policy was rated.
Renewal Status Flag	Renewal status code.
Renewal Premium	Premium to be charged upon renewal of the policy.
Fulfilment Cover Id	Go on Cover Option: Full, Docs O/S, Quote only
Payment Method Id	Method of payment.
Quote Type Id	It identifies if the quote is for new or MTA process.
Admin Fee	Administration fee in case of MTA process. It is fixed.
Adjustment Premium	Additional Premium(AP) or Return Premium (RP) in case of MTA process.
Total Premium	Total Annual Premium Due.
Reference Policy Nbr	Stamp of the original policy number in case of MTA process.
Ref Policy Version Nbr	The version number of the policy from which the quoted MTA was created.
Authorised Reference Num	Authorised Reference Number.
Authorising Staff Id	The user who authorised a referral case.
Propensity Lapse Disc Flag	Flag for propensity lapse discount.
PRQ Flag	The PRQ (Pre-Renewal-Query) flag is set whenever adverse claim info is recd.
Currency Id	The currency is which this particular policy was transacted in.
Term Days	The number of days in total that a policy term is scheduled to run.
Term Days Remaining	The number of days that a policy term is scheduled to run after the effective date of an MTA transaction.

Name	**Descriptions**
EDW Migrated Flag	This field indicates whether the record has been migrated from EDW.
Total Premium Old Rate	Total Annual Premium based on Current Risk and Old Rates at the time of renewal.
Premium Adjustment Factor	This field holds the Premium Adjustment Factor
Total Premium Excl Levy	Total Annual Premium (excluding fees and govt. levy)
Hdweb Reconciled Status	This column is added to identify reconcilation status of quote/policy.

Table B.5: Policy Attributes Details

Name	**Data Type**	**Applied**
Adjustment Premium	Number	Y
Admin Fee	Number	
Amount	Number	Y
Authorised Reference Num	Number	
Authorising Staff Id	Number	
BPS Transaction Type	Varchar	
Branch Team Id	Number	
Cert Returned	Varchar	
Client Id	Number	Y
Code	Varchar	Y
Cover Code	Varchar	Y
Create By	Varchar	
Create DTime	Time Stamp	
Currency Id	Number	
Description	Varchar	Y
Display Code	Number	
EDW Migrated Flag	Varchar	
Effective From Date	Time Stamp	Y
Effective To Date	Time Stamp	Y
Fulfilment Cover Id	Number	
Hdweb Reconciled Status	Number	
HDWeb Status Id	Number	
Incept DTime	Time Stamp	
Last Renewal DTime	Time Stamp	Y
Lk Transaction Id	Number	
Next Renewal DTime	Time Stamp	Y
Payment Method Id	Number	Y
Percentage	Number	Y
Pol Cancellation DTime	Time Stamp	Y
Pol Fulfilment DTime	Time Stamp	
Pol Suspension DTime	Time Stamp	Y

Name	**Data Type**	**Applied**
Policy Cancellation Desc	Varchar	
Policy Effective From Date	Time Stamp	Y
Policy Effective To Date	Time Stamp	Y
Policy Lapse DTime	Time Stamp	Y
Policy Nbr	Varchar	Y
Policy Owner Org Id	Number	
Policy Reason Code Id	Number	
Policy Reinstate DTime	Time Stamp	Y
Policy Version Nbr	Number	Y
Policy Year Nbr	Number	Y
PPP Confirmed Flag	Varchar	
Premium Adjustment Factor	Number	Y
Product Master Id	Number	Y
Propensity Lapse Disc Flag	Varchar	
PRQ Flag	Varchar	
QB Result Identifier	Number	Y
QB Result Type Identifier	Number	Y
Quote Expiry DTime	Time Stamp	
Quote Originator Id	Time Stamp	
Quote Owner	Varchar	
Quote Policy Link Id	Number	
Quote Policy Trans Id	Number	
Quote Reference Nbr	Number	Y
Quote Type Id	Number	Y
Quote Version Nbr	Number	Y
Rate Date	Time Stamp	
Reason	Varchar	Y
Ref Policy Version Nbr	Number	Y
Reference Policy Nbr	Varchar	Y
Renewal Premium	Number	Y
Renewal Status Flag	Varchar	
Review Date	Time Stamp	

Name	**Data Type**	**Applied**
Rounding Code	Varchar	Y
Running Total	Number	Y
Sales Node Id	Number	
Status Id	Number	Y
Term Days	Number	
Term Days Remaining	Number	
Total Premium	Number	Y
Total Premium Excl Levy	Number	Y
Total Premium Old Rate	Number	Y
Transaction Description	Varchar	
Transaction Type	Varchar	Y
Units	Varchar	
Update By	Varchar	
Update DTime	Time Stamp	
Vehicle Ref Num	Number	
WEB Quote Reference Nbr	Varchar	

Note: The "Y" in Used column means the listed attribute has applied for CLV Calculations.