Paper Title: Using Machine Learning to Improve Customer Credit Allocation in a Business to Business Context

Name: Derek Lukacsko

Paper Type: MSc Business Analytics dissertation

Contributions: I developed this dissertation in partnership with data science consultancy Satalia and a Fortune 500 hardware distributor. The goal was to demonstrate the potential to apply data science methods to better understand their business. I developed solutions using supervised learning (classification and regression) and unsupervised learning (clustering, i.e. customer segmentation) methods. In addition, I explored heuristic methods for evaluating customer segmentation including a package built in R programming language.

Using Machine Learning to Improve Customer Credit Allocation in a Business to Business Context

Derek Lukacsko

supervised by
Dr. Daniel J Hulme

This dissertation is submitted in partial fulfillment
of the requirements for the degree of
Master of Science in Business Analytics

Department of Computer Science
University College London

November 11, 2016

I, Derek Lukacsko, confirm that the work presented in this thesis is my own. Where information has been derived from other sources, I confirm that this has been indicated in the work. The report will be distributed to the internal and external examiners, but thereafter may not be copied or distributed except with written permission from the author and first academic supervisor (Daniel Hulme).

Abstract

There is a significant disconnect between practical business problems and how to apply and communicate complex machine learning solutions to solve those problems. The following dissertation empirically reduces this gap for a Fortune 500 company that wants to understand their customers' behavior and how this behavior can be measured in response to credit limit changes. I develop two solutions:

1). I leverage a marketing technique to characterize customer behavior, RFM analysis, for unsupervised customer segmentation. My results indicate customer behavior is almost exclusively driven by the volume of sales. Additionally, I create a unique heuristic validation technique to visualize cluster integrity in two dimensions, the results of which indicate RFM features are best normalized into quintiles in the presence of power laws. I also verify that cluster profile membership is stable over time using cluster transition matrices; the results of which provide evidence that a customer is likely to belong to the same cluster in the future.

2). I develop a method to create supervised predictive models to assess customer behavior following credit limit changes using significant features derived from the preceding solution. The method is generalizable for all customers, and is thus not limited to $n = 1$ prediction. Results indicate that current ad-hoc deliberation to allocate customer credit can be automated as a binary classification recommender system within 90% accuracy 9 days prior. More so, we can determine if the customer's behavior following the recommendation will result in desired behavior as defined by the customer with over 92% accuracy.

Such insight can augment the current deliberation process for allocating customer credit, providing a data-driven dimension to the process that has the potential to allow business users to more effectively distribute a multi-million pound source of capital.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter begins with a high level introduction to the main topics discussed in the dissertation: RFM analysis for customer segmentation and allocating credit to B2B customers. I then provide general dissertation context including my industry partners and the project objectives, both from the client's and from an academic perspective. The key contributions from these objectives are discussed in 1.4. I conclude this chapter with an outline of the dissertation and acknowledgments.

1.1 Topic Context

RFM Analysis: RFM analysis is the process of characterizing customers by when they last made a purchase (R), how often they make purchases (F), and how much total revenue they generate (M). It is an intuitive and effective method for segmenting customers into groups that behave similarly [1]. RFM features can be utilized and are effective for clustering and customer segmentation [2][3][4][5]. As my research empirically suggests, RFM features and features derived from RFM are also useful in training predictive models. I utilize RFM analysis to segment customers, determine which features are important in partitioning customers into logical groups, and finally as features to train models pertaining to customer behavior in response to credit limit changes. I further investigate the efficacy of the method in context of this dissertation: with large B2B customers.

Customer Credit Allocation: In a B2B context, customers require lines of credit to make purchases since few customers are willing to provide cash upfront

and large purchases often require financing [6]. A large B2B organization thus provides customers with lines of credit: a threshold on accounts receivable for a given period. Customers can only spend as much as their line of credit grants them, and the overall revenue generated from all customers is a function of how much credit is utilized amongst the entire customer base. Thus, effectively allocating credit optimizes revenue. Credit granted and not used by one customer takes away potential credit that another customer could have used. Likewise, a customer that goes bankrupt or is unable to pay for their purchases on account will cost their vendor money. Credit allocation is thus an optimization problem where the objective is maximize the amount of credit utilized while minimizing the number of customers that either default or don't utilize their credit. I create predictive models using RFM-derived features to determine if credit limit changes can be predicted and if we can predict if the change will result in desired behavior behavior. This method gives decision-makers a customizable machine learning tool for optimizing customer credit allocation.

1.2 Thesis Context

Industry Partners: This project was undertaken as a part of an engagement between Satalia, a London-base optimization and data science consultancy, and a Fortune 500 client. Satalia aims to help the client *increase overall customer yield* through data science. My role in this project is to demonstrate how machine learning can be utilized to understand the client's customers' behavior in response to credit limit changes through exploratory data science using 1 year of sales data and corresponding credit limit data. The methods and models that I create will be integrated into the project to increase customer yield.

SATALIA

Anonymity: I signed a non-disclosure agreement (NDA) with the client which places restrictions on how I can present the findings. Specific customer information is anonymized and only the insights permitted by the client for disclosure are presented. This does not compromise the dissertation deliverables. Extensive use of visualization is used to communicate and simplify findings for the business user while maintaining anonymity.

Project Inter-dependencies: My method for characterizing and predicting behavior in response to credit changes is intended for aggregate use (i.e. $n > 1$) and thus compliments the $n = 1$ change-point analyses provided by fellow MSc BUSICS student Maria Zervou and Satalia data scientist, Alistair Ferag. Additionally, the cluster visualization technique for evaluating RFM clusters (*see 4.2.3 and appendix D*) was also used on a much larger data-set of non-related smart meter data for a master's project by Jonathan Bourne in the UCL Bartlett School.

Project Complexity: The difficulty of this work is largely a factor of the topic's ambiguous nature; with no set question or prompt provided by the client, I create and develop methods that ultimately result in a demonstration of how advanced machine learning concepts can be applied to increase the yield and understanding of customers in the context of credit limits. I formulated the machine learning problem which involved creating data structures and feature engineering given the constraints of the data, ultimately resulting in a flexible, intuitive, and effective way to predict customer behavior. Adding further to the complexity, I extend the evaluation of RFM clustering beyond customary validation techniques.

1.3 Project Objectives

The overarching objectives of this dissertation are to determine the efficacy of RFM clustering as a means for segmenting the client's customers and to develop a method to predict customer behavior in response to credit limit changes. The following objectives relate to either the intentions of the client (a) or academic investigations that arose by nature of pursuing the clients objectives (b).

a) Client Requirements:

- Can current case-by-case deliberation for issuing credit be substituted or augmented by a predictive model?
  (*requirement fulfilled in the results, chapter 6*)
- Can we create a generalizable framework to assess and predict any customer's behavior following a credit limit change?
  (*requirement described in the methodology, chapter 4*)

b) Academic Investigations:

- Is RFM an appropriate method for segmenting B2B customers?
- Is RFM analysis effective for clustering the client's customers given they exhibit power law behavior (which was not present in the literature pertaining to RFM clustering)?
- How can we evaluate the effectiveness of clustering beyond traditional methods? Can we conclude if clustering is useful?
- Are RFM features and clusters useful as input features in creating predictive supervised models? Which features?

1.4 Contributions

The following section highlights the main results from this dissertation. The first section (a) outlines contributions that the client can utilize from the results of my experiments and models. The second section (b) contains two academic contributions that pertain to heuristic cluster evaluation.

a) Client Contributions:

- Customer segmentation for all available customers in the data (> 7000) using three features, RFM, normalized into quintiles. All customers belong to a cluster and are characterized and easily identified by their RFM attributes.
  (See chapter 5 and appendix C)

- A simple method to standardize and summarize customer sales data of differing lengths for the purpose of creating generalizable supervised models to predict credit limit changes and if such changes will be effective. This method permits predictive modelling for $n > 1$ cases so the client can create models that generalize over their entire customer base regardless of the frequency of purchasing behavior. It is specifically applicable to highly granular data (i.e. instance-level).
  (See chapter section 4.3)

- Four examples (i.e supervised classification and regression experiments) of how the client can utilize supervised learning via the aforementioned method to understand their customer's behavior in response to credit changes.
  (See chapter sections 4.3 and 6.2)

b) Academic Contributions:

- An addition to the literature on K-means cluster assessment: a method for visualizing the uniqueness of high dimensional cluster membership in two dimensions. This method uses a distance metric to compress high dimensional sets, describe the intra and inter cluster similarity to determine how strong the clustering is, and visualize the clustering clearly. The method can be used to (1) assess clustering integrity, (2) assess different ways to normalize RFM features and their effects on clustering, and (3) visualize high dimensional sets.
  (See chapter sections 4.2.3, 5.3, and appendix E)

- A method for forecasting cluster membership over time (i.e. will a customer in period X belong to a cluster with a similar profile in period Y) using cluster rank transition probabilities and network graph visualization.
  (See chapter sections 4.2.3 and 5.3)

1.5 Dissertation Outline

Chapter 2 (*Background*): Technical information, methods, and algorithms

Chapter 3 (*Literature Review*): High-level qualitative thesis context discussing CRM, RFM, and credit limit analysis

Chapter 4 (*Methodology*): Methods for processing the data, customer segmentation, and credit limit analysis

Chapter 5 (*Customer Segmentation Results*): Findings pertaining to segmenting the client's customers by RFM features

Chapter 6 (*Credit Limit Analysis Results*): Supervised classification and regression models developed for predicting a customer's future credit limit and their behavior in response to credit limit changes, as well as a discussion.

Chapter 7 (*Conclusion*): Thesis contributions, feedback, limitation, further work, and a link to the code

Appendix: Important supplementary material

1.6 Acknowledgements

I want to express my appreciation to Daniel Hulme, my course director and thesis advisor who provided on-going support from day one of my studies at University College London. To Alistair Ferag and Guangyan Song, Satalia data scientists: thank you for providing insight and recommendations throughout my dissertation. To Jonno Bourne, UCL data scientist: thank you for being my sounding board for ideas and programming techniques. To my peers, expert data scientists: Anindya Basu, Jonno Bourne, Barney Brien, Hermawan Budyanto, Henrik Ebenhag, James Hale, Mortiz Haller, Shane Kong, Miroslav Kral, Alex Lilburn, Jonathan Manfield, Andrew Mann, and Cyrus Parlin. Thank you for your friendship and support throughout this challenging yet rewarding year.

Chapter 2

Background

This chapter provides an overview of the algorithms and methods applied in the data processing, customer segmentation, and credit-limit analysis. The sections are ordered by appearance of the concept within the analyses.

2.1 Characterizing Pareto Distributions

In 1906, Italian economist Vilfredo Pareto discovered that roughly 80% of property in Italy was owned by 20% of the population. Intially highlighting the gravity of social inequalities, the infamous Pareto Principle has since been found to characterize data in other domains as well. One of these distributions is customer purchasing patterns; it has been observed and widely accepted that only 20% of customers account for 80% of an organizations sales[1] [7]. Moreover, the features used in this dissertation to characterize customer purchasing patterns (recency, frequency, *and* monetary value) indeed follow a power law distribution.

2.1.1 Feature Distribution Identification

Many empirical quantities are distributed around a particular central point with relatively predictable spreads. For example, 95% of a sample of men will fall within two standard deviation of the average male height (where $\mu \approx 178$cm and $\sigma^2 \approx 10$cm). However, assuming such normal/Gaussian behavior is confounding when analyzing variables and features that exhibit a power law. Assuming a Pareto

[1] See appendix B section 2 on how this is evident in the client's data

distributed data follows a Gaussian distribution leads to the miscalculation of extreme events.

2.1.2 Understanding Where the Power Law Begins

The first step in characterizing these distributions is to plot the functions and estimate the scaling factor and the point where the distribution becomes indicative of a power law. Thus, the method employed estimates these input parameters of the continuous power law probability distribution function (2.1) and cumulative distribution function (2.2), namely, α and $x_{\min}$ [8]. The former characterizes the behavior in the tail of the distribution and the latter is an estimate of where the power law behavior begins.

$$p(x) = \frac{\alpha - 1}{x_{\min}} \left(\frac{x}{x_{\min}}\right)^{-\alpha} \tag{2.1}$$

$$P(X \leq x) = 1 - \left(\frac{x}{x_{\min}}\right)^{-\alpha+1} \tag{2.2}$$

Where $\alpha > 1$ and $x_{\min} > 0$. A maximum likelihood estimator (MLE) is used to estimate α (2.3).

$$\hat{\alpha} = 1 + n \frac{1}{[\sum_{i=1}^{n} \ln \frac{x_i}{x_{\min}}]} \tag{2.3}$$

$x_{\min}$ is estimated using the following method 2.4 while validating fit using a log-log plot:

$$D(x) = \max_{x \geq x_{\min}} |S(x) - P(x)| \tag{2.4}$$

Where $S(x)$ and $P(x)$ are the cumulative distribution functions 2.2 of the data and power law fit respectively.

2.2 Dimensionality Reduction

Dimensionality reduction is a technique for reducing the amount of features used to a set of uncorrelated features. The primary purpose of dimensionality reduction

is to extract these features from high dimensional data sets, however it can also be utilized to understand the structure of the data and visually confirm groupings [9].

2.2.1 Principal Component Analysis

PCA is a statistical method that uses an orthogonal transformation to create underlying features called principal components derived from the original features. Each of these components contain underlying non-correlated data-affecting qualities. With highly correlated variables in a large data set, the top n principal components can be used for analysis, reducing the complexity of the computational requirement without sacrificing useful information. The result is a transformation of the original data into eigenvectors. Formally, principal component scores t_i are created by a transformation of the given weight vector w_k mapped to a row vector of instances x_i, where $t_{k(i)} = x_{(i)} \cdot w_{(k)}$. The n principal component can thus be formulated as:

$$w_{(n)} = \underset{\|w\|=n}{\operatorname{argmax}} \left\{ \sum_i (t_n)_i^2 \right\} = \underset{\|w\|=n}{\operatorname{argmax}} \left\{ \sum_i (x_{(i)} \cdot w)^2 \right\} \tag{2.5}$$

Only three features are used for clustering using RFM. Reducing the dimensionality of the data to extract features would thus only serve to obscure the deterministic inputs; with only 3 features, the dimensions would be reduced to 2 (or 1); this would alter our input from 3 understandable variables to 2 obscure components. However, in the context of supervised learning (specifically, to predict credit limit behavior using aggregated features that fall into the categories of R,F, and M), principal component analysis is utilized to potentially develop models with smaller feature sets.

2.2.2 t-SNE

T-distributed stochastic neighbor embedding is a more visually understandable dimensionality reduction technique; it is a machine learning method for non-linear dimensionality reduction that is specifically suitable for visualizing separation. The algorithm first computes a probability distribution p_{ij} proportional to the similarity of each object.

$$p_{j|i} = \frac{\exp(-\left\|x_i - x_j\right\|^2 / 2\sigma_i^2)}{\sum_{k \neq i} \exp(-\left\|x_i - x_j\right\|^2 / 2\sigma_i^2)} \tag{2.6}$$

$$p_{ij} = \frac{p_{j|i} + p_{i|j}}{2N} \tag{2.7}$$

The function learns a mapping $Y = \{y_1, ..., y_n\} \, | y_i \in R^d$ which best explains p_{ij}. The mapped similarities are measures similar to the original probability distribution:

$$q_{ij} = \frac{(1 + \left\|y_i - y_j\right\|^2)^{-1}}{\sum_{k \neq l} (1 + \left\|y_k - y_l\right\|^2)^{-1}} \tag{2.8}$$

t-SNE is highly useful as a visualization technique to detect underlying patterns between high dimensional feature sets because the algorithm tends to efficiently partition distinct groups [10]. The method is thus used for this purpose.

2.3 *K*-means Clustering

The *K*-means algorithm has been utilized successfully for a variety of unsupervised analytical tasks since its first practical application in 1982 [11]. Given a mapped representation of n instances as features, the algorithm finds k groups using a common measure of similarity that maximizes the intra-group homogeneity and maximizes the inter-group heterogeneity[2] [12].

2.3.1 Formulation

Given our instances (i.e. customers characterized by recency, frequency, and monetary value), where each observation is a real vector, the algorithm partitions the instances by minimizing the objective function (2.12). The objective function aims to reduce the within-cluster sum of squares (WCSS). This process is repeated until a point of convergence is found that minimizes the square distance between each individual point and the nearest cluster centroid. Formally, for a set of D-dimensional instances in the data-set N, each instance x_n is compared to a binary indicator $r_{nk} \in \left\{0, 1\right\}$ iteratively until the objective function is locally minimized

[2]The high dimensional heat map function in appendix E is used to visually assess this criteria

[13]. The objective function makes sets r_{nk} of instances in each k cluster that minimize the sum of squared errors.

$$x_1 = \left\{ x_1^1, ..., x_1^D \right\} \tag{2.9}$$

$$N = \left\{ x_1, ..., x_n \right\} \tag{2.10}$$

$$r_{nj} = \left\{ \begin{array}{l} 0; \text{if } x_n \rightarrow k \\ 1; \text{else } j \neq k \end{array} \right\} \tag{2.11}$$

$$\sum_{n=1}^{N} \sum_{k=1}^{K} r_{nk} \left\| x_n - \mu_k \right\|^2 \tag{2.12}$$

The Hartigan-Wong method [14] for K-means clustering is utilized because it has been found, both theoretically and empirically, to be superior than Lloyd's method [15]. Whereas Lloyd's optimization heuristically iterates to set the cluster centroid positions, $\mu(C_j)$, Hartigan-Wong takes into account the instance assignment, thus allowing instances to be re-assigned [16].

2.3.2 Clustering Algorithm Comparison

Pros: First and arguably most compelling from a pragmatic perspective, K-means is both intuitive and computationally efficient for low dimensional data. A hallmark of intricate mathematical methods and algorithms is their inability to be widely utilized by a non-technical users [17]. The efficiency and intuitive formulation of K-means lends itself to use by a wider audience. This benefit is especially pronounced when numerous experiments need to be completed. Moreover, with single cluster membership (i.e. each instance can only belong to one cluster), K-means permits easy membership identification validation [18]. In comparison

to agglomerative methods[3], the computational efficiency of K-means is especially pronounced [15].

Cons: K-means and other centroid-based partitioning algorithms do not effectively separate data that is not globular or hyper-spherical in two dimensions. Given that each item will be effectively negotiated into the most similar cluster, non-globular data will be partitioned by virtue of the point of initialization in the algorithm rather than by means of a globally optimal boundary. Also, K-means is sensitive to outliers. This was found to be one of the main drawbacks in a study that compared K-means to several other clustering algorithms [19].

Algorithm Comparison: Several common substitutes for K-means are the X-means [20], *fuzzy* K-means [19], and K-means++ algorithms [21]. Each of these algorithms are extensions of traditional K-means but offer potential solutions to the common disadvantages of the K-means clustering method. Pellag and Moore (X-means) for example optimized an introduced Bayesian Information Criterion that improved the prior likelihood estimation of cluster membership. Not dissimilar to hierarchical methods, *fuzzy* K-means permits cross-cluster membership, a useful quality when outliers are highly represented in the data set. The merit of such a method lends itself to the flexibility of instance cluster membership. This dissertation however utilizes clustering for customer segmentation purposes, of which case, distinct partitioning is valued over non-deterministic grouping.

The main appeal for the aforementioned methods is their ability to handle high dimensional data while maintaining or improving the WCSS. The Hardigan Wong method however has specifically addressed these issues while improving on clustering performance metrics and maintaining the exclusive partitioning structure [15].

[3]Unsupervised clustering algorithms are categorized by how they create clusters, either through agglomerative or partitioning methods. Agglomerative methods such as hierarchical clustering initiate the clustering per unit and then gather similar units into groups where each unit represents a group of instances that can overlap into multiple clusters. In contrast, partitioning methods (e.g. K-means) separate instances into *exclusive* clusters.

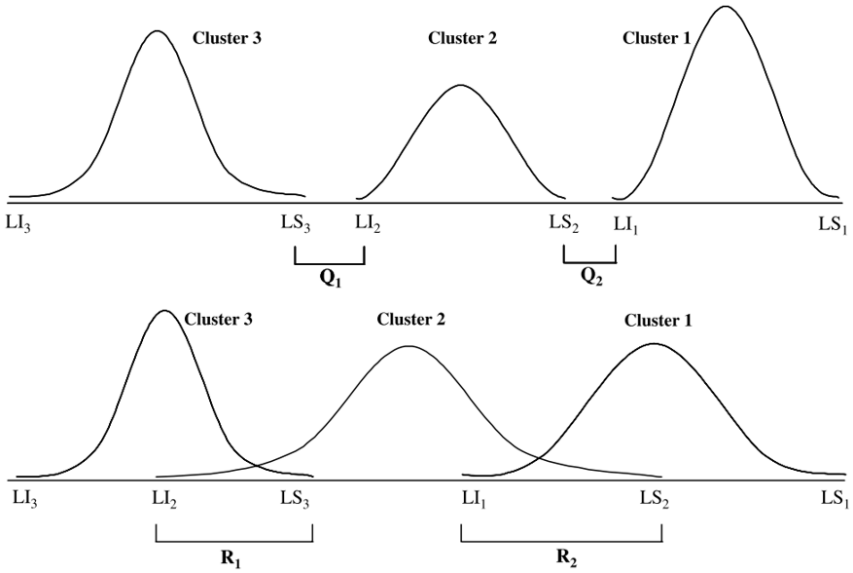


Figure 2.1: Exclusive vs. Overlapping Clustering: Fuzzy and hierarchical methods (*bottom*) permit cluster overlap R_i. The usage of exclusive clustering methods (*top*) such as K-means restricts membership at the cost of $R_i - Q_i$. The decreased level of intersection, $LS_i - LI_i$ is present by definition in exclusive clustering methods [19].

2.4 Clustering Validation

Validation for K-means is both a pre and post analysis issue; first, k must be defined, and after clustering is complete, the clusters must be assessed to attain holistic interpretation of clustering results [22]. It should be noted that the following methods are the traditional clustering validation methods utilized, however two other heuristic methods are used and will be discussed in the methodology.

2.4.1 Within Cluster Sum of Squares

The sum of the distance function for each instance culminates into the within cluster sum of squares metric. Minimizing this objective can be visualized by plotting the WCSS per each possible k and visually locating the elbow, or the point of inflection where a noticeable decrease in WCSS is present. This will signify the best choice for k. In absence of an obvious inflection point, a range of ks is tested and further validation is necessary to determine k.

$$\underset{N}{\operatorname{argmin}} = \sum_{i=1}^{K} \sum_{n \in N_i} \|x_n - \mu_i\|^2 \tag{2.13}$$

Minimizing the distance between an instance x_n and the centroid of a cluster μ_i ensures the most homogeneous cluster groupings. This is why the heat map cluster evaluation function is ordered by the diagonal in appendix E.

2.4.2 Gap Statistic

The goal of gap statistic validation is similar to WCSS method, where the elbow represents a preferred metric. The gap statistic metric is the within-cluster homogeneity, W_k, a metric that compares the clustering dispersion to a null reference distribution [23]. Formally, the sum of pairwise distances between objects in the r cluster is:

$$D_r = \sum_{ii' \in C_r} d_{ii'} \tag{2.14}$$

$d(i'i)$ is thus a vector of the Euclidean distances between vectors. With this vector, we can calculate the within-clusters homogeneity associated with the true cluster classification, W_k.

$$W_k = \sum_{r=1}^{k} \frac{1}{2n_r} D_r \tag{2.15}$$

The traditional statistic (2.16) has been found to be inferior to the non-log method in application to K-means clustering, especially when there is large variance between cluster [24]. Thus, (2.17) was used in this dissertation.

$$\text{Gap}_n(k) = E_n\{\log(W_k)\} - \log(W_k) \tag{2.16}$$

$$\text{Gap}_n(k) = E_n(W_k) - (W_k) \tag{2.17}$$

2.4.3 Silhouette Width

The silhouette method has been found to be robust particularly in application of assessing the K-means algorithm [25]. The silhouette width is another visual interpretation of cluster efficacy. Each cluster is represented as a silhouette where

the instances are shown by their distance[4] from their cluster centroid. High similarity/low distance indicates strong cohesion [27].

$$D(x,y) = \sqrt{\sum_{i=1}^{m}(x_i - y_i)^2} \tag{2.18}$$

The silhouette width, $-1 \leq s(i) \leq 1$, characterizes the average dissimilarity $a(i)$ of an instance between all instances in that cluster and the lowest average dissimilarity between that instance and the nearest disparate cluster $b(i)$.

$$s(i) = \frac{b(i) - a(i)}{\max\{a(i), b(i)\}} = \begin{cases} 1 - a(i)/b(i) & \text{if } a(i) < b(i) \\ 0 & \text{if } a(i) = b(i) \\ b(i)/a(i) - 1 & \text{if } a(i) > b(i) \end{cases} \tag{2.19}$$

$a(i) << b(i)$, indicates high intra-cluster cohesion; $s(i)$ near one is indicative of $a(i) << b(i)$ whereas $a(i) >> b(i)$ near -1 indicates an instance is more similar to another cluster and thus the clustering results are poor.

2.5 Supervised Learning Models

Supervised learning refers to machine learning problems that use *labelled* data to train models for the purpose of predicting a class (classification) or a value (regression). This is in contrast to the aforementioned unsupervised method, K-means clustering, which infers an underlying pattern or structure from *unlabelled* data.

2.5.1 Base Models

Base models provide a benchmark to improve upon when creating predictive models with supervised learning. These models are quick and easy to implement. For classification tasks, logit or logistic regression is a commonly used base classifier [28]. It aims to find a feature space to estimate $P(y_i = 1|X)$ and the result is a predicted class, 1 or 0. Formally, a linear function of a single independent feature x is mapped to $t = \beta_0 + \beta_1 x$ where β_i correspond to fitted parameters.

[4] K-means clustering using Euclidean distance (2.18) has been found to be superior to other popular distance metrics used in clustering such as Manhattan distance [26].

$$F(x) = \frac{1}{1 + e^{-(\beta_0 + \beta_1 x)}} \tag{2.20}$$

For regression tasks, linear models are common base models. Multiple linear regression was used as a base model for regression tasks. The function assumes a linear fit between the regressors or independent variable(s) X and the dependent predicted value, y_i. β and ε denote the weighting term and random error or noise.

$$y = X\beta + \varepsilon \tag{2.21}$$

2.5.2 C5.0 Algorithm

The C5.0 algorithm is a robust classifier derived from computational improvements in the C4.5 algorithm [29] [30]. It develops trees that create partitions at each node based on entropy or the amount of information gained by the given split. It has been ranked as the top classification algorithm at the International Conference on Data Mining in 2007 [31]. Moreover, an empirical comparison of 171 classifiers found that C5.0 had the best accuracy among boosting ensemble, decision trees, and rules based classifiers [32]. The objective function, information gain, is expressed as the change in entropy from one state[5] $H(T)$ to the next:

$$IG(T,a) = H(T) - H(T|a) \tag{2.22}$$

$$IG(T,a) = H(T) - \sum_{v \in vals(a)} \frac{|\{x \in T | x_a = v\}|}{|T|} \cdot H(\{x \in T | x_a = v\}) \tag{2.23}$$

2.5.3 Random Forests

Random forests are robust ensemble methods that can be used for regression or classification tasks. They grow a user-specified amount of trees N which each vote for class outputs and the algorithm chooses the classification with the most votes.

[5] In context of decision trees, these states are combinations of training examples T.

One key advantage of random forests is they curb overtraining of single complex decision trees. Each tree in the forest is constructed from a different bootstrap sample which thus computes error within the sample. Given a training set of predictors X and responses y, a random sample is selected N times with replacement to fit the trees. Hold-out data can thus be tested by:

$$\hat{f} = \frac{1}{N}\sum_{n=1}^{N} \hat{f}_n(x') \tag{2.24}$$

Partitioning methods are dictated by the integrity of their branching structure. Random forests output two metrics of variable importance which characterize the structural integrity and feature importance [33]: (1) the average out-of-bag error before and after permutation within training and (2) the Gini importance. Method 1 provides an indicator of how important a feature is by finding the mean decrease in accuracy without it. Method 2 evaluates the importance of a feature X_m by summing the decrease in impurity $p(t)\Delta i(s_t,t)$ for all nodes t averaged over all trees N_T.

$$Imp(X_m) = \frac{1}{N_T}\sum_{T}\sum_{t\in T: v(s_t)=X_m} p(t)\Delta i(s_t,t) \tag{2.25}$$

In the previously mentioned study [32], random forests were found to be superior in both classification and regression tasks compared to 171 other learning methods, achieving accuracy of over 90% in several tests. Moreover, recursive partitioning methods such as the random forest algorithm have been found to be superior for determining if credit should be granted in empirical studies [34] [35], lending itself to the application of this dissertation. Also, (like K-means) partitioning methods are easy to understand for the business user which lends itself to external validation.

2.5.4 Ensemble Learning

Supervised learning tasks attempt to find a hypothesis space to model the relationship between the independent and target variables. Ensemble learning combines several individual classifiers, creating a normalized hypothesis space. Three popular ensemble techniques proposed by Witten *et al* (2011). were utilized in this paper:

bagging, boosting, and stacking [36]. Bagging trains a random subset data in multiple models where each model votes for a classification outcome. Boosting aims to reduce the mis-classification in each model trained by focusing on those instances misclassified. Bagging and boosting are used in the training of Random Forests and C5.0 and have been found to increase model accuracy and flexibility [37]. Stacking entails combining final model outputs; such methods have been shown to increase classifier accuracy and have thus been employed in an attempt to improve final out of sample model accuracy for experiments 3 and 4 (discussed in chapter 5) [38].

2.6 Chapter Summary

This chapter discussed Pareto distributions which, although not mentioned in the literature on RFM clustering, have a significant effect on the client's data used in this dissertation. I then discuss dimensionality reduction; specifically, PCA is used primarily for feature engineering and t-SNE is utilized for visualization. I then discuss the traditional K-means clustering validation techniques because I create two heuristic validation techniques to compliment these techniques: a high dimensional heat map visualization to simultaneously assess within-cluster homogeneity and between cluster heterogeneity, and (2) an experimental method to assess cluster rank transitions. I then discuss Random Forest and C5.0 algorithms, two high performing supervised methods which I employ along with other CART methods to create models to predict credit limit allocation and whether or not a credit limit change will be effective.

Chapter 3

Literature Review

The following chapter provides a high level background of how RFM analysis can be used to characterize and segment a customer base and how these methods can inform credit limit decisions. K-means clustering has been found to be effective in segmenting customers into clusters based on their RFM scores. Clusters can be used to determine structure and commonalities among groups. In a B2B context, businesses can utilize clustering and features based on RFM metrics to inform how credit worthy customers are and consequently where to allocate credit.

3.1 Recency, Frequency, Monetary (RFM)

This section provides context for the dissertation with a brief overview on the role of customer relationship management (CRM) in organizations and how segmenting customers by RFM features has been demonstrated as a viable CRM and marketing tool.

3.1.1 Contemporary Context of CRM

Mass adoption of the Internet denotes a historic landmark in the B2B business environment. The *Age of Information* [39] opened-up the previously inaccessible network of customers and information; new database technologies allowed organization's to effectively capture the data of their clients. At the same time, it became easier for customers to find the products they needed and consequently it became easier for them to leave their current providers. The organization's value proposi-

tion model thus changed from a product-centered to a customer-centered paradigm [40], where relationships became a primary differentiator. Customer relationship management (CRM) is now widely accepted as "the key competitive strategy" [41]. CRM is a comprehensive business and marketing strategy that integrates technology, process, and all business activities around the customer and customer data [42].

3.1.2 CRM as a Strategic Imperative

Understanding the customer, their activities, preferences, purchasing patterns, and requirements and managing this relationship is now dictated by data. Although CRM systems have been around since the late 1980s, they have only recently been made scalable and easy-to-use across a standardized framework that enables manageable data capture [40]. This data helps explain the complex behavior of the customer, allowing organization's to make decisions targeted to their customers, consequently improving the relationship and thus the profitability of the organization [43].

The end goal of customer relationship management is to use actionable, descriptive customer data to cater solutions, services, and products to the customer so that they remain a customer. Loyal customers provide a steady stream of income by continuing to transact with the vendor in place of competitors [44]. Effective CRM can thus contribute to an organization's top line by maintaining historical streams of revenue through repeat customers. Additionally, acquiring new customers is five times more expensive than retaining existing customers [45]. Thus, maintaining a loyal customer base via CRM can also be viewed as a cost-reduction, bottom-line improving mechanism.

As a testament to the efficacy of CRM, Microsoft has just recently invested $23M in the CRM start-up and solutions provider, Helpshift [46] , and the Royal Bank of Scotland announced that they will increase their focus on CRM as a "push to be more customer-centric" [47]. In a dynamic business network where information, products, and clients are equally accessible, building relationships and developing the links within the network has clearly become a strategic priority. In context of this dissertation, I utilize essential CRM data to characterize behavior by RFM,

thus creating a better, data-informed understanding of the client's customers.

3.1.3 Using RFM to Measure CLV

The culminating metric to characterize the value of a customer is their lifetime value: a prediction of the customer's future net profit (i.e. the present value of their future cash flows) [45]. Customer relationship management aims to maximize this value and many methods exist to calculate CLV in monetary terms [48].

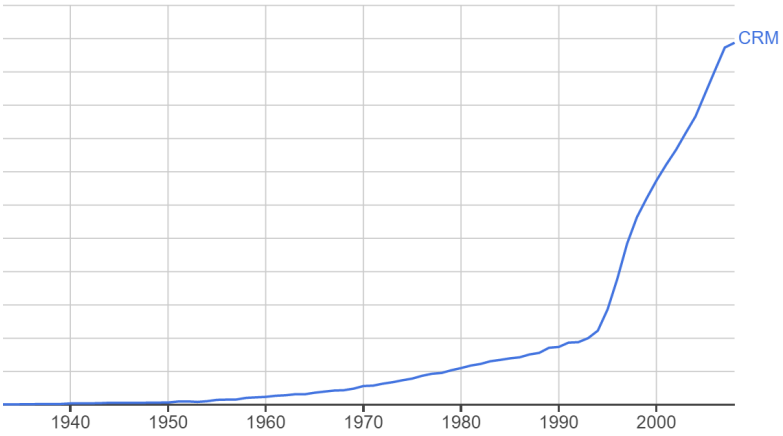


Figure 3.1: Google NGram: from 1940 to 2008, visualizing the percentage of printed texts containing the term "CRM". A noticeable peak around 1994 coincidentally corresponds to the emergence of "RFM" in the same year, suggesting RFM analysis may have contributed to the emergence of customer relationship management [49].

The emergence of RFM as a versatile metric offers an alternative method for calculating customer value that takes into account the recency, frequency, *and* the monetary value of the customer or customer segment over a specified period t (see equation 3.1). By adding two temporal dimensions to the financial return, the method provides a more holistic perspective of their customer's value. Recency corresponds to how recently the customer made a purchase; frequency, how many times the customer made a purchase during the given period; and monetary value, the total amount of revenue (or other financial KPI) generated by the customer in the given period.

$$\text{CLV} = w_R R_t + w_F F_t + w_M M_t \tag{3.1}$$

High recency indicates a valuable customer (this appears to hold true moreso in a B2C context). In consumer psychology and marketing alike it is widely accepted that "past behavior is the best predictor of future behavior" and thus, a return customer signifies loyalty and is therefore highly ranked (i.e. high recency $\Rightarrow$ high CLV) [1]. Depending on the industry, customer segments, or the nature of business in the organization utilizing the method, the "ideal" customer profile will differ. For example, my client sells large hardware and equipment and thus may consider frequency and recency as less important given that their customers traditionally make less regular large purchases. The sales or business development component of such a company's CRM system would thus place a higher weight on monetary value in their RFM measure. Indeed, even without weighting, the prominence of monetary value for the client became apparent during this dissertation.

Haenlein *et al.* (2007) advocate that CLV estimation must follow the following criteria: (1) CLV should focus on homogeneous segments of customers and (2) it should be easy to understand and parsimonious in nature [50]. Using RFM to calculate CLV (see equation 3.1) fulfils these requirements by offering a company-specific yet objective measure to understand and characterize customer value and is thus a viable method for quantifying customer group value.

3.1.4 Using RFM for Customer Segmentation

RFM has been shown to be an effective measure in customer relationship management for segmentation [51][52] largely because it can be used as a flexible input to characterize segments of customer groups. Indeed, Khajvand *et al.* (2011) have demonstrated empirically that RFM can be used as features to segment customers, predict customer future value, and apply segmentation to marketing decisions [2][3]. Moreover, although RFM was initially developed as an equally weighted measure [49] where recency, frequency, and monetary value each contribute equally to the comprised metric, it's functionality as an adjustable measure provides an element of specificity that can improve modelling [53]. Herein lies the crux of RFM analysis: the deterministic, customizable, and organization-specific nature of its formulation. RFM analysis inputs are highly flexible which allows a

wide range of individuals in an organization to utilize the method, however the output will be a function of the user's assumptions. As such, RFM analysis requires thorough understanding of the CRM data before any actionable decisions can be made to classify customer segments by their CLV calculated with RFM [52].

I use R,F, and M as three features to segment the client's customers into logical groups. These features are discretized into quintiles (discussed in chapter 4) that encompass percentage ranks that each customer falls into. As an intuitive numeric classifiers, the RFM attributes provide a standardized way to describe and reference individual customers throughout the credit analysis as features. I then use features that fall into one of the three RFM categories to train predictive models (discussed further in chapter 4).

3.2 Customer Segmentation

I use unsupervised clustering to uncover structure in the client's CRM data using RFM features. K-means clustering on RFM variables is used to create customer segments which can also be used as features in the credit limit analysis. These segments are characterized by their membership of RFM classes.

Jain (2010) noted three main purposes of clustering: (1) to identify the underlying structure, (2) to quantify the similarity and thus classify similar individuals together, and (3) to compress or summarize high dimensional data [12]. I leverage clustering for these three purposes as follows. I use K-means clustering first to identify structure through segments defined by RFM. Then, I quantify the similarity with a function I designed to visualize high dimensional cluster heterogeneity and within-cluster homogeneity. My function permits high dimensional summarizing by condensing vectors into a specified aggregated vector thus fulfilling Jain's criteria. See chapter sections 4.2.3, 5.3, and appendix E for further information.

The efficacy of K-means as a clustering method for CRM data on RFM attributes is supported and has been applied successfully to market basket analysis, retail sales analysis [54], recommender systems, and customer segmentation [55] [56]. Each of the these papers metricized customers using similar RFM attributes,

clustered them using K-means, and ultimately leveraged their clusters to create tangible value that could inform marketing decisions such as devising new marketing strategies around the clustered segments [57]. Clustering customers in segments using RFM as attributes or features is the first step in developing a characterization and understanding of the customer base.

3.3 Credit Limit Analysis

RFM features have been paired successfully in the literature with credit-related features to successfully segment customers and predict credit default [58], however this is more applicable in a B2C context. This dissertation utilizes the instances of credit limit *change* to characterize the normalized effectiveness of increasing and decreasing credit limits. I apply supervised regression and classification methods after clustering in two related experiments: to determine what the next credit limit change will be (confirmatory models; i.e. *are decision-maker choices to change credit limit predictable?*) and to see if after these decisions are made, we can predict if they will be effective (predictive models; i.e. *can we predict if increases or decreases in credit will lead to a desired outcome?*).

3.3.1 B2B Credit Allocation

The literature on credit limit analysis primarily focuses on whether to provide customers (usually B2C) with credit. Many papers and contemporary issues discuss credit in the context of risk or risk-level (e.g. the 2008 financial crisis) [59]. The literature on B2B credit analysis aims to identify if a line of credit should be granted, integrating prior probabilities as formulations of risk and quantifiable justifications [6]. However, ith large customers *how much* and *when* to provide credit are more important than *if* credit should be granted (this is because customers are normally repeat, long-term customers who already have lines of credit) [60]. In the context of customer relationship management, an organization wants to make the optimal decision as to who and how much credit should be granted. An under-utilized line of credit for one customer deprives another customer of valuable credit. Thus, this

dissertation aims to characterize and predict if a credit limit change will be effective

One of the main issues in assessing the credit-worthiness of a B2B customer is the excessive heterogeneity of a customer base [61]. With such diversity and typically handled with a case-by-case intuitive approach, assessing credit in B2B setting has been deemed a difficult area for research into optimization methods [62]. Nonetheless, I have found interesting parallels in several papers attempting to optimize B2B credit allocation decisions with RFM analysis; their results suggest RFM clustering can improve customer understanding as a means of homogenizing heterogeneous groups into actionable segments and as a result, improve credit allocation decisions. For example, Safi and Lin (2014) found that non-financial temporal factors such as how long a customer has been a customer were statistically significant in assessing credit worthiness [63]. Such features can be directly converted to recency or frequency metrics. Likewise, I use features related to RFM to train predictive models (results in section 6.2 and appendix D).

3.3.2 Informing Credit with Machine Learning

Recommender systems are models that predict/recommend an item given a user's past behavior (collaborative filtering) or similar item profiles (content-based filtering) [64]. Recommendation systems using K-means clustering and RFM features have typically been item-based, focused on recommending items based on the RFM features of clustered customers [56][65]. Ultimately, once the credit behavior of customers and customer segments is understood, learning models can be implemented into automated recommender systems. For example, my confirmatory models suggest that current ad-hoc credit changes are predicable and thus can be made automated by a recommender system. I then create the predictive models to demonstrate suggested target variables using static and aggregated features similar to how Shi *et al*. (2015) trained models to determine credit worthiness using static and dynamic features [66].

3.4 Chapter Summary

This chapter discusses RFM analysis as a tool used for customer relationship management and understanding customer behavior. I discuss the literature that has used RFM specifically for K-means clustering and how the clustering and features derived from the clusters can be leveraged for supervised learning experiments which I demonstrate in chapters 4 and 6. These include confirmatory experiments (i.e. *are current credit limit decisions predictable and thus able to be replaced by a recommender system?*) and predictive models (i.e. *given a credit limit change, can we predict if it will be effective?*). I also introduce how my high dimensional heat map visualization will be used to evaluate cluster similarity, a key factor in clustering efficacy.

Chapter 4

Methodology

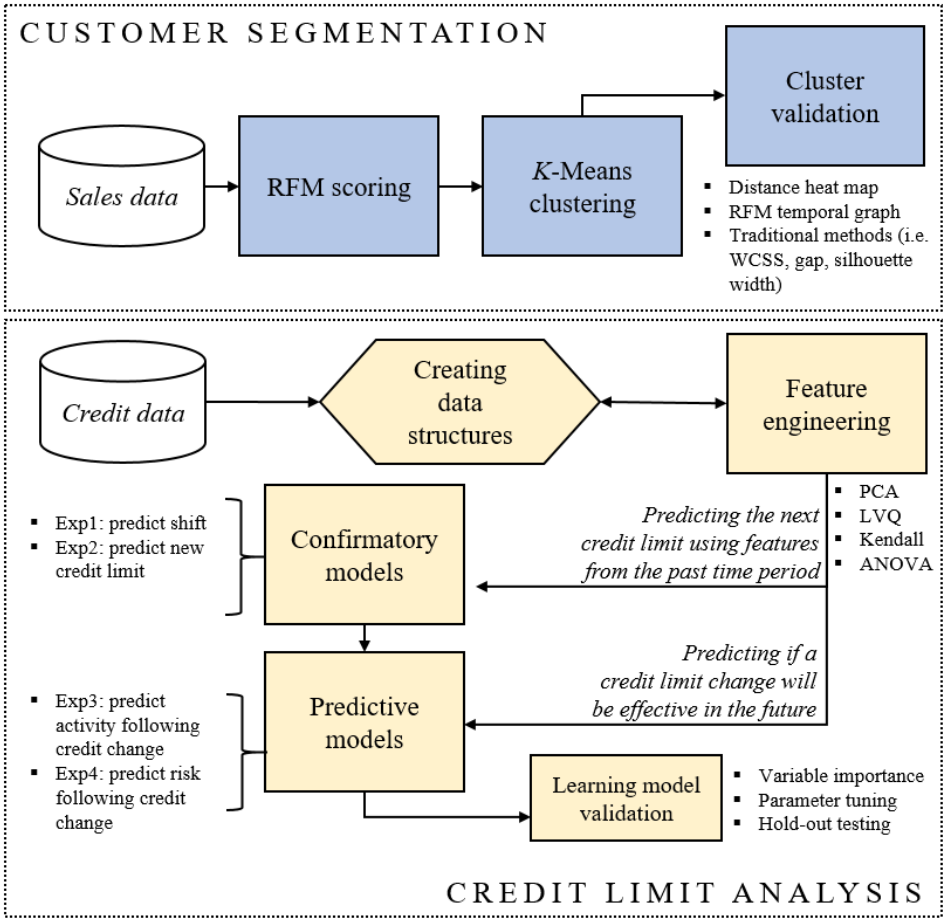


Figure 4.1: Methodology Overview Flowchart

This chapter is composed of three sections: pre-processing, customer segmentation, and analyzing credit allocation. The methodology for customer segmentation pertains to the results in chapter 5; the key information is how I validate the clus-

tering. The methodology for credit limit analysis pertains to the results in chapter 6; the key information is how I create data structures and the formulation of my supervised learning experiments.

4.1 Data Pre-processing

The following methods were employed before scoring the customers on RFM, clustering the customer using K-means, and creating supervised learning experiments to confirm credit limit changes (confirmatory models) and predict customer behavior in response to credit limit changes (predictive models).

4.1.1 Data Descriptions

I used CRM sales and credit data from my industry partner, a client of Satalia. Sales data was provided from their United Kingdom division. Each instance (rows) in the data set corresponds to an individual sale and the sale's related meta-data (columns) including financial variables, demographics, and product information.

Raw Features	Primary Descriptors	Extraneous	Unknown
used for RFM and training	largely unused	removed	removed
e.g. `Net.sell.price`	`Customer.group`	*blank*	`Logo`, `ISE`

Table 4.1: Initial Data Variable Cleaning: Variables were processed by determining their usefulness in each stage of the methodology. The working data was effectively reduced by 75% through manual user identification, consequently improving manageability and computer processing efficiency.

The most recent 11 months (01/06/15 - 30/04/16) of data was utilized for analyses and subsequent machine learning methods. By using the most recent temporal data, we avoid training the models on customers that are no longer active and erroneous past behaviors that are no longer relevant. Eleven months is a representative sample of the past fiscal year, where the most recent activity will be present in the data. Nonetheless, this cutoff directly affects the recency feature given that the minimum recency, R (i.e. *how long has it been since the customer's last purchase?*)

for each item will have a lower bound threshold defined by the maximum interval[1] $S = \{01/06/15, ..., 30/04/16\}, \min_{i \in S}(R_i)$.

4.1.2 Anomaly Detection

The client has several customers that account for a disproportionate amount of total net sales, skewing the significance and validity of the clustering methods (K-means clustering is susceptible to large distortions in the presence of outliers). I thus define an outlier as a customer that spends an amount within the range of the alpha of the corresponding power law distribution. The distributions for each feature used to score R, F, and M are characterized by the Pareto distribution, $p(x) = Cx^{-\alpha}$. A Kolmogorov Smirnov test confirmed which feature distributions had significant power law behavior and $X_{\min}$ and α were used to understand it (further details in Appendix C).

Statistical thresholds were used to remove the anomalies/outliers [5]. Thresholds from 95% to 99% were tested to see how much of the power law was removed, and it was for this reason that 98% threshold was used because it was near the tail end where the distribution exhibited power law behavior (i.e $>> X_{\min}$) but it did not completely discount the power law. The cut-off improves the clustering accuracy measured by the within cluster sum of squares (WCSS), however it continues to hold the majority of the long-tail (the defining feature of power law distributions) and therefore does not sacrifice a significant proportion of the original distribution structure. Indeed such massive customers would be best treated as $n = 1$ cases to reduce the risk in using predictive models for non-generalizable cases. In a review on the significance of outlier detection, Pearson *et al.* (2002) highlighted the importance of simultaneously addressing outliers without completely discounting their occurrence. A moderate threshold fulfills this claim [67].

[1]Note: the maximum interval for recency is typically defined as the time between the first instance and *now*. The day following the final day, 01/05/16 was used as the last day since the data was not dynamically updated. Doing so ensures *now* remains static.

4.2 Methods for Customer Segmentation

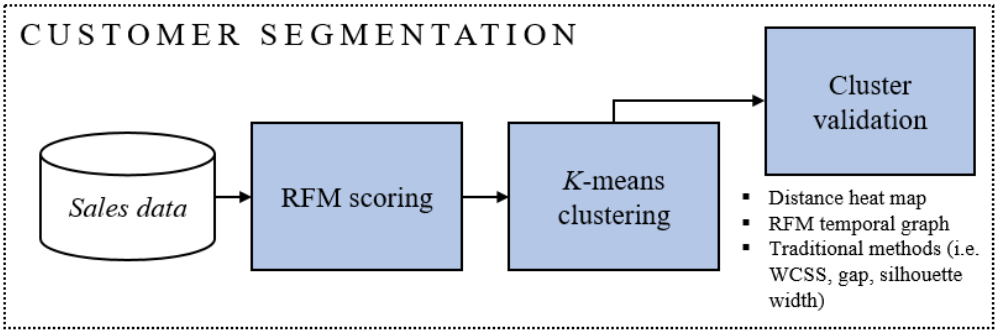


Figure 4.2: Customer Segmentation Methods Overview

4.2.1 RFM Scoring

Defining Recency, Frequency, and Monetary Value: The client provided a description of the financial variables in their CRM sales data set. This information was used to determine which features should represent Monetary Value.

- `Pocket_profit` = the margin including back-end rebates and additional operating costs (`Net_sell_price` - `Invoice_cost`).
- `Net_sell_price` = actual customer price. This is the revenue generated up front that the customer pays.
- `Back_end_rebate` = estimated rebate dollars received from the vendor for the sale. A vendor pays this to the client when the client sells their product.
- `Net_freight` = cost of shipping/freight to the client (`Actual_freight` - `Freight_cost`).
- `Logistics_cost` = cost of logistics activities. This amount is set by the sales system.
- `Financial_cost` = cost of financial activities. This amount is set by the sales system.

Thus, the ultimate profit to the client can be formed as: `Pocket_profit = [Net_sell_price - Inv_cost] - [Back_end_rebate + Net_freight +`

`Logistics_cost + Financial_cost] + POM`. From the following features, I deemed `Net_sell_price` to be the most suitable feature to use for M because it encapsulates how much the customer actually pays for the products without the contingent invoiced cost and other post-sale add-ons. A metric such as `Front_end_profit` may be more useful for measuring financial impact but only *once the invoice has been paid in full.* Consequently, large customers often have a large accounts payable via a line of credit granted by the client. Therefore, using `Front_end_profit` would be confounding since invoices are not always paid during the period of purchase. `Net_sell_price` captures the magnitude of the sale which is congruent with the purpose of the Monetary Value variable in RFM analysis [1].

Discretization of RFM Features: Researchers using RFM features to cluster have proposed alternative methods to standardize the raw RFM features. One suggested method is to scale the weighting of each metric using group decision making or analytical hierarchy processing (AHP) [55][68]. This method was not conducive to this disertation because it requires both an agreed upon weighting criteria of the RFM features and at least three decision-makers to weigh their opinions on the feature weighting, both of which were not available. Cheng *et al.* (2009) employed a more objective alternative method by normalizing the data on a discrete quintile scale [5]. This method was consequently utilized in the clustering for this dissertation because it is intuitive to model and explain and discretization does not disregard the underlying power law distributions of the data. Thus, I use discretization via quintile buckets to represent the features for recency, frequency, and monetary value.[2]

Quintile Normalization: Quintile normalization [1] is a method used to segment a features' distributions into five ordinal groups, thus creating logical

[2]In appendix E, I compare three discretization methods against the reference power law distribution and a normal distribution using the function (see 4.2.3) created by my myself and Jonathan Bourne to visualize high dimensional clustering similarity in two dimensions. The results of this both support the effectiveness of the visualization method *and* the use of quintile normalization for the reference power law distributions (i.e. for this dissertation).

bins from continuously distributed data. The raw data was normalized using this method, creating bins from 1 to 5 for each feature (RFM). The advantage of a quintile scale is it creates clear customer categories[3]. Formally [69], for x_j in ascending order for $j = 1,2,...,n$, $w_{(j)}$ corresponds to the weights of of x_j. To obtain the pth percentile, let $P = \frac{Np}{100}$ and $N = \sum_{j=1}^{n} w_{(j)}$

$$W_{(i)} = \sum_{j=1}^{i} w_{(j)} \tag{4.1}$$

The first index i such that $W_{(i)} > P$ attains the pth percentile by:

$$x_p = \begin{cases} \frac{x_{(i-1)}+x_{(i)}}{2} & \text{if } W_{(i-1)} = P \\ x_{(i)} & \text{otherwise} \end{cases} \tag{4.2}$$

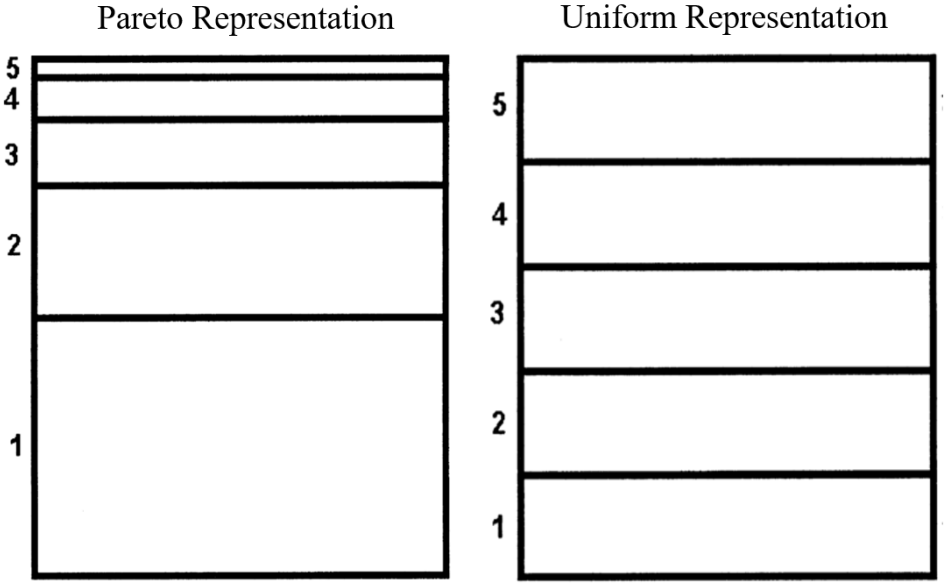


Figure 4.3: Quintile Binning of RFM Features: The actual distribution binning of the client's data (*left*) represents the binned data ranges given a power law distributions. The fat tail [Block 1] will have a larger range of values than a typical ordinal representation using uniformly distributed data (*right*). This is important to note when interpreting results.

[3]With 5 variables for 3 features, there are $5x5x5 = 125$ possible combinations of RFM. Note: only 108 of the possible 125 are present in the data.

4.2.2 *K*-means Clustering

Creating the Working Data Frame: The data was structured prior to clustering in a tabular structure $N \times M$ where $N = \{C_1, ..., C_n\}$ and $M = \{r_i, f_i, m_i\}_{i \in Q}$, and each instance, C_i, corresponds to a an aggregated characterization of a unique customer using scores on a quintile scale Q for R,F, and M as three features. With around 7,000 unique customers over the 11 month period and 3 features, the data frame for clustering contains around 21,000 elements.

Implementation: I then clustered the entire data-set using K-means clustering with the Hartigan Wong algorithm. Choosing k involved inspecting the elbow of a the within cluster sum of squares plot as a baseline gauge for what k would produce the optimal segmentation structure[4]. A gap statistic plot and silhouette was computed in parallel, each of which provided a visual interpretation of cluster integrity at k. I then used two heuristic validation methods to assess the clustering. Ultimately, clustering was considered on $k = \{2,3,4,5,6,7,8\}$, however $k = 6$ created the best substructure for differentiation. All clustering was done in `R.3.3.1`.

4.2.3 Cluster Assessment & Validation

Creating a Function to Visualize Cluster Similarity:[5] One issue with validating clustering results of high dimensional data is visualizing both the homogeneity within the cluster *and* heterogeneity between clusters simultaneously. I created a distance matrix heatmap visualization method that solves this problem with Jonathan Bourne, fellow UCL MSc Business Analytics student [70]. The function creates a distance matrix for each customer scored on RFM features. The matrix is fed into a heat map function that can visualize high dimensional data (a clear issue using built in matrix visualizations in R). Each instance of the similarity heat map can be scaled-down by a user-specified amount (e.g. 10,000 instances $\rightarrow$ 1,000 instances where each instance is a summary vector of 10). With clear cluster boundary lines, within and between cluster similarity can be compared. The

[4] See Section 2.4 for method formal explanations.

[5] See appendix E for a comparison between clustering methods using this technique.

method can be customized using different data, different features, and different distance metrics.

This method is more robust and versatile than traditional cluster heat map evaluation for three reasons: (1) high dimensional data sets can be condensed at the instance-level, which was previously mentioned; (2) metrics are extracted from each square of the symmetrical matrix which permits quantification of the similarity or dissimilarity between and within clusters; and (3) the method orders the diagonal of the clustering, permitting more precise intra-cluster homogeneity evaluation. Application and further explanation of this method are in appendix E.

Heuristic Cluster Temporal Integrity: I used several papers that researched predicting clustering using RFM [2][3] as inspiration to investigate whether my clusters maintained their structure and membership over time as a heuristic validation method. The studies used seasonal ARIMA models to plot the cluster ranks over time, using 6 quarters of data to predict the next quarter's cluster ranks. The ARIMA method requires normally distributed residuals and a significant amount of data, both of which requirements are questionable. I looked at the problem from a different angle; by scoring the *customers* (rather than using the cluster centroids) by RFM on a smaller time scale (monthly instead of quarterly) and investigating the cluster *transfer* from period to period as measured by what cluster rank a customer belong to each period, I was able to visualize the stability between cluster ranks as dictated by every customer present in each period. This method is useful because (1) more data is able to be utilized, (2) the ultimate goal of clustering (i.e. understanding where customers are and how they behave) is satisfied in a way that centroid averages don't fulfill, (3) the method can use for more granular clustering by clustering month by month instead of quarterly, and (4) no assumptions are made about the change in cluster profiles (i.e. the RFM centroids).

The method uses transition matrices on the cluster profiles (see 5.3 for more info). I built an Igraph network visualization in R and Gephi where the edges between customer nodes are 95% correlated cluster transfers [71]. Thus, highly correlated

customers grouped together indicate they would travel to the same clusters regardless of the rank. The final visualization suggests cluster membership is relatively stable over time; if we put a customer in arbitrary cluster in one period, in the next period, that customer would likely be clustered with many of the same customers they were clustered with in that initial period.

Transition Matrix Density: In conjunction with the investigation into cluster temporal integrity, I constructed a transition matrix from the cluster rank membership of each customer at each month. A squared transition probability of $1/5$ would indicate that the customer(s) randomly transition from cluster rank to cluster rank each period since there are five ranks ($0 =$ not present in the period; $1-4 =$ranks $1-4$ with 4 being the highest ranked cluster, i.e. $\text{Rank}_t = R_t + F_t + M_t$). Kernel density estimation is then utilized to estimate the probability density function of the transition. For the independent and identically distribution of probabilities $(x_1, ..., x_n)$ of an unknown density f, the shape of the distribution is estimated as:

$$\hat{f}_h(x) = \frac{1}{n}\sum_{i=1}^{n} K_h(x - x_i) = \frac{1}{nh}\sum_{i=1}^{n} K(\frac{x - x_i}{h}) \tag{4.3}$$

Plotting the density of the customer base's transition probabilities reveals how strong cluster rank membership is over time. Indeed this method has been shown to be efficacious for temporal data clustered on K-means [72].

t-SNE: t-SNE visualization was used to verify structural integrity of the clustering and which features (RFM) were more important in explaining the clusters. The raw feature data for RFM was used as an input into t-SNE which then created a similarity mapping of the dependencies between the data. When colored by R,F,M, and Cluster ID, the t-SNE visualization reveals how homogeneous the data is when partitioned by R,F,M, and Cluster ID, thus suggesting which of the four are more robust in explaining the segmentation of the customers.

Traditional *K*-means Validation: The methods outlined in chapter 2 section 2.4 were then utilized to assess the clustering. These include assessing the within cluster sum of squares, the gap statistic, and silhouette width.

4.3 Methods for Credit Limit Analysis

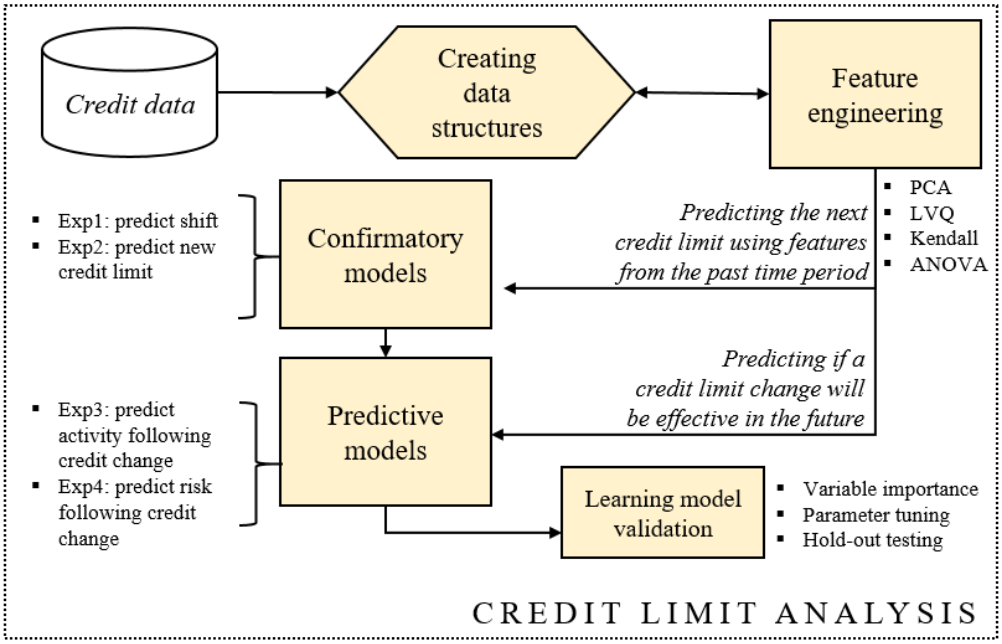


Figure 4.4: Credit Analysis Methods Overview

4.3.1 Creating Data Structures

Merging Data: The sales data utilized for clustering was combined with credit limit data for the same period. Credit limit changes were inferred by the change in credit limit from month to month in the data provided, not by specific indications of change. The resulting data structure (216338 x 27; instances are sales and columns are features and meta-data/descriptors) is altered and augmented through feature engineering before modelling.

Creating *Ex-post* and *Ex-ante* Segments[6] for Modelling: My first

[6]IMPORTANT: in reference to credit limit analysis and creating predictive models, *segments* refers to the data structures created to train the models. This is in contrast to segments referenced in customer segemntation and clustering which refer to clusters of customers with similar RFM features.

and second experiments explore if the next credit limit change is predictable (i.e. based on a set of aggregated features x over a certain amount of time B, can we predict that the credit limit will be moved up or down and by how much by the next period C?). These models can be characterized as *ex-post* or confirmatory because they confirm ad-hoc human decisions; the client makes case-by-case credit limit decisions that are not random, but a function of certain characteristics of the customer (i.e. *ex-post* features). They are useful for understanding what features or characteristics subconsciously affect these decisions which are inherently non-random (model accuracy $> 50\%$), and also for determining if such decisions can be automated by a recommender system model.

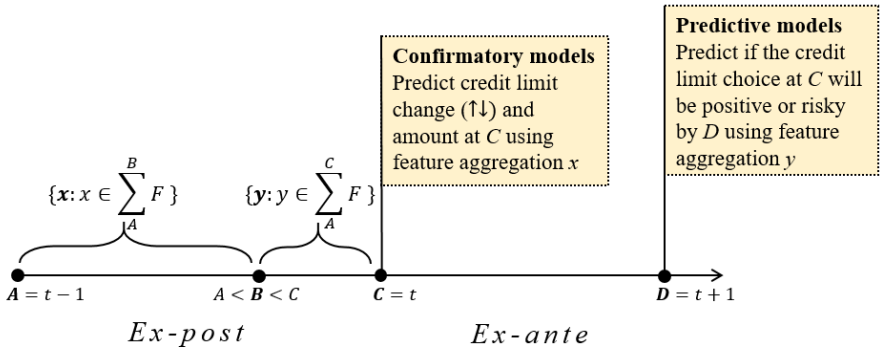


Figure 4.5: Data Structure Configuration for Experiments

My third and fourth experiments were to create a predictive models that fulfill the clients main prerogative: figure out how to measure the impact of a credit limit change. These experiments required an *ex-ante* configuration, where features y are aggregated until time C to predict the target at D. Creating segments collapses customer behavior into aggregated periods that are characterized by the credit limit at each period (e.g. a customer with three credit limit changes over an arbitrary period, $£1000_A x \rightarrow £2000_B \rightarrow £1000_C$ would be summarized as three sets/observations of aggregated features). In this way, generalized features can be engineered across the entire customer base regardless of customer behavior.

4.3.2 Feature Engineering

The feature engineering process begins with the client's motivation for the results (*How can we measure the impact of a credit limit changes?*). I am taking a macro-approach to providing insight into this question, modelling to understand customer behavior in response to credit changes that can be *generalized* (i.e. $n > 1$) across customer segments and clusters. I did this by **A** performing ANOVA tests on base RFM-related features following an increase and decrease in credit limit to determine which were significantly different provided the *direction* in shift (i.e. feature differentiation between segments following a credit limit decrease versus segments following an increase), **B** deriving features for confirmatory models based on the ANOVA testing, **C** developing the *ex-ante* structure by forming the target variable (i.e. what behavior, defined by a feature, constitutes an effective or ineffective credit limit change), and **D** deriving RFM-related features for predicting if credit changes would be effective using Kendall correlation, learning vector quantization, and principal component analysis.

(A) ANOVA Test Segmented Features: Each instance in the data structure is an aggregation of features characterized by the shift in credit limit prior to the aggregation (shift: -1 or 1; decrease or increase). I thus began feature engineering by testing the statistical significance of base features related to R,F, and M at shift $= 1$ versus at shift $= -1$. Features that are significantly different between shifts provide insight as to what further features will be useful in creating models. For example, a feature that does not have a significantly different distribution at shift$= 1$ versus shift$= -1$ will likely not be useful because they are not affected or differentiated between shifts.

(B) Creating *ex-post* features: Based on the significant features in **A**, I created features for experiments 1 and 2. These features were largely aggregated features, i.e. with each new sales instance, the features are updated to reflect the cumulative behavior regardless of how many times a customer made a purchase. This creates a generalizable structure to compare companies with different behav-

ior. For example, a company that makes 32 purchases after an increase in credit limit can be compared to a company that makes 5 purchases following an increase in credit limit because the features are aggregations (e.g. in simplest form, the mean of some variable of time). After removing those customers who are likely to have churned (`last_sale_diff`$>$ 100 days), the average number of days between the time of prediction and the last day of feature aggregation (i.e. between B and C in figure 4.5) is 9.5 days. Thus, these models can suggest if a credit limit will be decreased or increased 1.5 weeks earlier than current human deliberation.

(C) Creating Target Variable for Predictive Models: The predictive models (experiments 3 and 4) demonstrate two examples of how segments can be used to create recommendations based on desired outcomes. The target variables were formed by using the aggregated feature at the next time period (i.e. D in figure 4.5). The target variables, y for the predictive classification models are shown on the right side of figure 4.6. The following experiments are examples of how the customer can create customizable flag systems to augment their decision making. For a credit decrease or increase, these models will predcit if the decision will lead to desired/target behavior. The classifier rationale for desired behavior in experiments 3 and 4 is presented in figure 4.6. The description of target variables is as follows:

- **Experiment 3:** `Util` (utilization) is the amount of sales made during the segment over the total allocated credit during that segment. If the credit limit decreases, we assume the customer is not utilizing enough of their credit. Thus, if a customer's credit is lowered, it is desirable/positive that they spend at least as much as before or in proportion to the decrease which would correspond to an *increased* `Util`. Conversely, if the utilization decreases following a credit limit decrease, the customer is spending less than they were with higher credit which suggests their credit should have been lowered even further. This is undesirable behavior.

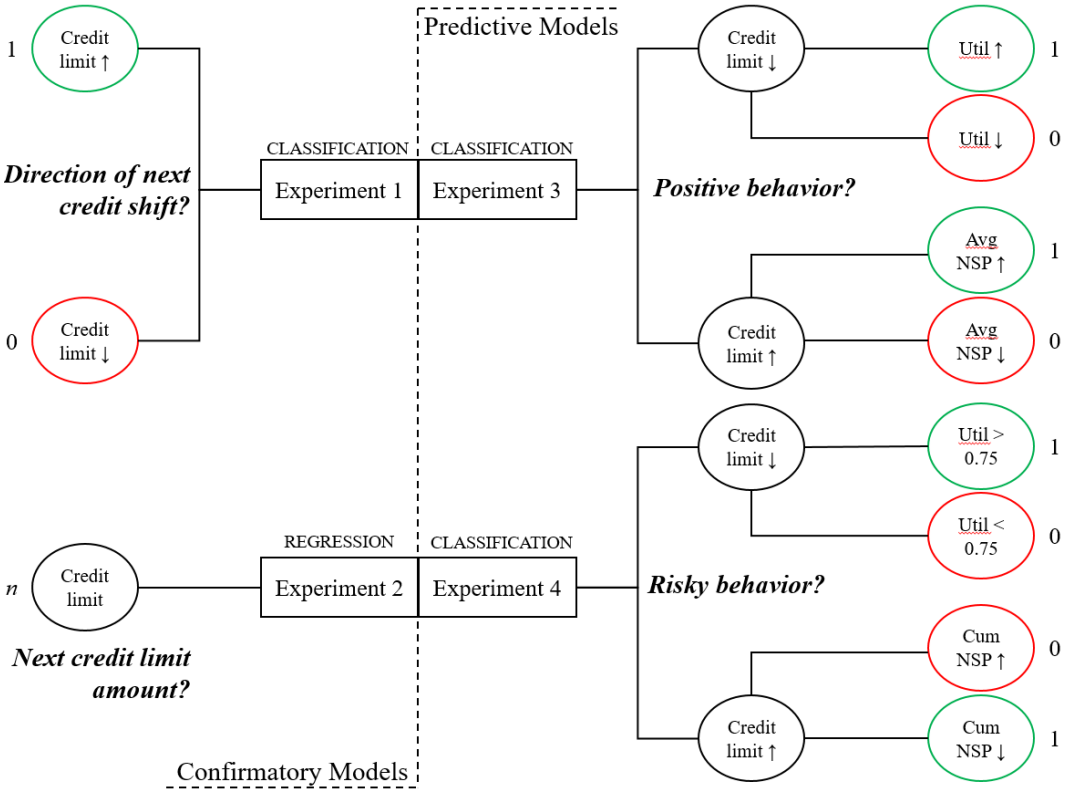
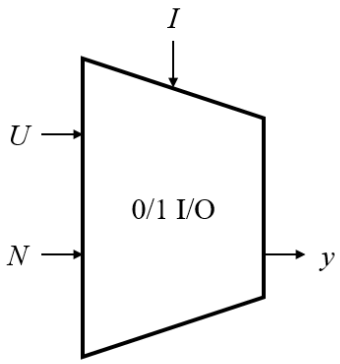


Figure 4.6: Formulation of Four Experiments

- **Experiment 3:** `Avg_NSP` (average net sales price) is the average revenue per sale made during the segment. An increase in credit limit is warranted when a customer will begin using more credit and thus spending more on average; a higher `Avg_NSP` following an increase in credits is thus desirable behavior. Conversely, decreased or maintained average sale amount is considered a negative behavior because the credit limit change did not incite increased sales activity.

- **Experiment 4:** Although an increase in utilization is expected and considered "positive" behavior in experiment 3, over-utilization (as defined by `util`> 0.75) can be considered risky because a customer may not have purchased as much since their total was approaching their allocated credit. This is undesirable/risky behavior. Also if a customer spends over the allocated credit amount, their accounts payable will exceed what has been budgeted which will cause the vendor (the client) money.

- **Experiment 4:** `Cum_NSP` (cumulative net sale price) is the total sales made during the segment. Following a credit limit increase, it is risky if a customer spends less in total because that means the credit was poorly distributed without inciting increases sales activity.



Experiment 3

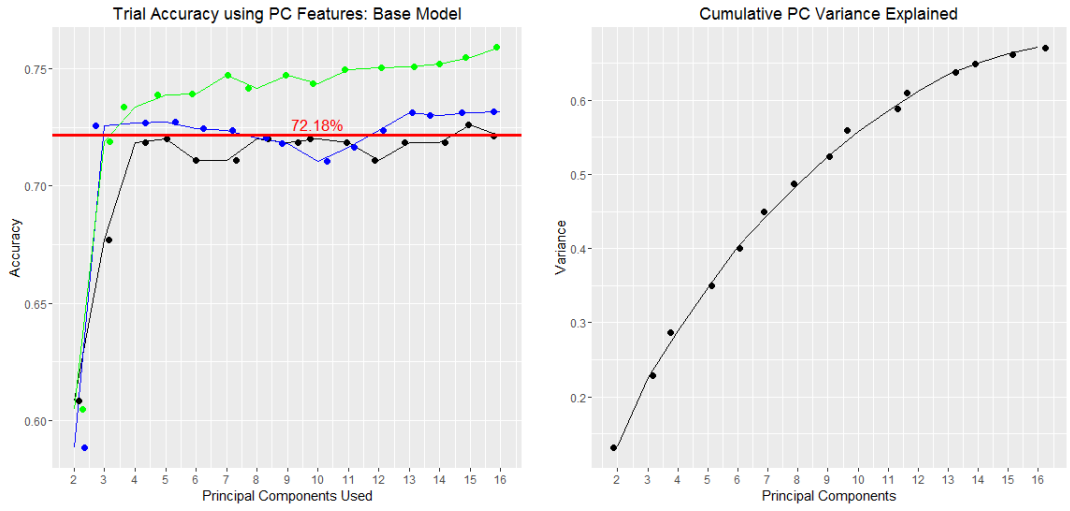
I	N	U	y
1	0	0	0
1	0	1	0
1	1	0	1
1	1	1	1
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1

Experiment 4

I	N	U	y
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	0
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	1

Figure 4.7: Mux Gate for Predictive Models Target Variables: Figure presents the conditional logic of how a target variable is selected for the predictive models. *I*,*N*,*U*,*y* respectively correspond to whether there was an increase in credit limit, the monetary feature (`Avg_NSP` and `Cum_NSP`), the utilization feature, and the binary target. Reference 4.6 for binary classes.

(D) Selecting *Ex-ante* Features: I performed principal component analysis on my raw features used for experiments 3 and 4. Principal components were found to not be as effective as training on raw features for three reasons: the out of bag accuracy was substantially worse using principal components (figure 4.8 **(a)**); it would take 10 principal components to explain 50% of the variance in the model, suggesting each raw feature holds a large amount of interesting information/variance (figure 4.8 **(b)**); and reducing the features to principal components makes the model less understandable and intuitive, especially when we want to know what features are important in explaining the target variable.



(a) **(b)**

Figure 4.8: Feature Selection with PCA: (a) Accuracy of random forest (green), C5.0 (blue) , and base generalized linear model (black) using 2-15 principal components. Red line indicates base model OOB accuracy *using raw features*; OOB accuracy does not exceed 75% with principal components which is more than 15% worse than results using raw features. Scree plot **(b)** indicates there is no logical principal component choice/elbow.

Correlation was then utilized on the raw features to isolate and remove highly correlated features since PCA was not used. Kendall rank correlation was utilized because it is versatile and robust for non-parametric correlation testing [73]. Naturally, RFM features were highly correlated with each other.

$$\tau = \frac{\text{number of concordant pairs} - \text{number of discordant pairs}}{n(n-1)1/2} \tag{4.4}$$

Learning vector quantization (LVQ) was then utilized to further substantiate which features would be useful in classification. LVQ introduces weighted factors of the feature inputs which are adapted to increase classification accuracy via feature pruning [74]. Formally, given a training set $\underset{i=1,\ldots,m}{X} = \{(x_i, y_i) \subset R^n \times \{1,\ldots,C\}\}$ where C denotes classes, the algorithm selects codebooks (i.e a finite amount of feature vectors from each class), $\{w_1,\ldots,w_K\}$. A class label C_i is attached to w_i iff w_i belongs to the correct class. The algorithm attempts to have each codebook explain each class as accurately as possible. The receptive codebook is defined by

$R_i = \{x \in \mathrm{X} | \forall w_j (j \neq i \rightarrow |x - w_i| < |x - w_j|)\}$. LVQ identifies potentially useful features for training classification models.

4.3.3 Machine Learning

Supervised Learning: The result of feature engineering was four separate data sets corresponding to the four experiments which were used for training. I used 67% of the data for training and cross-validation and the remainder for testing. Cross-validation randomly partitions the training data into K sub-samples of equal size and is a common method for reducing over training. Each 10-fold cross-validation was repeated ten times. Linear regression and logistic regression were used as baseline models for regression and classification experiments. Random forest, C5.0, and other CART models were then used to create the optimal, generalizable mapping of features,X to binary class, y.

Cloud Computing: The majority of models were made using `R.3.3.1` on Amazon AWS EC2 cloud computing service instances because the grid searches for optimal parameters (i.e. `mtry` & `boosts`)[7] were computationally expensive. Additionally, cloud computing enabled experiments to be run in parallel which further sped up the modelling process.

4.4 Chapter Summary

After binning the data into RFM quintiles and then clustering using K-means, I assess clustering using a high dimensional heat map and transition matrix cluster rank profiles. Then for credit limit analysis, I created segmented data structures characterized by changes in credit limit. I validated these segments using significance testing before creating the features for the *ex-post* configuration. Significant features from the *ex-post* configuration were used for the *ex-ante* configuration. Several other methods were used for feature selection including PCA which was deemed not useful for this dissertation.

[7] `mtry`: the number of variables randomly sampled as candidates at each split. The default values are different for classification ($\sqrt{p}$ where p is number of variables in x) and regression $(p/3)$[75];`boosts`: the number of boosting iterations.

Chapter 5

Customer Segmentation Results

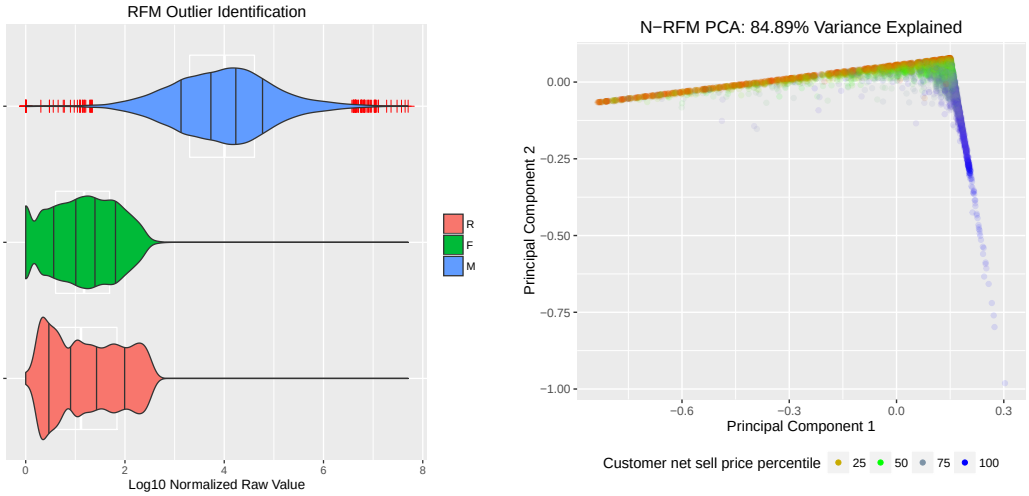
The following chapter presents the results and discussion for scoring customers on RFM, selecting an appropriate number of clusters, clustering the customers using K-means, and assessing the final clustering results using two unique assessment and validation techniques.

5.1 Characterizing Customers by RFM

Prior to standardizing the RFM features on an ordinal scale, the distributions of the raw variables that comprise RFM (i.e. `last_date_diff` (R), `sales_count` (F), and `total_sales` (M)) were inspected. The `total_sales` distribution exhibits a significant power law (see appendix B) where 937 customer have spent over £100k, 182 customers over £1MM, and 15 over £10MM. Figure 5.1 (a) confirms this behavior on a log scale[1]. The significance of M is further substantiated by Figure 5.1 (b); the two principal components represented as a perpindicular angle have a significant relationship with the amount spent/`net.sell.price`, thus suggesting that the defining factor in segmenting by RFM is the monetary value of the customer.

The aforementioned importance of `total_sales` (M) affects the quintile normalization and interpretation of the monetary value categorization; table 5.1's top quintile for M does not intuitively reflect the large range of the top 20% of customers. The discrepancy between a 5 and 4 M quintile score is much greater than a

[1] Although the range for dollars spent (M) would naturally be expected to be much greater than a count (F) or difference between days (R), the power law behavior is still much more significant for `total_sales`.



(a) Raw RFM variables violin plot split into quintiles. Red lines correspond to customers outside 1.5xIQR cutoff

(b) Raw RFM variables PCA colored by M reveal significant relationship between explained variance and `total_sales` (M)

Figure 5.1: Initial Inspection of Raw RFM Variables

5 and 4 R or F score. As noted in chapter 4 section 4.2.1, this also leads to a large number of customers being categorized on the extreme ends where the distribution exhibits a fat tail. Consequently, roughly 15% of customers fall into the two most extreme RFM categories 111 and 555. Out of 125 possible combinations of RFM scores (5 x 5 x 5), 109 are present in the data. This is because it is rare to observe a customer that purchases often while simultaneously spending a relatively small amount.

Table 5.1: Quintile Standardization Scales of Client's Customers using RFM

Quintile	(R) Days since last sale	(F) Number of sales	(M) Total revenue generated
1	>=97	<=3	<=£1,287
2	26-97	3-9	£1,287 - £5,029
3	9-26	9-23	£5,029 - £16,499
4	3-9	23-62	£16,499 - £55,439
5	<=3	>=62	>=£55,439

We may consider the first recency quintile as customers who have likely churned since they have not made a sale in over three months. The main benefit of segmenting by RFM quintiles rather than on raw features scores is it creates logical buckets of customers that are equally weighted[2]. These 1-5 quintile scores are

[2]See appendix E for my function which substantiates this statement.

used as features for *K*-means clustering and supervised methods in chapter 6.

5.2 Cluster Validation: Choosing *K*

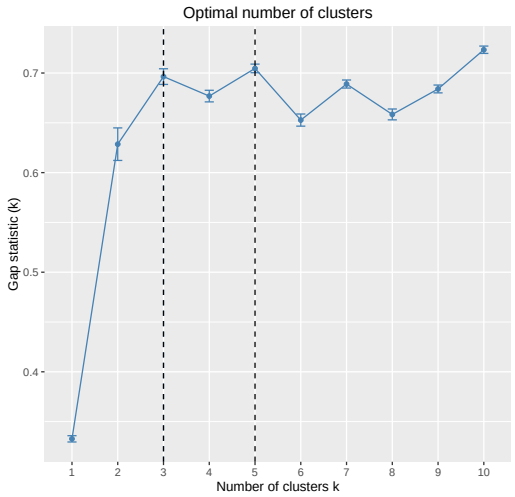
The following section presents the results for traditional *K*-means cluster validation. I present how the data is perhaps best separated into high value and low value segments, however this low *K* is not conducive for effective differentiation; thus, I select the highest *K* as suggested by these evaluation methods before assessing the clustering in 5.3.

5.2.1 Traditional Validation

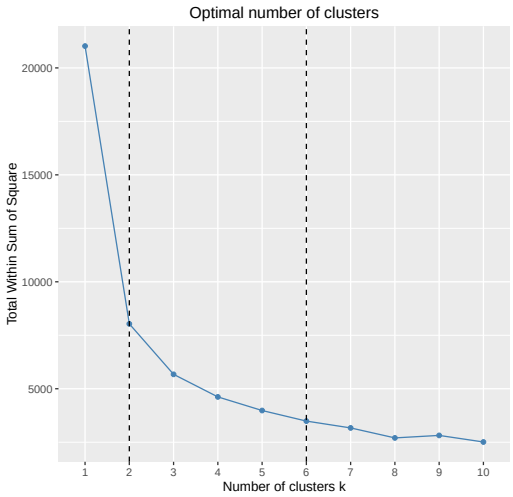
The gap statistic, within clusters sum of squared error, and average silhouette width were utilized to choose *K*. The results from the gap statistic test indicate the largest initial gap between the null distribution and the euclidean distance between clusters occurs when *K=3* or *K=5*. The elbow method, the within clusters sum of squares elbow, presents a clear elbow at *K=2*. This suggests that the data is best separated into two groups. Indeed, the parallel with the figure 5.1 PCA visualization is difficult to disregard; the temporal features R and F do not seem to contribute a significant amount of information for the algorithm to create clusters. The silhouette width also suggests *K=2* is optimal. The homogeneity of the clusters at *K=2* is 0.5 which is moderately strong, however it drops off after this point.

A small value of *K* is indicative that the data is best partitioned into two groups which are the "high value" (i.e. high RFM) and "low value" (i.e. low RFM, potential churn, one-timers). This is confirmed in section 5.2.2 where the cluster centroid RFM profiles at *K=2* are 111 and 555. A low *K* score also suggests the data is not globular in three dimensions (three dimensions: R,F, and M). Previously cited research using RFM for *K*-means clustering did not indicate a low *K* was optimal. The large Pareto distributions in this particular use-case could be affecting the partitioning of the algorithm which was not present in the previously cited papers. Indeed, this may be because many applications of RFM are in a B2C context where individual customer purchases are small ($M_{B2C} << M_{B2B}$), creating a more normal

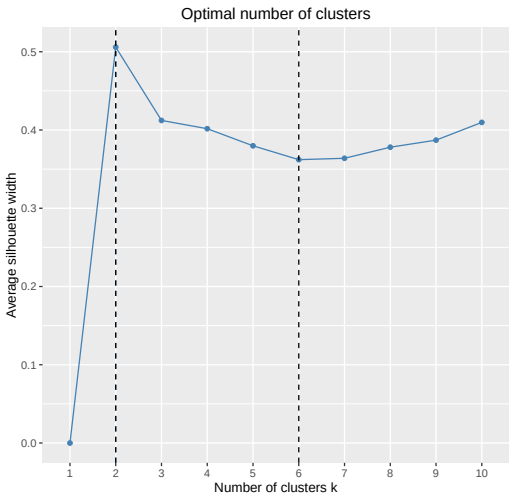
distribution of sales which consequently creates a less atypical separation between clusters with a larger *K*.



(a) Gap Statistic



(b) Within Clusters Sum of Squared Error



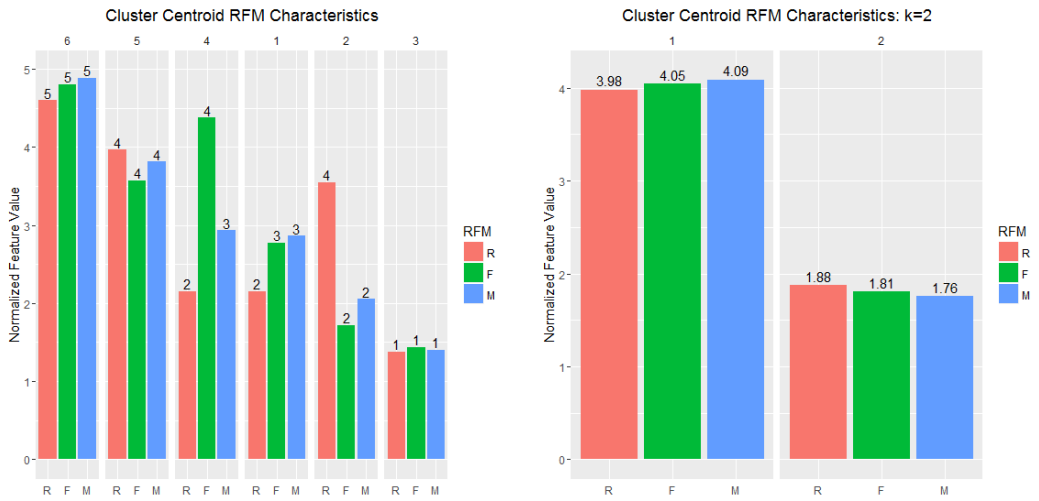
(c) Average Silhouette Width

Figure 5.2: Traditional *K*-means Validation Results: Validation using the gap statistic, WCSS, and silhouette width indicates the data is linearly separable best into 2-3 clusters.

5.2.2 Cluster Choice

Based on the previous validation results, I deemed a range of *K=2* to *K=6* to be appropriate for clustering. Figure 5.3 (b) confirms the previously observed phenomenon: partitioning the data into two clusters creates a high value and low value group. This is not useful for customer segmentation as a means of differentiation.

The choice between *K=2* and *K=6* stems from a common machine learning trade off: over-fitting with unique clusters versus under-fitting with highly generalized clusters (in the most extreme case, a single cluster). For prediction purposes in *supervised* models, employing validation, a cost parameter, or other techniques would be utilized to create the optimal balance in the presence of a model predisposed to over-fitting. However, the use of unsupervised clustering in application of this dissertation is to retrospectively describe past behavior rather than predict future behavior[3]. Thus, I favor creating distinguishable clusters at the expense of slightly over-fitting the model. *K=6* is chosen as the final cluster number because it creates distinguishable customer groups. These clusters are used as features in training supervised models to predict customer behavior in response to credit limit changes in 6.



(a) Upper Bound: *K*=6 **(b)** Lower Bound: *K*=2

Figure 5.3: Choosing *K* Range Comparison

[3]Note: However using clustering on a monthly scale in 5.2.3, my findings suggest customer cluster rank membership is relatively stable over time. Thus, the client may have good reason to assume a customer belonging to a certain cluster rank (e.g. $\{\uparrow \mathrm{R} \uparrow \mathrm{F} \uparrow \mathrm{M}\} = \text{Rank 1}; \{\downarrow \mathrm{R} \uparrow \mathrm{F} \uparrow \mathrm{M}\} \,|\, \{\uparrow \mathrm{R} \downarrow \mathrm{F} \uparrow \mathrm{M}\} \,|\, \{\uparrow \mathrm{R} \uparrow \mathrm{F} \downarrow \mathrm{M}\} = \text{Rank 2}$) will not deviate significantly from that profile from month to month.

5.2.3 Cluster Profiles

Table 5.2: Cluster Centroid Profiles

Clus	R	F	M	Rank	Size	Perc	Last day	Day count	Sales (£)
1	2.2	2.9	2.8	4	859	12%	-23	20	12,776
2	3.6	2.1	1.7	5	849	12%	-4	10	3,583
3	1.4	1.4	1.4	6	1869	27%	-61	5	2,410
4	2.2	4.4	2.9	3	527	8%	-23	146	13,528
5	4.0	3.8	3.6	2	1432	20%	-3	51	31,483
6	4.6	4.9	4.8	1	1472	21%	-2	212	583,914

Table 5.2 shows the centroid profiles for the clustering results from figure 5.3 a. `Clus` refers to the cluster ID. `RFM` refer to the centroid positions on the RFM quintile scales; `Last day, Day count` and `Sales` refer to the raw RFM value associated with the corresponding quintile. The cluster `Rank` is the unweighted CLV value as determined by equation 3.1 in chapter 3.

We observe with *K=6* that there is clear differentiation between clusters as seen in Figure 5.3. The high and low value clusters still exist (clusters 6 and 3), however there is also a high frequency low recency group (cluster 4) which may contain recently unfulfilled or displeased customers who as a result, have not made a purchase in 1 to 3 months. Customers in this cluster may have a higher propensity to churn. Additionally, cluster 2 contains relatively new customers on average; they have not made many purchases (↓F) or spent a large amount (↓M), but they have made a purchase within the last week (↑R). Customers in this cluster are perhaps suitable candidates for loyalty promotions. Cluster 3 contains customers who have likely churned (↓R) and were not previously active (↓F↓M). Cluster 3 also contains 27% of the total customer base which suggest that either the cluster contains a wide breadth of customer types that is centralized around ↓RFM or the seasonality of purchasing patterns means a significant number of customers last made a purchase around the holidays since the `last_day_diff` is defined as the last purchase from 01/05/16. See appendix C for further figures pertaining to the customer segments.

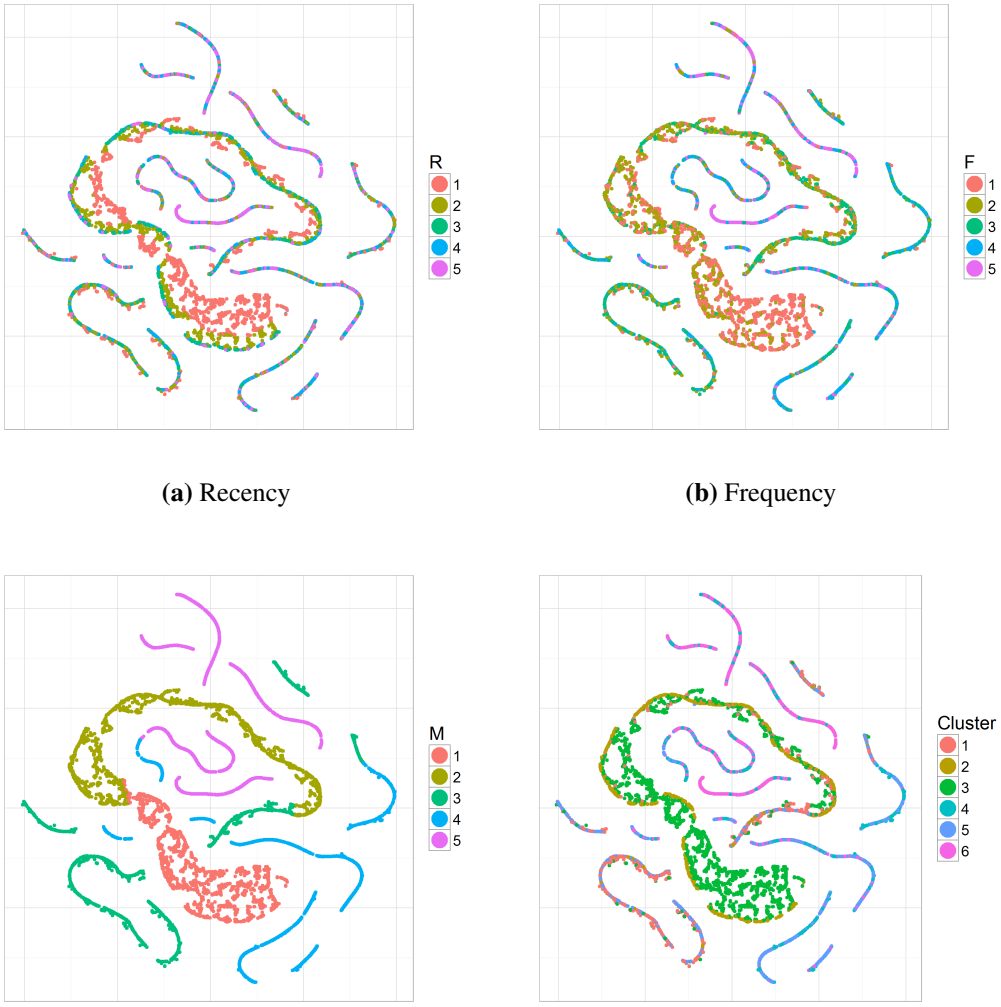
5.3 Heuristic Cluster Evaluation

Using the 6 clusters determined by section 5.2, I next verify the importance of monetary value using t-SNE visualization. Following this interpretation, I use my function to visualize the cluster heterogeneity between clusters and customer homogeneity within clusters using a heat map visualization at the instance level, the results of which verify the significance of the clustering. The final section reveals that clustering on RFM is stable over time (i.e. a customer can be expected to belong to the same cluster rank in the next period).

5.3.1 t-SNE Cluster Structure

The t-SNE visualizations in figure 5.4 support the previously suspected importance of the monetary value feature in characterizing clusters. The snake-like segments represent similar groupings of customers distinguished by the t-SNE algorithm. Each segment is colored by the RFM quintile of the corresponding feature. The segments colored by recency and frequency, although distinguishable by a majority quintile rank, are much less homogeneous than the segments colored by monetary value. However each segment colored by monetary value is exclusively represented by a single quintile rank. Indeed, the homogeneity using monetary value quintiles is more pronounced than when the segments are colored by cluster, which suggests that the amount a customer spends (M) may be a more robust metric for segmenting customers than temporal features such as recency and frequency.

Additionally, the clear separation in t-SNE segments suggests that the data is linearly separable in a high dimensional feature space. We can thus use this separation as an indicator that partitioning clustering such as K-means is valid as an alternative to hierarchically or fuzzy clustering methods. Nonetheless, it is important to note that this is a heuristic observation of which conclusions can only be drawn with supporting evidence from other objective methods.



(a) Recency

(b) Frequency

(c) Monetary Value

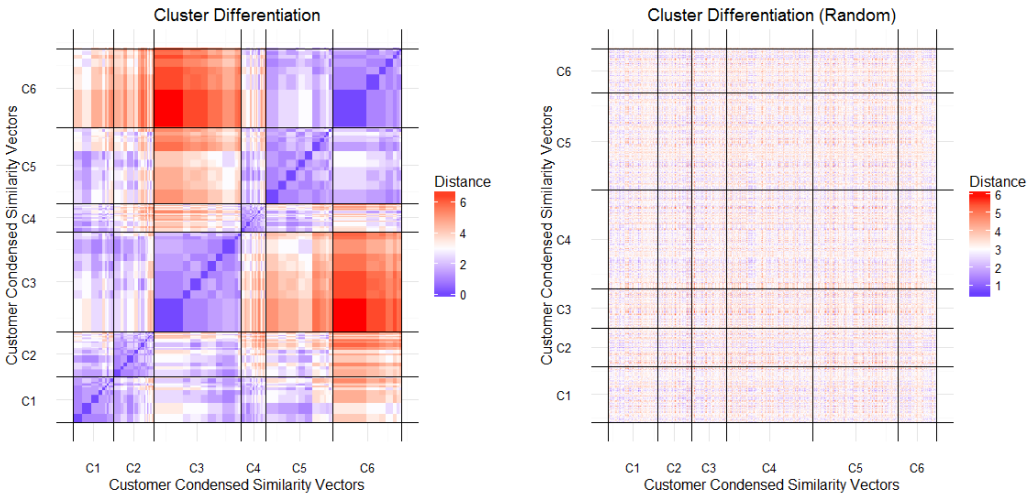
(d) Clusters

Figure 5.4: Clustering t-SNE Evaluation

5.3.2 Visualizing Cluster Similarity

Although the partitioning of the clustering is questionable given the choice of K and the importance of M over R and F, visualizing the distance between all customers reveals specifically *which* clusters are less stable and questionable. A positive takeaway from this visualization is that each cluster is more similar to itself (dark blue diagonal) across the gradient than other clusters, suggesting each cluster is representative of a *unique* combination of RFM not present in other clusters. Additionally, the *intra-cluster* homogeneity seems strong; the gradient of customers

for each cluster is stable within clusters which indicates that customers in each unique cluster exhibit similar RFM behavior. The high value and low value clusters (6 and 3) have the largest distance from the other clusters as seen in figure 5.5 a. Distinguishing homogeneity is even more pronounced with an ordered diagonal within the clusters as demonstrated in appendix E.



(a) Cluster Distance Matrix **(b)** Cluster Distance Randomized

Figure 5.5: Cluster Heatmap Evaluation

The method is useful for simultaneously inspecting inter-cluster heterogeneity and intra-cluster homogeneity since each instance of the gradient corresponds to an individual customer; both the cluster distances and customers within the cluster (the gradient) can be visualized simultaneously to evaluate the clustering. See appendix C for further details.

5.3.3 Heuristic Cluster Temporal Integrity

Predicting Cluster Profiles: Predicting cluster *profiles* over time is not a trivial forecasting problem because there is no measure to assert "this cluster in time 1 is the same cluster in time 2". Clusters are define dby their feature profile which means clusters with similar profiles in different periods can be considered the same cluster. Prediction using K-means usually implies fitting new instances to the established clusters [76] rather than predicting what the cluster profile will be in the future. Several researchers have specifically used K-means clustering in

a similar context to my dissertation application of the method; using RFM, they characterized customer groups into clusters using RFM as features to cluster on [2][3]. The studies aimed to predict the cluster *profiles* over time by characterizing each cluster by a single metric representing CLV, ranking the clusters by this metric, and plotting the metric using auto-regressive time-series analysis. I offer and use an alternative method where I construct a graph from customer cluster rank membership over 11 months from a transition matrix of cluster profile ranks.

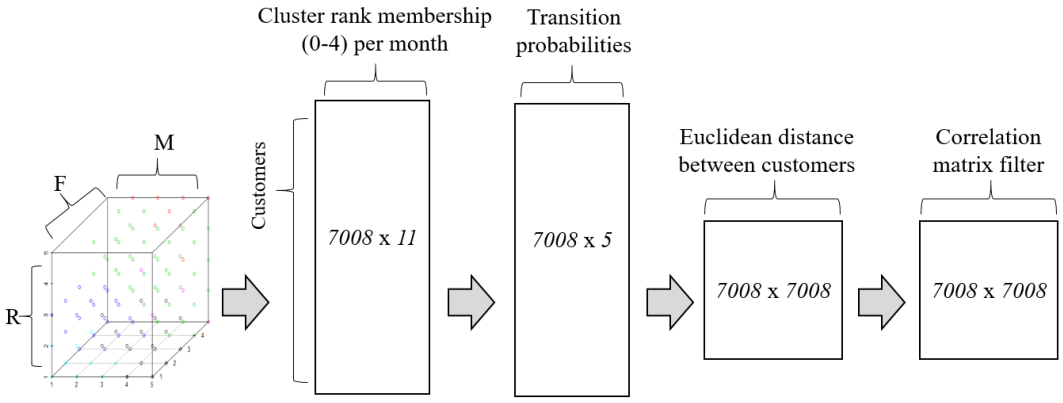
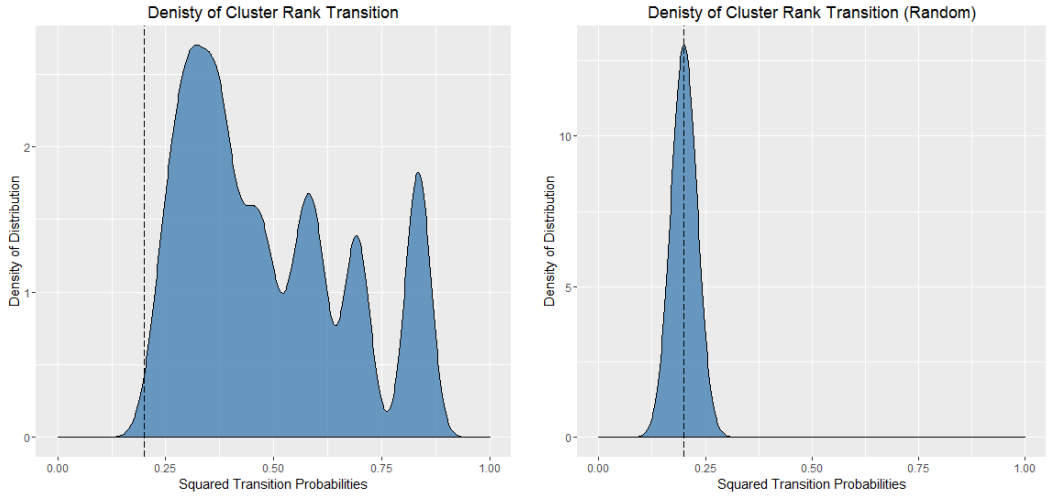


Figure 5.6: Temporal Cluster Data Structure Flow: K-means cluster of RFM produces ranks for each cluster which are assigned to customers in that cluster for each month. A transition matrix is created from cluster rank transition probabilities. The distance between the transition vectors is used to construct a graph. A correlation cut off of 95% is used to draw the edges, ensuring similar customers at each month are clustered together.

Formulation: The data is segmented by month creating 11 segment, a dataset for each month. RFM scores are created for each month for each customer present in the segment. Each segment is then clustered using K-means clustering with k=4[4]. Clusters are ranked by the score of their centroid cumulative RFM scores (i.e. $\text{CLV} = w_R R_t + w_F F_t + w_M M_t$) and these ranks are assigned to the customers/instances present in the clusters. The resulting transition matrix has a row for each customer and a cluster rank 0 to 4 per each segment column, where 0 means not present in that period and 4 is the top ranked cluster for that period. A distance matrix is then constructed from the cluster rank profiles across the 11

[4] K = 4 was chosen by WCSS plot and because clusters were used to rank transferring. It is easier to understand a 4-rank cluster transfer where K=4 than a K=6 cluster transfer, especially since there are only 11 segments of transfer. Similar transfer profiles are thus more easily assessed

segments for each customer. A correlation of the matrix is used to create edges



(a) 11 Month Transition Density **(b)** Transition Density Randomized

Figure 5.7: Cluster Transition Probability Density (a) reveals the cluster rank transitions of customers is not random. The x-intercept indicates the central tendency of randomly clustering data. A customer's rank appears relatively stable over time, suggesting RFM scoring may be useful for predictive modelling.

where an edge is drawn between vertices with 95% correlated vectors. Clustering behavior indicates that a customer's cluster rank is stable over time; customers are normally in a cluster of customers exhibiting a similar trend in cluster rank membership. The resulting square adjacency matrix is made into a graph. Using network visualization program Gephi, the weighted undirected graph is visualized in appendix section C.

5.4 Chapter Summary

Traditionally, RFM visualization has focused on how to leverage the RFM groups (i.e. 125 groups; 111 to 555) for marketing or other business-related purposes [77]. This paper takes these visualizations one step further by clustering the segments using unsupervised learning (K-means) and then visualizing the *clustering* rather than the individual RFM segments. These techniques can be used to assess the intra-cluster homogeneity, inter-cluster heterogeneity, and customer cluster membership over time. The following clusters can be used to identify potentially churned customers; cater marketing promotions based on the category of the segment, how new a customer is, how often a customer purchases, or how little or much a customer spends; and the clusters can be used as effective features to train predictive models as I demonstrate in the subsequent chapter.

Chapter 6

Credit Limit Analysis Results

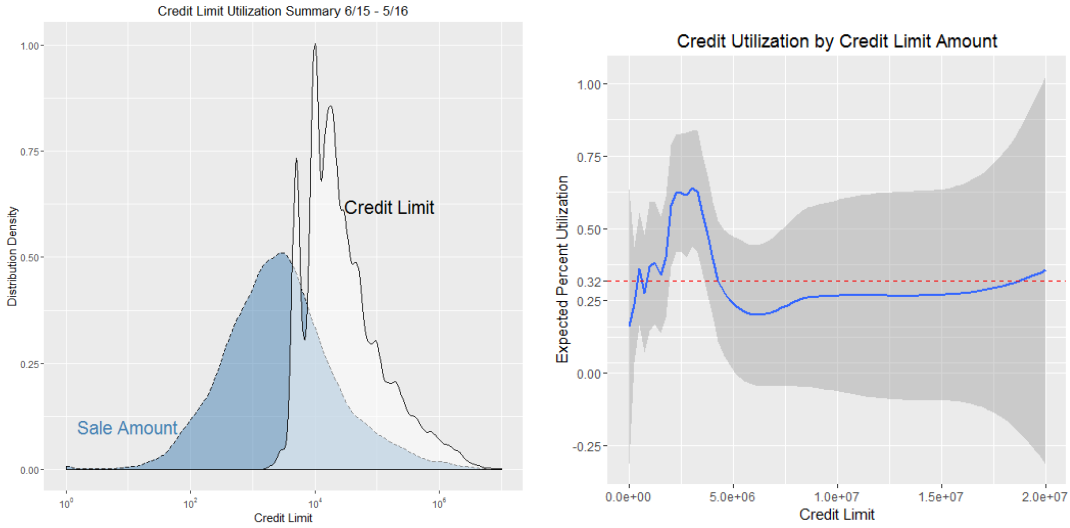
The following chapter begins by discussing the relationship between the client's customers' credit limits and the amount they spent. Section 6.2 presents which features are important and how they were found. The results for the final confirmatory (experiments 1 and 2) and predictive models (3 and 4) are presented in 6.3, highlighting the potential for supervised learning to augment current credit limit deliberation.

6.1 Credit Summary

There is log normal behavior in customer monthly sales which is complimented by a right skewed distribution in credit limits as shown in figure 6.1 a. A minimum threshold of credit is located near the center of the sales distribution. The gap between credit limit and realized sales represents the risk-mitigation zone, where excess credit is granted to cover the potential risk of a customer going bankrupt or defaulting on their payments.

The expected utilization is a measure of the percent of the customer's credit that was utilized as revenue ($1.0 \rightarrow$ Revenue = Credit Limit). When recalculating this value using monthly revenue as a percent of sales, the percent utilization is around 30% lower. Expected utilization is between 0.19 and 0.32 on average. Roughly 5% of customers each month utilize over 75% of their credit, indicating that very few customers take full advantage of their credit. This is interesting given that credit limits are normally increased by request of the customer to cover large upcoming

sales. Such increases may not be completley necessary. On the other hand, this figure may indicate efficient credit allocation and risk mitigation since the majority of customers do not go over their allocated credit or reach a point where they are unable to finance further purchases.



(a) (b)

Figure 6.1: Credit Limit Summary Activity

Table 6.1: Credit Behavior for 11 Months (Excluding November)

	Year	Mo	Credit Limit (MM)	Revenue (MM)	Avg Util	Util> .75	Util< .25
1	2015	Jun	£451	£99	0.22	0.05	0.71
2	2015	Jul	£453	£103	0.23	0.06	0.70
3	2015	Aug	£460	£78	0.17	0.04	0.75
4	2015	Sep	£454	£95	0.21	0.05	0.71
5	2015	Oct	£912	£99	0.11	0.01	0.90
6	2015	Dec	£459	£100	0.22	0.04	0.75
7	2016	Jan	£473	£103	0.22	0.04	0.72
8	2016	Feb	£477	£88	0.18	0.04	0.74
9	2016	Mar	£479	£106	0.22	0.05	0.71
10	2016	Apr	£988	£96	0.10	0.01	0.91
		Average:	*£561*	*£97*	*0.19*	*0.04*	*0.76*

Credit is typically utilized above the expected utilization rate until around £1MM in sales (see figure 6.1 b). This sudden decrease in utilization is largely due to the fact that the customers who spend more also pose the highest risk in terms of magnitude of potential loss. Indeed, in figure 6.1 b the smoothed fit (shaded grey area), which represents an expected utilization for a customer given their credit

limit, is significantly more variable for larger customers. For this reason, it is important to maintain a large gap between the credit limit threshold and the amount spent, which is done by hedging the risk of large customers with even larger (relatively speaking) credit limits. A conclusion we can draw from this is that an excessively high utilization may be considered *risky* behavior. This assumption is carried into experiment 4 in section 6.3.

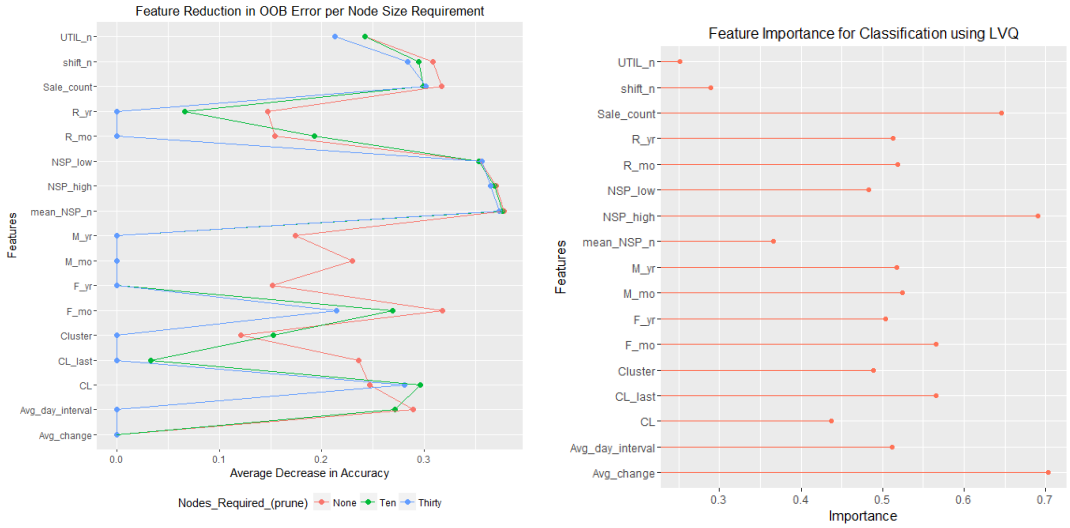
6.2 Feature Importance

Significance of *Segmented* Features[1]: Before modelling, I created data structures by aggregating customer data into segments defined by credit limit changes (see 4.3 for further details). I tested the statistical significance of these features and the results indicate that all features relating to monetary value are significant, and the amount of credit utilized is significant. This means that following an increase or decrease in credit, we can expect differing behavior in how much a customer spends and how much of their credit they utilize. This indicates three important findings: (1) splitting the data into segments characterized by if credit decreased or increased, regardless of the volume, is significant; (2) the amount spent by a customer may be largely influenced by the change in credit limit (which is intuitive), thus further substantiating previous findings that monetary value and features derived from monetary value are key drivers in understanding customer behavior; and (3) we should expect an increase in credit utilization following a credit limit decrease, which thus supports our assumption that an effective (i.e. results in positive behavior) decrease in credit limit will results in *increased* credit utilization (see experiment 3).

Important Features[2] in Predicting Behavior: Features derived from those tested for segmenting significance were used for training confirmatory models (i.e. $y =$*can we predict a credit limit change?*; experiments 1 and 2) and predictive models (i.e. $y =$*given the credit change and other features, can we predict the*

[1] See appendix table D.1 for a description of these features
[2] See appendix tables D.2 and D.3 for a description of these features

outcome in behavior following the change?; experiments 3 and 4). These features were created because they are each understandable to the non-technical user and they provide insight into customer behavior. Not including PCA, Gini coefficients, and Kendall correlation among un-trained features, I utilized two methods to determine the efficacy of *trained* features: (1) the reduction in model accuracy with each feature in 6.2 a and (2) learning vector quantization in 6.2 b. These methods indicate not only which features are important for supervised learning but more generally, which features are key drivers of understanding customer behavior in response to credit limit changes.



(a) Decrease in Accuracy **(b)** Learning Vector Quantization

Figure 6.2: Most Important Features in Explaining Model Predictability

The most useful features in 6.2 a continue to reduce accuracy even when a node minimum is required, indicating that they contribute in making decisive splits in the partitioning algorithms early in the tree. This can be seen because the accuracy decrease does not vary much when a node requirement is imposed. Those features which are also deemed important by LVQ may be considered even more robust indicators of changes in customer behavior. Their descriptions are in appendix 5. Noteworthy takeaways:

- Features related to monetary value (e.g. `NSP_high`) contribute significantly to both the tree partitioning and increasing model accuracy.

- Raw RFM features are more useful than generalized RFM quintile categories (e.g. `Sale_count` vs. `F_yr`) in predictive modelling because they are more specific.
- The `shift_n` (i.e. decrease or increase in credit) contributes significantly, substantiating the efficacy of credit limit shifts as an indicator of behavior.
- The presence or absence of variability in purchasing patterns (e.g. `Avg_change`) can indicate *risky* or *positive* behavior respectively.

6.3 Model Results

The following section includes the results of my four experiments (see section 4.3 for the formulation of these experiments) where were trained using the aforementioned features and a discussion of the most accurate models.

6.3.1 Summary of Models

Table 6.2: Models Results Summary

	Type	Experiment	Y	Model	Optimal Parameter	Test Accuracy
1	confirmatory	1 *(classification)*	Shift	LogitR	-	61.11%
2	confirmatory	1	Shift	C5.0	-	84.02%
3	confirmatory	1	Shift	C5.0	50 boosts	87.75%
4	confirmatory	1	Shift	RF	mtry=3	94.72%
5	confirmatory	2 *(regression)*	CL	LM	-	62.71%
6	confirmatory	2	CL	RRF	mtry=1	82.17%
7	confirmatory	2	CL	RF	mtry=2,	83.49%
8	predictive	3 *(classification)*	EF	LogitR	-	72.18%
9	predictive	3	EF	C5.0	20 boosts	86.88%
10	predictive	3	EF	RRF	mtry = 6	91.31%
11	predictive	3	EF	RRF	mtry = 5	94.58%
12	predictive	3	EF	ensemble	C5.0, RRF	95.11%
13	predictive	3	EF	RF	mtry = 5	96.29%
14	predictive	4 *(classification)*	RS	LogitR	-	85.85%
15	predictive	4	RS	RF	mtry = 6	96.93%
16	predictive	4	RS	RRF	mtry = 5	97.86%
17	predictive	4	RS	C5.0	20 boosts	98.21%

Table 6.2 contains the base (1,5,8, 15) and best (4,7,13,17) performing models for experiments 1-4. Accuracy for regression and classification is calculated on

unseen data using R-squared and the AUC respectively. Based on these models, we can conclude that credit limit changes can be automated (experiment 1) and behavior can be predicted from the change in credit (experiments 3 and 4). A minimum out of samples test accuracy of 90% was self-imposed, thus deeming the second experiment not robust enough for application. For all practical purposes, CART methods Random Forest and C5.0 variations were largely equivalent in creating generalizable predictive models. The next step is thus for the client to clearly define the problem, formulating and specifying their own experiments that can be made into predictive models. These results suggest that segments of behavior prior to credit limit changes are predictive of future behavior.

6.3.2 Models Discussion

Table 6.3: Best Performing Classification Models Summary

	Experiment 1	Experiment 3	Experiment 4
Algorithm	Random Forest	Random Forest	C5.0
Area Under Curve (AUC)	94.72%	96.29%	98.21%
Number of features	12	17	17
Sensitivity: $TPR = \frac{TP}{TP+FN}$	91.02%	92.26%	94.35%
Specificity: $SPC = \frac{TN}{FP+TN}$	81.80%	85.18%	90.77%
Precision: $PPV = \frac{TP}{TP+FP}$	91.38%	89.60%	95.00%
False Positive: $FPR = \frac{FP}{FP+TN}$	18.20%	14.83%	9.23%
False Negative: $FNR = \frac{FN}{FN+TP}$	8.98%	7.74%	5.65%
Accuracy: $ACC = \frac{TP+TN}{P+N}$	88.06%	89.29%	93.10%
$F1 = \frac{2TP}{2TP+FP+FN}$	91.20%	90.91%	94.67%

I will briefly discuss the practical implications of the model performance for these models. The criteria for evaluating a model depends on the decision threshold and sensitivity to mis-classifying a positive instance ($y = 1$) vs. a negative instance ($y = 0$). Since there is no pre-defined cost function provided by the client, a cutoff will not be predefined but rather inferred from that which creates the highest accuracy (see figure 6.3). On the right side of 6.3, I plot the potential cutoff as a function of the overall classification accuracy at that cutoff. The results suggest that these are relatively balanced class problems where the probability weighting of a positive

class is similar to a negative class with roughly a 50:50 weighting. However, at closer inspection of the sensitivity and specificity, each experiment favors the positive class slightly (i.e. sensitivity $>$ specificity). The difference is not large enough to warrant class weightings even though class $y = 1$ in experiment 4 was a dominate class with over 75% of instances.

The left side of 6.3 shows receiver operating characteristic curves (ROC) for each experiment, where sensitivity and $1-$specificity are plotted at each cutoff. For the ROC curve, where the model sensitivity S_n and specificity S_p characterize the axes, the intersection represents the minimum distance D from the optimal ROC (i.e $S_p = 1$, $S_n = 0$) and the actual as represented by the generic distance function $D = \sqrt{(1-S_n)^2 + (1-S_p)^2}$. The area under the ROC curve (AUC) is a metric that captures the classification accuracy. The AUC for each model indicates a strong tendency to select the correct class (i.e. S_n) in around $9/10$ of cases. This holds given that each supervised classification model is a relatively balanced class problem with similarly weighted classes as demonstrated by the suggested cut off onright side of figure 6.3. Each model has over 94% accuracy as determined by the area under the curve (AUC). It is important to note that this does not correspond to an equally high classification accuracy (*ACC*) on the hold-out test set. This is because the AUC represents potential true positive rates at certain cutoffs whereas the classification accuracy is determined by the number of instances correctly classified with the default cutoff of 0.5.

In terms of practical application, the results from experiment 1 suggest a recommendation system can be developed that can inform a business user of what the suggested next credit limit will be. A question that will arise is "what is the use of this recommendation given that a new credit limit is a decision that takes into account the customer's new sales needs, their credit score, and other criteria?" Such a system can learn what an *expected* change would be (reinforcement learning) and thus will not be a recommender system in the traditional application but rather, a flag system that will indicate when proposed behavior deviates from the norm. For example, after deliberation a manager may decide to raise a customer credit limit by

£10,000. The recommender system will either provide data-informed recommendation for ($y = 1$; yes) or against ($y = 0$; no) this decision. It is expected that the choice of the manager and the recommendation system would be congruent because the confirmatory model picks up underlying information of customer behavior (via features) that is taken into account when managers make credit change decisions. The model is trained on past credit limit decisions and resulting behavior. If the model does *not* agree with the manager's decision, this would indicate either a serious exception based on data/information not included in the model or a mistake was made. Under either condition, the system would make the business user reconsider their decision, adding an objective measure to the subject decision-making process.

Experiment 3 considers whether a credit limit change results in positive or negative behavior where negative behavior is decreased credit utilization following a credit decrease or decreased amount spent following an increase. Positive behavior is the opposite. Experiment 4 uses similar criteria to determine if a change will be *risky* (See section 4.3 for more information). The false positive and false negative rates are significantly lower for experiment 4, indicating the criteria used for training the models is more predictable. These criteria were whether or not the customer utilized over 75% of their credit and whether the cumulative amount spent by the customer decreased or increased. The takeaway from these results is not that predicting *risky* behavior is more suitable than predicting *positive* behavior, but rather (since these models use the same features) a simple set of RFM-related features can be used to customize a recommender or flag system that can augment the current credit allocation deliberation processes. These recommendation or flag models will vary in accuracy but based on experiments 3 and 4, we can expect their accuracy to be $> 90\%$. The ultimate purpose of these models is to demonstrate how the client can create deterministic predictive models to enhance their deliberation to change a customer's credit.

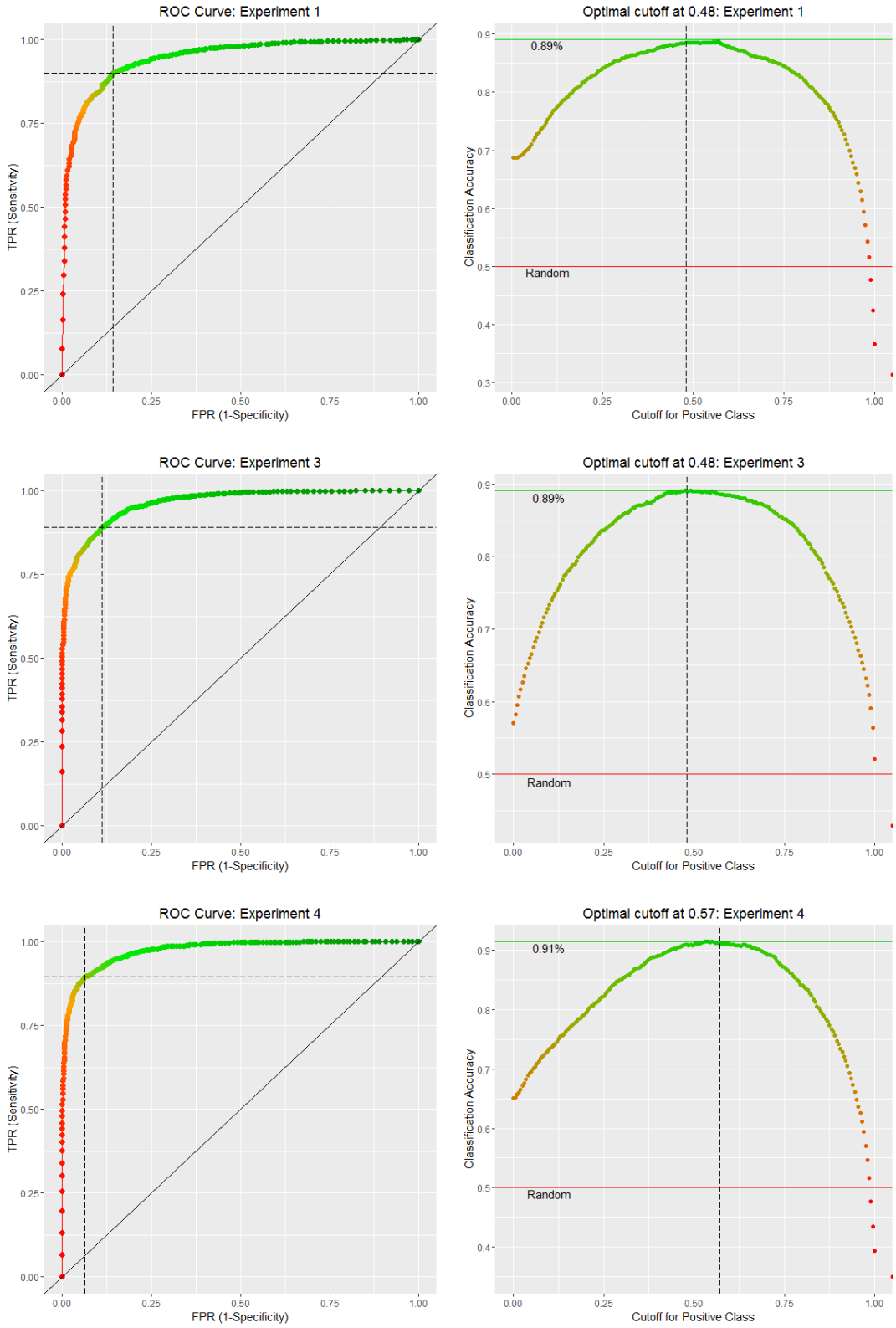


Figure 6.3: Best Random Forest ROC and Balanced Cutoff

6.4 Business Value

The following section is a discussion of the possible business value generated from the confirmatory models (experiments 1 and 2) and predictive models (experiments 3 and 4). Based on the decision maker's past decisions, we can see what an *expected* credit limit is and how this deviates from the new credit limit chosen. This will augment the current process by providing a point of reference based on past decisions. An A/B test can be implemented to determine if such recommendations results in better decisions based on the MOS for increasing and decreasing credit limits. Table 6.4 shows how a predictive model (in this case, experiment 3) can be translated into business value via more efficient capital allocation and distribution.

Table 6.4: Business Value Example from Experiment 3

Predicting if a change will result in positive behavior		
avg monthly credit limit totals	£550,000,000	
avg monthly revenue from customers with lines of credit	£97,000,000	
customers	7,000	
revenue per customer per month	£14,000	
segments characterized by credit decrease	1/3	*from methodology*
segments characterized by credit increase	2/3	*from methodology*
segments resulting in positive behavior	1/2	*from model*
segments resulting in negative behavior	1/2	*from model*

Based on table 6.4, 1/6 of credit limit changes are ineffective because the credit was decreased but the utilization of credit stayed the same or went down (negative behavior), indicating the client should have lowered it even further. Moreover, 1/3 of changes are ineffective because the credit was increased but average amount spent stayed the same or went down (negative behavior), indicating the client should not have increased the customer's credit limit. This indicates that roughly half of the credit allocation (i.e. 1/6 + 1/3 or £225,000,000) is allocated without the intended response in customer behavior. With a recommender system, less credit would be needed because ineffective allocations would often be deemed unnecessary. Although credit should naturally be much larger than the amount spent to account for unforeseen risks and large purchases, even a percentage decrease in the credit pool would free up several million pounds for financing elsewhere.

Chapter 7

Conclusion

The following chapter revisits the project objectives from chapter 1 in context of the results in chapters 5 and 6. Some feedback from both the client and Satalia are provided in section 7.2. The limitations of my work are mentioned in section 7.3. I briefly mention further work that can be completed both for the client and in an academic context. The chapter concludes with a reference to my code repository.

7.1 Project Objectives Recap

The following objectives relate to either the intentions of the client (a) or investigations that arose by nature of pursuing the clients objectives (b). Each requirement from chapter 1 is followed by the key takeaway from the project results.

a) Client Requirements:

- Can current case-by-case deliberation for issuing credit be substituted or augmented by a predictive model?
  Yes. With over 90% accuracy 1.5 weeks prior, we can predict what credit limit will be determined for a customer. Moreso, we can create a robust (>92% ACC.) recommendation or flag system that can determine if a credit limit change will be effective.
- Can we create a generalizable framework to assess and predict any customer's behavior following a credit limit change?
  Yes. By grouping features into segments characterized by a unique credit

limit, we can train predictive models that generalize accross all customer regardless of differences in sales frequency.

b) Academic Investigations:

- Is RFM an appropriate method for segmenting B2B customers?
  In context of this dissertation, the monetary value that the customer generates and derived features from M are most characteristic of the difference between customers. The temporal dimensions do not add significant value to customer segmentation but are still useful for deriving features to train predictive models.
- Is RFM analysis effective for clustering the client's customers given they exhibit power law behavior (which was not present in the literature pertaining to RFM clustering)?
  The presence of power laws affects the normalization method. In this case, the quintile binning creates highly skewed bin representations which hold the tail of the distribution. The data is thus more accurately binned into two dichotomous group, however this does not permit differentiation which is the key prerogative for normalizing in the context of segmentation.
- How can we evaluate the effectiveness of clustering beyond traditional methods? Can we conclude if clustering is useful?
  Through transition matrices and corresponding visualization, we can heuristically verify that a customer is likely to belong to the same cluster profile in the future. Within cluster homogeneity and between cluster heterogeneity can be assessed by extracting metrics from each square of a heat map visualization and associated metrics. We can also assess what normalization method is most appropriate using this technique.
- Are RFM features and clusters useful as input features in creating predictive supervised models? Which features?
  A customer's RFM quintiles are not useful for training predictive mod-

els, however features derived from these RFM attributes are informative. The important features depend on the formulation of the experiment but often include the past credit limit, sales amount variance, credit utilization, and other monetary value-derived features.

7.2 Industry Partner Feedback

The work Derek has undertaken over the last 3 months has been valuable for both Satalia and the client. The abstract problem that Derek has addressed, namely identifying what behavioural traits of a customer would allow us to predict their response to certain stimuli (in this case a change to credit limit), is applicable across a number of different problems faced by both this and other clients. Furthermore the visualizations he has produced, particularly around understanding the distances between and within clusters (referred to as function for visualizing cluster similarity) will be directly applied to a current project. The benefits of his work are far from immaterial.

Data Scientist (Satalia)

From my viewpoint this project is progressing fantastically well and could be an absolute game changer...to sum up, a very exciting project which is out of the blocks and progressing and a role model example of business and IT working together for high business impact.

CEO, UK (client)

The work was very good and certainly gave me a different view of how we can use that type of data to focus the sales and marketing areas of our business. The models used can be difficult to interpret and the underlying mathematical formulas are not within my area of knowledge but Derek's explanations were helpful and enabled understanding.

Director, UK Credit Services (client)

7.3 Limitations

One of the main limitations of this work is that it is not reproducible given that the data used is private. However, the method to create segments characterized by credit limit amounts *is* reproducible for a B2B business with customers that make a significant amount of purchases (i.e. a single purchase would not permit aggregated behavioral features used to train the models). Additionally, K-means clustering using the client's Pareto-distributed RFM features favors the tail of the distributions, which segments clustering best at $K = 2$: a high and low value cluster. However as was mentioned in chapter 5, this does not permit differentiation which is a key prerogative of clustering for segmentation. For this reason, exploration in hierarchical clustering may create more useful clustering because we could compare clusters at different hierarchical levels stemming from the high and low value clusters.

7.4 Further Work

The following section briefly discusses further work that can be done both on the client's end and from an academic perspective.

7.4.1 Client-based

For case-by-case credit limit decisions, the client makes great use of a third party credit agency report that contains forecasted credit scores. Each customer in the segmented data (see chapter 4 section 4.3) has additional information from these forecasted reports that is already used by the client to determine credit. With the addition of this information as features, the models would likely become more robust.

Moreover, the client makes credit decisions often in response to suggestions from the sales team. It would be useful to compare how the following models match-up to the sales team's suggestions for increasing and decreasing credit. To implement this, we would train supervised models where the *target* variable is the credit limit amount (similar to experiments 1 and 2; the confirmatory models) and the features would be behavioral variables (possibly RFM-related) that the sales team uses to recommend credit changes. A highly predictive model in this case

would result in a recommender system that could be used to suggest credit limits to the credit limit team on behalf of the sales team. Such a system would streamline the communication pathway between the sales and credit team. We may also consider what products, types, and product families contribute to the predictability of a supervised model that predicts an effective credit limit change.

On the segmentation part of the analysis, clustering with RFM may not be optimal for segmentation with large B2B customers that exhibit power-law behavior. There are two ways we can tackle this issue: we can make a different cumulative metric aside from RFM, perhaps with a variance-in-sale (RFM-V) which was found to be effective for training models in chapter 6; or, as was mentioned in section 7.3, hierarchical clustering may be considered for clustering. Hierarchical clustering would be more difficult to explain to the business user, but may be more efficacious.

7.4.2 Academic

I plan to publish this work or section of this work as a case study in one of the following journals: *Information Systems and e-Business Management*, *Society for Industrial and Applied Mathematics*, *Journal of Interactive Marketing*, or *Journal of Marketing Research*. There are several angles I could take in such a publication. The RFM clustering can be used as a case study in B2B segmentation for a large multi-national company, perhaps focusing on power law distributions. The RFM segmentation and clustering validation techniques can be highlighted and further explored. The methodology for predicting effective credit limit allocation can be leveraged as a generalizable framework for applying supervised learning to a traditionally ad-hoc decision. I also plan to make the high dimensional heat map method more flexible, integrate it into an interactive heat map API such as `plotly`, and ultimately make it available for public use and improvements.

7.5 Code

The final scripts can be accessed via: https://github.com/lukadw11/Github. `Dissertation_script` contains code for data cleaning and customer segmen-

tation. `Dissertation_script2` and `Dissertation_script3` correspond to the heuristic cluster rank graph and visualization. `Dissertation_script4` contains the processing of credit limit data and the creation of segments for supervised learning experiments. `Caret_models` contains the code for the experiments. `Heuristic_Cluster_Eval` contains the code for high dimensional cluster distance heat map visualization.

7.6 Chapter Summary

This chapter concludes the dissertation by first recapping how I fulfilled the project objectives initially set forth in chapter 1. I then mention some feedback I received from both my indsutry partners. I conclude this chapter with a few notable limitations of the work and further work that can be done both for the client and in pursuit of an academic publication. The final section mentions the link to my code and what is contained in each script.

Appendix A

Computational Details

A.1 Environment

- (*Hardware*): Windows Surface Pro 3
- (*OS*): Microsoft Windows 10 Pro
- (*Processor*): Intel Core i5-4300U CPU @ 1.90GHz, 2501 Mhz, 2 cores
- (*RAM*): 8GB

A.2 Programs

- (*For initial data extraction*): MySQL Workbench 6.3 CE
- (*For visualizations*): Tableau 9.1 and ggplot (R)
- (*For cloud and parallel computing*): Amazon Web Services EC2: m4.xlarge with 4vCPU and 16GiB memory.
- (*For programming*): R v3.3.1 - R.Studio IDE
- (*For network graph visualization*): Gephi 0.9.1
- (*For creating figures*): Microsoft Excel, PowerPoint, and LucidChart
- (*For writing report:*) ShareLaTeX Editor

Appendix B

RFM Distributions

B.1 Assessing Power Laws

Distributions of RFM vraibles were assessed as follows:

- The base variables provided were converted to a usable type and transformed into three RFM features for each customer as follows: Recency is the difference in the max `Calendar.day` of the entire data-set (i.e. the last purchase day) and the max `Calendar.day` for the customer. Frequency is the count of sales instances. Monetary value is the sum of `Net.sell.price`.
- Each feature is plotted on a histogram distribution with 50 bins.
- Using the `poweRlaw` package [78] in R, a continuous power law distribution (`conpl`) and log normal (`conlnorm`) are fitted to the feature vectors. The point where the normal plot deviates from the power law (i.e. where the power law behavior begins), X_{min}, is estimated using the `conpl` fit.
- The α for each feature is estimated from the `conpl` fit. This value characterizes the tail of the power law distribution.
- A Kolmogrov Smirnov goodness of distribution fit test is used to verify if the power law fit is significant. The test statistic, KS compares the fitted distribution with the feature vector. A smaller statistic corresponds to a better fit. Where $P<0.05$, H0 is rejected that the fit of the power law and feature vector are equivalent.

Indeed the stability of the RFM feature curves are not predictable on a Gaussian or Poisson distribution but each seem to exhibit power law behavior. Such distributions are largely confounding for predicting customer behavior in terms of how much they will buy, when, and how often, however the aim of these features in context of this dissertation is to create logical buckets to cluster and train learning models. This is fulfilled with or without a predictable distribution. Nonetheless, when considering the behavior of quintiles$> X_{\text{min}}$ (as seen in B.1 right side), the extent of the variability within those quintiles must taken into account. For instance, central moments such as the mean and median are not robust for the entire distributions of R and M since $\alpha < 2.0$. However an estimate of the moments *prior* to X_{min} may provide a guage of central tendencies of the customers' behavior, but we would still not conclude that these are robust summary metrics.

	R	F	M
X_{min}	3	38	99,870
α	1.47	2.19	1.75
KS	0.136	0.121	0.038
Pval	0.00	0.00	**0.128**

R	F	M
1	*1*	*1*
2	*2*	*2*
3	*3*	*3*
4	*4*	*4*
5	*5*	*5*

Table B.1: RFM Distribution Summary (left) reveals significant power law fit for `Total_sales` (M) since the test statistic is small and the P value indicates a failure to reject of H0 at a 95% confidence interval, suggesting that the total amount of revenue generated per customer lies on a Pareto distribution. The red quintiles (right) correspond to intervals where we may expect highly variable scoring in the tails of the given distributions beyond X_{min} and the green quintiles exhibit more normal behavior.

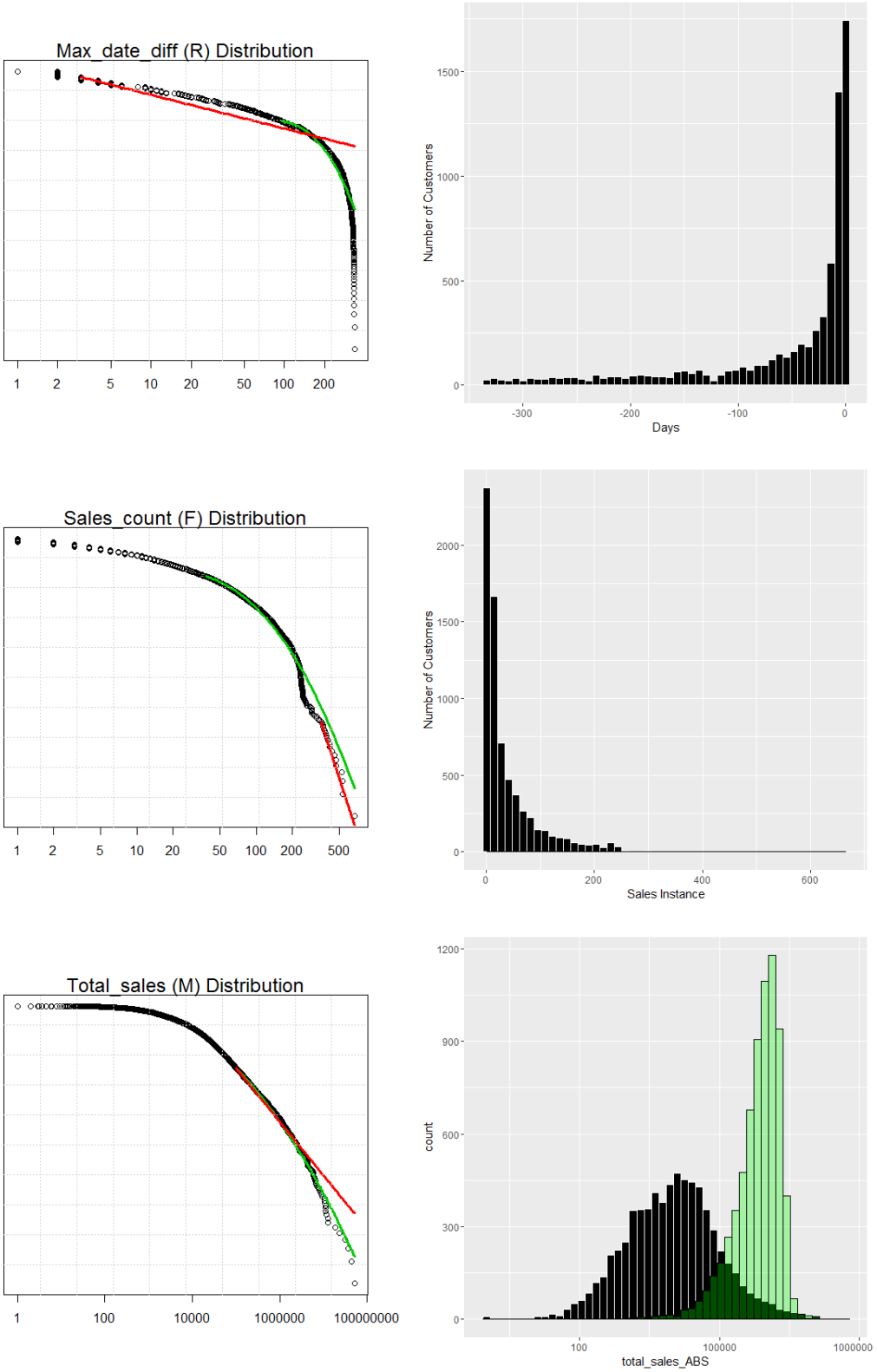


Figure B.1: RFM Pareto Distributions: Exponential dot plot (left) and histogram of raw variables distribution (right) that comprise the features Recency (top), Frequency (middle), and Monetary Value (bottom). The green and red curves correspond to a log-normal fit and exponential power law fit. The M histogram is on a log scale with a fitted normal distribution (skewed towards the mean of the actual Pareto distribution) to highlight the significant of the power law.

B.2 The 80/20 Rule

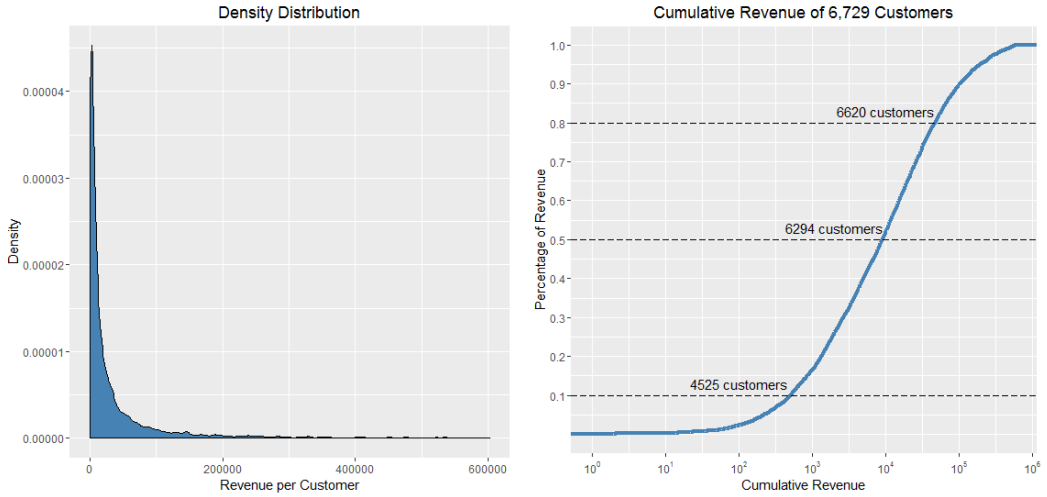


Figure B.2: Pareto Distribution of Customer Revenue

Figure B.2 demonstrates the aforementioned power law behavior for M in reference to the client's customer base. The famous 80/20 rule is significantly present; 20.54% of the client's customers accounted for 80% of the revenue from 06/15 to 05/16, almost perfectly fulfilling the Pareto principle [79]. What the results from the previous section reveal however, is that the tail of this distribution is longer than a typical Pareto distribution fitting the 80/20 rule. These findings are present in the clustering results and significantly affect the customer segmentation in chapter 5. Additionally, it is important to note that the right side of figure B.2 does not visually prove that 20.54% of the customers account for 80% of the sales because the customers are ordered from least to most valuable and the x-axis is on a log scale.

Appendix C

Extra Figures:

Customer Segmentation

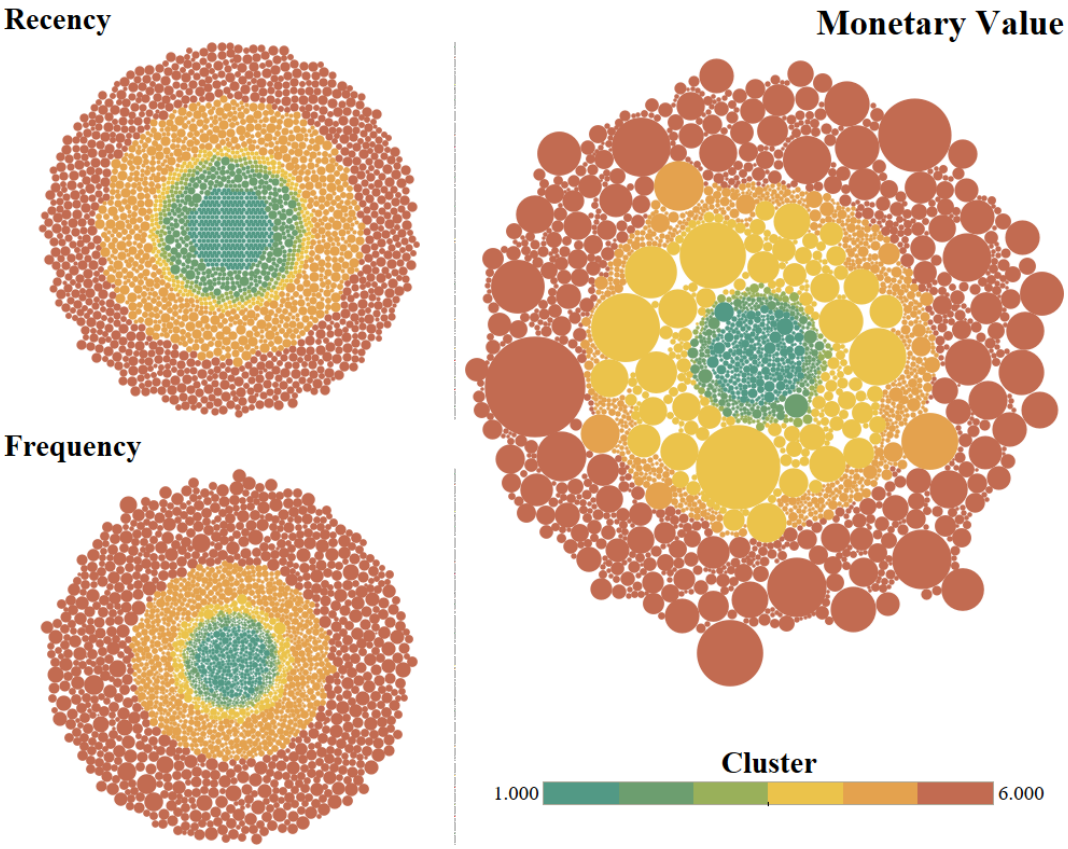


Figure C.1: Clusters Characterized by RFM: 7008 customers colored by cluster and sized by raw RFM variables, highlighting the distribution of cluster profiles at the instance level.

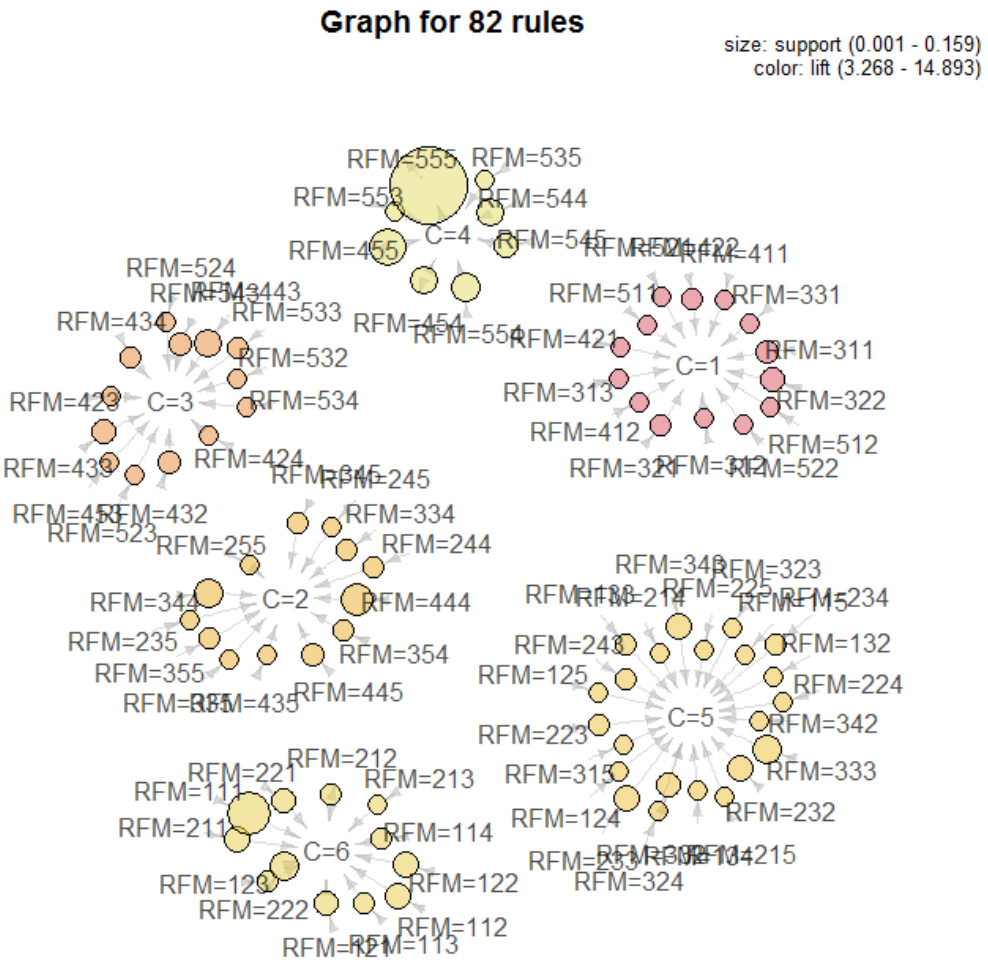


Figure C.2: Cluster RFM Membership Association Rules: each RFM category has 1.0 confidence meaning it only appears in that particular cluster. The size of each RFM cateogry corresponds to the amount of customers in that particular category. The lift for each rule is significant suggesting that the relationships are not random; the lighter categories are more homogeneous (less random).

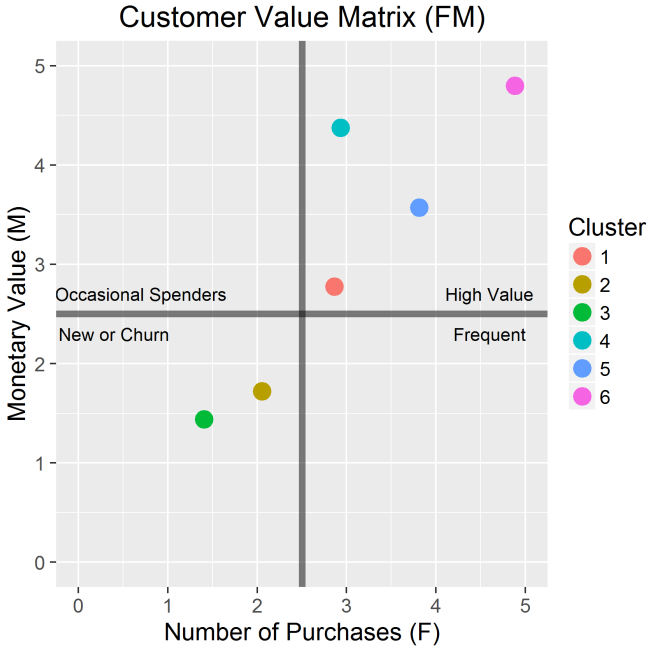
High-Level Overview		*Data Description*	
Total number of purchases	267742	Start date	6/1/2015
Total number of customers	7008	End date	4/29/2016
Average number of purchases	38.21	Region	UK
Total revenue	£1,078,435,318		
Average purchase amount	£4,028		
Average last purchase (days)	2 weeks		

Receny, Freuency, and Monetary Value Summary						
Cluster	1	2	3	4	5	6
R	2.15	3.55	1.38	2.15	3.96	4.6
F	2.86	2.05	1.41	4.37	3.81	4.88
M	2.78	1.72	1.44	2.93	3.57	4.8
Cumulative RFM value	7.79	7.32	4.23	9.45	11.34	14.28
Rank	4	5	6	3	2	1
Totals per Cluster						
Number of customers	859	849	1869	527	1432	1472
Percentage of customers	12.26%	12.11%	26.67%	7.52%	20.43%	21.00%
Total revenue	£6,716,942	£2,124,067	£2,695,515	£47,411,824	£34,723,084	£984,763,886
Percentage of total revenue	0.62%	0.20%	0.25%	4.40%	3.22%	91.31%
Number of purchases	122249	6245	5718	10540	51547	181443
Percentages of all purchases	32.36%	1.65%	1.51%	2.79%	13.65%	48.03%
Number customers purchased in last week	0	326	0	4	964	1419
Number customers purchases in last month	304	849	21	201	1432	1472
Number customers likely churn (>3mo)	157	0	1203	127	0	0

Table C.1: Cluster High Level Overview [80]

(a) Frequency-Recency

(b) Recency-Monetary Value



(c) Frequency-Monetary Value

Figure C.3: Cluster Value Matrices compare cluster centroids plotted against RFM. There is a minor correlation between how often a customer purchases and their last purchase, and between how frequently a customer purchases with how much they purchase. Intuitively this makes sense since if a customer tends to purchase quite often, they are more likely to have purchased more recently. Moreover, more frequenct purchasing implies more overall spend.

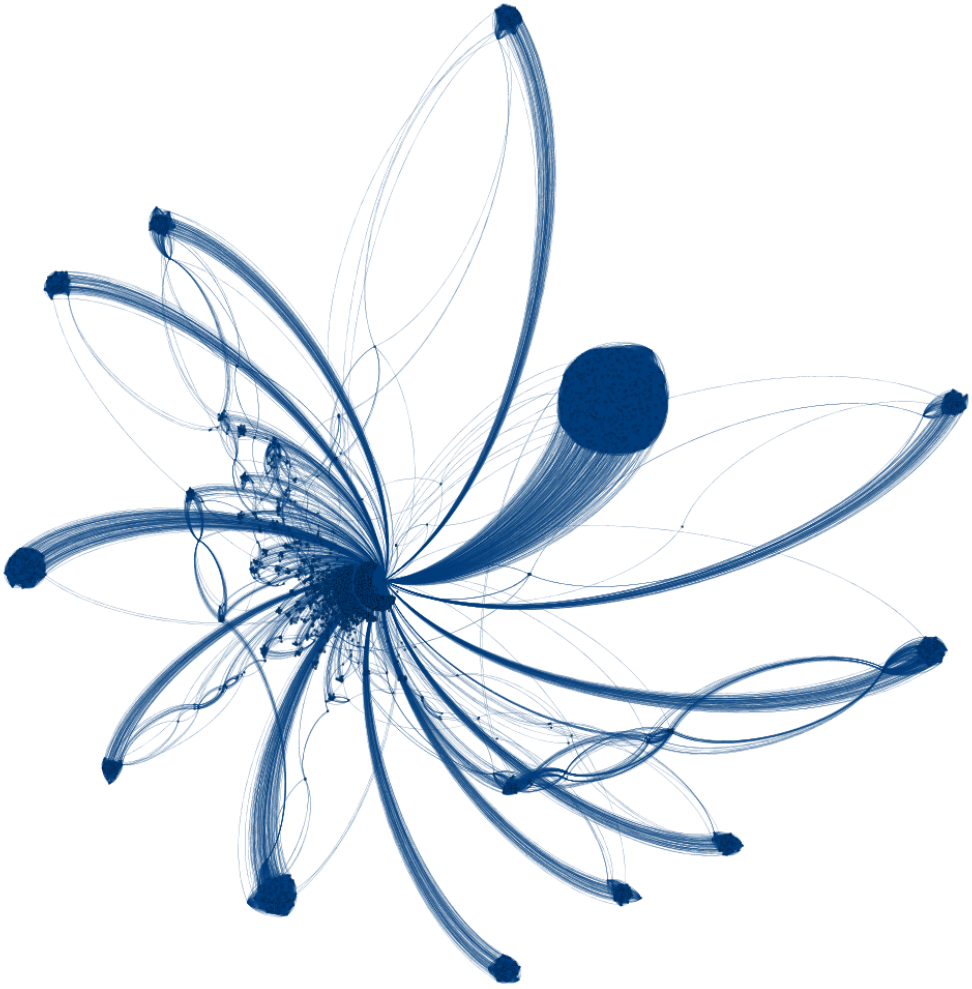


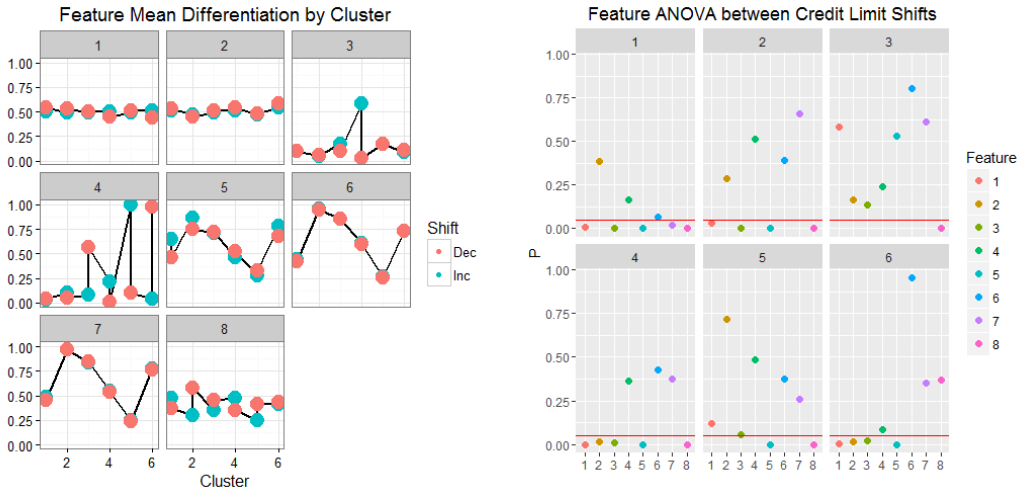
Figure C.4: Temporal Cluster Network: Each node corresponds to a customer's cluster rank vector. The edges are drawn between customer rank transition vectors that are 95% correlated. Clear strong structure is apparent; customers are likely to belong to a similar rank cluster and if not, will likely transfer to a cluster with other similar customers.

Appendix D

Extra Figures:

Credit Limit Analysis

D.1 Feature Engineering



(a) Scaled difference between segmented features **(b)** Statistical significance test of features

Figure D.1: Significance of Segmenting Features Characterized by the Direction Shift in Credit Limit: Plots indicate that features can be segmented and features relating to M (from RFM) are most significant. Features 1 through 8 are as follows: `cum_NSP`, `mean_NSP`, `avg_util`, `avg_change`, `R_mo`, `F_mo`, `M_mo`, and `sale_count`.

Table D.1: Original Features ANOVA Tested: Features tested for segmenting significance and used to determine which types of features (RFM) are significant. Category O corresponds to non-RFM related features.

	Feature	Category	Description
1	R_mo	R	The monthly recency score: the last day of the month minus the last sale for that customer on that month. The value is made normalized on an ordinal quintile scale for all customers during that month.
2	F_mo	F	The monthly frequency score: a raw count of how many sales were made during that month. The value is made normalized on an ordinal quintile scale for all customers during that month.
3	Sales_count	F	A raw count of sales during the period. This is used to create F_mo and other frequency-related features.
4	cum_NSP_seg	M	A rolling cumulative sum of revenue generated. Longer segments (i.e. those with no credit changes) will be more likely to have larger cum_NSP values.
5	mean_NSP	M	The average Net.Sell.Price over the segment. This is a useful feature because it is not affected by segment length and is thus easily generalizable between disparate segments and customers.
6	avg_util	M	A rolling cumulative sum of how much credit limit is being utilized. This is defined as the percentage of credit used for sales via Net.Sell.Price.
7	avg_change	M	The average cumulative percentage change in net sales price over the period. This is a feature that measures the variance in Net.Sell.Price over the period.
8	M_mo	M	The monthly monetary value score: a sum total of Net.Sell.Price over the month. The value is made normalized on an ordinal quintile scale for all customers during that month.

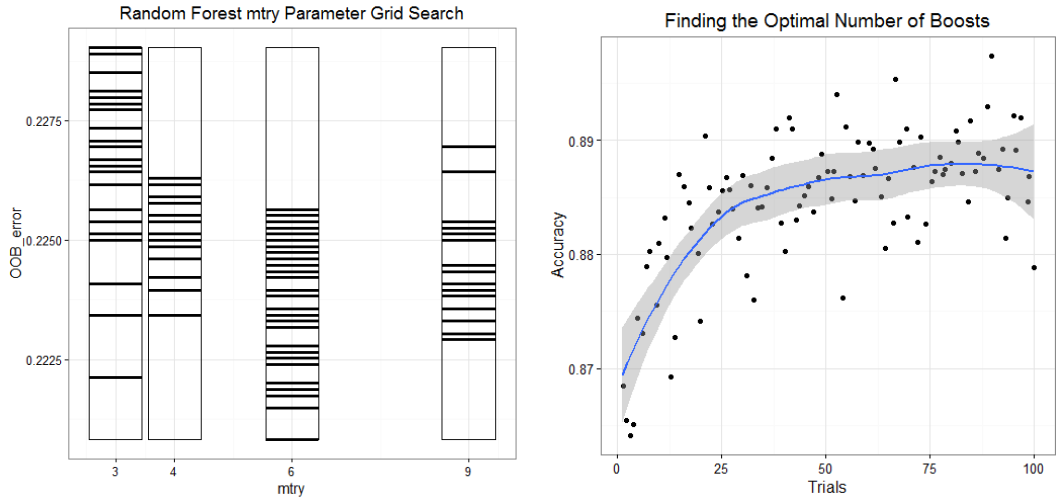
Table D.2: *Ex-post* Features Used for Experiments 1 and 2

	Feature	Category	Description
9	R_mo	R	The monthly recency score: the last day of the month minus the last sale for that customer on that month. The value is made normalized on an ordinal quintile scale for all customers during that month.
10	last_sale_diff	R	How many days between the first sale of this period and the last sale of last period.
11	Sale_count	F	A raw count of sales during the period. This is used to create F_mo and other frequency-related features.
12	Day	F	What day of the month was the sale made.
13	Net.Sell.Price	M	Revenue (Actual customer price). This can be the system generated price but a lot of the time it is difference for various reasons (most common because sales negotiated the price)
14	Avg_cum_NSP_perc_change	M	The average cumulative revenue percent change. This is a measure of deviation, i.e. how stable are purchases throughout the period.
15	NSP_expected	M	The mean revenue for the period
16	NSP_deviation	M	How much the revenue deviates throughout the period. This is a standardized feature derived from Avg_cum_NSP_perc_change and it takes into account the all sales, not just the last segment.
17	Perc_CL_sale	O	Net.Sell.Price over the credit for that month. This should be a value <0.1 per sale.
18	Cluster	O	What cluster the cluster is in as determined by K-Means clustering algorithm
19	Last_cl	O	The previous credit limit.
20	CL_per_clust	O	The customer's credit limit over the average credit limit for their cluster each month.

Table D.3: *Ex-ante* Features Used for Experiments 3 and 4

	Feature	Category	Description
21	R_yr	R	The customers recency score as determine by the intial K-means clustering
22	R_mo	R	The monthly recency score: the last day of the month minus the last sale for that customer on that month. The value is made normalized on an ordinal quintile scale for all customers during that month.
23	Avg_day_interval	R	The average amount of time (in days) between sales.
24	F_yr	F	The customers frequency score as determine by the intial K-means clustering
25	F_mo	F	The monthly frequency score: a raw count of how many sales were made during that month. The value is made normalized on an ordinal quintile scale for all customers during that month.
26	Sale_count	F	A raw count of sales during the period. This is used to create F_mo and other frequency-related features.
27	cum_NSP_seg	M	A rolling cumulative sum of revenue generated. Longer segments (i.e. those with no credit changes) will be more likely to have larger cum_NSP values.
28	M_yr	M	The customers monetary value score as determine by the intial K-means clustering
29	M_mo	M	The monthly monetary value score: a sum total of Net.Sell.Price over the month. The value is made normalized on an ordinal quintile scale for all customers during that month.
30	Avg_change	M	The average cumulative percentage change in net sales price over the period. This is a feature that measures the variance in Net.Sell.Price over the period.
31	NSP_high	M	The segment's highest Net.Sell.Price
32	NSP_low	M	The segment's lowest Net.Sell.Price
33	Util	O	The percent of credit utilized as determined by Net.Sell.Price/CL
34	CL	O	The customer's credit limit.
35	CL _last	O	The previous credit limit.
36	Cluster	O	What cluster the cluster is in as determined by K-Means clustering algorithm

D.2 Parameter Tuning



(a) *Mtry* for random forests **(b)** Boosting iterations for C5.0

Figure D.2: Parameter Tuning for Random Forest and C5.0

For the random forest variations, the number of variables sampled at each split was determined by iterating over 50 models at certain *mtry* values for each experiment's algorithm. Although the decrease in error was not completely clear, the plots provide an indication of an optimal parameter which was thus used in training and 10-fold cross validation. Likewise, the number of boosting iterations seems to level after 50 for the example above, however this differed from model to model.

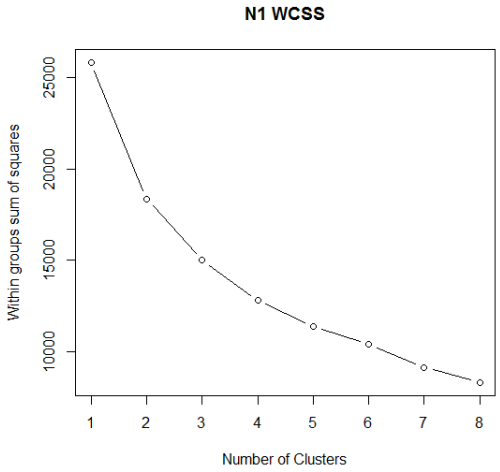
Appendix E

Clustering Evaluation Visualization

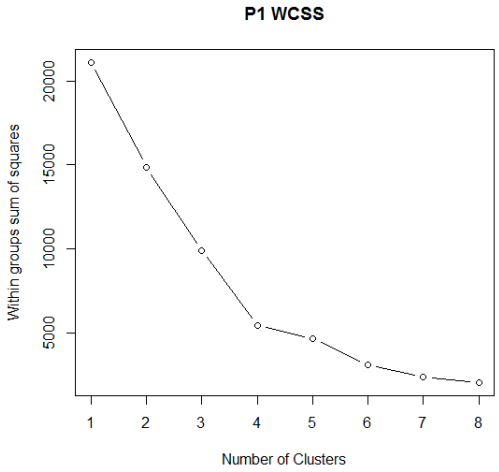
I used a heuristic visualization method from a function I made to compare RFM cluster membership of high dimensional data sets. I assess the inter-cluster heterogeneity (i.e. how different are clusters?) and intra-cluster homogeneity (i.e. how alike are customers within a cluster?) using euclidean distance. Using these assessment methods, I compared clustering my data set where each RFM feature exhibits a power law (**P**) to a data set where each feature is on a normal distribution (**N**). These were compared using the RFM raw features (**1**)[2][3], binary RFM scoring method (**2**)[81], and quintile RFM scoring [81][1][4]. Result suggests the binary method is best for clustering RFM on *K*-means; the binary method has the least variance and strongest clustering however it limits the amount of clustering. The quintile method however finds more diverse groups without sacrificing intra-cluster homogeneity across distributions, whereas clustering with raw or binary features creates significantly different clusters with different distributions.

	Intra-cluster homogeneity					*Inter-cluster heterogeneity*			
	Mean	Sd	Range	SError	Size Var	Mean	Sd	Range	SError
N1	0.6641	0.7476	0.6690	1.8546	110.1944	1.0780	0.9011	0.8471	2.2252
N2	0.1504	0.4179	0.7275	0.8999	295.0236	1.1486	0.2724	0.5295	0.6533
N3	0.5580	0.6401	0.6385	1.5703	104.0736	1.0874	0.8092	0.7963	1.9903
P1	2.2424	1.7040	0.4710	42.6454	879.5667	6.9191	2.8889	0.6945	46.4347
P2	0.1178	0.3414	0.7581	1.1112	757.4458	1.2567	0.1882	0.5013	0.6762
P3	0.4074	0.4297	0.6884	1.1073	239.6829	1.1184	0.4458	0.7253	1.1553

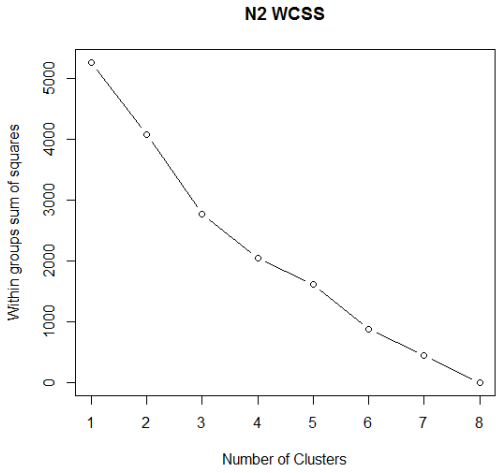
Table E.1: RFM Feature Clustering Comparison: Results comparing *K*-means clustering using the reference Pareto distribution (**P**) compared to a normal distribution of features (**N**) across three types of RFM features (**1,2,3**).



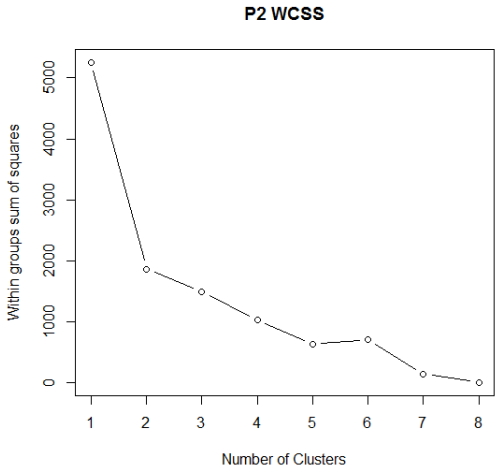
(a) Normal distribution with raw RFM



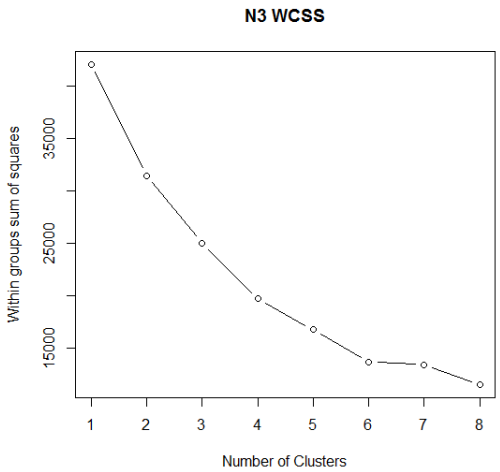
(b) Pareto distribution with raw RFM



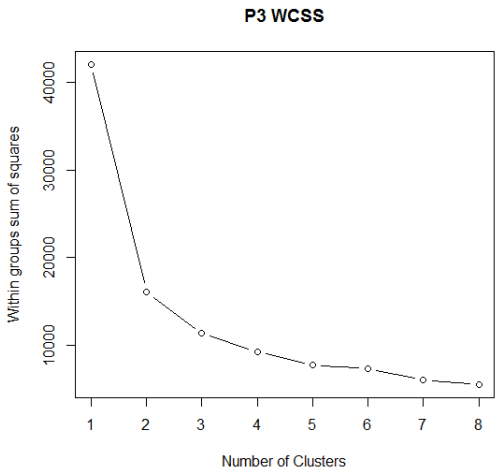
(c) Normal distribution with binary RFM



(d) Pareto distribution with binary RFM

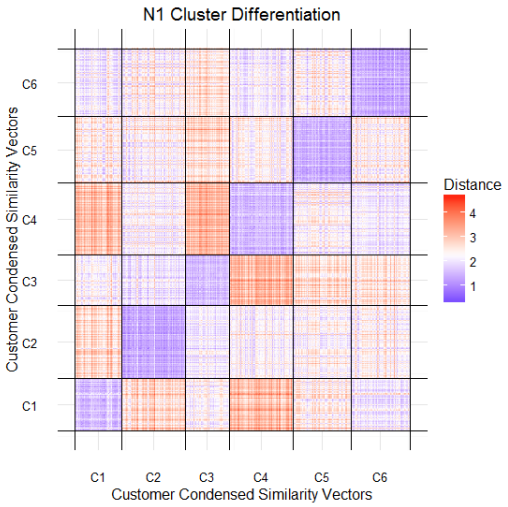


(e) Normal distribution with quintile RFM

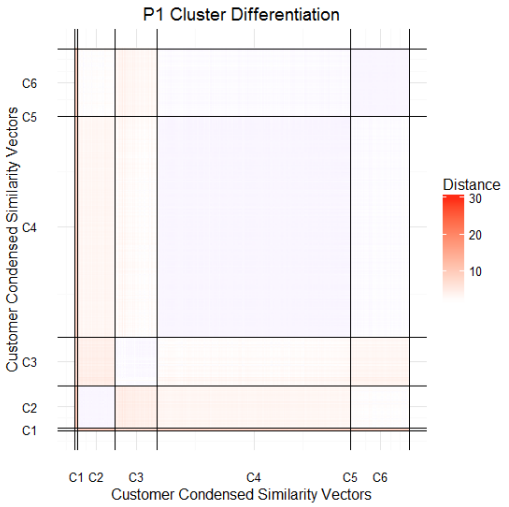


(f) Pareto distribution with quintile RFM

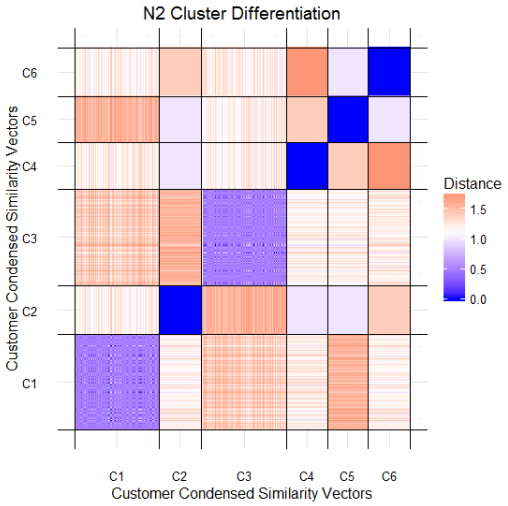
Figure E.1: RFM Clustering Method Comparison WCSS



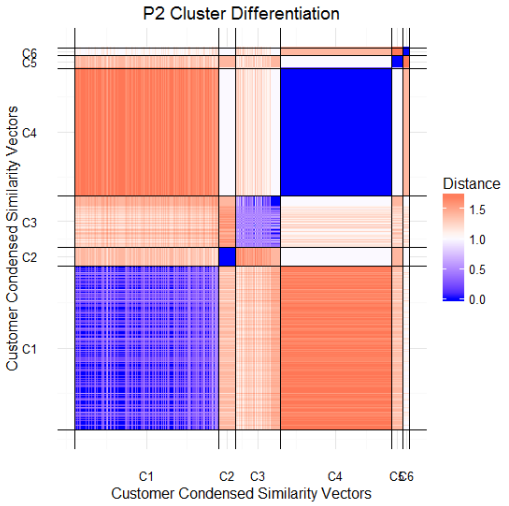
(a) Normal distribution with raw RFM



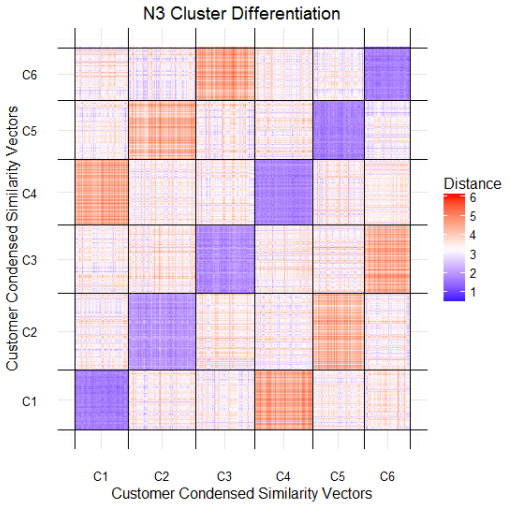
(b) Pareto distribution with raw RFM



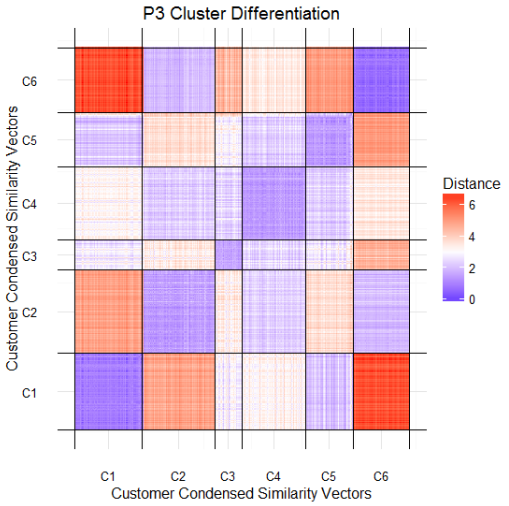
(c) Normal distribution with binary RFM



(d) Pareto distribution with binary RFM



(e) Normal distribution with quintile RFM



(f) Pareto distribution with quintile RFM

Figure E.2: RFM Clustering Method Comparison Heatmaps

Table E.2: Clustering Using Raw RFM Features: Each square, *s* corresponds to the square of the matrix (6x6 squares) for the corresponding visualization in Figure E.2 a and b.

	N1					P1				
	Intra-cluster Homogeneity									
s	Mean	Sd	Range	SError	Size	Mean	Sd	Range	SError	Size
1	0.704	0.810	0.742	2.184	963	2.235	1.365	0.340	15.274	56
8	0.618	0.679	0.622	1.555	1336	0.418	0.373	0.255	1.119	779
15	0.708	0.829	0.748	2.281	926	0.667	0.452	0.152	1.267	891
22	0.697	0.753	0.604	1.733	1323	0.329	0.199	0.089	0.261	4047
29	0.643	0.705	0.608	1.692	1216	9.431	6.897	0.994	235.716	6
36	0.615	0.710	0.690	1.682	1249	0.375	0.938	0.996	2.234	1234
	Inter-cluster heterogeneity									
2	1.219	0.933	1.000	2.516		4.945	1.287	0.423	14.395	
3	1.000	0.926	0.896	2.499		3.533	1.458	0.410	16.310	
4	1.348	0.854	0.838	2.303		4.367	1.384	0.359	15.484	
5	1.094	0.974	0.983	2.626		13.499	5.616	0.882	62.822	
6	0.889	0.842	0.811	2.270		4.617	1.423	0.880	15.916	
7	1.219	0.933	1.000	2.136		4.945	1.287	0.423	3.859	
9	0.910	0.812	0.754	1.860		2.463	0.432	0.245	1.294	
10	0.987	0.864	0.776	1.979		1.800	0.356	0.202	1.067	
11	1.020	0.924	0.780	2.116		16.196	7.010	0.955	21.026	
12	1.026	0.959	0.836	2.197		1.108	0.761	1.000	2.282	
13	1.000	0.926	0.896	2.549		3.533	1.458	0.410	4.089	
14	0.910	0.812	0.754	2.234		2.463	0.432	0.245	1.210	
16	1.399	0.852	0.897	2.345		1.325	0.531	0.137	1.490	
17	1.173	0.978	0.850	2.691		15.624	6.946	0.960	19.479	
18	1.130	0.932	0.830	2.565		1.768	0.758	0.980	2.127	
19	1.348	0.854	0.838	1.965		4.367	1.384	0.359	1.821	
20	0.987	0.864	0.776	1.989		1.800	0.356	0.202	0.468	
21	1.399	0.852	0.897	1.961		1.325	0.531	0.137	0.699	
23	1.014	0.895	0.846	2.059		15.820	7.371	0.994	9.700	
24	0.921	0.804	0.750	1.850		0.780	0.695	0.994	0.915	
25	1.094	0.974	0.983	2.337		13.499	5.616	0.882	191.926	
26	1.020	0.924	0.780	2.217		16.196	7.010	0.955	239.584	
27	1.173	0.978	0.850	2.348		15.624	6.946	0.960	237.378	
28	1.014	0.895	0.846	2.148		15.820	7.371	0.994	251.919	
30	1.039	0.968	0.859	2.323		15.942	7.306	0.996	249.700	
31	0.889	0.842	0.811	1.994		4.617	1.423	0.880	3.391	
32	1.026	0.959	0.836	2.272		1.108	0.761	1.000	1.813	
33	1.130	0.932	0.830	2.208		1.768	0.758	0.980	1.807	
34	0.921	0.804	0.750	1.904		0.780	0.695	0.994	1.657	
35	1.039	0.968	0.859	2.292		15.942	7.306	0.996	17.412	

Table E.3: Clustering Comparison Using Binary RFM Features: Each square, *s* corresponds to the square of the matrix (6x6 squares) for the corresponding visualization in Figure E.2 c and d.

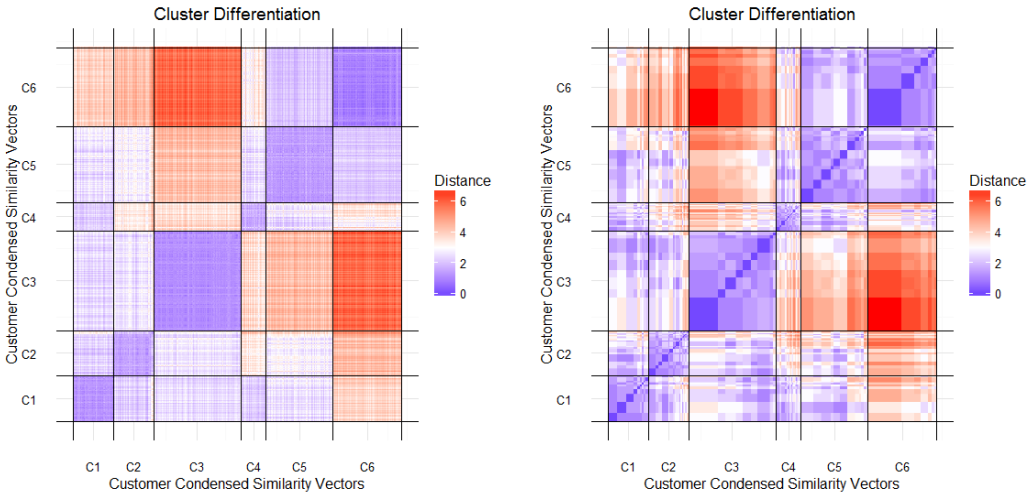
	N2					P2				
	Intra-cluster homogeneity									
s	Mean	Sd	Range	SError	Size	Mean	Sd	Range	SError	Size
1	0.446	1.017	0.577	2.037	1747	0.252	0.659	0.577	1.007	3003
8	0.002	0.096	0.577	0.268	897	0.008	0.160	0.816	0.718	349
15	0.447	1.019	1.000	2.024	1777	0.415	0.746	1.000	2.038	940
22	0.003	0.139	0.816	0.398	854	0.001	0.044	0.577	0.076	2345
29	0.003	0.140	0.816	0.401	847	0.009	0.140	0.577	0.779	227
36	0.002	0.096	0.577	0.270	891	0.022	0.299	1.000	2.049	149
	Inter-cluster heterogeneity									
2	1.073	0.425	0.816	0.852		1.310	0.246	0.816	0.375	
3	1.245	0.528	0.423	1.057		1.304	0.335	0.423	0.512	
4	1.078	0.421	0.239	0.844		1.637	0.172	0.184	0.263	
5	1.400	0.324	0.423	0.650		1.315	0.227	0.423	0.346	
6	1.074	0.422	0.423	0.845		1.036	0.227	0.239	0.347	
7	1.073	0.425	0.816	1.189		1.310	0.246	0.816	1.100	
9	1.398	0.332	1.000	0.928		1.468	0.233	1.000	1.044	
10	0.893	0.058	0.423	0.162		0.975	0.060	0.423	0.270	
11	0.892	0.028	0.239	0.079		1.374	0.041	0.423	0.185	
12	1.260	0.040	0.239	0.111		0.975	0.051	0.239	0.228	
13	1.245	0.528	0.423	1.048		1.304	0.335	0.423	0.916	
14	1.398	0.332	1.000	0.659		1.468	0.233	1.000	0.636	
16	1.071	0.425	1.000	0.844		1.095	0.288	0.816	0.787	
17	1.072	0.421	0.239	0.837		1.252	0.287	0.239	0.785	
18	1.081	0.421	0.239	0.837		1.097	0.288	0.239	0.785	
19	1.078	0.421	0.239	1.207		1.637	0.172	0.184	0.298	
20	0.893	0.058	0.423	0.167		0.975	0.060	0.423	0.104	
21	1.071	0.425	1.000	1.218		1.095	0.288	0.816	0.498	
23	1.259	0.099	0.816	0.283		0.968	0.100	0.816	0.174	
24	1.543	0.055	0.423	0.158		1.373	0.052	0.239	0.091	
25	1.400	0.324	0.423	0.933		1.315	0.227	0.423	1.259	
26	0.892	0.028	0.239	0.081		1.374	0.041	0.423	0.229	
27	1.072	0.421	0.239	1.212		1.252	0.287	0.239	1.596	
28	1.259	0.099	0.816	0.284		0.968	0.100	0.816	0.558	
30	0.891	0.085	1.000	0.245		1.672	0.214	1.000	1.192	
31	1.074	0.422	0.423	1.183		1.036	0.227	0.239	1.557	
32	1.260	0.040	0.239	0.112		0.975	0.051	0.239	0.349	
33	1.081	0.421	0.239	1.181		1.097	0.288	0.239	1.972	
34	1.543	0.055	0.423	0.155		1.373	0.052	0.239	0.359	
35	0.891	0.085	1.000	0.239		1.672	0.214	1.000	1.471	

Table E.4: Clustering Comparison Using Quintile RFM Features: Each square, *s* corresponds to the square of the matrix (6x6 squares) for the corresponding visualization in Figure E.2 e and f.

	N3					P3				
	Intra-cluster homogeneity									
s	Mean	Sd	Range	SError	Size	Mean	Sd	Range	SError	Size
1	0.523	0.562	0.433	1.419	1100	0.303	0.370	0.354	0.824	1411
8	0.631	0.736	0.645	1.639	1413	0.478	0.451	0.829	0.968	1520
15	0.581	0.675	0.707	1.603	1241	0.525	0.475	0.707	1.682	559
22	0.549	0.617	0.661	1.487	1208	0.463	0.427	0.816	0.974	1345
29	0.536	0.618	0.677	1.570	1086	0.468	0.502	0.595	1.339	986
36	0.528	0.632	0.707	1.704	965	0.208	0.353	0.829	0.856	1192
	Inter-cluster heterogeneity									
2	1.014	0.826	0.866	2.084		1.595	0.384	0.866	0.855	
3	0.994	0.787	0.722	1.987		0.922	0.570	0.685	1.271	
4	1.377	0.671	0.796	1.694		1.038	0.438	0.533	0.975	
5	1.007	0.718	0.633	1.812		0.692	0.507	0.563	1.129	
6	1.048	0.862	0.722	2.177		2.060	0.310	0.567	0.691	
7	1.014	0.826	0.866	1.839		1.595	0.384	0.866	0.824	
9	1.065	0.891	1.000	1.985		1.105	0.522	0.866	1.121	
10	1.033	0.839	0.780	1.868		0.751	0.436	0.780	0.937	
11	1.315	0.810	0.856	1.804		1.220	0.427	0.780	0.916	
12	0.997	0.875	0.722	1.948		0.633	0.338	0.780	0.726	
13	0.994	0.787	0.722	1.870		0.922	0.570	0.685	2.019	
14	1.065	0.891	1.000	2.118		1.105	0.522	0.866	1.848	
16	1.044	0.833	0.866	1.981		0.788	0.492	0.829	1.744	
17	1.026	0.859	0.722	2.041		0.938	0.614	0.856	2.174	
18	1.359	0.762	0.856	1.812		1.485	0.485	0.677	1.716	
19	1.377	0.671	0.796	1.617		1.038	0.438	0.533	0.999	
20	1.033	0.839	0.780	2.020		0.751	0.436	0.780	0.996	
21	1.044	0.833	0.866	2.008		0.788	0.492	0.829	1.124	
23	1.036	0.843	0.866	2.031		0.752	0.464	0.777	1.059	
24	1.027	0.760	0.697	1.831		1.143	0.369	0.455	0.843	
25	1.007	0.718	0.633	1.824		0.692	0.507	0.563	1.351	
26	1.315	0.810	0.856	2.058		1.220	0.427	0.780	1.138	
27	1.026	0.859	0.722	2.182		0.938	0.614	0.856	1.637	
28	1.036	0.843	0.866	2.142		0.752	0.464	0.777	1.237	
30	0.969	0.801	0.842	2.035		1.656	0.332	0.866	0.885	
31	1.048	0.862	0.722	2.324		2.060	0.310	0.567	0.752	
32	0.997	0.875	0.722	2.358		0.633	0.338	0.780	0.819	
33	1.359	0.762	0.856	2.055		1.485	0.485	0.677	1.175	
34	1.027	0.760	0.697	2.048		1.143	0.369	0.455	0.895	
35	0.969	0.801	0.842	2.159		1.656	0.332	0.866	0.805	

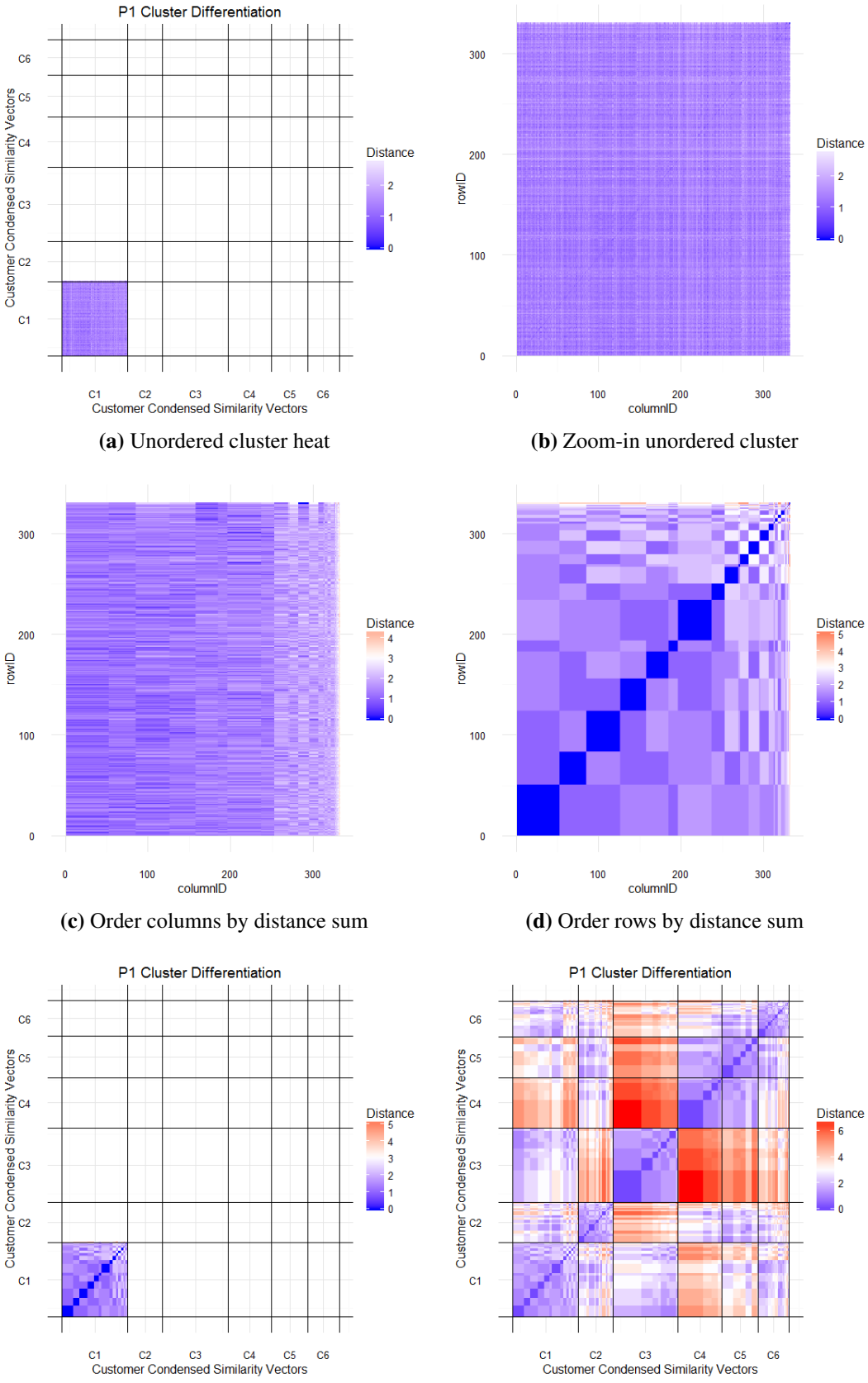
As previously mentioned, the function has two main purposes: assess the homogeneity of customers within clusters (i.e. how similar are customers in the same cluster) and assess the heterogeneity or differentiation between clusters. It does this by visualizing and extracting metrics from each square in the symmetrical heat map. We select the clustering method that maximizes both of these metrics as determined by distance mean, standard deviation, and standard error. This translates to minimizing the intra-cluster metrics and maximizing the inter-cluster metrics in tables E.2,E.3, and E.4.

In order to better assess the within-cluster similarity, I added a method to order the customer vectors *within* each cluster (i.e. the diagonal of the heat map). The results of which reveal the level of dispersion within the cluster. Customers within the diagonal of each diagonal (i.e. the most similar customers in each cluster) reveal how similar and how many customers are nearly identical. This does not affect the results of the between cluster metrics; it only orders and makes more clear the instance similarity dispersion within clusters. Figure E.4 shows the process of how the diagonal of the heat map is extracted and ordered to improve the heat map interpretation. Figure E.3 demonstrates the function used empirically on the client's clusters. We can identify the density of completely identical and disparate customers after ordering the diagonal (i.e. the within-cluster similarity).



(a) Unordered within-cluster diagonal **(b)** Ordered within-cluster diagonal

Figure E.3: Client's Clusters Ordered Heatmap



(a) Unordered cluster heat

(b) Zoom-in unordered cluster

(c) Order columns by distance sum

(d) Order rows by distance sum

(e) Cluster ordered

(f) Full output

Figure E.4: Function for Ordering Cluster Heat: Improving within-cluster homogeneity visualization

Appendix F

Dissertation Project Workflow

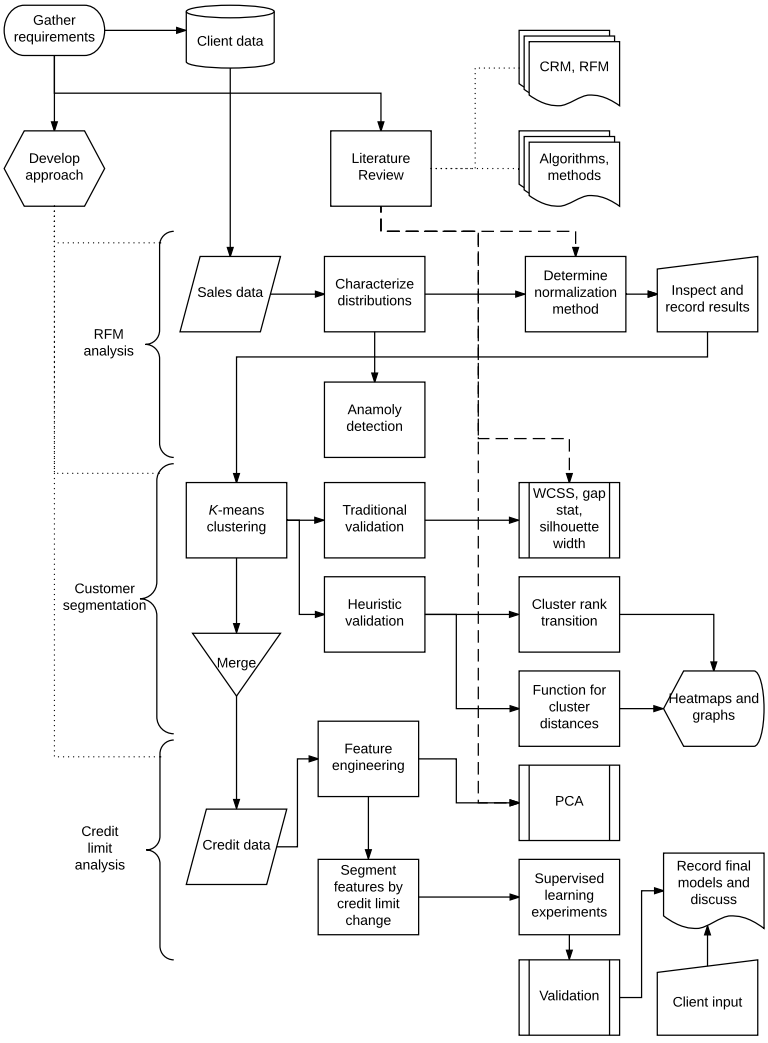


Figure F.1: Dissertation Summary Flowchart

Bibliography

[1] J R Miglautsch. Thoughts on rfm scoring. *Journal of Database Marketing & Customer Strategy Management*, 8(1):67–72, 08 2000.

[2] Mahboubeh Khajvand and Mohammad Jafar Tarokh. Estimating customer future value of different customer segments based on adapted rfm model in retail banking context. *Procedia Computer Science*, 3:1327–1332, 2011.

[3] 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.

[4] Q Razieh. Developing a model for measuring customer s loyalty and value with rfm technique and clustering algorithms. *The Journal of Mathematics and Computer Science*, 4(2):172–181, 2012.

[5] Ching-Hsue Cheng and You-Shyang Chen. Classifying the segmentation of customer value via rfm model and rs theory. *Expert Systems with Applications*, 36(3):4176–4184, 04 2009.

[6] Robert A. Eisenbeis. Credit granting: A comparative analysis of classification procedures: Discussion. *The Journal of Finance*, 42(3):681, 07 1987.

[7] Richard Koch, Tijmen Roozenboom Engels, and Carla Prins. *Het 80/20-principe: Het geheim van meer bereiken met minder moeite*. Academic Service economie en bedrijfskunde, Schoonhoven, 1998.

[8] Yogesh Virkar and Aaron Clauset. Power-law distributions in binned empirical data. *The Annals of Applied Statistics*, 8(1):89–119, 03 2014.

[9] S. T. Roweis. Nonlinear dimensionality reduction by locally linear embedding. *Science*, 290(5500):2323–2326, 12 2000.

[10] Laurens Van Der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

[11] EW Forgy. Cluster analysis of multivariate data: efficiency versus interpretability of classifications. *Biometrics*, 1965.

[12] Anil K. Jain. Data clustering: 50 years beyond k-means. *Pattern Recognition Letters*, 31(8):651–666, 06 2010.

[13] Chen Yu. Kmeans clustering. Technical report, 2010.

[14] J. A. Hartigan and M. A. Wong. Algorithm as 136: A k-means clustering algorithm. *Applied Statistics*, 28(1):100, 1979.

[15] Matus Telgarsky and Andrea Vattani. Hartigan's method: K-means clustering without voronoi. Technical report.

[16] David J. Strauss and J. A. Hartigan. Clustering algorithms. *Biometrics*, 31(3):793, 09 1975.

[17] Thomas H Cormen. *Algorithms unlocked*. MIT Press, Cambridge, MA, 03 2013.

[18] M Young. K-means clustering overview, 2004.

[19] Sueli A. Mingoti and Joab O. Lima. Comparing som neural network with fuzzy c-means, k-means and traditional hierarchical clustering algorithms. *European Journal of Operational Research*, 174(3):1742–1759, 11 2006.

[20] D Pellag and A Moore. x-means: Extending k-means with efficient estimation of the number of clusters. Technical report, 2000.

[21] David Arthur and Sergei Vassilvitskii. K-means++: The advantages of careful seeding. Technical report, Proceedings of the eighteenth annual ACM-SIAM

symposium on Discrete algorithms. Society for Industrial and Applied Mathematics Philadelph, 2006.

[22] Boris Mirkin. Choosing the number of clusters. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, 1(3):252–260, 03 2011.

[23] Mingjin Yan and Keying Ye. Determining the number of clusters using the weighted gap statistic. *Biometrics*, 63(4):1031–1037, 04 2007.

[24] Mojgan Mohajer, Karl-Hans Englmeier, and Volker J Schmid. A comparison of gap statistic definitions with and with- out logarithm function. 2011.

[25] Renato Cordeiro de Amorim and Christian Hennig. Recovering the number of clusters in data sets with noise features using feature rescaling factors. *Information Sciences*, 324:126–145, 12 2015.

[26] Archana Singh, Avantika Yadav, and Ajay Rana. K-means with three different distance metrics. *International Journal of Computer Applications*, 67(10):13–17, 2013.

[27] Peter J. Rousseeuw. Silhouettes: A graphical aid to the interpretation and validation of cluster analysis. *Journal of Computational and Applied Mathematics*, 20:53–65, 11 1987.

[28] Joseph M Hilbe. *Logistic regression models*. Chapman and Hall/CRC, Boca Raton, 07 2016.

[29] Steven L. Salzberg. C4.5: Programs for machine learning by j. ross quinlan. morgan kaufmann publishers, inc., 1993. *Machine Learning*, 16(3):235–240, 09 1994.

[30] Rutvija Pandya and Jayati Pandya. C5. 0 algorithm to improved decision tree with feature selection and reduced error pruning. *International Journal of Computer Applications*, 117(16):18–21, 2015.

[31] Xindong Wu, Vipin Kumar, Ross J. Quinlan, Joydeep Ghosh, Qiang Yang, Hiroshi Motoda, Geoffrey J. McLachlan, Angus Ng, Bing Liu, Philip S. Yu, Zhi-Hua Zhou, Michael Steinbach, David J. Hand, and Dan Steinberg. Top 10 algorithms in data mining. *Knowledge and Information Systems*, 14(1):1–37, 2007.

[32] Manuel Fernández-Delgado, Eva Cernadas, Senén Barro, Dinani Amorim, and Russ Greiner. Do we need hundreds of classifiers to solve real world classification problems? *Journal of Machine Learning Research*, 15:3133–3181, 2014.

[33] Andy Liaw and Matthew Wiener. Classification and regression by randomforest. Technical report, 2002.

[34] Venkat Srinivasan and Yong H. Kim. Credit granting: A comparative analysis of classification procedures. *The Journal of Finance*, 42(3):665, 07 1987.

[35] Tian-Shyug Lee, Chih-Chou Chiu, Yu-Chao Chou, and Chi-Jie Lu. Mining the customer credit using classification and regression tree and multivariate adaptive regression splines. *Computational Statistics & Data Analysis*, 50(4):1113–1130, 02 2006.

[36] I. H. Witten, Eibe Frank, and Mark A Hall. *Data mining: Practical machine learning tools and techniques, Third edition (the Morgan Kaufmann series in data management systems)*. Morgan Kaufmann Publishers In, United States, 3 edition, 02 2011.

[37] Kristina Machova, Miroslav Puszta, Frantisek Barcak, and Peter Bednar. A comparison of the bagging and the boosting methods using the decision trees classifiers. *Computer Science and Information Systems*, 3(2):57–72, 2006.

[38] David H. Wolpert. Stacked generalization. *Neural Networks*, 5(2):241–259, 01 1992.

[39] Martin Hilbert. Big data for development: From information - to knowledge societies. *SSRN Electronic Journal*, 2015.

[40] Yurong Xu, David C. Yen, Binshan Lin, and David C. Chou. Adopting customer relationship management technology. *Industrial Management & Data Systems*, 102(8):442–452, 11 2002.

[41] Stanley A. Brown and PricewaterhouseCoppers. *Customer relationship management: A strategic imperative in the world of e-business /editor and contributor Stanley A brown*. John Wiley & Sons Canada, New York, 06 2000.

[42] Jon Anton and Michael Hoeck. *E business customer service*. Purdue University Press, United States, 01 2002.

[43] Joe Peppard. Customer relationship management (crm) in financial services. *European Management Journal*, 18(3):312–327, 06 2000.

[44] Shun Y. Lam, Venkatesh Shankar, M. Krishna Erramilli, and Bvsan Murthy. Customer value, satisfaction, loyalty, and switching costs: An illustration from a business-to-business service context. *Journal of the Academy of Marketing Science*, 32(3):293–311, 07 2004.

[45] Philip Kotler. Marketing during periods of shortage. *Journal of Marketing*, 38(3):20, 07 1974.

[46] Microsoft and salesforce invest in new crm chat tool. *Business Insider*, 06 2016.

[47] M McCabe. Rbs moves crm into lida as part of customer-centric push. 06 2016.

[48] Paul D. Berger and Nada I. Nasr. Customer lifetime value: Marketing models and applications. *Journal of Interactive Marketing*, 12(1):17–30, 01 1998.

[49] A Hughes. *Strategic Database Marketing*. Probus Publishing Company, Chicago, IL, 1994.

[50] Michael Haenlein, Andreas M. Kaplan, and Anemone J. Beeser. A model to determine customer lifetime value in a retail banking context. *European Management Journal*, 25(3):221–234, 06 2007.

[51] Frederick Newell. *The new rules of marketing: How to use one-to-one relationship marketing to be the leader in your industry*. Irwin Professional Publishing, New York, 06 1997.

[52] Peter S. Fader, Bruce G.S. Hardie, and Ka Lok Lee. Rfm and clv: Using iso-value curves for customer base analysis. *Journal of Marketing Research*, 42(4):415–430, 11 2005.

[53] Bob Stone, Ron Jacobs, and Johna A. Greco. *Successful direct marketing methods: Interative, database, and customer-based marketing for digital age*. McGraw-Hill Companies, The, United States, 8 edition, 01 2008.

[54] Gordon S. Linoff and Michael J Berry. *Data mining techniques: For marketing, sales, and customer relationship management*. *Wiley Computer Publishing, Indianapolis, IN, 3 edition, 04 2011.

[55] Duen-Ren Liu and Ya-Yueh Shih. Integrating ahp and data mining for product recommendation based on customer lifetime value. *Information & Management*, 42(3):387–400, 03 2005.

[56] YS Cho. Lecture notes in electrical engineering. pages 353–362. Springer Science + Business Media, 2012.

[57] B Sohrabi and A Khanlar. Customer lifetime value (clv) measurment based on rfm model. *Iranian Acc. Aud. Rev.*, 14, 2007.

[58] R Mohammadi, B Bidabad, T Nourasteh, and M Sherafati. Credit ranking of bank customers (an integrated model of rfm, fahp and k-means), 2014.

[59] Yuliya Demyanyk and Otto Van Hemert. Title introduction data analysis summary understanding the subprime mortgage crisis. Technical report, 2008.

[60] Ravi S. Achrol and Philip Kotler. Marketing in the network economy. *Journal of Marketing*, 63:146, 1999.

[61] Paul N. Wilson. The economic nature of network capital in b2b transactions. *Agribusiness*, 23(3):435–448, 2007.

[62] Marius Pretorius. Defining business decline, failure and turnaround: A content analysis. *The Southern African Journal of Entrepreneurship and Small Business Management*, 2(1):1, 12 2009.

[63] Roozmehr Safi and Zhangxi Lin. Association for information systems ais electronic library (aisel) using non-financial data to assess the creditworthiness of businesses in online trade. Technical report, 2014.

[64] Recommender systems handbook. pages 1–35. Springer Science + Business Media, 2010.

[65] Young-Sung Cho, Mi-Sug Gu, and Keun-Ho Ryu. Implementation of personalized recommendation system using k-means clustering of item category based on rfm. *Journal of the Korea Society of Computer and Information*, 17(6):163–172, 06 2012.

[66] Jinzhao Shi, Jue Guo, Shubin Wang, and Zhanhao Wang. Credit risk evaluation of online supply chain finance based on third-party b2b e-commerce platform: An exploratory research based on china's practice. *International Journal of u- and e-Service, Science and Technology*, 8(5):93–104, 05 2015.

[67] R.K. Pearson. Outliers in process modeling and identification. *IEEE Transactions on Control Systems Technology*, 10(1):55–63, 2002.

[68] Seyed Mohammad Seyed Hosseini, Anahita Maleki, and Mohammad Reza Gholamian. Cluster analysis using data mining approach to develop crm methodology to assess the customer loyalty. *Expert Systems with Applications*, 37(7):5259–5264, 07 2010.

[69] P Ryan. Title stata.com pctile create variable containing percentiles. Technical report, 2013.

[70] Bourne Jonno. Bigheat visualization. Github, 07 2016.

[71] Wuyang Ju, Jianxin Li, Weiren Yu, and Richong Zhang. Igraph: An incremental data processing system for dynamic graph. *Frontiers of Computer Science*, 10(3):462–476, 04 2016.

[72] T. Warren Liao. A clustering procedure for exploratory mining of vector time series. *Pattern Recognition*, 40(9):2550–2562, 09 2007.

[73] G NOETHER. Why kendall tau, 1981.

[74] Barbara Hammer and Thomas Villmann. Generalized relevance learning vector quantization. *Neural Networks*, 15(8-9):1059–1068, 10 2002.

[75] Andy Liaw, Matthew Wiener, Andy Maintainer, and Liaw. Title breiman and cutler's random forests for classification and regression description classification and regression based on a forest of trees using random inputs. Technical report, 2015.

[76] O J Oyelade, O Oladipupo, and I C Obagbuwa. Application of k-means clustering algorithm for prediction of students' academic performance. *IJCSIS) International Journal of Computer Science and Information Security*, 7(1), 2010.

[77] Ron Kohavi, Neal J. Rothleder, and Evangelos Simoudis. Emerging trends in business analytics. *Communications of the ACM*, 45(8), 08 2002.

[78] Colin S. Gillespie. Fitting heavy tailed distributions: The powerlaw package. *Journal of Statistical Software*, 64(2), 2015.

[79] M E J Newman. Power laws, pareto distributions and zipfs law. 2006.

[80] Claudio Marcus. A practical yet meaningful approach to customer segmentation. *Journal of Consumer Marketing*, 15(5):494–504, 10 1998.

[81] Derya Birant and Alp Kut. St-dbscan: An algorithm for clustering spatialtemporal data. *Data & Knowledge Engineering*, 60(1):208–221, 01 2007.