OPTIMIZING QUOTATION BASED BUSINESS TO BUSINESS PRICING

KAAN APAK

ÖZYEĞİN UNIVERSITY

FEBRUARY, 2025

OPTIMIZING QUOTATION BASED BUSINESS TO BUSINESS PRICING

by
Kaan Apak

Thesis
Submitted in Partial Fulfillment of the
Requirements for the Degree of

Master of Science

in
Data Science

Advisor: Assoc.Prof.Enis Kayış

Graduate School of Science and Engineering
Özyeğin University
İstanbul

February, 2025

OPTIMIZING QUOTATION BASED BUSINESS TO BUSINESS PRICING

Approved by:

Assoc.Prof.Enis Kayış, Advisor
Department of Industrial Engineering
Özyeğin University

Prof.Erhun Kundakcıoğlu
Department of Industrial Engineering
Özyeğin University

Prof.M.Güray Güler
Department of Industrial Engineering
Istanbul Technical University

Approval Date: December 30, 2024

To all who dedicate themselves to change

DECLARATION OF ORIGINALITY

I hereby declare that I am the sole author of this thesis and that this is the true copy of my thesis, including the final revisions, approved by my thesis committee. All data and information have been obtained, produced, and presented in accordance with the rules of research ethics and principles of academic honesty. As required by these rules, to the best of my knowledge I have acknowledged ideas, thoughts, and any copyrighted material in accordance with the standard referencing rules. I certify that any part of this thesis has not been submitted for a degree or diploma in another educational institution.

Kaan Apak

ABSTRACT

This paper focuses on quotation based pricing for a Business to Business (B2B) firm, using a real world case of a distributor engaged in aftermarket sales. The study assigns quotation groups into clusters, each associated with distinct logistic regression curves, to enhance overall performance and define the elasticities of various quotation groups. The objective of this study is to determine the optimal assignment of groups to clusters while simultaneously identifying the corresponding cluster wise regression coefficients. Coupled with the combinatorial nature of the task, finding an optimal solution is particularly challenging and computationally intensive. Additionally, we introduce a complex teacher model designed to improve the performance while using logistic regression model to enhance interpretability. To generate clusters, apart from using solvers, we propose heuristic algorithms such as the Recursive Partitioning Algorithm and the Iterative Refit Classifier Algorithm to address the problem. The performance of these approaches is evaluated using both real world and simulated datasets. Furthermore, we demonstrate the impacts of implementing quotation based pricing strategies using the real dataset.

Keywords: Quotation Based Pricing, After Market Sales, Elasticity, Clustering, Logistic Regression, Optimization, Heuristic Algorithms

ÖZET

Bu makale, bir İşletmeden İşletmeye (B2B) firmanın fiyat teklifi tabanlı fiyatlandırmasına odaklanmakta ve satış sonrası hizmetler alanında faaliyet gösteren bir distribütörün gerçek bir vakasını ele almaktadır. Çalışmada, fiyat teklifi grupları kümelere atanmış ve her küme, farklı lojistik regresyon eğrileriyle ilişkilendirilerek genel performansı artırmak ve çeşitli fiyat teklifi gruplarının elastikiyetlerini tanımlamak hedeflenmiştir. Bu çalışmanın amacı, grupların kümelere en uygun şekilde atanmasını sağlamak ve aynı anda her kümeye ait regresyon katsayılarını belirlemektir. Görev, kombinatoryal yapısı nedeniyle çözüm bulmayı hem oldukça zorlu hem de hesaplama açısından yoğun bir süreç haline getirmektedir. Ayrıca, kümeleme metodolojisinin performansını iyileştirmek ve karmaşıklığını azaltmak amacıyla bir öğretici model geliştirilmiştir. Kümeler oluşturmak için çözücüler kullanmanın yanı sıra, çeşitli sezgisel algoritmalar önerilmiştir. Bu yaklaşımların performansı, hem gerçek dünya verileriyle hem de simüle edilmiş veri setleriyle değerlendirilmiştir. Ayrıca, gerçek veri seti kullanılarak fiyat teklifi tabanlı fiyatlandırma stratejilerinin uygulanmasının olası etkileri gösterilmiştir.

Anahtar Kelimeler:Teklif Bazlı Fiyatlandırma, Müşteri Destek Satışları, Elastikiyet, Lojistik Regresyon, Optimizasyon, Sezgisel Algoritmalar

ACKNOWLEDGEMENTS

Throughout the implementation of this project, I encountered various challenges and limitations. Overcoming these difficulties was only possible through the commitment and invaluable guidance of my advisor, Assoc.Prof.Enis Kayış. I am deeply grateful for his support and insight.

This project also aimed to address real business problems. I sincerely thank my manager Yeşim Temizer, who gave me the opportunity to undertake this project and develop it as my thesis. Her encouragement played a pivotal role in my personal and professional growth.

My master's education would not have been possible without the inspiration and mentorship of Prof.Burcu Balçık. From her, I gained a deep understanding of project development, and the importance of continuous improvement. Her dedication has been a constant source of motivation.

Lastly, I am grateful to my family, who have always been by my side, offering unwavering support and love throughout this journey.

TABLE OF CONTENTS

LIST OF TABLES

LIST OF FIGURES

LIST OF ACRONYMS AND ABBREVIATIONS

B2B	Business to Business
CRM	Customer Relationship Management
B2C	Business to Consumer
AI	Artificial Intelligence
IRC	Iterative Refit Classification
GM	Gross Margin
OML	Output Teacher Model
MRP	Model-Based Recursive Partitioning
KPI	Key Performance Indicator
MLL	Average Log Loss
RCS	Refitted Construction Solution
ARCS	RCS Achievement Ratio
CIR	Cluster Impact Ratio
GB	Gigabyte
ID	Identifier
SKU	Stock Keeping Unit
LL	Log Loss
EPP	Expected Percentage Profit
SVM	Support Vector Machine
OP	Opportunity Focus Percentage

VIP	Very Important Person
CPU	Central Processing Unit
RSM	Random Subspace Method
MML	Mean Margin Loss
GPU	Graphics Processing Unit
ML	Margin Loss
CI	Cluster Increase Rate

1. INTRODUCTION

For firms, it is crucial to match their products and services with coherent pricing strategies since a 5% increase in pricing leads to a 22% improvement in earnings before tax [1]. In addition, relevant pricing strategies can only achieve high market share and high prices at the same time. For Business to Business (B2B) companies, customers evaluate the product based on its contribution to their value chain, so it is important to decide the right pricing strategies that meet the needs of the consumer [2]. Therefore, the literature underlines the need for market-based pricing in B2B business [3]. Fortunately, B2B companies have access to tools such as Customer Relationship Management (CRM) systems, which are not typically available to Business to Consumer (B2C) firms.

CRM software is a great data source that highlights the entire sales funnel: opportunities, leads, quotations, and at the end, invoiced sales. CRM data underlie the history of the relationship between manufacturers and consumers and show a network of interactions [4]. In sales funnel data, companies can view sales together with rejected quotations that signal false pricing approaches. Only with Artificial Intelligence (AI), B2B companies can transform those big data from their sales funnel into pricing strategies. With AI, firms can personalize their services with the power of sales funnel data [4].

Manufacturers with aftermarket services are B2B companies that work intensively on pricing, as aftermarket sales are not only a liability but have become their main source of income [5]. Therefore, in this thesis, we will focus on the dynamic spare parts pricing case of a distributor with aftermarket services. The aftermarket is an important part of the customer journey and greatly influences the customer experience. In addition, aftermarket performance is an important factor for customer loyalty and determines future main product agreements.

Pricing analysts within these firms prefer models that are easy to understand and can provide actionable insights. In practice, professionals may choose models that, while

not the most accurate, offer explainability [6]. In this study, pricing analysts need an interpretable model to adjust the gross margin (m) of a quotation, which is the profit-to-sales ratio. Firms must find the correct Gross Margin (GM) for each of the quotations considering all these factors to maximize their profitability. Due to their inherent interpretable [7] nature, logistic regression models can be an option.

However, in spare parts pricing, there is a vast number of units. For instance, the commercial airplane Airbus A380 contains around 2.5 million individual parts (Airbus Press Office, 2017). Furthermore, sufficient part-wise historical transaction data is not available to decide on pricing. Fortunately, firms have segmentation of customers and part levels that indicate which component a part belongs to. For instance, a part level indicates that a given part is placed in an engine, undercarriage, hydraulics, or filter that should be replaced periodically. That gives important clues for the customer needs and urgency of purchasing that part. In addition, it can be an indicator of competition, as there is high competition between distributors, pirates, and parallel importers for some of the parts levels [8].

Furthermore, there can be many part levels in a single quote, and clients accept the quotations when they are comfortable with the overall price [2]. Quotations are bundles that customers don't criticize parts-wise. Consequently, a single part in a quotation may determine the acceptance of the whole quotation, and to evaluate the elasticity of a quotation, all part levels in a quotation should be considered together.

In addition, B2B companies have customers with various profiles and necessities. In aftermarket sales cases, for some customers and some situations, price is not even a factor; however, for some cases, in a machine breakdown, the consumer needs to reach the spare parts as soon as possible. Therefore, firms have customer segmentation to differentiate customers.

Consequently, B2B firms need to consider characteristics of product levels and

customer characteristics in quotations at the same time. In this study, quotation groups are defined by the segment of the quotation's customer and the levels of parts included in the quotation. However, there is no evidence of elasticity or the right pricing strategy for these predefined quotation groups [8]. Most of the time, these groups do not contain the necessary amount of data with price variation to evaluate price elasticity. With only a limited amount of data, determining pricing strategy for a quotation group is not possible.

Since funnel data are heterogeneously elastic, firms need to generate clusters from these groups that signal the elasticity of the groups, using data from the previous sales funnel. Each of these clusters' logistic regression curves can have cluster-wise determined intercept and coefficient values.

Within each cluster, logistic regression models are used to represent the connection between the features and the acceptance of a quotation. All quotation groups within the same cluster share the same regression curve and coefficient values. These curves underlie the acceptance probabilities of quotations toward a range of possible gross margin.

This study aims to assign quotation groups into clusters with different logistic regression curves to increase overall log likelihood performance and define the elasticities of various quotation groups. The regression model also uses another more complex ensemble model (teacher) that replaces all other features except m with a single feature, Output Teacher Model (OML), and increases accuracy while preserving explainability.

In this study, a mixed integer nonlinear model is proposed to define the logistic regression clustering problem. The results of the solver are used as a benchmark to assess the heuristic algorithms developed in this study. The solution approaches include the Recursive Partitioning algorithm and two variants of the Iterative Refit Classification (IRC) algorithm, which are effective in clustering quotation groups within real datasets.

As a result, firms can use generated clusters to match quotation groups with pricing strategies. Moreover, a logistic regression curve and Expected Percentage Profit

(EPP) curves can be used to find the optimal m of a given quote. Crucially, this approach enables B2B firms to adopt data-driven and quotation-based pricing strategies.

The remainder of the thesis is organized as follows; Section 2 presents a view of existing studies and underlies distinctions of our study. Afterwards, Section 3 defines the problem and presents a formulation. Section 4 covers our methodology: the teacher-student model, the exact solution, and other solution methods as proposed heuristics. Section 5 presents results and presents how clustering captures elasticity through simulated and case data. Section 6 includes a discussion that underlines the outcomes for B2B firms.

2. LITERATURE REVIEW

Generally, price elasticity has been studied extensively and assessed as a coefficient of a regression equation. Regression equations for the estimation of price elasticity are used in various fields, such as the price elasticity of municipal water [9] the demand for gasoline [10] the price elasticity of electricity consumption [11] and market level appliance [12]. Wang [13] applied the Support Vector Machine (SVM) model for clustering before using regression to estimate the price elasticity in the electricity market. In addition, Michinaka [14] applied the k-medoids method to generate regression clusters.

Thus, to generate clusters that can maximize overall accuracy, clustering should be done together with regression to capture the dynamics of acceptance of the quote. Due to the highly combinatorial nature of this problem, Spath [15] introduces a heuristic that iteratively relocates data points. Qian [16] used a similar approach and proposed the least squares criterion to decide the number of clusters. This algorithm iteratively refits regression models and relocates data points based on their distances from the regression curve, forming clusters while treating outlier clusters as separate clusters.

In addition, global optimization of the clusterwise regression is possible via mathematical programming [17] nevertheless, these models can be highly complex. Although big M reformulation may accelerate solutions in some problems, it carries the risk of yielding non-optimal or infeasible results. In addition, Kayış [18] introduced a gradient descent-based heuristic for regression clustering of data with predefined groups. However, none of these studies are focused on logistic regression clustering problems with non-linear objective function that we need to apply to our data with binary target features.

As pricing applications, to estimate price elasticity through multiple regressions, Sui [19] introduced approaches that calculate elasticities with breakpoints and with different quantity-price combinations using Quantile on Quantile Regression. Kayış [20] proposed heuristics for regression clustering to group different product levels with similar

price elasticities. In this study, similar to our approach, cluster wise linear regressions explain the variability in sales quantity using variables that include price.

As a contribution, our article incorporates the rejected sales in the sales funnel data into decision-making processes. Distinctively, elasticities in this study are neither Stock Keeping Unit (SKU) nor product level, quotation group-based elasticities are defined. Furthermore, our problem uniquely requires an explainable model for gross margin and a complex model for other features to increase performance. With these new requirements, regression clustering of data with binary target features requires the development of new approaches.

3. PROBLEM DESCRIPTION

In this section, details of the problem and mathematical formulations are presented. Our mixed integer non-linear model that presented in Section 3.1, uses historical quotation data, assigns quotation groups to clusters, and finds cluster-wise regression coefficients simultaneously. In addition, the solution of this model can be utilized to determine the Expected Percentage Profit (EPP) curve for a quotation, as described in Section 3.2.

3.1 Quotation Group Clustering Problem

Let $y_i \in \{0, 1\}$ present the acceptance of the quotation i and x_i^j denote the quotation data that include $|J|$ independent variables associated with the quotation. The first variable of x (x_i^1) is the gross margin (m), which is the percentage of the overall profit of the firm from the quotation and is used as an indicator of the pricing decision. Generally, a B2B firm's goal is to maximize m without increasing rejected quotations. The remaining $|J| - 1$ variables x_i^j (for $j = 2, \ldots, |J|$) provide additional indicators to assess the elasticity of a quotation. The diversity of variables increases the likelihood of capturing the variability of the acceptance of the quotation. However, this strength comes with a burden: high dimensionality.

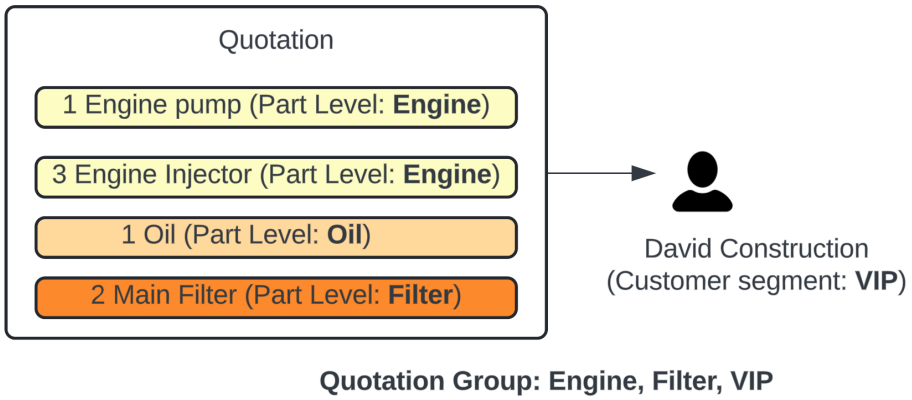
These variables increase the complexity of the logistic regression model and clustering approaches. Moreover, for these variables, rather than an easily interpretable model, we require one that can capture the complex relationships among these variables and the acceptance of a quotation. At the same time, we need a highly interpretable model for gross margin that gives a differentiable function, which can be used in our mathematical model to give insight for decision makers.

This model will be used in our methods to categorize quotations, taking into account the customer's segment and the levels of the included parts. Since a quotation

can have multiple parts with various part levels, we define quotation groups by customer segment with the first and second highest quantity part level in the quotation. With a predetermined set of rules, each quotation i is labeled with a quotation group $g \in G$. The binary matrix $\alpha_{ig} \in \{0, 1\}$ denotes which group g quotation i belongs to.

In Figure 3.1, we represent an example quotation to a customer, David Construction. It is a customer with a high number of construction equipments and a high sales frequency, therefore it is in the Very Important Person (VIP) segment. The quotations consist of numerous items (referred to as parts in our case), and in the given example, there are four distinct parts. Similar to customers, parts are classified with part levels. In the provided quotation, both the engine pump and injector are categorized under the Engine part level, which is the level with the highest quantity in that quotation. Therefore, represented quotation, belongs to the same quotation group α_{ig} as other quotations with a VIP customer segment, Engine as the highest level part, and Filter as the second highest level part.

Figure 3.1: *Representation of example quotation from the case data*



We assume that elasticity differentiates between various quotation groups. Therefore, it's essential to cluster groups with similar relationships to develop targeted, cluster-based strategies. For each cluster, all groups within the cluster share the same values of logistic regression coefficients. The objective of this study is to find the best possible

assignment of groups to clusters ($\tilde{\gamma}$) and to find corresponding cluster-wise coefficients ($\tilde{\beta}$) that maximize the overall log likelihood. The binary matrix $\tilde{\gamma}_{gk} \in 0, 1$ represents the assignment of group g to cluster k , while $\tilde{\beta}_k \in \mathbb{R}$ denotes the coefficient of the corresponding cluster, for $k \in K$. Note that in our problem, a group can only be in one cluster (3.3) and each cluster should contain at least one group (3.4).

In this paper, the binary matrix α denotes the labeling of quotations in groups, and the binary matrix $\tilde{\gamma}$ represents the assignment of groups to clusters and $\tilde{\beta}$ includes the values of the cluster wise coefficients values of each variable. Therefore, the quotation-wise coefficient values can be expressed with the matrix multiplication of these matrices. Moreover, $\tilde{z}$ can be described as the sum of the element-wise product of coefficient values with x (3.2) (in this paper, y^T denotes the transpose of the matrix y and $\circ$ represents Hessian multiplication).

The probability of acceptance of quotations can be expressed with a sigmoid function $\sigma(\tilde{z})$ and the objective function of our mathematical model is the total log-likelihood of logistic regression curves (3.1). We use log likelihood instead of accuracy because it is not a piecewise function and provides a measure of the strength of the prediction rather than just a binary classification outcome. The simplification of the log-likelihood formulation presented in Appendix A. Additionally, our objective function indirectly maximizes log loss, which is derived by dividing the log likelihood by the size of the data set and will be used as an additional metric to assess the performance of a solution.

The mathematical model of the described problem is presented below. The matrix notation is used for the formulation of $\tilde{\boldsymbol{z}}$ (3.2) for the compact representation (the boldface font is used to denote vector representations), while indexed summation is used for rest of the model.

Sets

I: Set of quotations ;

K: Set of clusters ;

G: set of groups;

J: set of variables in x;

Parameters

$$\alpha_{ig} = \begin{cases} 1, & \text{if quotation i} \in \text{I is in group g} \in \text{G} \\ 0, & \text{otherwise} \end{cases}$$

x_i^j: Value of variable j for quotation $i \in I$

$$y_i : \begin{cases} 1, & \text{if quotation } i \in I \text{ is accepted} \\ 0, & \text{quotation } i \in I \text{ is rejected} \end{cases}$$

Decision Variables

$$\tilde{\gamma}_{gk} : \begin{cases} 1, & \text{if group g} \in \text{G is in cluster k} \in \text{K} \\ 0, & \text{otherwise} \end{cases}$$

$\tilde{\beta}_k^j$: Coefficient value of variable $j \in J$ for cluster $k \in K$

$\tilde{z}_i$: Linear multiplication of coefficient values and variable values of quotation i $\in$ I

$$\max \sum_{i \in I} y_i \left(-\log\left(1 + e^{-z_i}\right)\right) + (1 - y_i)\left(-\log\left(1 + e^{-z_i}\right) - z_i\right) \tag{3.1}$$

$$\tilde{\boldsymbol{z}} = \left(\left(\boldsymbol{\alpha}\,\tilde{\boldsymbol{\gamma}}\,\tilde{\boldsymbol{\beta}}\right) \circ \mathbf{x}\right)\mathbf{1}^T \tag{3.2}$$

$$\sum_{k \in K} \tilde{\gamma}_{gk} = 1 \quad \forall g \in G \tag{3.3}$$

$$\sum_{g \in G} \tilde{\gamma}_{gk} \geq 0 \quad \forall k \in K \tag{3.4}$$

$$\tilde{\gamma}_{gk} \in \{0, 1\} \tag{3.5}$$

As every quotation within a quotation group must be grouped together in the same cluster, with $|G|$ groups and with a predetermined number of clusters (R) there are $|G|^R$ potential ways to partition them. Since both the clustering and the regression coefficients

are interdependent, they influence each other and the final partition. The proposed problem involves a mixed-integer nonlinear and non-convex model, combined with the combinatorial nature of the problem, finding an optimal solution is particularly challenging and computationally rigorous.

3.2 Optimal Gross Margin Values - Maximization of EPP

Using the coefficient matrix and the cluster assignments, we can determine the Expected Percentage Profit (EPP_i) by multiplying the m by the probability of accepting a quotation ($\tilde{p}(m)$), following formulation used in many studies, such as Guillen [21], Carvalho [22], and Qu [23]. The equation of EPP_m with only the gross margin and the intercept is presented as follows:

$$\begin{aligned} E\tilde{P}P_i(m) &= m\tilde{p}(m) \\ &= m\frac{1}{1+e^{-(\sum_g^G \sum_k^K \alpha_{ig}\tilde{\gamma}_{gk}\tilde{\beta}_k^I + \alpha_{ig}\tilde{\gamma}_{gk}\tilde{\beta}_k^m m)}} \\ &= \frac{m}{1+e^{-(\sum_g^G \sum_k^K \alpha_{ig}\tilde{\gamma}_{gk}\tilde{\beta}_k^I + \alpha_{ig}\tilde{\gamma}_{gk}\tilde{\beta}_k^m m)}}. \end{aligned} \tag{3.6}$$

Firms need to apply strategies that maximize their profit without compromising the number of expected quotation. Consequently, the value of m that provides the highest $E\tilde{P}P_i$ is the optimal value m and is denoted as $\tilde{\theta}_i$. The mathematical model that finds $\tilde{\theta}_i$ is presented in Appendix B.

Here, our goal is to find the best $\tilde{\theta}_i$ for each of the quotations. As we are not addressing a resource allocation problem involving shared resources, in our independent quotation problem, achieving the highest possible value EPP_i is equally crucial for all quotations, regardless of the value of the quotation.

If the regression curve is already determined, with an exhaustive enumerative approach, one can try all possible values of m with an acceptable step size to find $\tilde{\theta}_i$. Therefore, a challenging aspect lies in deciding on clusters of quotation groups.

4. METHODOLOGY

Our mathematical model is challenged with numerous features and can still not achieve high accuracy with interpretable models. Furthermore, our problem requires both interpretability and complex models to represent the complex relationship of our indicators. To address this challenge, in Section 4.1, a teacher model is presented that simultaneously improves the performance of the model and decreases the complexity of our clustering methodologies. This teacher model replaces all the variables except the gross margin with a single feature: OML.

In this study, our main focus is assigning quotation groups to clusters and finding coefficient values for each of the clusters. This will be a critical input for B2B firms in the reshaping of their pricing strategies. Even after using OML, this is challenging due to the complexity of our model and the large number of combinations. In Section 4.2 for the exact solution of the problem, we apply strategies to improve solver solutions and present another variant of our model to reduce complexity. Still, commercial solvers are not developed for non convex models and use heuristic algorithms, therefore we need to develop alternative approaches.

Therefore, in Section 4.3, we use the recursive partitioning approach to generate decision trees, proposed by Zeileis [24]. Also, for cases where this algorithm generates a higher number of clusters than the requested number of clusters, we present a pruning algorithm. Although this approach finds competitive results, it cannot work with a large dataset with a high number of groups.

Thus, in Section 4.4, we present the iterative refit classifier (IRC) heuristic, which generates an initial solution and in each iteration moves groups while refitting regression curves. Our base IRC algorithm presented in Section 4.4.1 makes moves efficiently and achieves a high overall likelihood, since it moves groups to a cluster that maximizes their log likelihood. However, this approach leads to local optimum solutions in some

of the datasets. The Algorithm successfully exploits, although, it hardly explores new solutions. Consequently, in Section 4.4.2, we introduced a Simulated Annealing Inspired IRC, which improves upon the basic IRC solution utilizing a meta heuristic algorithm formulated by Kirkpatrick [25].

4.1 Teacher - Student Model (OML)

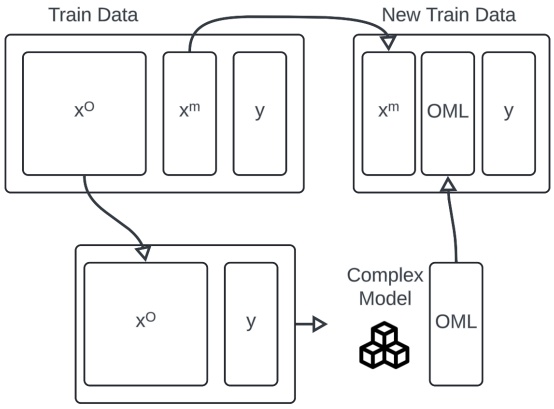
In x_i^j, there is a gross margin variable (x_i^1) and the remaining $|J| - 1$ variables x_i^j (for $j = 2, \ldots, |J|$). As discussed in the description of the problem (Section 3), our problem requires interpretable models for x_i^1 and a complex model that increase log likelihood of our approach for other variables. Thus, we separate x_i^j into two components: the gross margin (x_i^m) of the quotation i, which encompasses only x_i^1, and x^O, which includes the remaining elements of x_i^j.

x^O includes variables that influence the acceptance of a quotation independently of the decided pricing strategy. We use a complex black-box teacher model that processes features in x^O and generates a probability of acceptance (OML). That model can learn complex relationships of quotation acceptance. Afterward, as illustrated in Figure 4.1, the interpretable student model (logistic regression) uses only OML instead of x_i^O, together with the x_i^m. As a result, distilling data lowers the number of coefficients in logistic regression, increases the accuracy of our approach, and lowers the complexity of the mathematical model while preserving interpretability. The resultant logistic curve underlies different acceptance probabilities under all possible m values. Our reformulated mathematical model with x_i^m and OML_i is presented in Appendix C.

4.2 Exact Solution

Since the mathematical model of our problem (introduced in Section 3), includes binary decision variables and non-convex nonlinear objective function, to find an exact

Figure 4.1: *The structure of OML generation and model with OML, x^m*



solution, it can only be solved with specialized solvers.

However, most commercial solvers are designed for convex problems and for non-convex models they apply heuristic algorithm and can find local-optimal solutions. Therefore, the coefficients derived from the solvers are not optimally calibrated. To increase the performance of the solver, using the solver's assignment ($\tilde{\gamma}_{gk}$), we can easily refit the coefficients of each cluster to minimize log-likelihood.

In addition, we introduce a variant of our base model in which we relax the binary variable ($\tilde{\gamma}_{gk}$) of the model to reduce complexity (4.4). In this relaxation $\tilde{\gamma}_{gk}$ becomes a continuous variable in the range [0,1] and used a penalty (ρ^q_{gk}) value to hold $\tilde{\gamma}_{gk}$ close to 0 or 1. We add summation of penalty values to the objective with a multiplier of l (4.1) and also limit the penalty with limit of l^2 (4.2). As a function of ρ^q_{gk}, we use a quadratic penalty function (4.3) which is first introduced by Giannessi [26]. Essentially, ρ^q_{gk} increases as it approaches 0.5.

The relaxed model presented is shown below. The constraints (3.2), (3.3), and (3.4) are preserved in the relaxed model without any modification.

$$\max \sum_{i=i}^{I} [y_i \left(-\log\left(1+e^{-z_i}\right)\right) + (1-y_i)\left(-\log\left(1+e^{-z_i}\right) - z_i\right)] - (l * \sum_{g}^{G} \sum_{k}^{K} \rho^q_{gk}) \quad (4.1)$$

$$\rho^q_{gk} <= l^2 \quad \forall g \quad \forall k \tag{4.2}$$

$$\rho^q_{gk} = \tilde{\gamma}_{gk}(1 - \tilde{\gamma}_{gk}) \quad \forall g \quad \forall k \tag{4.3}$$

$$\tilde{\gamma}_{gk} \in \mathbb{R} \tag{4.4}$$

Afterwards, we round the values of $\tilde{\gamma}_{gk}$ to the nearest binary value and refit each cluster as we did with the solutions of the base model.

4.3 Recursive Partitioning

To divide groups into various clusters, splitting groups recursively can be a great strategy. Zeileis [24] introduces a Model-Based Recursive Partitioning (MRP) algorithm that can form a decision tree with a fitted logistic regression model at each of its leaf (terminal) nodes, and these leaf nodes can be used as clusters in our proposed model.

MRP algorithm starts with the entire data and in each iteration selects a partitioning variable by evaluating the stability of the target feature (y). In our case, there is only one partitioning variable, namely the quotation groups. To determine the split point, an exhaustive search is performed by fitting a model in each child node for all possible split points. By applying the same procedure, the algorithm recursively splits each child node until stability is achieved.

However, this approach may give leaf nodes more than the requested number of clusters (R). Therefore, we need to develop an algorithm that prunes some of the leaves of this tree until we reach the target number of clusters. As described in the Pseudo Code 1, in the first phase, we use the MRP algorithm with quotation groups as a classifier.

Then, in Phase 2 until we reach to R clusters, in each iteration, we select a source node and combine that node with the target node. In the combination procedure, we combine a leaf node with its sibling leaf node (nodes that share the same parent node) so that the number of leaf nodes decreases by one. Therefore, as a source node, we need to select a leaf node that has a sibling. In addition, this should be a node that is

Algorithm 1 Iterative Regression Tree Pruning

Phase 1: Initial Tree Construction

Run MRP algorithm with categorical Group Identifier (ID). Each group g is assigned to a leaf node (N_g) of the decision tree.

Each node of the tree has coefficient values of m, represented as $\tilde{\beta}_n^m$.

Phase 2: Pruning Iteration

while the tree has more than R leaf nodes **do**
  for all non-leaf node p **do**
    Let V_p denote the set of descendant nodes of p.
    Calculate the average value of coefficient m ($\tilde{\beta}_n^m$) of nodes in V_p:

$$S_p^m = \frac{1}{|V_p|} \sum_{n \in V_p} \tilde{\beta}_n^m.$$

  end for
  for all leaf node n with a sibling **do**
    Calculate the difference:

$$\Delta_n^r = \left| \tilde{\beta}_n^m - S_{n'}^m \right|,$$

    where n' is the parent node of n.
  end for
  Select the leaf node with the minimum Δ_n^r as the *source node*.
  Select the sibling node of the source node as the *target node*.
  Reassign groups associated with the source node to the target node.
  Re-run MRP algorithm with the updated N_g.
end while

least significantly differentiated from its sibling. For that, we first define S_p^m for each non-leaf node p, that is, the average coefficient values of the descendant nodes of node p. Afterwards, for every candidate node n, we calculate the difference (Δ_n^r) between its coefficient values ($\tilde{\beta}_n^m$) and the S_p^m value of its parent node (n').

In essence, Δ_n^r increases when node n has coefficient values that differ from those of its siblings. For example, combining a node with a significant value of $\tilde{\beta}_n^m$ with a node that contains groups whose acceptance is not influenced by the gross margin will result in a loss of information. Therefore, we select a node with minimum Δ_n^r as a source node and assign all groups of the source node to the target node (sibling node). Afterwards, at the end of the iteration we use the MRP algorithm again with new node ids, rather than group ids, as we did in the initial run. The new tree derived, will have one less leaf node, however, if there remain more than R leaf nodes, the algorithm will proceed.

This approach enables us to use the MRP algorithm with a predetermined number of leaf nodes and creates a great alternative to generate clusters considering cluster-wise logistic regression curves.

4.4 Iterative Refit Classifier (IRC) Heuristic

This study presents variants of the IRC algorithm to classify quotation groups considering the log likelihood of the cluster-wise regression curve. IRC algorithm can solve a wide range of instances. Because, instead of trying all possible combinations, IRC starts with a feasible solution and moves groups through iterations. IRC improves the log likelihood of the groups and, as a result, increases the overall log likelihood.

4.4.1 Base IRC

In the first phase of the Base IRC algorithm, we create an initial solution. As described in Pseudo Code 2, we calculate group-wise intercept ($\tilde{\beta}_g^I$), coefficient of gross margin ($\tilde{\beta}_g^m$), and coefficient of the OML ($\tilde{\beta}_g^{OML}$) by fitting separate logistic regression

curves for each of the groups. Then, we use $\tilde{\beta}_g^m, \tilde{\beta}_g^{OML}$, as features of the k-center algorithm to generate K clusters.

Algorithm 2 Base Iterative Refit Classifier

Phase 1: K-Center Initialization

Fit group-wise regressions. For each g, determine coefficient values: $\tilde{\beta}_g^I, \tilde{\beta}_g^m, \tilde{\beta}_g^{OML}$.
Initialize k clusters with the k-center algorithm using $(\tilde{\beta}_g^m, \tilde{\beta}_g^{OML})$ as features.
Each cluster's group set is represented by S_k, and each group's assigned cluster is denoted as C_g.

Phase 2: Improving Initial Solutions

for $t = 1$ to 10 iterations **do**
  Randomly shuffle the order of groups in G.
  for all Group $g_i \in G$ **do**
    if $|S_{C_{g_i}}| > 1$ **then**
      Phase 2.1: Evaluation with Refit Trials
      for all Cluster $k_i \in K$ **do**
        if $g_i \in S_{k_i}$ **then**
          Fit coefficients for combined data of groups S_{k_i}: $\tilde{\beta}_{k_i}^I, \tilde{\beta}_{k_i}^m, \tilde{\beta}_{k_i}^{OML}$.
        else
          Fit coefficients for combined data of groups $S_{k_i} \cup \{g_i\}$: $\tilde{\beta}_{k_i}^I, \tilde{\beta}_{k_i}^m, \tilde{\beta}_{k_i}^{OML}$.
        end if
        Compute log-likelihood $L_{g_i k_i}$ for group g_i, when placed in cluster k_i with coefficients $\tilde{\beta}_{k_i}^I, \tilde{\beta}_{k_i}^m, \tilde{\beta}_{k_i}^{OML}$.
      end for
      Select the cluster k with the maximum log-likelihood $L_{g_i k_i}$: n_1.
      Phase 2.2: Move
      if $n_1 \neq C_{g_i}$ **then**
        Move g_i to cluster n_1. Update S_k, C_g.
      end if
    end if
  end for
end for

Afterwards, in the second phase, we improve this initial solution by moving groups. Over a number of iterations, in sequence, we evaluate the best-fit cluster for each group. In this context, the sequence of group evaluations can influence the outcome, so we randomly shuffle the order of group evaluations in each iteration. The evaluation of group g_i takes place in two stages: first, identifying the most suitable cluster (Phase 2.1) and then

moving g_i to this selected cluster (n_1) in Phase 2.2.

During Phase 2.1, to determine the optimal assignment of the cluster for g_i, we must calculate the log likelihood values for that group as a member of each possible cluster choice. If g_i is already part of cluster k_i, we can determine the coefficient values using it's current groups. Otherwise, the addition of a new group will have an impact on the cluster's regression curve. Therefore, we refit the cluster-wise coefficients before calculating the log-likelihood values of g_i. Consequently, $L_{g_i k_i}$ represents the log likelihood of g_i when moved to the group k_i. In the base heuristic IRC, we select the cluster (n_1) with the highest $L_{g_i k_i}$, that is, the local optimal movement for that group. As a result, in the Phase 2.2, if that cluster wasn't that gorup's current cluster ($n_1 \neq C_{g_i}$), we move that group to cluster n_1.

As a result of this algorithm, we need to construct exactly R clusters; therefore, we need to implement a strategy to avoid solutions with fewer clusters. Therefore, we avoid evaluating the movement of a group that is the only member of its cluster ($|S_{C_{g_i}}| = 1$), as moving that group would leave the cluster empty.

Furthermore, in Appendix E we present other variants of IRC and show a different approach to finding an initial solution and preserving the number of clusters. Also, there is a variant that we did not refit coefficients during iterations, and in another variant, we allow the algorithm to have empty clusters.

In the rest of this thesis, the base IRC algorithm that we have presented in this section (in Pseudo Code 2) is used. It can efficiently move groups through iterations while preserving the number of clusters and provides competitive solutions.

4.4.2 Simulated Annealing Inspired IRC

The IRC algorithm cannot reach global optimal solutions since it can find local optimal solutions for some of the datasets, and we need to explore the feasible region

more for these instances. Since there is a trade-off between exploit and explore, it is better to explore more in the first iterations and then improve the solution.

To achieve this, Kirkpatrick [25] argues that there is a connection with multivariate combinatorial optimization problems and statistical mechanics. It is inspired by the annealing process in metallurgy, where a material is heated to a high temperature and then slowly cooled. Initially, the algorithm starts at a high temperature (melting phase) in which molecules move rapidly. In this phase, it can select worse solutions. The system maintains the same temperature for a predefined number of epochs (e) before reducing the temperature using a cooling schedule. As the temperature lowers, the selection of worse solutions becomes less likely. At low temperature, the system becomes stable (freezing phase). With that analogy, the algorithm uses a temperature parameter to adjust exploration and exploitation during iterations.

Through iteration, to accept a new solution, simulated annealing uses the Metropolis criterion [27]. The acceptance probability is calculated as $e^{-\Delta/T}$, where Δ represents the change in the value of the objective function and T is the current temperature. If a randomly generated number is less than this probability, the move is accepted.

We use this approach to create a new variant of IRC. In the generic simulated annealing algorithm, the cost function of the new solution is compared to the current solution. However, instead of comparing with the current solution's cost function, our approach evaluates group's log likelihood in different cluster alternatives. In our simulated annealing-inspired approach, we preserve the strengths of iterative refits in the IRC algorithm.

In the base IRC algorithm, after evaluating different groups for g_i, we select the cluster with the highest value of $L_{g_i k_i}$. In the annealing inspired algorithm, we refit different alternatives with the same approach as in the base algorithm. However, this time we also record the second highest cluster (n_2) option for that group and calculate

$\Delta_{g_i}([(L^{g_i}_{n_2} - L^{g_i}_{n_1})/L^{g_i}_{n_1}]u)$. Since we calculate the ratio of the change in the likelihood values, our determined initial temperature and cooling schedule can work with groups with various data sizes. This division is multiplied by u to rescale. The likelihood values are typically negative and $L^{g_i}_{n_1}$ is always greater than $L^{g_i}_{n_2}$ because n_1 represents a better solution. Consequently, always Δ_{g_i} takes a positive number, and it increases as the logarithmic likelihood difference increases.

To implement simulated annealing for our problem, as shown in Pseudo Code 3, we start with the solution of the IRC algorithm. We start with a given initial temperature, then decrease it by the cooling rate after a certain number of epochs.

Higher Δ_{g_i} values represent more risky moves, and with the presented metropolis criterion they are only accepted at high temperatures. With that metropolis criterion in Phase 2.2 of the algorithm n_1 (best possible) or n_2 (second possible) selected as a cluster (n_s) for g_i. Then, similar to the base algorithm, we move that group to n_s if that cluster was not that group's existing cluster.

Algorithm terminates when the temperature converges to zero. Through iterations, algorithm can find worse solutions, since simulated annealing algorithm can accept movements that decrease likelihood, hence, our final solution can be worse than the initial solution (Base IRC). Consequently, the best solution reached through iterations is accepted as the final solution.

Algorithm 3 Iterative Refit Classifier with Inspired Annealing

Phase 1: IRC Initialization

Initialize k clusters with the IRC algorithm.

Each cluster's group set is represented by S_k, and each group's assigned cluster is denoted as C_g.

Phase 2: Improving Initial Solutions

while Temperature $T > 0$ **do**
  for $e = 1$ to epoch size e **do**
    Randomly shuffle the order of groups in G.
    for all Group $g_i \in G$ **do**
      if $|S_{C_{g_i}}| > 1$ **then**
        Phase 2.1: Evaluation with Refit Trials
        for all Cluster $k_i \in K$ **do**
          if $g_i \in S_{k_i}$ **then**
            Fit coefficients for combined data of groups S_{k_i}: $\tilde{\beta}^I_{k_i}, \tilde{\beta}^m_{k_i}, \tilde{\beta}^{OML}_{k_i}$.
          else
            Fit coefficients for combined data of groups $S_{k_i} \cup \{g_i\}$: $\tilde{\beta}^I_{k_i}, \tilde{\beta}^m_{k_i}, \tilde{\beta}^{OML}_{k_i}$.
          end if
          Compute log-likelihood $L_{g_i k_i}$ for group g_i, when placed in cluster k_i with coefficients $\tilde{\beta}^I_{k_i}, \tilde{\beta}^m_{k_i}, \tilde{\beta}^{OML}_{k_i}$.
        end for
        Phase 2.2: Move
        Select k with the maximum and second maximum log-likelihood $L_{g_i k}$: n_1, n_2.
        Calculate the log-loss difference and multiply with a constant multiplier u:

$$\Delta_{g_i} = \frac{L^{g_i}_{n_2} - L^{g_i}_{n_1}}{L^{g_i}_{n_1}} u.$$

        if Metropolis Criterion (T, Δ_{g_i}) **then**
          $n_s = n_2$.
        else
          $n_s = n_1$.
        end if
        if $n_s \neq C_{g_i}$ **then**
          Move g_i to cluster n_s. Update S_k, C_g.
        end if
      end if
    end for
  end for
  Reduce T by $CoolingRate$.
end while
Select the best solution from all iterations.

5. RESULTS

In this section, we first explain the implementation of our proposed approaches in Section 5.1. Then, after explaining the preparation of the simulated data set (Section 5.2), we present the performance of our methodologies (Section 5.2.1) with various Key Performance Indicator (KPI). With a simulated data set, we can evaluate our solution approaches with many different data characteristics such as size and price elasticity (Section 5.2.2). Additionally, utilizing the simulated dataset allows us to employ the Expected Percentage Profit (EPP_i) values of the solutions (Section 5.2.3) to demonstrate the profit outcomes of different solutions.

In contrast, with real data sets we can use the case data and show how our algorithms developed can perform with large data, where the price relationship is not visible as simulated data sets (Section 5.2.2). In addition, we analyze the potential outcomes of employing solutions for pricing decisions (Section 5.3.2). Finally, we demonstrate solutions with different number of clusters (Section 5.3.3).

5.1 Computational Studies

To implement our proposed approaches, we utilized some of the available packages. First, to generate OML from x_j^O as presented in Section 4.1, we use the supervised AutoML library in Python. AutoML adds golden features, makes feature selections, develops ensemble models, and applies hyperparameter tuning. It makes a model selection according to the k-fold log loss. An overview of these steps and the AutoML results of our dataset is presented in Appendix D. From these results, we train a CatBoost model as a teacher model.

After OML generation, for exact solutions (in Section 4.2) we use Knitro 14 as a solver. Knitro is designed for large-scaled nonlinear problems and increases its flexibility by integrating the interior point approach with the active set approach [28]. It is capable

of solving a variety of problems and comparative numerical studies of Andrei [29] proved distinctive efficiency. Therefore, Knitro creates a great benchmark for evaluating other methodologies in this study.

In Section 4.3, we use the model-based recursive partitioning (MRP) algorithm and then iteratively prune the decision tree until we have a predetermined number of (R) leaf nodes. To implement the MRP algorithm introduced by Zeileis [24], we utilize the mob function of the Partykit library in R.

The IRC variants (Section 4.4) are implemented in Python. Parameters in Simulated Annealing Inspired IRC (Section 4.4.2); initial temperature, cooling rate, and multiplier (u), are determined with the basic grid search trials in simulated dataset. All computing is done on a machine with an Apple M4 chip, featuring an 8-core Central Processing Unit (CPU), 10-core Graphics Processing Unit (GPU), and 16 Gigabyte (GB) of unified memory.

5.2 Simulated Dataset

There are three different reasons for generating a simulated dataset. First, in the real data, it is not possible to determine the optimal portioning. However, for the simulated dataset, optimal clusters are known by construction, and hence we can evaluate the proposed approaches by using cluster assignments used in creation. Moreover, we can create simulated datasets with many different characteristics to show the performance of algorithms in different circumstances. In addition, most of our approaches are not capable of solving large instances. Therefore, it is not possible to show differences in real data. With only smaller data, we can show weaknesses and strengths of different solution approaches.

To evaluate the performance of our approaches, our simulated dataset includes 360 different data. In this section, i_d refers to the quotation i in data d. Consequently,

the value z_i in dataset d is represented as z_{i_d}. Furthermore, for each simulated data d, it is assumed that the acceptance of a quotation depends only on gross margin, denoted $x^m_{i_d}$. Therefore, similar to the formulation of our mathematical model (3.2) , the element-wise product (z_{i_d}) of the coefficient values for the data d can be represented as in Equation (5.1). To use (5.1), the construction values of the coefficients ($\beta^m_{k_d}$,$\beta^I_{k_d}$) and the assignment matrix (γ_{gk_d}) must be determined. In data d, the intercept values in the construction of the data ($\beta^I_{k_d}$) are randomly generated in the interval [0.5,1.5] and the values $x^m_{i_d}$ in the interval [0,1.5] with a uniform distribution. The regression error ϵ_i is generated with a mean of 0 and a standard deviation of s. The probability of acceptance of the quotation can be expressed with a sigmoid function $\sigma(z)$, and in our dataset, quotations with values of $\sigma(z)$ greater than 0.5 are assumed to be accepted quotations; otherwise, they are labeled as rejected.

Sets

D: Set of data;

I_d: Set of quotations in data $d \in D$;

K_d: Set of clusters in data $d \in D$;

G_d: set of groups in data $d \in D$;

Parameters used for data generation

α_{ig_d}: Assignment of quotation $i_d \in I_d$ to group $g_d \in G_d$ in data $d \in D$.

$x^m_{i_d}$: Value m of the quotation $i_d \in I_d$ in the data $d \in D$.

$\beta^I_{k_d}$: Intercept coefficient used in generation of cluster $k_d \in K_d$ in data $d \in D$.

γ_{gk_d}: Assignment matrix of groups to clusters in data $d \in D$.

$\beta^m_{k_d}$: Coefficient of gross margin (m) used in generation of cluster $k_d \in K_d$ in data $d \in D$.

ϵ_{i_d}: Regression error for quotation $i_d \in I_d$ in data $d \in D$.

$$z_{i_d} = \sum_{g_d}^{G_d} \sum_{k_d}^{K_d} \alpha_{ig_d}\gamma_{gk_d}\beta^I_{k_d} + \alpha_{ig_d}\gamma_{gk_d}\beta^m_{k_d}x^m_{i_d} + \epsilon_{i_d} \quad \forall i \quad \forall d \tag{5.1}$$

To generate clusters, we first create a cluster with a value of β_k^m of -1. This coefficient represents a low price elasticity, which fluctuations of the m have a minor effect on the acceptance of a quotation. For example, in clusters with high β_k^I and coefficient values of -1, quotations are accepted regardless of m. These groups represent the quotations where customers urgently need to reach that part and pricing is not a primary criterion in customer decision. Then other clusters are generated with values β_k^m with a gap of g^m ($\beta_k^m = (1-k)g^m - 1$).

As represented in Figures 5.1 and 5.2, as g^m increases, the difference in the price elasticity of the clusters increases. To generate a data, we must choose a s value from the options [0.1,0.2], a g^m value from the range of [-0.5,-1,-1.5], the number of clusters ($|K|$) among options of [2,3,4], the number of groups within each cluster ($|S_k|$) from [2,3,4], and the number of quotations ($|I_k|$)within each group from options [50,100,250,500].

Figure 5.1: *Plot of simulated data with g^m=0.5*

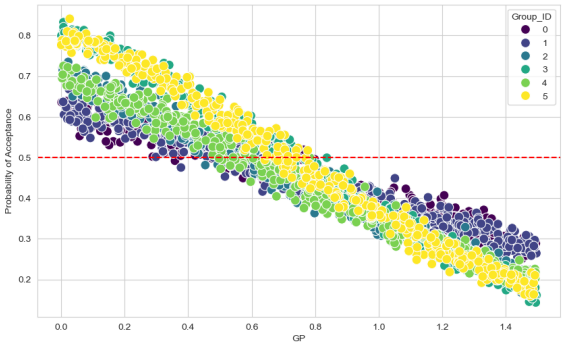
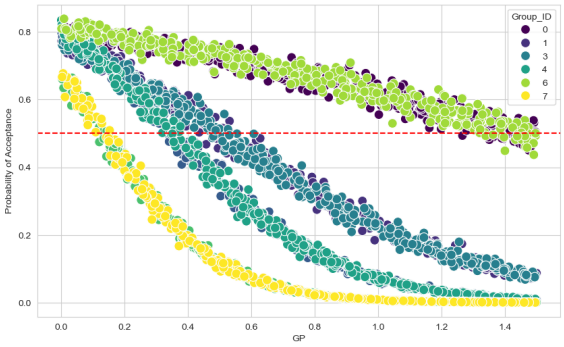


Figure 5.2: *Plot of simulated data with g^m=1.5*



We generate many data sets with different values of these parameters to evaluate our approaches under different conditions. For generating data, 3 different approaches are used, and data are generated in 3 batches. These approaches are described in the Appendix F. Following the generation of the data set, each data set is divided into training and testing sets (%20 of data in the test set) using stratified splitting.

5.2.1 Computational Results with Simulated Dataset

The objective of our mathematical model is the log likelihood, and without normalizing, it is not possible to compare log likelihood of data with different data sizes; in this section, to evaluate of the results of our approaches, we use log loss. Log Loss (LL) is the division of log likelihood into the number of instances in the data set. Since the only difference is a constant divisor, our mathematical model in Section 3.1 also optimizes log loss. The value of z_{i_d} for a given solution approach can be calculated as presented in Equation (5.3) and the log loss ($\tilde{LL}_d$) of the data d can be represented by the following formulation (5.2).

$$\tilde{LL}_d = \frac{\sum_{i_d}^{I_d} y_{i_d} \left(-\log\left(1+e^{-\tilde{z}_{i_d}}\right)\right) + \left(1-y_{i_d}\right)\left(-\log\left(1+e^{-\tilde{z}_{i_d}}\right) - \tilde{z}_{i_d}\right)}{|I_d|} \quad \forall d \quad (5.2)$$

$$\tilde{z}_{i_d} = \sum_{g_d}^{G_d} \sum_{k_d}^{K_d} \alpha_{ig_d} \tilde{\gamma}_{gk_d} \tilde{\beta}^I_{k_d} + \alpha_{ig_d} \tilde{\gamma}_{gk_d} \tilde{\beta}^m_{k_d} x^m_{i_d} \quad \forall i \quad \forall d \quad (5.3)$$

The first KPI, known as the Average Log Loss (MLL), is calculated as the mean of all values of $\tilde{LL}_d$ within the dataset. In our case, there are 360 ($|D|$) data in the dataset and the division is scaled by 100 to increase the explainability of the results.

To provide more insight, we generated other KPIs that compare log loss values with alternative solutions. Since we have created the simulated dataset ourselves, we already know the assignment of the groups (γ_{gk}) and the coefficient values (β^m_k,β^I_k) by construction. However, we also used an error term ϵ_i with a standard deviation of s. Therefore, to present the best possible solution, we need to refit coefficients for each cluster. With that we create the Refitted Construction Solution (RCS), which has a matrix of assignment of γ_{gk} and refitted coefficients ($\hat{\beta}^m_k$ and $\hat{\beta}^I_k$). With these values, we can easily find the z values (z_{i_d}) of RCS.

$$z_{i_d} = \sum_{g_d}^{G_d} \sum_{k_d}^{K_d} \alpha_{ig_d} \gamma_{gk_d} \hat{\beta}^I_{k_d} + \alpha_{ig_d} \gamma_{gk_d} \hat{\beta}^m_{k_d} x^m_{i_d} \quad \forall i \quad \forall d \tag{5.4}$$

With RCS, the log loss (LL_d) of the data d can be simply calculated using the same formula as presented in Equation (5.2) with a z value of z_{i_d}. We assume that LL_d is the almost highest log-loss value that a solution approach can reach. (There may be some special cases where an algorithm finds an even better solution than RCS). Afterwards, we can compare the solution of our methodology with RCS. In this context, we introduce a binary variable ($A_d \in \{0, 1\}$), which is assigned the value of 1 only when the solution approach achieves a log loss equal to or greater than that of RCS ($LL_d \leq \tilde{LL}_d$). After finding A_d, the RCS Achievement Ratio (ARCS) is simply calculated as a percentage of the data with $A_d = 1$.

The third KPI, named Cluster Impact Ratio (CIR), compares the log-likelihood values of solution approaches with having all quotation groups in same cluster. As RCS shows an upper bound, a solution with only one cluster presents the worst possible solution. In addition, the performance of one cluster is an important benchmark, since it represents the conventional approach that one model is trained with the entire data. This study mainly presents methodologies for developing quotation-based pricing strategies. Therefore, with this metric, we present the increase in performance with our approaches compared to not having any clusters. The z-value with only one cluster can be easily calculated with the following equation (5.5). $\dot{\beta}^I_d$ and $\dot{\beta}^m_d$ denotes coefficients fitted with the entire data.

$$\dot{z}_{i_d} = \dot{\beta}^I_d + \dot{\beta}^m_d x^m_{i_d} \quad \forall i \quad \forall d \tag{5.5}$$

The log loss ($\dot{LL}_d$) for a single cluster containing data d can be determined using the same equation where the value of $\dot{z}_{i_d}$ is used as the z value. Then with the Cluster Increase Rate (CI) we show the increase in the log loss of data d (5.6). Therefore, for a

given solution approach, our last KPI, the CIR can be calculated as the mean of the values CI_d.

$$CI_d = \frac{\dot{LL}_d - LL_d}{\dot{LL}_d} \quad \forall d \tag{5.6}$$

We have tested our approaches with 360 simulated data with a time limit of 10 min, and the results are presented in Table 5.1. We can observe that the algorithm inspired by simulated annealing has the best results on almost all metrics. IRC Annealing Inspired algorithm reaches the RCS likelihood in almost all data sets (% 92 of them). Having cluster-wise regression curves instead of a single curve significantly improves the log likelihood and we can observe high CIR values. Furthermore, also IRC Base algorithm can find near results to Annealing Inspired in a shorter run time. Recursive partitioning algorithm can also give competitive results. It could achieve the highest average log loss in the test data. Even though the base model is unable to reach solutions of heuristics, refitting the coefficients enhances the solver's results. However, the relaxed model produces distinctively worse solutions even with refitted coefficients.

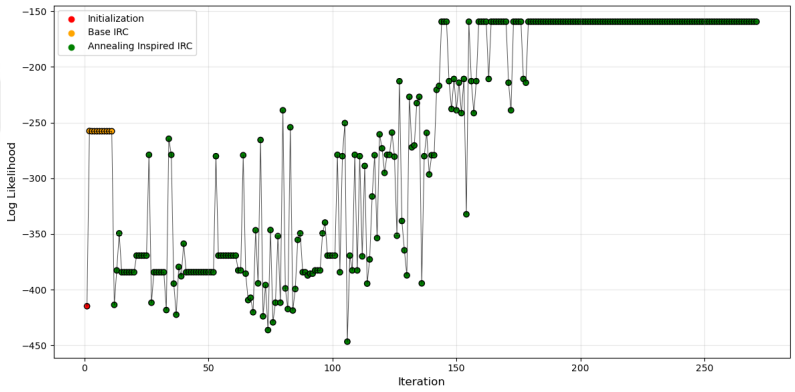
Table 5.1: *Results of proposed approaches with generated simulated datasets*

Solution Approach		MLL		$ARCS(\%)$		$CIR(\%)$		Average Run Time (sec)
		Train	Test	Train	Test	Train	Test	
IRC	Base	-10.91	-11.67	88.05	**77.55**	62.25	53.49	**0.52**
	Annealing Inspired	**-10.83**	-11.49	**92.71**	**77.55**	**62.52**	56.35	9.45
Recursive Partitioning		-10.95	**-11.35**	76.09	77.26	61.94	**59.55**	12.09
Base Model	Solver Coefficients	-11.75	-12.40	77.84	65.89	60.91	54.69	112.73
	Refitted Coefficients	-11.69	-12.85	82.80	70.26	60.77	48.02	
Relaxed Model	Solver Coefficients	-33.66	-31.21	11.8	9.62	6.30	11.98	1.19
	Refitted Coefficients	-26.63	-27.17	13.99	16.91	25.90	24.48	

To understand how annealing inspired IRC could achieve higher metric scores then

the Base IRC, we record log likelihood values through iterations for a sample simulated dataset. As shown in Figure 5.3, algorithm starts with k-center initializations. Then the base IRC improves the solution. From the almost first iteration it converges to a solution and sticks at that point. With only exploiting and only making moves that improve the group's log likelihood, it cannot make any change. Afterwards, Annealing Inspired IRC starts to make moves that decrease log likelihood (accepts movements with high Δ_{g_i}). However, it gives the algorithm the opportunity to explore feasible region more. Then, as temperature decreases, Annealing Inspired IRC makes smaller moves and exploits. With that approach, for a given simulated data set, the annealing-inspired algorithm can achieve a higher log-likelihood then only Base IRC.

Figure 5.3: *Log Likelihood through iterations of Simulated Annealing IRC (with a simulated dataset)*



5.2.2 Simulation Analysis: Parameter Values of Data

We generate many data sets with different parameter values to measure the performance of our solutions approaches under different conditions. In this section, we will be using the test results of the metric $ARCS$ to compare the performances, and $ARCS$ represents the percentage of instances that achieved a log loss of the solution RCS. Since relaxed models can't give competitive results in almost all conditions, that solution approach is not included in this section.

As shown in Figure 5.4, the performance of all approaches decreases as the number of clusters increases. Because an increase in number of clusters increases the number of possible group cluster combinations exponentially.

Figure 5.4: *ARCS value by various number of clusters*

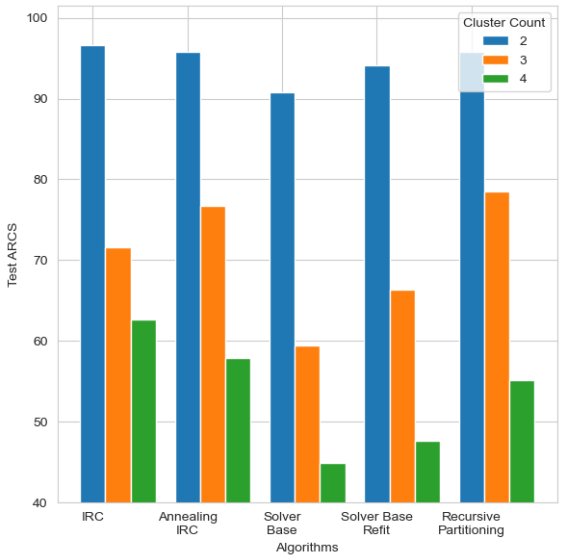
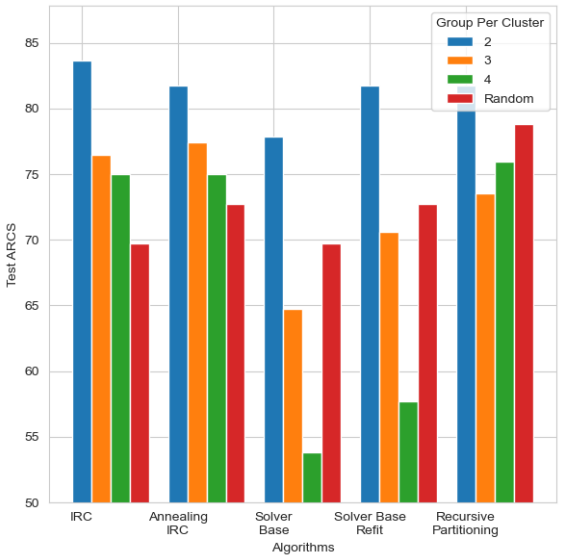


Figure 5.5: *ARCS value by various group per clusters*

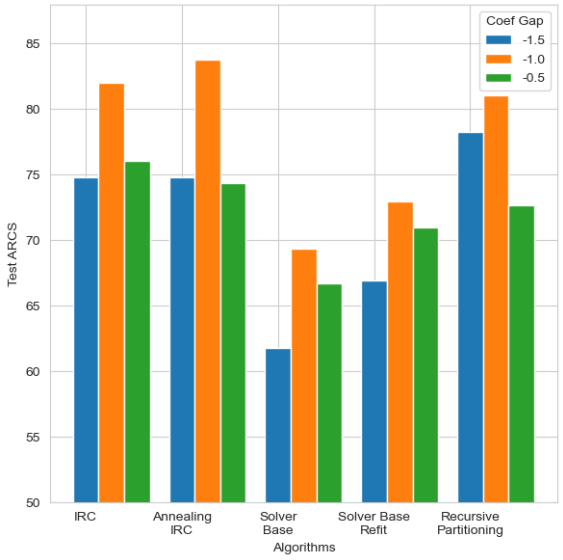


In 4 clusters, the difference between the performance of the solvers and the heuristics increases, and the solvers cannot reach the log loss value of RCS in almost all of the simulated data. For data with 4 clusters, also the performance of recursive partitioning is comparatively low, since it uses exhaustive search to split groups. Interestingly, the base IRC has a higher test score for low and high number of clusters compared to the annealing inspired IRC. For the number of groups per cluster, similar pattern is observed and performance decades in increase of number of groups. However, as shown in Figure 5.5, when the number of groups per cluster is not the same for all clusters, more complex approaches such as annealing IRC and recursive partitioning manage to assign clusters better. Again in a large number of groups, model-based approaches give results with low performance.

Another important parameter in the generation of the data set is g^m, which decides

the difference in the price coefficient between the clusters. As g^m rises, the regression lines of the clusters move further apart. The IRC variant can work better in a setting where g^m is neither too high nor too low (Figure 5.6). However, recursive partitioning performs comparatively better than IRC when clusters have close values.

Figure 5.6: *$ARCS$ value by g^m values*



As presented in Figures 5.4 and 5.5, the computational complexity of the problem increases as the number of clusters and groups increases. This also has an impact on the run-time of our approaches. Both the number of groups (Figure 5.7) and the number of quotations in the data (Figure 5.8) play a major role in the compilation times. In Figure 5.8, there is a fluctuation due to the variability in the number of clusters and clusters.

5.2.3 Simulation Analysis: Expected Percentage Profit Values

The proposed KPIs in Section 5.2.1 are crucial to test how our approaches can learn from the data to classify quotation groups. However, the main purpose of this study and the pricing analysts of B2B firms is not only to achieve high likelihood values. To implement a pricing strategy, pricing analysts should be convinced that the proposed methodology increases the number of accepted quotations and profits. This requires the

Figure 5.7: *Run time (min) of approaches by various number of groups*

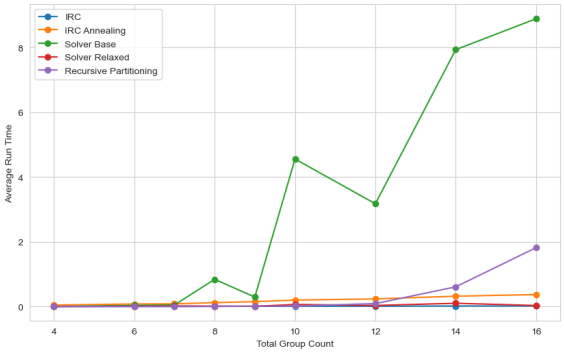
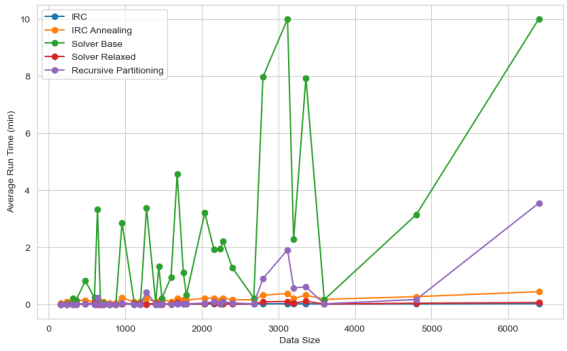


Figure 5.8: *Run time (min) of approaches by various number of quotations*



development of new measurements.

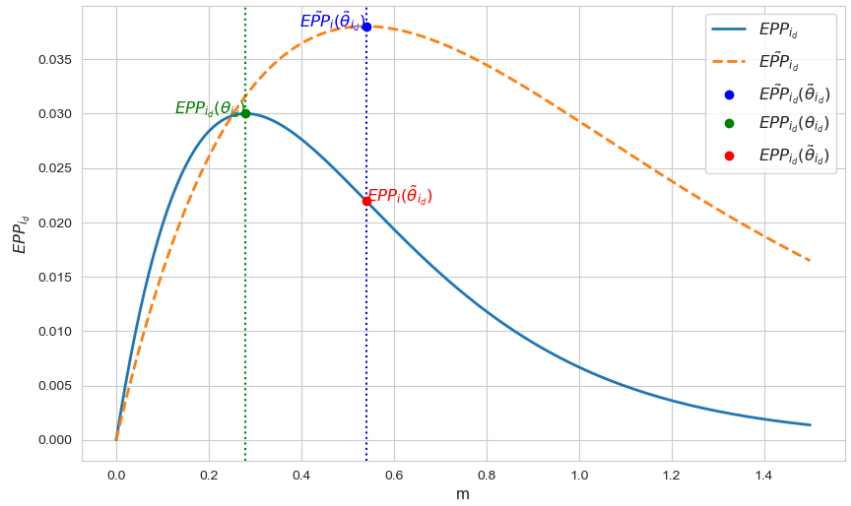
As explained in the problem description (Section 3.2), we can find the Expected Percentage Profit ($E\tilde{P}P_{i_d}$) of a quotation with assignments ($\tilde{\gamma}_{gk_d}$) and coefficients ($\tilde{\beta}^m_{k_d}$,$\tilde{\beta}^I_{k_d}$) from our solution approaches. It is obtained by multiplying the acceptance probability by the value of the variable m of the quotation (3.6), and firms need to apply strategies that maximize the expected percentage profit (EPP_{i_d}). With each solution approach, it is possible to find the value m that maximizes ($\tilde{\theta}_{i_d}$) the $E\tilde{P}P_{i_d}$. With an exhaustive enumeration, it is possible to find m that maximizes $E\tilde{P}P_{i_d}$ in less than 1 second.

The sample curve $E\tilde{P}P_{i_d}$ in Figure 5.9 is generated with incorrect coefficient values that do not reflect the elasticity of that quotation. Therefore, $\tilde{\theta}_{i_d}$ of that example curve is also misleading.

Within simulated datasets, we can assess the EPP_{i_d} curves identified by our solution approaches. It is possible to use RCS to calculate the correct EPP values of the given m value. The following equation uses γ_{gk_d} (assignment matrix used to generating data sets) for the assignment of groups and $\hat{\beta}^m_{k_d}$,$\hat{\beta}^I_{k_d}$ (refitted coefficients) for coefficient values.

$$EPP_{i_d}(m) = \frac{m}{1 + e^{-(\sum_{g_d}^{G_d} \sum_{k_d}^{K_d} \alpha_{ig_d}\gamma_{gk_d}\hat{\beta}^I_{k_d} + \alpha_{ig_d}\gamma_{gk_d}\hat{\beta}^m_{k_d} m)}} \quad \forall i \quad \forall d \tag{5.7}$$

Figure 5.9: *Example curve of EPP_{i_d} and $\tilde{EPP}_{i_d}$*



We can use this function to find the real EPP value of the optimal gross margin ($\tilde{\theta}_{i_d}$) from our solution approach. In Figure 5.9, we can observe that $EPP_{i_d}(\tilde{\theta}_{i_d})$ has a lower value than the value estimated by the solution approach. We also represent the optimal value of m that maximizes EPP_{i_d} as θ_{i_d} (real optimal gross margin). In Figure 5.9, θ_{i_d} is much lower than $\tilde{\theta}_{i_d}$. The solution approach gave too high gross margin suggestion to pricing analysts, and that may lead to a rejection of a quotation. The loss of EPP at there can be calculated as $EPP_{i_d}(\theta_{i_d}) - EPP_{i_d}(\tilde{\theta}_{i_d})$. Therefore, we calculate the Margin Loss (ML) of the data d using the following formulation (5.8).

Sets

D: Set of data;

I_d: Set of quotations in data $d \in D$;

Functions

$\tilde{EPP}_{i_d}(m)$: For quotation $i_d \in I_d$ in the data $d \in D$, the expected percentage profit value of the given m (using the solution method)

$EPP_{i_d}(m)$: For quotation $i_d \in I_d$ in the data $d \in D$, the expected percentage profit value of the given m (using RCS)

Parameters used for ML_d

$\tilde{\theta}_{i_d}$:For quotation $i_d \in I_d$ in the data $d \in D$, an optimal value m maximizes $E\tilde{P}P_{i_d}(m)$

θ_{i_d}:For quotation $i_d \in I_d$ in the data $d \in D$, an optimal value m maximizes $EPP_{i_d}(m)$

$$ML_d = \frac{\sum_d^D EPP_{i_d}(\theta_{i_d}) - EPP_{i_d}(\tilde{\theta}_{i_d})}{|I_d|} \forall d \tag{5.8}$$

Then, for each approach to solution for the test data, the average values of ML_d after scaled (with 10^4) are represented as Mean Margin Loss (MML) in Table 5.2. In these solutions, the annealing inspired algorithm achieves the lowest average loss. Regarding the log-loss metric (MLL) in the test data, the recursive partitioning algorithm performs best. In terms of average margin loss, the recursive partitioning ranks just after the annealing IRC method. The findings indicate that the IRC algorithm is not just effective in terms of log likelihood but also offers significant potential for enhancing companies' gross margins. Furthermore, Table 5.2 reveals that not having cluster-wise curves results in an exceptionally high MML value.

Table 5.2: *MML values of solution approaches in test data*

IRC		Recursive Partitioning	Base Model		Relaxed Model		One Cluster
Base	Annealing Inspired		Solver Coefficients	Refitted Coefficients	Solver Coefficients	Refitted Coefficients	
19.46	**12.27**	16.39	84.52	82.14	1308.70	16326	1755.6

5.3 Real Dataset

Our approaches are designed for B2B firms that use CRM systems. Manufacturers with aftermarket sales have suitable cases as they have sales with high frequency, many customer segments, and various parts levels. Therefore, we use real case data from a manufacturer to evaluate our solution approaches.

We encode all of the pricing related features to save sensitive data. Then we summarize the data, as each row will present a unique quotation (i). Subsequently, to produce

α_{ig} representing the quotation group of quotation i, we use three different features: the customer's segment, the highest and the second highest quantity part level within the quotation. Initially, we create different groups for all combinations of these 3 features. Afterwards, if a group contains fewer quotations than a set threshold, we merge it with another group that only differs in the second highest part level. Consequently, with only quotation group we can present quotation's parts and customer in real case data.

Case data also has many features that are important identifiers for the acceptance of a quotation other than the gross margin (x_i^m). In Section 4.1, these features are denoted by x_i^O. x_i^O include facts about the customer, the customer's previous transactions, and the parts levels included in the quote. For example, the frequency of past sales and prior discounts given to that customer should be considered, since discounts have long-term effects on customer behavior and a nonlinear impact on customer pricing expectations [30]. Furthermore, the data set covers the list prices, as quotations of higher value show distinct characteristics.

The data is split to train and test with stratified splitting. Afterwards, as represented in Section 4.1, all the features in x_i^O are replaced by a single feature (OML_i) with a complex teacher model. The model is trained with the training dataset utilizing the AutoML library, as detailed in Section 5.1.

As a result, our dataset includes 47,230 quotations in the training set and 10,000 quotations in the test set, organized into 41 quotation groups. The test and training data include x_i^O, x_i^m, α_{ig} and the corresponding binary quotation acceptance variable (y_i).

5.3.1 Computational Results

We also utilize our proposed approaches with the real case data. However, due to the large data size and the high number of groups, it is not possible to use recursive partitioning. Even with extremely high run-time limits as 10 hours, the recursive partitioning

algorithm could not converge to a solution. In addition, our mathematical model-based approaches cannot give competitive results with a large data set. Base models could only obtain a log loss of a single cluster (could not achieve higher), while the relaxed model performed even more poorly, despite the refitting of coefficients. In these results, we set the number of clusters as 4 and used a time limit of 20 minutes due to the large size of the data. These results are presented in Table 5.3. Here, as we introduced in Section 5.2.1, LL represents log loss of dataset (5.2) and CI denotes the increase in the cluster (5.6) and presents the impact of the clustering strategy. Here in real data there isn't a KPI as $ARCS$ since it is not a simulated data and we don't know the RCS solution.

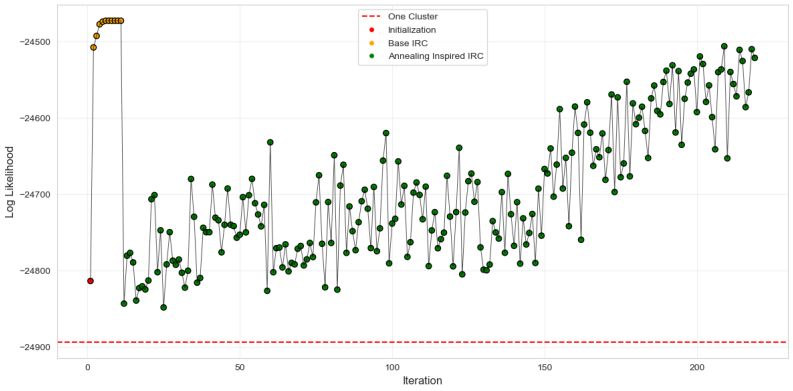
Table 5.3: *Results of proposed approaches with real case data*

Solution Approach		LL		CI (%)		Run Time
		Train	Test	Train	Test	
IRC	Base	**-51.8**	**-61.7**	**1.7**	**1.1**	**15**
	Annealing Inspired	**-51.8**	**-61.7**	**1.7**	**1.1**	285
Base Model	Solver Coefficients	-52.6	-62.41	0	0	1213
	Refitted Coefficients	-52.64	-62.36	0	0	
Relaxed Model	Solver Coefficients	-60.9	-72.7	-1.79	-0.17	48
	Refitted Coefficients	-52.44	-62	0	0	

The annealing inspired IRC could not achieve a log-likelihood higher than Base IRC in our real dataset. As shown in Figure 5.10, it starts with an initial solution and improves distinctively in the iterations of Base IRC. The red line in the figure represents the log likelihood of not having any clusters and even from initial solution it started with a better solution then that. The annealing iterations explore different areas of the feasible region and could improve its solution. However, it could not reach a better result.

As we can observe from the CI values, in a real dataset, these approaches cannot create extreme increases in log loss. Nevertheless, the section 5.3.2 highlights even more

Figure 5.10: *Log Likelihood through iterations of Simulated Annealing IRC with Real Data*



significant advantages.

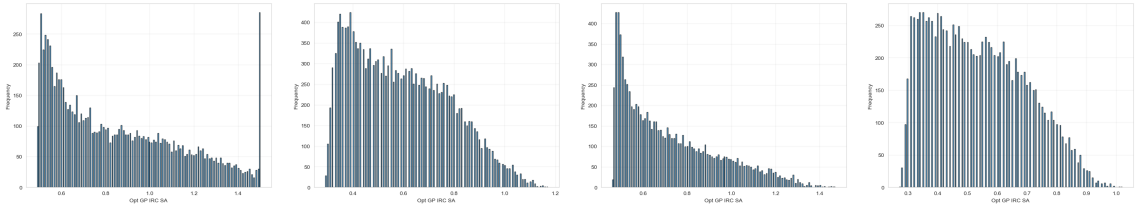
5.3.2 Real Data Analysis: Optimal Gross Margin Values

The base IRC algorithm generates 4 different clusters from the real dataset. Our objective in generating clusters is to apply different strategies for each of the clusters. With real data, the logistic regression curves use variables x_i^m and OML_i. Consequently, we expect that the acceptance of quotations in each cluster will have a different relationship with the gross margin values. To test this, with coefficient values ($\tilde{\beta}_k^m$, $\tilde{\beta}_k^I$, $\tilde{\beta}_k^{OML}$) and quotation group assignment $\tilde{\gamma}_{gk}$ of the base IRC, we find $\tilde{EPP}_i$ curves. Then, as with the same approach with the simulated dataset (Section 5.2.3), we find m values that maximize these curves ($\tilde{\theta}_i$).

Cluster-wise histograms are presented in Figure 5.11. The $\tilde{\beta}_k^m$ values for clusters 0 to 3 can be expressed as [-4, -3.08, -371, -2.2]. The lower coefficient value indicates acceptance of the quotation is more dependent on the gross margin. These quotations may include part levels with high competition or special customer segment. For Cluster 0, the coefficient value of m is low; therefore, we can observe quotations with low values of $\tilde{\theta}_i$, for these quotations profit can be compromised for acceptance. However, in that cluster, there are still quotations with $\tilde{\theta}_i = 1.5$ (highest possible value m) due to the impact of

OML on the regression quotation. In contrast to Cluster 0, in Cluster 3, the coefficient of gross margin is high. Therefore, pricing is not the main criteria. We can note that within that cluster, the distribution of quotations is almost uniform.

Figure 5.11: *Histograms of $\tilde{\theta}_i$ values for Clusters 0-3*



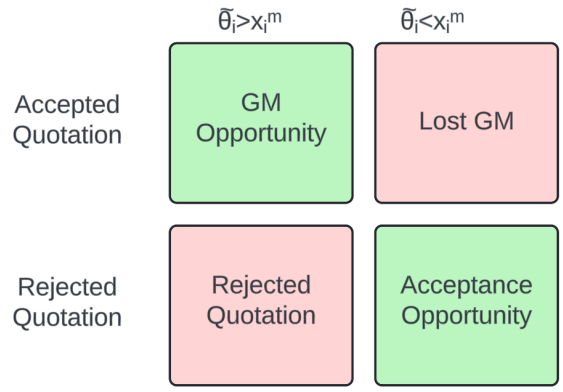
In the case of a single cluster, the gross margin coefficient for the data is -2.13. This is even higher than the highest $\tilde{\beta}_k^m$ value of our clusters. In a case with only using one coefficient, we assume that all quotations are not affected from gross margin. This again shows importance of our solution approaches.

Although these results give important insights, it is not possible to evaluate the correctness of $E\tilde{P}P_i$ as in the simulated data set (Section 5.2.3). We don't have a RCS solution to find the real EPP_i curve. However, we know the values of x_i^m and whether a quotation is accepted or rejected (y_i) with that gross margin.

Therefore, we can create Focus Categories to evaluate our approaches and connect these results to strategies as presented in Figure 5.12. In test data, if our approach found an optimal gross margin ($\tilde{\theta}_i$) higher than x_i^m and we know that the quotation is accepted, we can infer that this quotation probably sold with a lower gross margin than it can reach. It is not possible to determine with certainty whether the customer will agree to the quotation with that specific margin. Therefore, these quotations ($\tilde{\theta}_i > x_i^m$ and $y_i = 1$) are labeled as GM opportunity (GM denotes gross margin).

If our model gives an optimal m value lower than the existing and that quotation is rejected, this shows that the model correctly learned the train data and suggested a lower

Figure 5.12: *Representation of Focus Categories*



gross margin value. If the model's GM is applied as an alternative to the current one, there is a chance that the quotation might not be rejected. Consequently, these quotations are labeled as Acceptance Opportunity ($\tilde{\theta}_i < x_i^m$ and $y_i = 0$).

In contrast, our solution approaches can make mistakes in some of the quotations. For example, if the model proposed a higher value m and it is known that the quote is rejected with a lower value m, it can be confidently concluded that it will be rejected with the suggested gross margin. This signals that there is an inaccuracy in the model's $E\tilde{P}P_i$ curve of that model. We label these quotations as Rejected Quotation ($\tilde{\theta}_i > x_i^m$ and $y_i = 0$). Again, if the model suggests a lower m, however, the quotation is accepted with higher gross margin, the model's suggestions will decrease firm's profitability and these quotations are labeled as Lost GM ($\tilde{\theta}_i < x_i^m$ and $y_i = 1$).

Table 5.4 represents the percentage of quotations by focus categories presented in this section.

Table 5.4: *Percentage of Focus Categories*

	Train		Test	
	IRC	One Cluster	IRC	One Cluster
GM Opportunity*	45%	52%	42%	49%
Rejected Quotation	28%	43%	31%	45%
Acceptance Opportunity*	20%	5%	19%	5%
Lost GM	7%	0%	8%	1%

The total count of quotations with the GM Opportunity and the Acceptance Opportunity (marked with * in Table 5.4 and shaded with green in Figure 5.12) represent actionable quotations in which pricing analysts should focus on adjusting their strategy. The total percentage of these categories is represented as Opportunity Focus Percentage (OP). In addition, 1-OP is the percentage of quotations in which the solution approach provided misleading suggestions. Therefore, we can use OP values to compare different approaches. Our solution approach, IRC, achieved a OP value of %65 in train and %61 in the test data. It achieves %7 higher OP value in test data, compared to not clustering quotations. The model with only one cluster gives incorrect suggestions for 707 more quotations than the IRC base.

5.3.3 Real Data Analysis: Various Number of Clusters

In this section with using base IRC, we will be analyzing the performance of our solutions with respect to generation of different number of clusters than 4 clusters. We show CI values in Figure 5.13 and OP values in Figure 5.14.

Figure 5.13: *CI (%) value by number of clusters*

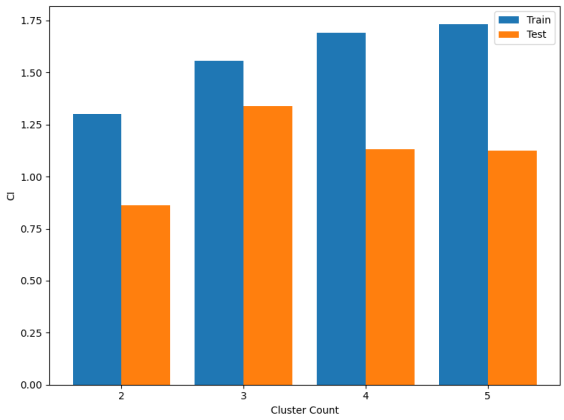
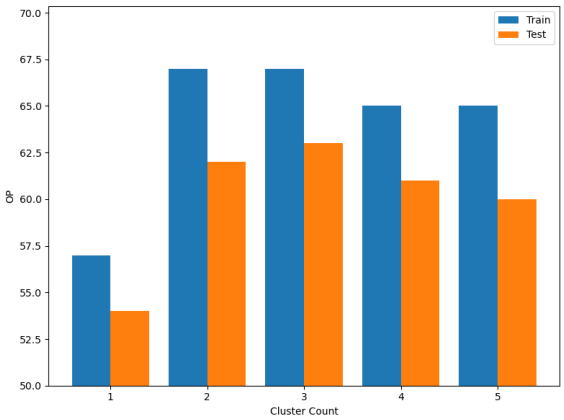


Figure 5.14: *OP (%) value by number of clusters*



As we can observe in Figure 5.13, the increase in the log likelihood performance of the test data stops at 3 clusters. In Figure 5.14, the value of OP increases markedly when

transitioning from a single cluster to two clusters. However, this increase rate decades later, it slightly increases in 3 clusters, and starts to decrease afterwards. Only by analyzing the OP values, we can conclude that it is reasonable to categorize the quotations into only two distinct clusters. However, an increase in logloss performance with 3 clusters should not be ignored.

6. CONCLUSIONS

This research introduces a quotation-based approach for B2B companies, illustrated through a case study of a B2B firm involved in aftermarket sales. These companies must determine the gross margin of a quotation taking into account the customer segment of the quotation and the levels of the included parts. Consequently, we propose a method that allows pricing analysts to develop distinct pricing strategies customized to various clusters containing quotation groups.

Moreover, through an innovative approach in this study, we additionally consider the gross margin of rejected quotations. Furthermore, as a contribution, we define elasticity by quotation groups instead of part-wise or customer-wise classification. Furthermore, our case demands a model that is both complex and interpretable, a requirement that we satisfy using OML.

Furthermore, this assignment problem is highly complex and requires the generation of new algorithms. For instance, solvers and recursive partitioning algorithms can't solve real world datasets. Therefore, we introduce various heuristics, such as Recursive Partitioning and Iterative Refit Classifier algorithms. Only IRC and IRC Annealing Inspired algorithms could achieve high performance in both the simulated and real data sets.

In addition, to make the proposed strategy convincing, our case requires newly defined metrics. Only high log loss is not convincing for pricing analysts, B2B firms try to increase their accepted quotation and their gross margin at the simultaneously. Therefore, we calculate the Expected Percentage Profit (EPP) of our solution approaches. Based on our simulated datasets, we find that not using a cluster-specific pricing strategy results in an average EPP loss 143 times greater than the solution provided by Annealing IRC.

Also, in real data we observe that pricing relationship can't be observed with a

single cluster and we observe that each of the clusters of IRC have different price elasticities. Also, we generated focus groups from real data, to show quotations that model give misleading suggestions and characteristics that pricing analysts should focus on. Based on these findings, we conclude that a single cluster strategy finds 7% more inaccurate gross margin values that lead to rejections of quotations or gross margin losses.

In particular,the focus of this study is to find clusters of quotation groups. As a further study, with newly developed approaches that utilize IRC solutions, even better optimal gross margin values can be found.

In this study, we use the case of a firm with aftermarket sales. However, this problem is applicable for many firms that have a CRM system. It introduces an additional dimension as a quotation and suggests novel solutions for B2B firms, enabling them to expand their customer base while generating greater profit.

REFERENCES

[1] A. Hinterhuber, "Towards value-based pricing—an integrative framework for decision making," *Industrial Marketing Management*, vol. 33, no. 8, pp. 765–778, 2004.

[2] R. Farres, "Optimal pricing models in b2b organizations," *Journal of Revenue and Pricing Management*, vol. 11, no. 1, pp. 35–39, 2012.

[3] K. Indounas, "Market-based pricing in b2b service industries," *Journal of Business Industrial Marketing*, vol. 34, no. 5, pp. 1030–1040, 2019.

[4] A. D. Stein, M. F. Smith, and R. A. Lancioni, "The development and diffusion of customer relationship management (crm) intelligence in business-to-business environments," *Industrial Marketing Management*, vol. 42, no. 6, pp. 855–861, 2013.

[5] L. Li, M. Liu, W. Shen, and G. Cheng, "A dynamic spare parts ordering and pricing policy based on stochastic programming with multi-choice parameters," *Mathematical and Computer Modelling of Dynamical Systems*, vol. 23, no. 2, pp. 196–221, 2017.

[6] M. T. Ribeiro, S. Singh, and C. Guestrin, ""why should i trust you?" explaining the predictions of any classifier," 2016. arXiv preprint arXiv:1602.04938.

[7] P. J. Phillips, C. A. Hahn, P. C. Fontana, A. N. Yates, K. Greene, D. A. Broniatowski, and M. A. Przybocki, "Four principles of explainable artificial intelligence," Tech. Rep. NIST IR 8312, National Institute of Standards and Technology (U.S.), 2021.

[8] J. Andersson and J. Bengtsson, "Spare parts pricing - pre-study for a pricing strategy at pon," 2013.

[9] R. A. Young, "Price elasticity of demand for municipal water: A case study of tucson, arizona," *Water Resources Research*, vol. 9, no. 4, pp. 1068–1072, 1973.

[10] M. Brons, P. Nijkamp, E. Pels, and P. Rietveld, "A meta-analysis of the price elasticity of gasoline demand. a sur approach," *Energy Economics*, vol. 30, no. 5, pp. 2105–2122, 2008.

[11] X. Zhu, L. Li, K. Zhou, X. Zhang, and S. Yang, "A meta-analysis on the price elasticity and income elasticity of residential electricity demand," *Journal of Cleaner Production*, vol. 201, pp. 169–177, 2018.

[12] Fujita and K. Sydney, "Estimating price elasticity using market-level appliance data," 2015.

[13] Y. Wang, "Price-load elasticity analysis by hybrid algorithm," in *The Third International Conference on Electric Utility Deregulation and Restructuring and Power Technologies*, (Nanjing, China), 2008.

[14] T. Michinaka, S. Tachibana, and J. A. Turner, "Estimating price and income elasticities of demand for forest products: Cluster analysis used as a tool in grouping," *Forest Policy and Economics*, vol. 13, no. 6, pp. 435–445, 2011.

[15] H. Späth, "Algorithm 39 clusterwise linear regression," *Computing*, vol. 22, no. 4, pp. 367–373, 1979.

[16] G. Qian and Y. Wu, "Estimation and selection in regression clustering," *European Journal of Pure and Applied Mathematics*, vol. 4, no. 4, pp. 455–466, 2011.

[17] R. A. Carbonneau, G. Caporossi, and P. Hansen, "Globally optimal clusterwise regression by mixed logical-quadratic programming," *European Journal of Operational Research*, vol. 212, no. 1, pp. 213–222, 2011.

[18] E. Kayış, "A gradient descent based heuristic for solving regression clustering problems," in *Proceedings of the 9th International Conference on Data Science, Technology and Applications*, pp. 102–108, 2020.

[19] M. Sui, W. R. Erick, F. Viole, and K. Jetta, "Modeling elasticity: A brief survey of price elasticity of demand estimation methods," *Journal of Research in Marketing*, 2019.

[20] E. Kayış, "Regression clustering for estimating product-level price elasticity with limited data," *Journal of Business Research - Turk*, vol. 12, no. 4, pp. 3319–3332, 2020.

[21] M. Guillén, A. M. Perez, and M. Alcañiz, "A logistic regression approach to estimating customer profit loss due to lapses in insurance," 2011. SSRN Electronic Journal.

[22] A. X. Carvalho and M. L. Puterman, "Learning and pricing in an internet environment with binomial demands," 2005. Journal of Revenue and Pricing Management, 3(4), 320–336.

[23] H. Qu, I. O. Ryzhov, and M. C. Fu, "Learning logistic demand curves in business-to-business pricing," 2013. 2013 Winter Simulations Conference (WSC), 29–40.

[24] A. Zeileis, T. Hothorn, and K. Hornik, "Party with the mob: Model-based recursive partitioning in r," 2006. Unpublished manuscript.

[25] S. Kirkpatrick, C. D. Gelatt, and M. P. Vecchi, "Optimization by simulated annealing," *Science*, vol. 220, pp. 671–680, 1983.

[26] F. Giannessi and F. Tardella, "Connections between nonlinear programming and discrete optimization," in *Handbook of Combinatorial Optimization* (D.-Z. Du and P. M. Pardalos, eds.), pp. 149–188, Springer US, 1998.

[27] N. Metropolis, A. W. Rosenbluth, M. N. Rosenbluth, A. H. Teller, and E. Teller, "Equation of state calculations by fast computing machines," *The Journal of Chemical Physics*, vol. 21, no. 6, pp. 1087–1092, 1953.

[28] R. H. Byrd, J. C. Gilbert, J. Nocedal, and R. A. Waltz, "Knitro: An integrated package for nonlinear optimization," in *Large-Scale Nonlinear Optimization* (G. Di Pillo and M. Roma, eds.), vol. 83, pp. 35–59, Springer US, 2006.

[29] N. Andrei, "Numerical studies: Comparisons," in *Continuous Nonlinear Optimization for Engineering Applications in GAMS Technology*, vol. 121, pp. 437–447, Springer International Publishing, 2017.

[30] M. J. Del Rio Olivares, K. Wittkowski, J. Aspara, T. Falk, and P. Mattila, "Relational price discounts: Consumers' metacognitions and nonlinear effects of initial discounts on customer retention," *Journal of Marketing*, vol. 82, no. 1, pp. 115–131, 2018.

APPENDICES

APPENDIX A. SIMPLIFICATION OF OBJECTIVE

In this section simplification of the log-likelihood formula in the mathematical model of the problem is described. To start with, for a logistic regression model, the probability of y=1 can be formulated with the sigmoid function (A.1).

$$\sigma(z) = \frac{1}{1 + e^{-z}} \tag{A.1}$$

The formulation of likelihood is presented (A.2).

$$\prod \left(\sigma(z)^y (1 - \sigma(z))^{1-y}\right) \tag{A.2}$$

Therefore, the formula of log-likelihood can be expressed as (A.3).

$$y^T \log\left(\frac{1}{1 + e^{-z}}\right) - (1 - y)^T \log\left(\frac{e^{-z}}{1 + e^{-z}}\right) \tag{A.3}$$

After that, we apply some simplifications (A.4).

$$y^T \left(-\log(1 + e^{-z})\right) - (1 - y)^T \left(\log(e^{-z}) - \log(1 + e^{-z})\right) \tag{A.4}$$

That leads to our objective function of the mathematical model (A.5).

$$\max y^T \left(-\log(1 + e^{-z})\right) - (1 - y)^T \left(-z - \log(1 + e^{-z})\right) \tag{A.5}$$

APPENDIX B. MATH MODEL OF GROSS MARGIN OPTIMIZATION

This model utilizes coefficient values and the assignment matrix of our solution approaches as parameters and finds a gross margin that maximizes $E\tilde{P}P_i$

Sets

I: set of quotations;
K: set of clusters;
G: set of groups;

Parameters

$\alpha_{ig} = \begin{cases} 1, & \text{if quotation i} \in \text{I is in group g} \in \text{G} \\ 0, & \text{otherwise} \end{cases}$

$\tilde{\gamma}_{gk} : \begin{cases} 1, & \text{if group g} \in \text{G is in cluster k} \in \text{K} \\ 0, & \text{otherwise} \end{cases}$

$\tilde{\beta}_k^I$: Coefficient value of the intercept, for cluster $k \in K$
$\tilde{\beta}_k^m$: Coefficient value of the gross margin, for cluster $k \in K$
$\tilde{\beta}_k^{OML}$: Coefficient value of the OML, for cluster $k \in K$

Decision Variables

$\tilde{z}_i$: Linear multiplication of coefficient values and variable values of quotation $i \in I$
$\tilde{\theta}_i$: Optimal Gross Margin value of the quotation $i \in I$

Function

$E\tilde{P}P_i(m)$: For quotation $i \in I$, the expected percentage profit value of the given gross margin

$$\max \frac{\sum_{i \in I} E\tilde{P}P_i(\tilde{\theta}_i)}{|I|} \tag{B.1}$$

$$\tilde{z}_i = \sum_{g \in G} \sum_{k \in K} \alpha_{ig} \tilde{\gamma}_{gk} \tilde{\beta}_k^I + \alpha_{ig} \tilde{\gamma}_{gk} \tilde{\beta}_k^m \tilde{\theta}_i \quad \forall i \in I \tag{B.2}$$

$$E\tilde{P}P_i(\tilde{\theta}_i)) = \frac{\tilde{\theta}_i}{1 + e^{-\tilde{z}_i}} \quad \forall i \in I \tag{B.3}$$

$$\tilde{\theta}_i \gg 0 \quad \forall i \in I \tag{B.4}$$

APPENDIX C. MATH MODEL OF REGRESSION CLUSTERING WITH OML

Sets

I: Set of quotations ;
K: Set of clusters ;
G: set of groups;

Parameters

$\alpha_{ig} = \begin{cases} 1, & \text{if quotation } i \in I \text{ is in group } g \in G \\ 0, & \text{otherwise} \end{cases}$
x_i^m: Value of gross margin of quotation $i \in I$
OML_i: Value of OML of quotation $i \in I$
$y_i : \begin{cases} 1, & \text{if quotation } i \in I \text{ is accepted} \\ 0, & \text{otherwise} \end{cases}$

Decision Variables

$\tilde{\gamma}_{gk} : \begin{cases} 1, & \text{if group g} \in \text{G is in cluster k} \in \text{K} \\ 0, & \text{otherwise} \end{cases}$
$\tilde{\beta}_k^I$: Coefficient value of the intercept, for cluster $k \in K$
$\tilde{\beta}_k^m$: Coefficient value of the gross margin, for cluster $k \in K$
$\tilde{\beta}_k^{OML}$: Coefficient value of the OML, for cluster $k \in K$
$\tilde{z}_i$: Linear multiplication of coefficient values and variable values of quotation $i \in I$

$$\max \sum_{i \in I} y_i \left(-\log\left(1 + e^{-z_i}\right)\right) + (1 - y_i)\left(-\log\left(1 + e^{-z_i}\right) - z_i\right) \tag{C.1}$$

$$z_i = \sum_{g \in G} \sum_{k \in K} \alpha_{ig} \tilde{\gamma}_{gk} \tilde{\beta}_k^I + \alpha_{ig} \tilde{\gamma}_{gk} \tilde{\beta}_k^m x_i^m + \alpha_{ig} \tilde{\gamma}_{gk} \tilde{\beta}_k^{OML} OML_i \quad \forall i \in I \tag{C.2}$$

$$\sum_{k \in K} \tilde{\gamma}_{gk} = 1 \quad \forall g \in G \tag{C.3}$$

$$\sum_{g \in G} \tilde{\gamma}_{gk} \geq 0 \quad \forall k \in K \tag{C.4}$$

$$\tilde{\gamma}_{gk} \in \{0, 1\} \tag{C.5}$$

APPENDIX D. TEACHER MODEL SELECTION

To develop our teacher model with data x^O , we use the Python library AutoML. In this section, an overview of the steps AutoML applied to our dataset and the results will be presented.

AutoML starts with training simple algorithms (Baseline, Decision Tree, and linear predictors). The baseline predicts the entire dataset as the most frequent label. Theis simple algorithm is used as a threshold to evaluate other default algorithms. Default algorithms (Random Forest, Extra Trees, Xgboost, LightGBM, CatBoost, and neural network) are trained with the default hyperparameters. All models are evaluated with their k-fold log-loss values. Then, hyperparameter tuning is applied for default algorithms and trained again with tuned parameters (AutoML Mljar-Supervised Documentations).

Afterward, AutoML creates golden features that have high predictive power. These features are created from randomly selected pairs of features and random ratios. To make a feature selection, AutoML first adds a random feature. Then drops all features that are less important than that random feature.

To create an ensemble algorithm, as Caruana (2004) introduced, AutoML applies greedy search through models and iteratively tries adding new models to the stack. After ensemble stacking, with weight values, ensemble models are generated.

For our dataset, the performance of various models is show in Figure D.1.

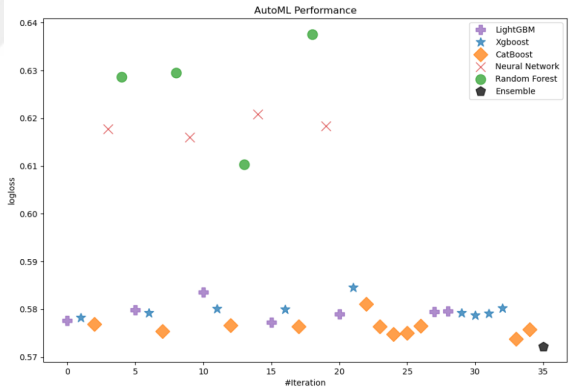


Figure D.1: *AutoML performance scatter plot*

On this basis, AutoML selected CatBoost as the best model with the following parameters: a learning rate of 0.025, a depth of 9, and an Random Subspace Method (RSM) of 0.7. The resultant CatBoost model is used to estimate quotation-wise acceptance probabilities and is expressed as OML in this paper.

APPENDIX E. OTHER IRC VARIANTS

E.0.1 Random Initialization

In this initialization, instead of using gorup-wise coefficients and the k-center algorithm (as in the pseudo code 2), we randomly initialize clusters. To find the number of groups per cluster in the case with random assignment, we divide the number of groups by the number of cluster. From this division, if there is a reminder, at the last cluster there will be one more group. Then, as a straightforward random assignment approach, we randomly shuffle the order of groups and assign groups to clusters with determined sizes.

This approach gives worse initializations than the k-center method. However, in some cases, it may be an opportunity for IRC. Because it can explore different regions of the feasible region while improving log-likelihood of groups with moves. Consequently, for some cases with that, IRC can reach better results with worse initialization.

E.1 Retaining Number of Clusters with Min Log Loss

IRC retains the number of groups by not moving the groups that is the only group in it's cluster ($|S_{C_{g_i}}| = 1$). In this variant, we remove the condition of $|S_{C_{g_i}}| > 1$. Consequently, the algorithm can evaluate all groups. Instead of limiting the movement of some groups, as shown in the pseudo code 4, we add another phase (Phase 3). In this phase, until all clusters include a group, we select a group with min log loss and move that group to an empty cluster.

E.2 Not Retaining Number of Clusters

With the IRC algorithm, we find a solution with exactly K clusters. It is worth considering whether relaxing this constraint, by allowing the algorithm to include empty clusters, might enhance the likelihood of solutions. Therefore in this variant, we evaluate all of the group (removing rule $|S_{C_{g_i}}| > 1$ in Pseudo Code 2) and we don't include any balancing phase as Phase 3 in the pseudo code 4.

This method expands the problem's feasible region, but this does not necessarily enhance the performance of the IRC. For instance, when the algorithm does not retain the number of clusters and there are empty clusters, during Phase 2.1, it evaluates fewer number of combinations. For example, in an extreme scenario where just one cluster contains all groups, during subsequent iterations, when the algorithm tries to move a group, it only evaluates whether it remains in the current cluster with all other groups or stands alone if it was moved to an empty cluster.

Algorithm 4 IRC-Balance

Phase 1: Initialization

Fit group-wise regressions. For each g, determine coefficient values: $\tilde{\beta}_g^I, \tilde{\beta}_g^m, \tilde{\beta}_g^{OML}$.
Initialize k clusters with the k-center algorithm using $(\tilde{\beta}_g^m, \tilde{\beta}_g^{OML})$ as features.
Each cluster's group set is represented by S_k, and each group's assigned cluster is denoted as C_g.

Phase 2: Improving Initial Solutions

for $t = 1$ to 10 iterations **do**
  Randomly shuffle the order of groups in G.
  for all Group $g_i \in G$ **do**
    Phase 2.1: Evaluation with Refit Trials
    for all Cluster $k_i \in K$ **do**
      if $g_i \in S_{k_i}$ **then**
        Fit coefficients for combined data of groups S_{k_i}: $\tilde{\beta}_{k_i}^I, \tilde{\beta}_{k_i}^m, \tilde{\beta}_{k_i}^{OML}$.
      else
        Fit coefficients for combined data of groups $S_{k_i} \cup \{g_i\}$: $\tilde{\beta}_{k_i}^I, \tilde{\beta}_{k_i}^m, \tilde{\beta}_{k_i}^{OML}$.
      end if
      Compute log-likelihood $L_{g_i k_i}$ for group g_i, when placed in cluster k_i with coefficients $\tilde{\beta}_{k_i}^I, \tilde{\beta}_{k_i}^m, \tilde{\beta}_{k_i}^{OML}$.
    end for
    Select the cluster k with the maximum log-likelihood $L_{g_i k}$: m_1.
    Phase 2.2: Move
    if $m_1 \neq C_{g_i}$ **then**
      Move g_i to cluster m_1. Update S_k, C_g.
    end if
  end for
  Phase 3: Re-balance
  while there exists a cluster k_i such that $|S_k| < 1$ **do**
    Find group g_i with minimum log-loss.
    Move g_i to k_i. Update S_k, C_g.
  end while
end for

E.3 Not Refitting

During evaluation of a group, IRC tries different cluster alternatives. When the group moves into a different cluster, it changes the cluster's price elasticity, which consequently affects the logistic regression curve. Therefore, during evaluation (Phase 2.1 in base IRC algorithm), we refit coefficients. However, because of this flow, the order of group evaluations affects the solution. Because groups that evaluated earlier and movements of these groups, effects coefficient values at later evaluations. Therefore in the base IRC, in each iteration we shuffle order of groups.

As shows in psuedo code 5, in this variant, we do not refit coefficients during evaluations, we only update coefficient values at the end of the iteration. During evaluation, we use the pre-determined cluster-wise coefficient values. Therefore, order of evaluation does not have any impact and we do not shuffle order of groups in this variant.

Algorithm 5 Iterative Classifier - IRC Without Refit

Phase 1: Initialization

Fit group-wise regressions. For each g, determine coefficient values: $\tilde{\beta}_g^I, \tilde{\beta}_g^m, \tilde{\beta}_g^{OML}$.
Initialize k clusters with the k-center algorithm using $(\tilde{\beta}_g^m, \tilde{\beta}_g^{OML})$ as features.
Each cluster's group set is represented by S_k, and each group's assigned cluster is denoted as C_g.

Phase 2: Improving Initial Solutions

Fit cluster-wise regressions. For each k, determine coefficient values: $\tilde{\beta}_k^I, \tilde{\beta}_k^m, \tilde{\beta}_k^{OML}$.

for $t = 1$ to 10 iterations **do**
  for all Group $g_i \in G$ **do**
    if $|S_{C_{g_i}}| > 1$ **then**
      for all Cluster $k_i \in K$ **do**
        Compute log-likelihood $L_{g_i k_i}$ for group g_i, when placed in cluster k_i with pre-determined coefficients $\tilde{\beta}_{k_i}^I, \tilde{\beta}_{k_i}^m, \tilde{\beta}_{k_i}^{OML}$.
      end for
      Select the cluster k with the maximum log-likelihood $L_{g_i k}$: m_1.
      if $m_1 \neq C_{g_i}$ **then**
        Move g_i to cluster m_1. Update S_k, C_g.
      end if
    end if
  end for
end for

APPENDIX F. SIMULATED DATASET GENERATION

To test our approaches, with simulated datasets, we generate 360 dataset in 3 batches. To create these datasets, different combinations of parameters ($|K|$,$|S_k|$,g^m,s) are used.

In the first batch, we assumed that each cluster will have the same number of groups and each group will have the same number of groups. We created a data set for all combinations of options.

Then, for the second batch, we create datasets where group sizes are not equal. For each combination of $|K|$ and $|S_k|$ (9 combinations), we generate 4 different sequences of group sizes (36 sequences). Then, with each of these sequences, we generate data sets with all combinations of g^m and s.

In the last batch, we also have groups with different sizes. For this, for each of the $|K|$ options (3 options), we create 4 different random sequences of $|S_k|$ and $|I_k|$ (12 pair of sequences). With these pairs, we create a dataset with combinations of g^m and s.