# MERCHANT SERVICE FEE OPTIMIZATION FOR CREDIT CARD TRANSACTIONS

A Thesis

by

Berk Cihan Deniz

Submitted to the
Graduate School of Sciences and Engineering
in Partial Fulfillment of the Requirements for
the Degree of

Master of Science

in the
Department of Industrial Engineering

Özyeğin University
January 2024

# MERCHANT SERVICE FEE OPTIMIZATION FOR CREDIT CARD TRANSACTIONS

Approved by:

Asst. Prof. Enis KAYIŞ
Dept. of Industrial Eng.
*Özyeğin University*

Asst. Prof. Dilek GÜNNEÇ DANIŞ
Dept. of Industrial Eng.
*Özyeğin University*

Asst. Prof. Murat KAYA
Dept. of Industrial Eng.
*Sabancı University*

Date Approved: November 13, 2023

*To my beloved family and friends......*

# ABSTRACT

This thesis aims to enhance the current pricing policies set to optimize customer retention by proposing enhanced methodologies, focusing on finding efficient algorithms that apply to large-scale businesses. Customer retention studies have a vital effect on companies since they affect the generated revenue and market share. This study mainly investigates pricing factors and their relationship with customer churn. The aim is to find commission rates that will be used to charge merchants when making transactions with Point of Sale (POS) devices. An important aspect to remember is the price sensitivity of those merchants as there happens to be a negative relationship between their retention, and pricing policies. We designed a nonlinear mathematical model to solve this problem. However, solving nonlinear models for large-scale problems is not effective when computational power and current solving techniques are considered. As an approach to solving this phenomenon, we discretized the nonlinear form of the model and proposed an alternative integer programming model. By doing so, solvers used to optimize the integer programming models were able to solve higher scales of the problem when compared with global nonlinear solvers. However, it is observed that IP solvers fail to solve the proposed mathematical model after a certain size. To solve this dimensionality problem, stratification techniques are used to divide the problem and optimize subparts of the whole problem separately while limitations and the objective of the problem are considered.

# ÖZETÇE

Bu tez projesinde fiyatlandırma prosedürü üzerinde geliştirmeler yapılarak, müşteri tutundurma optimizasyonu üzerinde çalışmalar yapılmış ve geliştirilen algoritmaların büyük kitleler üzerinde uygulanabilirliği dikkate alınmıştır. Müşteri tutundurma optimizasyonu, gelir ve market hacmi gibi metrikleri direkt etkilediği için firmalar için ciddi bir öneme sahiptir. Bu çalışma genel olarak fiyatlama ve bunun kayıp oranına etkisini inceleyip, en iyi fiyatlamanın nasıl yapılacağını araştırmaktadır. Amaç banka ile çalışan üye işyerlerine, POS cihazlarını kullandıklarında ödemekle yükümlü oldukları komisyon oranlarını sağlamaktır. Bu işlem yapılırken, üye işyerlerinin fiyat hassasiyeti de dikkate alınmalıdır çünkü komisyon oranları kayıp oranlarını etkileyebilmektedir. Bu problemi çözebilmek için önce doğrusal olmayan bir matematiksel model tasarlanmıştır. Doğrusal olmayan modelleri çözmek için günümüzde kullanılan çözüm tekniklerinin yüksek hesaplama gücü istemesinden kaynaklı, tasarlanan modelin büyük kitleler üzerinde uygulanamayacağı gözlemlenmiştir. Ölçeklendirme problemini çözebilmek için tasarlanan model ayrıklaştırılmış ve doğrusal forma dönüştürülmüştür. Doğrusal modellerde kullanılan çözüm algoritmaları, büyük kitleler ile çalışmaya uygun olsa dahi, bu çalışmada kullanılan büyüklükte bir kitle için yetersiz kalmış ve bu problemin çözümü için katmanlı örnekleme teknikleri kullanılmıştır.

# ACKNOWLEDGEMENTS

Getting this far and completing my project successfully would have not been possible without the continued and valuable support of my advisor Enis Kayış. For that, it would be shameful for me not to confess the gratitude I have for him. He was not just an academic advisor to me, rather he guided me with other aspects of my life. I would like to also thank Özyeğin University for facilitating the opportunity to receive a top-quality level of education that helped me develop various skills and gave me a competitive advantage which will assure that I thrive in what will come ahead of me in life. If given the chance to go back in time before joining Özyeğin University and pursuing my graduate degree, I would still take the same step and enroll in the university and program. The reason behind that is that I compare myself now and once I joined the program and all I can say is that I feel like a different person. I person who is more capable and achievable. Finally, I want to express my appreciation to my family for being there for me in my hard times and being patient with me throughout my education years.

# TABLE OF CONTENTS

# LIST OF TABLES

# LIST OF FIGURES

# CHAPTER I

# INTRODUCTION

The credit card industry plays a crucial role in financial systems by facilitating secure transactions for both consumers and merchants in cashless transactions. The industry has grown considerably in recent years and changed the consuming behavior of individuals [1]. According to a recent study by Custom Market Insights, market of the global credit card payments was worth about 152.2 Billion USD in 2022 and it's predicted to reach 165.6 Billion USD in 2023. Furthermore, the projected net worth is expected to reach 286.5 billion USD by 2032, with a steady growth rate of 9.1% from 2023 to 2032 [2]. Although consumers still use cash, especially for small-size transactions, the level of usage differs from country to country and it is strongly correlated with merchant card acceptance, demographics, and the sales location [3]. A recent survey conducted by Forbes Advisor in February 2023 revealed that the majority of purchases are made by debit and credit cards in the US. Specifically, 54% of consumers use debit cards and %36 of them use credit cards while just %9 use cash for purchases as shown in Figure 1 [4].

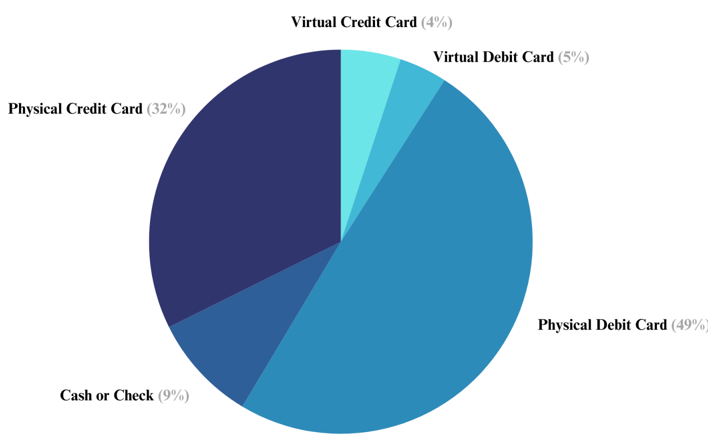


**Figure 1:** How Consumers Pay for Purchases in the US

Cashless transactions have gained massive popularity due to their ease of use, and the convenience that comes with it [5, 6]. A simple electronic card can speed up the payment process between parties exchanging products or services for money. Not only that, but it makes the activities done more transparent, scalable, and secure due to their trackability features [5]. It also diminishes the worry that comes with the fear of the physical currency getting lost or finding a suitable place to store it. Also, the evolving features being rapidly developed like contactless payments make the use of them even more attractive [7]. Whether physically available or located continents away, completing transactions is still possible thanks to the power that comes with being able to purchase online using them [8]. One can also think of the environmental impact it has. Carrying transactions using credit cards that can be issued by banks eliminates the use of physical currency. With this elimination, producing and transporting the cash is no longer needed leading to less carbon emissions and other aspects negatively impacting the environment [9]. This also leads to less consumption of resources that can be invested in more beneficial products to mankind.

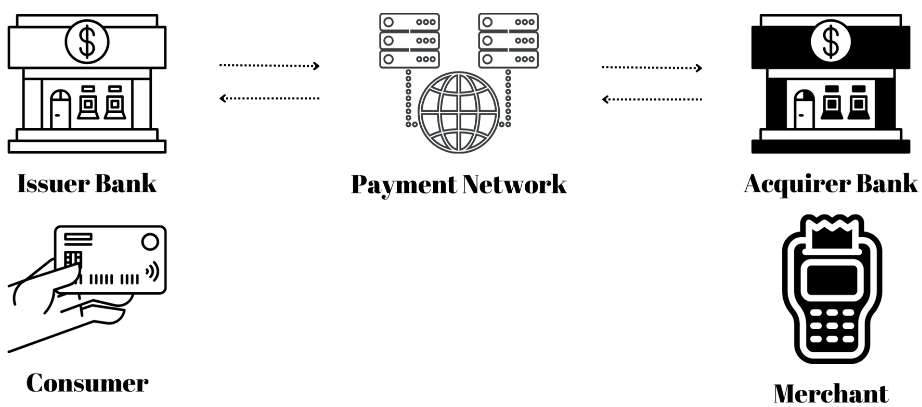


**Figure 2:** Players in Payment Industry

After mentioning all the perks that come with using electronic cards, switching to them instead of using physical currency becomes undeniably attractive to buyers and sellers in different systems. Here is where acquiring banks come in and try to execute their best set of actions. By providing the needed infrastructure for such operations to various parties, huge amounts of profits can be generated.

To understand how the ecosystem works, we should know all the players in the payment industry. Credit card issuers refer to banks that provide credit cards to consumers. Merchants need to work with acquiring banks that provide the infrastructure to carry out cashless transactions. Payment networks have a connection role between acquiring banks and credit card issuers. When consumers use their credit cards to purchase products or services, the payment will be made by the issuer bank to the merchant. The consumer is obliged to pay this transaction to the issuer bank on a predetermined payment date.

Various fees need to be paid by both consumers and merchants. Some credit cards require an annual fee which will be paid by the cardholder to the issuer bank for having the privilege of using a card that allows them to make purchases and pay for them on further dates. Also, consumers will be charged an interest fee that must be paid if they do not pay their debt to the issuer bank within an agreed time.

A fee referred to as an interchange fee is also paid by acquiring banks to the payment networks and issuer banks(consumer's banks) [10]. The fee is paid to the payment networks for providing them with the foundation that allows the transfer of funds electronically. For the issuer banks on the other hand, a fee must be paid since while carrying out these activities, these banks are faced with the risk of non-payment. In other words, there is a possibility that consumers will not pay their debt to the issuer bank, and this puts the issuer bank in a situation of uncertainty. Interchange fees can be considered as safety measures for issuer banks in case of the occurrence of undesirable events.

Acquiring banks compensate for this fee by charging merchants a fee called the Merchant Service Fee (MSF), which includes both interchange and additional fees. Additional fees mainly include the cost of using the infrastructure and operational expenses that come along with it. Infrastructure expenses include all the hardware and services that are provided by acquirer banks to make merchants able to use credit cards for payments. The payment network is also paid with a part of the Merchant Service Fee.

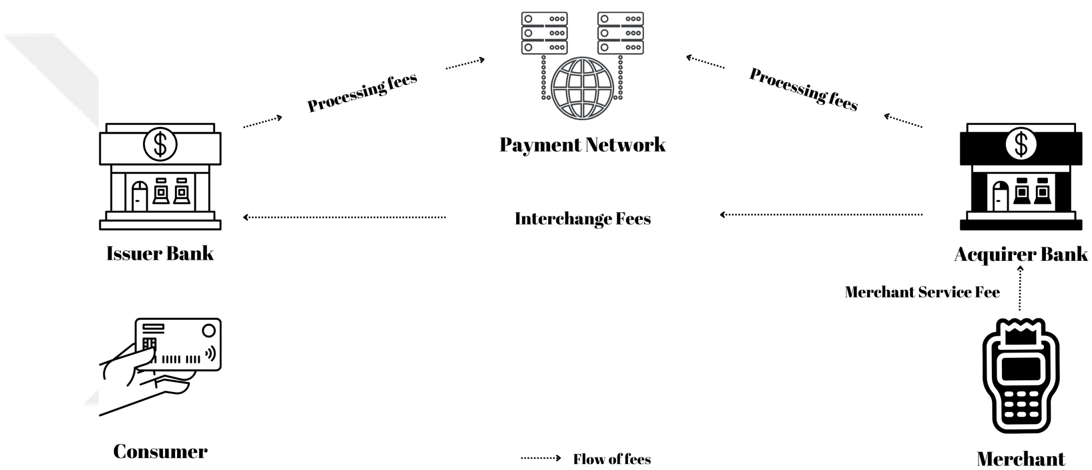


**Figure 3:** General structure of Payment Industry

MSF has a very vital role in the payment system since higher fees may force merchants to sell their products/services at higher prices, offer discounts to their customers for cash payments or they can negotiate with another acquirer bank that offers a lower MSF. Higher prices will affect the consumer and the overall transaction volume might decrease. Also, offering cash payment discounts might decrease transactions made by credit cards as well. Despite all of these throwbacks, some merchants accept paying higher fees for credit card transactions because they can sell their products or services, even if their customers do not have sufficient balance. However, [11] states that some merchants are complaining about the fees that they are charged by the acquirer banks as well. This supports the idea that the price sensitivity of the merchants might differ from one to another.

The main reasons for these out-of-proportion fees are interchange fees and transfer fees. In recent years, merchant associations have sued card networks in the United States due to high interchange fees, and in some countries, regulations regarding interchange fees have been started to be made. The parameters considered to determine MSF have expanded due to regulations and the price sensitivity of the customers, transforming the pricing problem into a more complex task. Acquirer banks must carefully consider numerous parameters while determining this fee, to guarantee its profitability.

In the banking framework, churn refers to a phenomenon where customers discontinue their relationship with the financial institution [12, 13]. The risk of churn is a significant concern, and this study aims to reduce that risk by calculating the MSF that needs to be paid by merchants when they use POS devices for payments. Customer Retention studies mainly consist of data mining techniques since the price sensitivity of customers needs to reflect reality as much as possible. Without knowing how the customers will react to price regulations, it is not possible to have an impact on churn. Applications that aim to decrease churn, are used in various sectors such as banks, insurance companies, telecommunication companies, and much more [14, 15, 16]. Usage of these applications can be easily extended if there is sufficient information about the customers' price sensitivity.

In this thesis study, we are trying to retain merchants, that are simply clients of the acquirer banks, by pricing them with suitable MSF. In this setting, MSF stands for the fee that needs to be paid by the merchants to the acquirer bank if they use POS devices to sell their products or services. As mentioned, there are several factors to consider during the pricing step. One of the key considerations in this pricing process is finding a balance that ensures the satisfaction of both parties involved. While higher fees might initially appear financially attractive, they pose the risk of acquirer banks losing clients. Conversely, lower fees may reduce the bank's profitability, even though it might lower the risk of churn.

Beyond the profit and churn considerations, there's a broader regulatory landscape that influences the pricing strategy. In this problem setting, MSF is limited by regulations of the Central Bank of the Republic of Turkey(CBRT). Additionally, acquirer banks have internal policies, that forbid offering an MSF that is higher than merchants' existing fee.

Although it seems like a pricing problem, acquirer banks are essentially pricing the same service with varying rates. This makes it different than typical pricing scenarios where the same products/services are sold to various customers at a unique price. Pricing for one merchant does not affect the churn risk of another merchant, unlike traditional pricing where there is a concept of substitute and complementary products that affects the overall demand.

To the best of our knowledge, studies that aim to decrease churn by integrating mathematical optimization with prediction algorithms are very limited. To solve this pricing problem we developed a nonlinear optimization model with logistic regression integration but this model was not suitable for the setting due to its size, to solve the dimensionality problem, we developed an alternative Integer Programming(IP) model and enhanced it by designing a matheuristic that combines stratification and discretization techniques to further scale-up our solution technique. Our study's main contributions can be summarized as follows:

- A new variant of the product pricing problem was introduced.
- A non-linear programming(NLP) model with logistic regression integration was developed.
- Proposed model modified by discretization techniques and an Integer Programming(IP) model developed to solve the problem with high-scale instances.

- Performance of the IP presented by comparing it with the results obtained from the NLP model at small dimensions.

- Proposed IP model enhanced by integrating stratification techniques to further scale up our solution technique.

- Performance of the stratified IP compared with the solutions obtained from its first state and a promising performance detected.

The rest of the thesis is organized as follows. In section 2, existing literature is reviewed and categorized into three subsections, each focusing on a different aspect of our study. In section 3, we define the problem and provide a nonlinear programming model that is integrated with logistic regression. In section 4, we propose an integer programming model by discretizing the developed non-linear programming model and further developing it with stratification techniques. In section 5, we present several computational analyses to show the performance of each development made during the progression. Finally, in Section 6, we wrap up our study by providing a summary.

# CHAPTER II

# PREVIOUS WORK

In this section, concepts related to customer retention prediction algorithms, pricing-demand models and studies that combined churn prediction with optimization models will be introduced, along with the objective of this work.

## *2.1 Customer Retention Prediction Algorithms*

Churn is a crucial metric for profitability since it is five to six times more costly to obtain new customers when compared with retaining the existing ones [17]. Churn prediction models play a vital role in customer retention-related studies but we are not concerned with applying and analyzing the churn prediction models in this study. We are trying to leverage the insights, that are obtained after the application of these models, in an optimization frame. Data mining algorithms are extensively applied to a wide range of fields. For instance, they are used to forecast stock prices, predict disease outbreaks, understand customer preferences, and many more. To predict churn, algorithms like Naive Bayes, Support Vector Machines, Decision Trees, Artificial Neural Networks, and Logistic Regression are commonly used [12, 18, 14, 19, 20, 21, 22]. According to [23], regression is the frequently chosen approach for the churn prediction because of its highly accurate results and interpretability. In the next section, we will explain pricing and demand models that are designed by strategic utilization of these algorithms in detail.

## 2.2 Pricing and Demand Models

### 2.2.1 Pricing Models

Revenue Management(RM) organizes various tactics and strategies to control and optimize the demand within a business frame. It aims to connect the business to the market by dealing with very complex selling decisions to optimize revenues [24]. As stated in [25], maximizing the profit is solely dependent on pricing decisions since pricing strategies are used to manipulate the demand in various business frames with different methodologies. The reason for using various pricing methodologies is that the demand for the product is influenced by several parameters that have distinct dynamics in different business frames. Even though demand for the product has a negative relationship with its price in most of the pricing models, there are many different approaches developed in the literature but they can be grouped based on two fundamental characteristics: whether there is competition in the market and pricing strategy is fixed or dynamic [25]. Various extensions and assumptions are made within these groups but it is convenient to explain the general structure of these groups to understand how demand is manipulated in distinct problem frames and what are the differences in this thesis study.

In static and non-competitive models, multiple products need to be simultaneously priced by the seller and each product must have a unique price. In a similar study, belonging to Botimer and Belobaba [26], where airline pricing is the subject, the revenue function is designed as:

$$R = \sum_{i=1}^{N} P_i(1 - \sum_{j=i+1}^{N} d_{ij})Q_i + \sum_{i=1}^{N-1} \sum_{j=i+1}^{N} P_j d_{ij} Q_i$$

where passengers are allowed to choose the product that they want to use. There are N many products and $d_{ij}$ stands for the percentage of customers that divert their choice from the expensive product i to cheaper product j. $Q_i$ simply stands for the overall demand for the product i under the condition that when it's priced with $P_i$.

The first section of the summation simply calculates the revenue that is obtained from the customers that stick to their initial plan and don't switch to another product, the second section stands for the revenue that is obtained from the passengers that divert to product j. Even though the compact form of the function seems easy on the eye, there are complex relations between the expressions shown in the notation since the price of one product will directly affect both its and other products' demand and diversion ratio.

In the dynamic non-competitive models, the only difference is that the price of a product can be changed over the selling horizon(can be continuous or discrete-time) when compared with the static models. Gallego and Van Ryzin [27] used the following revenue maximization function in their paper for a similar problem frame:

$$\max \left\{ \int_0^t p(s)\lambda(p(s))ds \,\middle|\, \int_0^t A\lambda(p(s))ds \leq x; \quad p(s) \in P(s); \quad s \in [0,t] \right\}$$

In this expected revenue maximization equation, $p(s)$ stands for the price of the products, it is dependent on the timing horizon(shown as $s \in [0,t]$) and limited by the set $P(s)$, while the demand of the products is denoted by $\lambda(p(s))$ which is dependent to the time-specific price of the products(shown as $p(s)$). Multiplication of these terms is summed over the time horizon $[0,t]$ to find expected revenue. Matrix $A$ represents the amount of resource i needed to produce one unit of product j, while vector $x$ stands for the available resource i on hand.

When there is competition in the market, as the name suggests, demand for the product will be affected by the prices of the other sellers' products since these goods are highly substitutable or almost identical. It means the demand function must take the product prices of other sellers as an additional input.

[25] states that the Bertrand oligopoly game is often considered as one of the frequently studied and simple competitive models. In the Bertrand model, multiple firms sell substitutable products and compete on prices. In this model prices of the products can be cut down to the point where it is almost the same as the cost of

production since consumers will choose to buy the cheapest product in the market. This intense competition is also named as Bertrand Paradox. In this concept seller i optimizes its profit by using the following function:

$$\pi_i = (p_i - c_i)D_i(p_i, p_{-i})$$

As mentioned, the demand function($D_i()$) takes the product prices of other sellers(denoted as $p_{-i}$) as an additional input, even though the only decision the seller will make is the prices of its products(denoted as $p_i$). Since $c_i$ stands for the cost of product i, this value is subtracted from the price of the product to find a profit, which is multiplied by the expected demand to find the expected profit. In some variations, there might be some limitations for the price decision variable [28, 29]

### 2.2.2 Demand Models

As already mentioned, without knowing how the customer will react to price regulations, it is not logical to design pricing models. These models must be incorporated with each other to obtain realistic results. There are various demand models with distinct assumptions studied in the literature.

Demand models grouped with input parameter characteristics by Soon et al. [25] in their review article. Certain research work focuses on demand functions that only account for price variables to predict demand, while others consider additional parameters as well. Some demand models are structured on deterministic assumptions, whereas others have more realistic stochastic designs. As the name suggests, in deterministic models, the reactions of the customers to the price(or other if any) modifications were known. Even though, the deterministic demand functions are commonly structured in linear and concave form, functions like Logit, Cobb-Douglas and CES are studied as well. Although they are simpler models compared to stochastic ones, it is observed that the results obtained by them are good approximations of stochastic models.

The general structure for the deterministic linear demand function can be formalized as:

$$d_i(p) = b_i - a_{ii}p_i + \sum_{j \neq i} a_{ij}p_j$$

In this equation $d_i(p)$ stands for the demand of the product i when products are priced according to price vector $p$. The constant term $b_i$ can be considered as the maximum demand for product i when all of the prices are zero. The term $a_{ii}$ regulates how the demand for product i is affected by its price, when it's greater than zero, it indicates there is a negative relationship between demand and the price. On the other hand, $a_{ij}$ term regulates the price effect of product j on the demand of product i. When the products are substitutes of each other, it must be greater than zero, since the price increase in product j will increase the demand for product i. Conversely, the value must be lower than zero for the complementary products.

The more complex form of a deterministic model that uses the Logit function to calculate expected demand is formulated in [30] as follows :

$$D_{jt} = s_{jt} \cdot H, \quad s_{jt} = \frac{e^{\alpha(v_{jt} - p_{jt})}}{\sum_{i \in I} e^{\alpha(v_{it} - p_{it})} + 1}$$

In this formulation $D_{jt}$ stands for the demand for product j at time t, which is found by multiplying the probability of product j purchased at time t (denoted as $s_jt$) with a total number of households(denoted as $H$). The term $(v_{jt} - p_{jt})$ can be considered a metric for the satisfaction that consumers receive (a.k.a. consumer surplus). $v_{jt}$ stands for the typical consumer's willingness to pay for product j at time t, which is subtracted by the price of the product j at time t ($p_{jt}$) to calculate consumer surplus. The term $v_{jt}$ calculated by using following function:

$$v_{jt} = \frac{\beta_{0j} + \beta X_{jt} + \psi_{jt}}{\alpha}$$

Before explaining the notations, it will be convenient to simplify the equation since the $s_{jt}$ equation has several components which makes it hard to track the effect of parameters. If the term $\alpha(v_{jt} - p_{jt})$ represented as $\alpha(\frac{\beta_{0j}+\beta X_{jt}+\psi_{jt}}{\alpha} - p_{jt})$, $\alpha$ value in the denominator will cancel out and equation will be $(\beta_{0j} + \beta X_{jt} + \psi_{jt}) - \alpha p_{jt}$. After this conversion, we can write $s_{jt}$ as:

$$s_{jt} = \frac{e^{\beta_{0j}+\beta X_{jt}+\psi_{jt}-\alpha p_{jt}}}{\sum_{i \in I} e^{\beta_{0i}+\beta X_{it}+\psi_{it}-\alpha p_{it}} + 1}$$

$B_{0j}$ is the product-specific intercept value and summation of the terms $\beta X_{jt} + \psi_{jt}$ simply stands for the utility that is obtained from both observable and unobservable product attributes. Lastly, $\alpha$ is the coefficient of price variable.

## 2.3 Optimization-Prediction Algorithm Integration

Studies that combine mathematical optimization with data-mining techniques are very limited. Data mining and optimization combined with a 3-stage design in [31] to find optimal insurance premium pricing while considering the balance between profitability and market share.

Before explaining the stages of the developed method, it will be convenient to describe Markowitz's portfolio theory [32]. In portfolio theory, the aim is minimizing the variance(risk) while achieving the desired expected return. Risk can be associated with market share and the risk of churn, while profitability is with expected return.

Markowitz's model formulated in [31] as follows:

$$
\begin{aligned}
\text{Minimize} \quad & \sum_{i=1}^{N}\sum_{j=1}^{N} w_i w_j \sigma_{ij} \\
\text{s.t.} \quad & \sum_{i=1}^{N} w_i \mu_i = R^* \\
& \sum_{i=1}^{N} w_i = 1 \\
& 0 \leq w_i \leq 1, \quad i = 1, \ldots, N
\end{aligned}
$$

To minimize total variance, proportions of the budget used for assets i and j($w_i, w_j$) multiplied with the covariance between them($\sigma_{ij}$) and summation, over combinations of the assets (set $N$), minimized. In the first constraint, the multiplication of proportion and expected return of the assets($\mu_i$) summed and equalized the desired expected revenue target($R^*$) to achieve the expected return. Further, the summation of the proportions equalized to 1 and each of them is limited in range(0-1). The proposed model solved with various $R^*$ values to track risk-return tradeoff (efficient frontier).

[33, 34] focuses on clustering customers of an insurance company to categorize them into distinct risk groups and predict claim costs. Subsequently, in [15] neural networks were utilized to predict the price sensitivity of these clusters. Lastly in [31], a mathematical model was developed to determine optimal pricing based on the data obtained from these studies.

Insurance companies assess the risk of their customers by looking at the claim volume and frequency. Frequent and high-cost claims are associated with high-risk customers. Since risky customers will be charged with higher premiums, their price sensitivity will likely be different to price modifications, when compared to low-risk customers. To illustrate, if the premium charged for a customer that has almost no claims, is set too high compared to last year's offer, it is more likely that the customer will work with another insurance company. Conversely, a customer with frequent high-cost claims is likely to be insensitive to price modifications, at least up to some point.

The described idea was successfully supported by enhancing the performance of the neural networks, that are used to predict price sensitivity of the customers, by training each of them separately for each cluster. Since the application and analysis of data-mining techniques are out-of-scope for this study, we will focus on the section where the results obtained from the neural network are integrated into a mathematical optimization frame to calculate optimal premiums to charge customers.

In [31], the following non-linear integer programming model was developed to maximize expected revenue, while the overall termination rate was limited in the constraints:

**Table 1:** Decision Variables and Parameters

| **Variable** | **Description** |
|---|---|
| $x_{ij}$ | $\begin{cases} 1 & \text{if cluster } i \text{ has a termination rate } \alpha_j \\ 0 & \text{otherwise} \end{cases}$ |
| **Parameter** | **Description** |
| $N$ | Number of clusters |
| $P_i$ | Number of policies in cluster $i$ |
| $\rho_{ij}$ | Premium of cluster $i$ to achieve termination rate $a_j$ |
| $C_i$ | Average claim cost of cluster $i$ |
| $\alpha_j$ | Set of r termination rates |
| $\alpha_i$ | Desired termination rate for portfolio |

$$\text{Maximize} \quad \sum_{i=1}^{N} \left(1 - \sum_{j=1}^{r} \alpha_j x_{ij}\right) P_i \left(\sum_{j=1}^{r} \rho_{ij} x_{ij} - C_i\right) \tag{1}$$

$$\text{s.t.} \quad \sum_{j=1}^{r} x_{ij} = 1 \quad \forall i \tag{2}$$

$$x_{ij} \in \{0, 1\} \quad \forall i, j \tag{3}$$

$$\sum_{i=1}^{N} P_i (1 - \alpha_i) \leq (1 - \alpha_t) \sum_{i=1}^{N} P_i \tag{4}$$

Information of the overall termination rate($\alpha_j$) when the cluster i priced with the premium $p_{ij}$, is obtained from the neural network by using various premium pricing as an input for each cluster. In the objective function, the multiplication of retention rate $(1-\alpha_j)$, number of policies in cluster i ($P_i$), and the revenue $(p_{ij}-C_i)$ summed to maximize expected revenue. Usage of the binary variable can be changed to convert the non-linear model into linear form by converting the objective function to the following form:

$$\sum_{i=1}^{N}\sum_{j=1}^{r}(1-\alpha_j)P_i(\rho_{ij}-C_i)x_{ij}$$

Constraint 2 ensures that each cluster has just one termination rate which directly forces the model to price each cluster with a unique premium. Constraint 3 is an integrality constraint. Lastly, constraint 4 is designed to ensure that the overall termination rate is lower than the desired termination rate ($\alpha_t$) with a minor mistake which can be corrected by multiplying the left-hand side of the constraint with $x_{ij}$ and adding proper summation as follows:

$$\sum_{i=1}^{N}\sum_{j=1}^{r}P_i(1-\alpha_i)x_{ij} \leq (1-\alpha_t)\sum_{i=1}^{N}P_i$$

The results show that there is an increase in the expected revenue without impacting the overall termination rate (market share). This suggests that such an integration can enhance the performance of methods that are solely reliant on data-mining techniques.

In our problem frame, the objective is optimizing the pricing of merchant service fees, aiming to maximize expected profits while meeting specific company objectives. Additionally, we must consider some predefined fee limits influenced by company policies and governmental regulations. The differences and similarities between the existing studies can be summarized as follows:

- We are concerned about the price of the one and only service, provided by the acquirer bank. Merchants can not divert their preference with another service, since there are no substitute or complementary services. In the pricing models, whether the pricing is static or dynamic, the seller needs to consider the prices of the substitute and complementary products/services since these factors might manipulate the demand.

- Each merchant can be priced with different fees for the same service, similar to the dynamic pricing models. The only difference is, in the dynamic models pricing of the products/services changes over some time interval but in our study, we are not concerned about time-specific pricing. The same service can be priced differently for each merchant at the same time, which does not apply to dynamic pricing models.

- There is competition in the market but competitive pricing models are not quite applicable. Due to confidentiality reasons, accessing the merchant-specific pricing data of the other banks is impossible. The only factor that affects the demand is the price of the service itself and the price sensitivity of the merchants.

# CHAPTER III

# PROBLEM DEFINITION AND FORMULATION

Acquirer banks need to charge merchants when they use POS devices since there is a cost associated with the usage of these machines. In this thesis study, we are interested in an acquirer bank's problem, finding the optimal merchant service fees to maximize expected profit. We assume that the acquirer bank has 100,000 merchants (represented with the set N, where N = 1, 2, ..., 100,000) with various properties. Each of these merchants are existing member and already priced with a merchant service fee($kf_i$). Our objective is to re-offer a fee($k_i$) for the upcoming year's contract.

This fee will be charged to merchants when they use POS devices to make purchases. We have the information on each merchant's upcoming year's forecasted volumes for both debit($rd_i$) and credit card($rc_i$) transactions. The cost of using a POS device is not the same for all merchants and it changes with the type of the transaction. There are two types of transactions, which are made by a debit card and a credit card. The cost of using a credit card($cc$) is higher than the cost of a debit card($cd$). This is because when consumers use their credit card, the payment will be made by the issuer bank to the acquirer bank. This situation keeps the issuer bank in a risky position since there is a chance that the consumer might not pay this amount back to the issuer. That's why acquirer banks need to pay an extra fee to the issuer bank as explained in the payment industry eco-system.

There is no such risk when consumers use a debit card since it is almost similar to using cash. The payment will be directly paid by the issuer bank from the balance of the consumer, which makes debit card transactions cheaper compared to credit card transactions.

The industry has distinct regulatory limitations for various countries. This study is specifically designed for the framework that is used by the acquirer banks in Turkey. In Turkey, the maximum MSF($u$) is limited by the Central Bank of the Republic of Turkey (CBRT) [35] and the minimum fee is assumed to be limited by the bank itself. The limit on the minimum fee is simply the cost percentage of the merchant($Cp_i$), which is calculated by taking the weighted average of debit and credit card transaction volumes with the following formula:

$$Cp_i = \frac{rd_i \cdot cd + rc_i \cdot cc}{rd_i + rc_i}$$

Similar to the expected revenue target in Markowitz's portfolio theory [31], the acquirer bank desires to achieve some percentage of expected revenue($erp$) after the pricing as well. The reason for keeping the expected revenue percentage at some predetermined level can be associated with the bank's long-term growth. Even though this revenue target might lead to lower profit margins, it might be beneficial for keeping a strong relationship with its merchants.

Solution techniques proposed to solve this pricing problem must be capable of considering the merchant's response to price regulations to calculate "expected" revenue and profit. We combined the logistic regression technique, which calculates the price sensitivity of the merchant, with a mathematical model to optimize profitability. Logistic regression uses one or more independent variables to predict the probability of an event to occur by using the sigmoid function. We used this technique by defining the MSF as an independent variable to calculate the churn score. The sigmoid function and MSF variable were used as follows:

$$\text{Churn score}_i = \frac{e^{\beta_0+\beta_1 k_{i1}+\beta_2 k_{i2}+\dots}}{1+e^{\beta_0+\beta_1 k_{i1}+\beta_2 k_{i2}+\dots}}$$

The sigmoid function calculates the churn score of merchant $i$ by using various independent variables, but in our problem frame, all of the independent variables other than the MSF stay the same. It means that if $k_{i1}$ is treated as the MSF given for merchant $i$, the remaining part of the summation $\beta_0 + \beta_2 k_{i2} + \ldots$ becomes a constant value for each merchant, and this value can be symbolized as $I_i$ to make the function look simpler:

$$\text{Churn score}_i = \frac{e^{I_i + \beta_1 k_{i1}}}{1 + e^{I_i + \beta_1 k_{i1}}}$$

The following nonlinear programming model was designed to solve the described problem:

**Table 2:** Decision Variables and Parameters(NLP model)

| **Variable** | **Description** |
|---|---|
| $k_i$ | MSF given to customer $i$ |
| $c_i$ | Churn score for customer $i$ |
| **Parameter** | **Description** |
| $\text{kf}_i$ | Existing MSF of customer $i$ |
| $\text{rd}_i$ | Debit card transaction volume of customer $i$ |
| $\text{rc}_i$ | Credit card transaction volume of customer $i$ |
| cd | Cost rate for debit card |
| cc | Cost rate for credit card |
| $u$ | CBRT limit |
| erp | Expected Revenue Percentage |
| $I_i$ | Member-specific logistic regression value for customer $i$ |
| $B$ | Logistic regression coefficient for pricing variable |
| $\text{Cp}_i$ | Cost percentage of customer $i$ |

$$
\begin{aligned}
\max \quad & \sum_i (1-c_i)(\mathrm{rd}_i(k_i-\mathrm{cd})+\mathrm{rc}_i(k_i-\mathrm{cc})) \quad (1)\\
\text{s.t.} \quad & \frac{\sum_i(\mathrm{rd}_i+\mathrm{rc}_i)(k_i)(1-c_i)}{\sum_i(\mathrm{rd}_i+\mathrm{rc}_i)} \geq \mathrm{erp} \quad (2)\\
& c_i = \frac{e^{I_i+\beta k_i}}{1+e^{I_i+\beta k_i}} \quad \forall i \quad (3)\\
& k_i \leq u \quad \forall i \quad (4)\\
& k_i \leq \mathrm{kf}_i \quad \forall i \quad (5)\\
& k_i \geq \mathrm{Cp}_i \quad \forall i \quad (6)\\
& k_i, c_i \geq 0 \quad \forall i \quad (7)
\end{aligned}
$$

In the objective function(1), $(1-c_i)$ represents the retention score of the merchant $i$ and it's calculated by subtracting the churn score($c_i$) of the merchant from 1. $(rd_i(k_i-cd)$ represent the profit obtained from the debit transactions of the merchant $i$, it's calculated by multiplying the forecasted debit transaction volume of the merchant $i(rd_i)$ with the profit rate that will be obtained from the merchant's debit transactions. The profit rate($k_i - cd$) is simply calculated by subtracting the cost rate of using a debit card($cd$) from the MSF($k_i$) given to merchant $i$. Profit obtained from credit card transactions is calculated by using a similar formula($rc_i(k_i - cc)$). Expected profit from merchant $i$ calculated by multiplication of the retention score and the profit obtained from merchant's both debit and credit card transactions. Summation of the expected profit over all merchants, maximized in the objective function to ensure optimal expected profitability.

The left-hand side of the constraint(2) calculates the expected revenue percentage obtained after the pricing. The numerator part is similar to the objective function with just one distinction, which is the subtraction of costs. Retention score and total transaction volume($rd_i + rc_i$) are directly multiplied with the MSF($k_i$) and summed over all merchants to find expected revenue. This value is divided by the total transaction volume(denominator) of all merchants to find the percentage of the expected revenue which is limited by the desired expected revenue percentage($erp$).

Constraint (3) dynamically calculates the churn probability of the customer $i(c_i)$ with varying MSFs by using the sigmoid function. Constraint (4) limits the given MSF by the regulations of the CBRT($u$). Constraint (5) ensures that the given MSF is not higher than the merchant's initial MSF($kf_i$). Constraint (6) ensures that the member is not priced lower than its associated cost($Cp_i$). Finally, non-negativity restrictions are represented in Constraint (7).

The multiplication of the churn score($c_i$) and MSF($k_i$) in the objective function, and the division of the MSFs in the constraint(3) are the reasons why the proposed mathematical model is non-linear. Solution techniques, developed to solve these models demand substantial computational resources, which makes it impractical to handle the proposed model with large-scale instances.

# CHAPTER IV

# PROPOSED SOLUTION APPROACH

In this section, we will represent an alternative integer programming model to solve the scaling problem and develop a matheuristic by integrating the proposed IP with stratification techniques, to further scale up our solution technique.

## *4.1 Integer Programming Model*

Nonlinear programming models are not applicable to large-scale problems due to inefficient solution techniques. We proposed an alternative integer programming model, to solve the scaling problem, by using discretization techniques. Before presenting the developed model, it would be convenient to present the discretization technique used in IP.

The main reason for the non-linearity in the proposed model can be considered as the sigmoid function which is used to calculate the probability of churn with varying MSFs in a dynamic manner. When the structure of the sigmoid function is examined, we can see that the churn score and given MSF are interconnected. If we use pre-defined sets for the MSF and its corresponding churn score, we can easily get rid of non-linearity by simply assigning one value from each set to each merchant. To illustrate, let's say MSF is limited between 1.4%(0.014) to 2.2%(0.022) for the merchants and we want to offer MSFs with 3 decimal precision. It means only values that can be priced for the merchants must be in the following set:

$k_i \in \{1.4\%, 1.5\%, 1.6\%, 1.7\%, 1.8\%, 1.9\%, 2.0\%, 2.1\%, 2.2\%\}$

We can calculate corresponding churn scores($c_i$) for each merchant by using the sigmoid function since every value used in the function becomes a simple parameter:

$$c_i = \frac{e^{I_i+\beta k_i}}{1 + e^{I_i+\beta k_i}}$$

The tuning methodology for the values of the parameters ($I_i$ and $\beta$) will be explained in the computation section, but let's assume that for a specific merchant, the value of $I_i$ used as -2 and $\beta$ used as 300. For each MSF following churn scores will be found when given values placed into the sigmoid function:

$c_i \in \{90.02\%, 92.41\%, 94.27\%, 95.69\%, 96.77\%, 97.59\%, 98.20\%, 98.66\%, 99.00\%\}$
It means, that if the given MSF is 1.4%, the merchant's churn score will be 90.02% and the same pattern applies for all MSF-churn rate pairs. If we repeat the churn score calculation for all merchants by using merchant-specific parameters, the only thing remaining left to do is the assignment of these pairs for each merchant. Which converts the problem structure into an assignment problem.

The following IP was designed by using the defined discretization technique to scale up the proposed NLP:

**Table 3:** Decision Variables and Parameters(IP model)

| Variable | Description |
|---|---|
| $x_{ij}$ | $\begin{cases} 1 & \text{if merchant } i \text{ priced at section } j \\ 0 & \text{otherwise} \end{cases}$ |
| **Parameter** | **Description** |
| $\mathrm{kf}_i$ | Existing MSF of merchant $i$ |
| $c_{ij}$ | Churn score of merchant $i$ when priced at section $j$ |
| $k_j$ | MSF at section $j$ |
| $\mathrm{rd}_i$ | Debit card transaction volume of merchant $i$ |
| $\mathrm{rc}_i$ | Credit card transaction volume of merchant $i$ |
| cd | Cost rate for debit card |
| cc | Cost of credit card |
| $u$ | CBRT limit |
| erp | Expected Revenue Percentage |
| $\mathrm{Cp}_i$ | Cost percentage for merchant $i$ |

$$\max \quad \sum_i \sum_j (1 - c_{ij})(\mathrm{rd}_i(k_j - \mathrm{cd}) + \mathrm{rc}_i(k_j - \mathrm{cc}))x_{ij} \tag{1}$$

$$\text{s.t.} \quad \frac{\sum_i \sum_j (\mathrm{rd}_i + \mathrm{rc}_i)(1 - c_{ij})k_j x_{ij}}{\sum_i (\mathrm{rd}_i + \mathrm{rc}_i)} \geq \mathrm{erp} \tag{2}$$

$$\sum_j k_j x_{ij} \leq u \quad \forall i \tag{3}$$

$$\sum_j k_j x_{ij} \leq \mathrm{kf}_i \quad \forall i \tag{4}$$

$$\sum_j k_j x_{ij} \geq \mathrm{Cp}_i \quad \forall i \tag{5}$$

$$\sum_j x_{ij} = 1 \quad \forall i \tag{6}$$

$$x_{ij} \in \{0, 1\} \quad \forall (i, j) \tag{7}$$

In the IP model, MSFs are divided into n many sections(represented with the set K, where K=1,2,...,n ), and corresponding churn scores are calculated for each section in a merchant-specific manner. All of the calculated values are stored to be used as parameters in the IP model. It means that we have the data of possible MSFs($k_j$) and what will be the merchants' churn scores after the pricing($c_{ij}$). A binary decision variable ($x_{ij}$) is defined to assign these MSF sections to merchants while limitations are handled by the IP model presented above.

Structural differences between the IP and NLP model can be summarized as:

- Churn prediction calculation, with varying MSFs, constraint is no longer needed since these calculations were made beforehand and used as parameters in the IP.

- Assignment constraint (6) added to ensure that each merchant is priced with a unique MSF in IP, which was not a requirement for the NLP.

- Decision variables used in the NLP were continuous, while they are binary in the IP.

Constraint (6) simply limits the summation of binary variable $x_{ij}$ to 1, over all MSF sections, for each merchant, to force the model to assign unique MSFs. Which makes multiplication of $x_{ij} * k_j$ acts like $k_i$ decision variable in the NLP, when it's summed over MSF sections. The remaining part of the IP is designed identically to the NLP.

Various commercial solvers can solve IP models with very large-scale instances. In the developed IP model, the only storage capacity-consuming parameter is $c_{ij}$. The size of the $c_{ij}$ matrix (N*K) increases with the number of merchants and MSF sections. If we keep increasing the size of the instance, it will become impossible to initialize all the matrices and store them as parameters. That's why, the number of MSF sections needs to be well calculated to solve an instance with 100,00 merchants. Making too few divisions may reduce the expected profit while making too many divisions will make it impossible to solve due to computational requirements. Numerical studies on the number of divisions will be represented in the computational studies, but it has been noticed that the necessary division amount demands a significant amount of computational capacity.

## 4.2 Matheuristic

After realizing that the computational capacity is exceeded to solve the proposed IP for whole merchants with the required division amount, stratification techniques are integrated into the solution methodology. Dividing the population into many samples and solving the defined IP separately will decrease the required computational capacity but there might be some throwbacks.

- Summation of the expected profits obtained from each sample might be lower than the value that obtained by solving the IP for the population.
- Problem might become infeasible for some of the samples, due to the expected revenue percentage target. It means, that if the division is not properly set, the desired *erp* obtained from the population might not be obtained from each sample, which makes it impossible to solve the IP with the same *erp* target for each sample.

Certain characteristics of each sample must reflect the population as much as possible to decrease infeasibility risk. To find these characteristics and properly divide the population, the structure of the *erp* constraint needs to be examined:

$$\frac{\sum_i (\mathrm{rd}_i + \mathrm{rc}_i)(k_i)(1 - c_i)}{\sum_i (\mathrm{rd}_i + \mathrm{rc}_i)} \geq \mathrm{erp}$$

If summation at the denominator is moved to the right-hand side and the retention score of merchant i is symbolized as $r_i$ instead of $(1 - c_i)$, all merchant-specific characteristics will be gathered to the left-hand side:

$$\sum_i (\mathrm{rd}_i + \mathrm{rc}_i)(k_i)(r_i) \geq \mathrm{erp} \cdot \sum_i (\mathrm{rd}_i + \mathrm{rc}_i)$$

In this equation, revenue obtained from both debit and credit card transactions($rd_i + rc_i$) is known for each merchant but the given MSF ($k_i$) and retention score of the merchant after the pricing($r_i$) are decision variables, and vary for each merchant. Ignoring these decision variables and stratifying the population, by only using each merchant's total transaction volume, will simply not work for our objective. There will be a risk of distributing churners unevenly, which will contradict with obtaining the same *erp* from each sample.

Other than the total transaction volumes of the merchants, churners need to be distinguished to be divided evenly between each sample. The status of a merchant, labeled as "churner", can be altered by offering reduced MSFs but if the merchant is not highly sensitive to price modifications, its churn score will remain relatively stable. This makes it even more challenging to understand which merchants are already lost and which ones can be retained after proper pricing. When it comes to merchants with low initial churn scores, the situation is somewhat simpler. We cannot reclassify them as churners since pricing higher than their initial MSF is restricted by the bank.

Even after, all of these parameters are considered to stratify the population, there will be still a risk of infeasibility. It's not possible to guarantee that, the same *erp* target can be achieved from each sample and the population. Also, there is no guarantee that the expected profit will not be affected when the same *erp* target is obtained from each sample and the population.

To handle this infeasibility risk and enhance the robustness of our solution technique, further analysis related to the *erp* parameter is still required. As an initial step, it appears appropriate to start by stratifying the population by merchants' total transaction volumes and initial churn scores and develop this technique if the desired results are not obtained.

### 4.2.1 Stratification

By stratification, the population is partitioned into homogeneous subgroups with certain characteristics. Characteristics under consideration are continuous, as they represent the initial churn score and total transaction volumes of the merchants. To use these continuous values in stratification, the following steps are used:

- Merchants divided by their initial churn score and total transaction volume characteristics, into n*n many clusters.

Let's consider a scenario with 100 merchants that need to be divided into 4 clusters(n is 2). To achieve this, we need to sort merchants' churn scores to separate first 50 from the last 50. We will repeat the same procedure by sorting transaction volumes for each cluster. After that, there will be 4 clusters with 25 distinct merchants.

- By randomly selecting a single merchant from each cluster, until the desired sample size is achieved, k many samples are created.

The value of k requires calculations to ensure each sample has the same amount of distinct merchants. If k is used as 4, each sample must have 25 merchants but it is not possible to randomly select 6.25(25/4) distinct merchants from each cluster. If k is used as 8, each sample must have 12.5 merchants, which is not possible as well. That's why the amount of merchants in each sample must be multiple of 4 and divisible by 100 without a remainder. When k is used as 5, there will be 20 merchants in each sample and there will be 5 distinct merchants in each sample.

Numerical calculations for stratifying 100,000 merchants will be provided in the computational studies section.

### 4.2.2 Bounds for expected revenue percentage

There is a tradeoff between expected revenue and expected profit. If there is no *erp* target constraint, the model can be decomposed and solved separately for each merchant since other constraints independently limit the offered MSF for each merchant. In the objective function, the retention score of the merchant i when it's priced with MSF section j $(1 - c_{ij})$ multiplied by the profit obtained from merchant i after the pricing $(rd_i(k_j - cd) + rc_i(k_j - cc))$ to find expected profit. When there is no revenue target, all merchants will be priced to maximize expected profit within feasible MSF sections. By following the algorithm, we can easily calculate the optimal MSFs and maximum expected profit, when there is no *erp* target:

**Algorithm 1:** Maximum Expected Profit Without Revenue Target

```
Input  : Set of all merchants,c,rd,rc,cd,cc
Output: Maximum Expected Profit and MSF
max_expected_profit ← 0;
for each merchant i do
    temp_msf ← 0;
    highest_expected_profit ← 0;
    for each k in feasible MSF (for merchant i) do
        temp_expected_profit ← expected_profit(k, c[i, k], rd[i], rc[i], cd, cc);
        if highest_expected_profit < temp_expected_profit then
            temp_msf ← k;
            highest_expected_profit ← temp_expected_profit;
        end
    end
    MSF[i] ← temp_msf;
    max_expected_profit ← max_expected_profit + highest_expected_profit;
end
return max_expected_profit and MSF;
```

Parameters$(c_{ij}, rd_i, rc_i)$ used in IP formulation, given in matrix form to use as an input in Algorithm 1 with the same notation. The inner loop cycles over feasible MSFs to find the highest expected profit that can be obtained from the merchant and keeps the corresponding fee. Feasible MSFs mean the cycled fee values are higher than merchant's cost percentage$(Cp_i)$ and lower than merchant's initial MSF$(kf_i)$ or governmental limit($u$) to satisfy limitation constraints. Expected profit for a given fee is calculated by using the same equation in the objective function:

$$(1-c)(rd(k-cd)+rc(k-cc))$$

Outer loop cycles over each merchant to calculate the maximum expected profit gained from all the merchants by summing the highest expected profits. Also, saves optimal fees in the MSF vector for each merchant to calculate obtained expected revenue and percentage by:

**Algorithm 2:** Expected Revenue Percentage

```
Input  : Set of all merchants, MSF, c, rd, rc
Output: Expected Revenue and Expected Revenue Percentage
total_expected_revenue ← 0;
expected_revenue_percentage ← 0;
total_transaction_volume ← 0;
for each merchant i do
    total_expected_revenue ← total_expected_revenue +
      expected_revenue(MSF[i], c[i, MSF[i]], rd[i], rc[i]);
    total_transaction_volume ← total_transaction_volume + rd[i] + rc[i];
end
expected_revenue_percentage ← total_expected_revenue / total_transaction_volume;
return total_expected_revenue and expected_revenue_percentage;
```

The for loop iterates over each merchant and sums the expected revenue gains to calculate the total expected revenue. It also aggregates the debit and credit card transaction volumes of each merchant to determine the total transaction volume. Expected revenue calculation is made by using the following equation:

$$(rd+rc)(1-c)k$$

The value of MSF[i] is simply used as $k$ to calculate the expected revenue gain from merchant i when it's priced with no revenue target. Once the for loop has completed its execution, the total expected revenue is divided by the total transaction volume to find the expected revenue percentage.

The expected revenue percentage value found by Algorithm 2 represents the minimum *erp* bound that makes expected revenue constraint binding. Even when the merchants priced to maximize expected profit, without any revenue consideration, this percentage of expected revenue will be achieved. Targeting an *erp* that is lower than this value, will make *erp* constraint non-binding.

To find an upper bound for the *erp* target, the following algorithm, that maximizes expected revenue, can be used:

**Algorithm 3:** Calculate Max Expected Revenue

```
Input  : Set of all merchants, c, rd, rc
Output: Maximum Expected Revenue
max_expected_revenue ← 0;
for each merchant i do
    highest_expected_revenue ← 0;
    for each k in feasible MSF (for merchant i) do
        temp_expected_revenue ← expected_revenue(k, c[i, k], rd[i], rc[i]);
        if highest_expected_revenue < temp_expected_revenue then
            highest_expected_revenue ← temp_expected_revenue;
        end
    end
    max_expected_revenue ←
      max_expected_revenue + highest_expected_revenue;
end
return max_expected_revenue;
```

Algorithm-3 has a very similar structure with the algorithm-1. The only difference is that instead of maximizing expected profit, it's maximizing expected revenue.

Following this computation, the maximum expected revenue can be divided by total transaction volume to find an upper bound for the *erp* target. It will be infeasible to use higher *erp* values since this percentage is derived by calculating the maximum expected revenue.

After these calculations, the lower bound, makes the expected revenue constraint binding, and the upper bound, makes the problem infeasible, of the *erp* target will be known. Once this stage is reached, our task is to find an effective approach for deciding *erp* targets for samples, to lower the expected profit gap between solutions found by solving the IP for the entire population and each cluster. The designed approach will be represented in the computational study section, supported by numerical justifications.

# CHAPTER V

# COMPUTATIONAL STUDY

This section includes 2 subsections. In the first section, data generation steps for parameters, numerical studies to determine sample size in stratification, and expected revenue percentage ($erp$) targets for each sample will be explained. In the second section, the applicability of the NLP model, the optimality gap between the NLP and IP model, the performance of the matheuristic, and a comparison between models with static and dynamic churn scores will be presented. Lastly, the hardware and software-related detailed information will be provided.

## 5.1 Data Generation and Analysis

In the payment industry, it is exceedingly challenging to access and report customer-specific data due to confidentiality issues. Also, generating purely random data might not reflect reality and results might give misleading information. To avoid any conflicts, the data generation methodology is designed with the collaboration of a bank with a confidentiality agreement that forbids to use and reporting of any real data. The parameters, required to apply presented models, and techniques used to generate this information can be summarized as follows:

- $cd$ and $cc$ (Cost rate for debit and credit card)

These rates are provided by the bank itself. It includes the cost of the infrastructure, fees given to payment networks and issuer banks, and operational costs. As mentioned, the cost of using a credit card is higher than using a debit card due to extra fees paid to accept credit cards for transactions.

- $u$ (Max MSF limit)

This rate is announced by the CBRT in [35].

- $kf_i$ (Existing MSF)

In reality, most of the merchants priced with the maximum limit. The reason for observing that kind of pricing pattern is associated with the limited application of optimization algorithms during the pricing stage. Instead of using the max limit as existing MSF, values are randomly distributed between the cost rate of accepting credit cards($cc$) and the maximum limit($u$) to create diversity.

- $rd_i$ and $rc_i$ (Debit and Credit card transaction volumes)

After eliminating the outliers from the actual data, minimum and maximum transaction volumes are used as lower and upper boundaries to randomly generate the values.

- $erp$ (Expected Revenue Percentage Target)

There exists a balance between the expected revenue target and expected profit. In the previous section, the methodology to find a lower bound, that makes the expected revenue constraint binding, and an upper bound for $erp$ target is explained. Pursuing higher expected revenues is mostly associated with the bank's long-term strategy. Despite the risk of obtaining lower profit margins, the relationships between the bank and merchants can get stronger. As such, the value of this parameter can be changed by the dynamic nature of the bank's strategy. The detailed numerical analysis will be provided in the $erp$ target analysis section.

- $I_i$ and $B$ (Churn and Price Sensitivity)

Training logistic regression models with real data and reporting these results are restricted by the bank due to confidentiality agreements. Also, the application and analysis of such prediction algorithms are out of the scope of this study. That's why a methodology to tune these parameters must be designed to reflect the merchant's price sensitivity properly.

Sign of the pricing coefficient($B$ parameter) can be easily found when the structure of the sigmoid function is analyzed:

$$c_i = \frac{e^{I_i+\beta k_i}}{1+e^{I_i+\beta k_i}}$$

In this study, it is assumed that lower pricing will yield lower churn scores. That's why, the sign of the pricing coefficient must be positive. To illustrate, when the values of $B$ and $I_i$ are used as 50 and 0, the churn score of the merchant will decrease from 78% to 67% if the given MSF changes from 2.5% to 1.4%. But if the value of $B$ changed to -50, the churn score will increase from 22% to 33%, which will contradict with our assumption.

Values of the $I_i$ parameter will be randomly generated within an interval to create merchants with different price sensitivities. When deciding these bounds, the effect of the pricing term ($\beta k_i$) must be considered as well. If the range is not calculated properly, it can undermine or overestimate the impact of the pricing. Since the MSF rates are relatively small numbers, using a larger $B$ value might seem appropriate to increase its impact but the same thing holds for the value of this parameter as well. Using too big values will overestimate the impact of pricing while using relatively small numbers will underestimate the effect of pricing.

Balance between these values is very important to mimic reality as much as possible. If the data is generated by using inappropriate values, two issues may arise. Firstly, the maximum churn score among merchants can be too low, which makes the entire population consist of very low-risk churners. Secondly, the churn scores of the merchants might stay relatively stable, even when the given MSF decreases to its lowest point, which makes the population consist of price-insensitive merchants.

Through a process of hand-picking, various combinations of these parameters were examined and found that setting the value of $B$ as 300 and defining the bounds for the $I$ within -2 to -7 led to the creation of a diverse population as represented in Table-4.

**Table 4:** Tuning Parameters($B$ and $I$)

| I\MSF | 2.70% | 2.60% | 2.50% | 2.40% | 2.30% | 2.20% | 2.10% | 2.00% | 1.90% | 1.80% | 1.70% | 1.60% | 1.50% | 1.40% |
|---|---|---|---|---|---|---|---|---|---|---|---|---|---|---|
| -2 | 99.8% | 99.7% | 99.6% | 99.5% | 99.3% | 99.0% | 98.7% | 98.2% | 97.6% | 96.8% | 95.7% | 94.3% | 92.4% | 90.0% |
| -3 | 99.4% | 99.2% | 98.9% | 98.5% | 98.0% | 97.3% | 96.4% | 95.3% | 93.7% | 91.7% | 89.1% | 85.8% | 81.8% | 76.9% |
| -4 | 98.4% | 97.8% | 97.1% | 96.1% | 94.8% | 93.1% | 90.9% | 88.1% | 84.6% | 80.2% | 75.0% | 69.0% | 62.2% | 55.0% |
| -5 | 95.7% | 94.3% | 92.4% | 90.0% | 87.0% | 83.2% | 78.6% | 73.1% | 66.8% | 59.9% | 52.5% | 45.0% | 37.8% | 31.0% |
| -6 | 89.1% | 85.8% | 81.8% | 76.9% | 71.1% | 64.6% | 57.4% | 50.0% | 42.6% | 35.4% | 28.9% | 23.1% | 18.2% | 14.2% |
| -7 | 75.0% | 69.0% | 62.2% | 55.0% | 47.5% | 40.1% | 33.2% | 26.9% | 21.4% | 16.8% | 13.0% | 10.0% | 7.6% | 5.7% |

The first column and row represent the values of $I$ and MSFs, the resulting churn scores are located at the intersection of these values. There are merchants with distinct price sensitivities and initial churn scores, as desired. Specifically, the rows, corresponding to the calculations when the value of $I$ is used as -2 and -4, show that even the merchants' churn scores are initially similar, when their MSF is set at 2.7%, their reactions to price adjustments differ significantly.

### 5.1.1 Sample size

Using larger sample sizes when dividing the population through stratification will increase the risk of getting lower expected profit. Conversely, using the entire population at once is not possible due to computational requirements. Other than the size of the sample, the MSF division amount affects the requirement of computational resources as well. Numerical analysis related to the required division amount will be presented in the next section by showing how the optimality gap is affected by the division amount. It is observed that solving the IP for more than 10,000 merchants is not possible when the necessary division amount is used.

To ensure that each sample created by stratification, does not exceed 10,000 merchants, the population(100,000 merchants) should be divided into 10 samples. To achieve this, the population is first divided into 25 clusters(n set to 5), each having

4000 merchants with distinct characteristics. Subsequently, by randomly selecting 400 merchants from each cluster, 10 samples with 10,000 merchants were created.

### 5.1.2 Expected revenue target for samples

Feasible *erp* bounds can be obtained by using a bound calculation technique for the population and each sample but using the same *erp* target for each sample does not necessarily guarantee the optimality. Also, the desired *erp* target might not be feasible for each sample as well.

It will be convenient to present the proposed *erp* tuning technique with expected revenue, instead of percentage.

$$\frac{\sum_i (\mathrm{rd}_i + \mathrm{rc}_i)(k_i)(1 - c_i)}{\sum_i (\mathrm{rd}_i + \mathrm{rc}_i)} \geq \mathrm{erp}$$

When the denominator in the right-hand side of the equation, represents total transaction volumes for all merchants, moved to right-hand side, multiplication of *erp* and $\sum_i (\mathrm{rd}_i + \mathrm{rc}_i)$ can be defined as expected revenue target. When this conversion is applied to *erp* bounds, expected revenue bounds can be found for the population and each sample.

In order to tune expected revenue targets for each sample, the following algorithm is designed:

**Algorithm 4:** Tuning Expected Revenue

```
Input  : max_expected_revenues, min_expected_revenues, T,
         expected_revenue_target, N
Output: tuned_expected_revenues

tuned_expected_revenues ← max_expected_revenues;
sum_tuned_expected_revenue ← ∑(tuned_expected_revenues);
decrement_vector ← (max_expected_revenues − min_expected_revenues)/T;
check ← False;
while expected_revenue_target < sum_tuned_expected_revenue do
    for i ← 1 to N do
        if expected_revenue_target < sum_tuned_expected_revenue then
            tuned_expected_revenues[i] ←
              tuned_expected_revenues[i] − decrement_vector[i] ;
            sum_tuned_expected_revenue ← ∑(tuned_expected_revenues);
        end
        if expected_revenue_target > sum_tuned_expected_revenue then
            tuned_expected_revenues[i] ←
              tuned_expected_revenues[i] + decrement_vector[i] ;
            check ← True;
            break;
        end
    end
    if check then
        break;
    end
end
```

Maximum and minimum expected revenues represent the bounds for each sample. nth indice of the maximum expected revenue vector represents the maximum bound for the nth sample and the same holds for the minimum expected revenue vector. N represents the number of samples and the expected revenue target is the expected revenue desired to obtain from the population.

Tuned expected revenues are initialized by using the maximum expected revenue vector. Values inside the vector are summed to calculate what will be the expected revenue obtained from the population when the maximum expected revenue bounds used for each sample. Using these values as expected revenue targets in IP will yield the lowest possible expected profit from each merchant since targeting higher expected revenues will lower the obtained expected profit.

Values in the tuned expected revenues vector will be decreased in consecutive order until the sum of these values gets close to the expected revenue target. The values in the decrement vector will be used to decrease these revenue targets and uniquely calculated for each sample by dividing their maximum and minimum bounds with the number T. Using a relatively large number for T will be sufficient. In every decrement step, the summation of the tuned expected revenues will be getting close to the expected revenue target. If the value of T is relatively small, the values in the decrement vector will increase. Bigger decrement values will increase the convergence rate but the difference between the expected revenue target and summation of tuned expected revenues might increase as well. Since this calculation does not require extensive computational power, the value of T can be used relatively large to converge these values as much as possible.

## 5.2 Model Evaluation

In this section, the following points are checked in presented order to validate the solution technique progression:

- Whether the NPL model is applicable to large instances or not. If not, what is the largest instance size that can be solved to optimality within a desired time limit?

- Is the gap between the solutions obtained by applying NLP and IP models to the same instances within the desired range?

- Whether the IP model is applicable to large-scale problems or not. If it is, what is the maximum number of merchants that can be solved with the required division amount?

- Is the gap between the solutions obtained by solving the same instance with IP and Matheuristic within the desired range?

### 5.2.1 Applicability of NLP model

It is observed that the NLP model is impractical for large-scale datasets. On average, it takes nearly one hour to solve an instance with 10 merchants to optimality, and the solution time exponentially increases as the instance size grows, as illustrated in Table 5

**Table 5:** Runtimes for Instances with Different Numbers of Merchants(in minutes)

| **Instance** | **5 merchants** | **10 merchants** | **15 merchants** |
|---|---|---|---|
| 1 | 35 | 76 | 142 |
| 2 | 28 | 63 | 131 |
| 3 | 22 | 38 | 84 |
| 4 | 30 | 66 | 127 |
| 5 | 40 | 94 | 147 |
| 6 | 24 | 65 | 134 |
| 7 | 16 | 38 | 92 |
| **Average** | 27.86 | 62.86 | 122.43 |

### 5.2.2 Optimality gap between the NLP and IP model

As a next step, the same instances are solved by IP, with varying MSF sections, to compare the gap, between the optimal solutions of the IP and NLP model. When the division amount is set to 2 decimal precision, the model becomes infeasible. 2 decimal precision will create MSFs with 1% increments which will be pointless since the MSF fluctuates between 1.4% and 2.7% and only MSF can be priced to merchants will be 2%. The results when the decimal precision starting from 3 to 6, represented in Tables 6,7,8 and 9.

**Table 6:** Comparison of NLP-IP Solutions (3 decimal precision)

| Instance | Expected Profit (NLP) | Expected Profit (IP) | Difference |
|---|---|---|---|
| 1 | $230,232 | $228,867 | $1,365 |
| 2 | $173,891 | $173,437 | $454 |
| 3 | $227,679 | $227,439 | $240 |
| 4 | $312,580 | $312,341 | $239 |
| 5 | $274,852 | $274,664 | $188 |
| 6 | $340,835 | $340,583 | $252 |
| 7 | $247,657 | $245,343 | $2,314 |

**Table 7:** Comparison of NLP-IP Solutions (4 decimal precision)

| Instance | Expected Profit (NLP) | Expected Profit (IP) | Difference |
|---|---|---|---|
| 1 | $230,232 | $230,222 | $10 |
| 2 | $173,891 | $173,888 | $3 |
| 3 | $227,679 | $227,677 | $2 |
| 4 | $312,580 | $312,579 | $1 |
| 5 | $274,852 | $274,849 | $3 |
| 6 | $340,835 | $340,831 | $4 |
| 7 | $247,657 | $247,639 | $18 |

**Table 8:** Comparison of NLP-IP Solutions(5 decimal precision)

| Instance | Expected Profit (NLP) | Expected Profit (IP) | Difference |
|---|---|---|---|
| 1 | $230,232 | $230,232 | $0 |
| 2 | $173,891 | $173,891 | $0 |
| 3 | $227,679 | $227,679 | $0 |
| 4 | $312,580 | $312,580 | $0 |
| 5 | $274,852 | $274,852 | $0 |
| 6 | $340,835 | $340,835 | $0 |
| 7 | $247,657 | $247,657 | $0 |

**Table 9:** Comparison of NLP-IP Solutions(6 decimal precision)

| Instance | Expected Profit (NLP) | Expected Profit (IP) | Difference |
|---|---|---|---|
| 1 | $230,232 | $230,232 | $0 |
| 2 | $173,891 | $173,891 | $0 |
| 3 | $227,679 | $227,679 | $0 |
| 4 | $312,580 | $312,580 | $0 |
| 5 | $274,852 | $274,852 | $0 |
| 6 | $340,835 | $340,835 | $0 |
| 7 | $247,657 | $247,657 | $0 |

Using more than 5 decimal precision will be unnecessary. Even if the differences are relatively small with 4 decimal precision, it might lead to greater amounts of expected profit loss when the IP algorithm is used with 100,000 merchants.

### 5.2.3 Sample Size and Performance of the Matheuristic

The applicability of the IP model to large-scale datasets was checked after finding an appropriate MSF division size(5 decimal precision). The RAM capacity of the hardware was found to be insufficient to handle more than 10,000 merchants. To show the performance of the matheuristic, populations must be divided into at least 10 samples since the actual problem aims to price 100,000 merchants.

In Table 10, the performance of the matheuristic, when the population size is used as 10,000 merchants and the division amount as 10, represented:

**Table 10:** Comparison of Expected Profit

| Instance Number | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| Expected Profit-IP | $224,429,557 | $220,371,752 | $225,518,229 | $227,869,484 | $231,478,347 | $229,930,057 | $230,402,008 | $228,245,314 |
| Expected Profit-Matheuristic | $224,429,557 | $220,370,156 | $225,518,226 | $227,869,484 | $231,478,347 | $229,930,057 | $230,402,008 | $228,245,314 |
| Difference | $0 | $1,596 | $4 | $0 | $0 | $0 | $0 | $0 |

To understand the benefit of using tuning mechanism, same *erp* target is used for population and each sample, and represented in Table 11.

**Table 11:** Comparison of Expected Profit (No Tuning)

| Instance Number | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| Expected Profit-IP | $224,429,557 | $220,371,752 | $225,518,229 | $227,869,484 | $231,478,347 | $229,930,057 | $230,402,009 | $228,245,315 |
| Expected Profit-Same Target | $224,262,157 | $220,281,936 | $225,478,368 | $227,785,586 | $231,478,347 | $229,929,984 | $230,402,009 | $228,029,018 |
| Difference | $167,400 | $89,817 | $39,861 | $83,898 | $0 | $73 | $0 | $216,296 |

The results prove that using the same *erp* target does not guarantee optimality. Solutions obtained by matheuristic yield better results(if any) when the same *erp* targets are used for each sample. Another benefit of using this mechanism is that it makes the solution technique more robust. If the *erp* target used for the population is not feasible for one or more samples, it is not possible to obtain a solution. However, this is not the case for the proposed solution technique.

Time requirements of the matheuristic are represented in Table 12.

**Table 12:** Comparison of Times (in seconds)

| Instance Number | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 | Average |
|---|---|---|---|---|---|---|---|---|---|
| IP | 2,190s | 2,405s | 2,238s | 2,277s | 2,925s | 2,887s | 2,887s | 2,896s | 2589s |
| Matheuristic | 2,106s | 2,298s | 2,143s | 2,166s | 2,960s | 2,970s | 2,970s | 2,992s | 2576s |
| Difference | 84s | 107s | 95s | 111s | -35s | -84s | -84s | -96s | 12s |

On average, it takes a similar time(approximately 43 minutes) to price 10,000 merchants at once by IP or matheuristic. It is observed that, on average, it takes 7.1 hours to price 100,000 merchants using the matheuristic method. Since using this solution technique once or twice in a year, is sufficient to price merchants, the time requirement of the algorithm is in the desired range. Also, using more advanced hardware will decrease the solution time as well.

### 5.2.4 Comparison between models with static and dynamic churn scores

To understand, the benefit of using dynamic churn scores, the following Linear Programming(LP) model, that uses static churn scores, designed:

**Table 13:** Decision Variables and Parameters for Static Model

| Variable | Description |
|---|---|
| $k_i$ | MSF given to customer $i$ |
| **Parameter** | **Description** |
| $c_{int}$ | Initial Churn score of customer $i$ |
| $\mathrm{kf}_i$ | Existing MSF of customer $i$ |
| $\mathrm{rd}_i$ | Debit card transaction volume of customer $i$ |
| $\mathrm{rc}_i$ | Credit card transaction volume of customer $i$ |
| cd | Cost rate for debit card |
| cc | Cost rate for credit card |
| $u$ | CBRT limit |
| erp | Expected Revenue Percentage |
| $\mathrm{Cp}_i$ | Cost percentage of customer $i$ |

$$\max \quad \sum_i (1 - c_{int})(\mathrm{rd}_i(k_i - \mathrm{cd}) + \mathrm{rc}_i(k_i - \mathrm{cc})) \quad (1)$$

$$\text{s.t.} \quad \frac{\sum_i (\mathrm{rd}_i + \mathrm{rc}_i)(k_i)(1 - c_{int})}{\sum_i (\mathrm{rd}_i + \mathrm{rc}_i)} \geq \mathrm{erp} \quad (2)$$

$$k_i \leq u \quad \forall i \quad (3)$$

$$k_i \leq \mathrm{kf}_i \quad \forall i \quad (4)$$

$$k_i \geq \mathrm{Cp}_i \quad \forall i \quad (5)$$

$$k_i \geq 0 \quad \forall i \quad (6)$$

In this model, the churn scores of the merchants are not dynamically changing. That's why directly comparing the results obtained by this model will be inappropriate. To make such a comparison, MSFs, found by the LP, need to be used in a sigmoid function to calculate the resulting churn score. It means, that even though the churn scores are static in the LP model, actual churn scores are calculated after solving the model, to make such a comparison.

The first observation is the *erp* targets, that were feasible, become infeasible due to static churn scores. After this observation, the structure of the objective function:

$$\sum_i (1 - c_{int})(\mathrm{rd}_i(k_i - \mathrm{cd}) + \mathrm{rc}_i(k_i - \mathrm{cc}))$$

and expected revenue constraint:

$$\frac{\sum_i (\mathrm{rd}_i + \mathrm{rc}_i)(k_i)(1 - c_{int})}{\sum_i (\mathrm{rd}_i + \mathrm{rc}_i)} \geq \mathrm{erp}$$

re-analyzed and the following differences were observed:

- In the objective function, all of the terms other than the $k_i$ decision variable are simply pre-known numbers. That's why pricing merchants with the maximum MSF will yield the maximum expected profit.

- The same holds for the expected revenue constraint as well. When the value of the $k_i$ variable increases, the resulting *erp* increases as well.

- Since there is no reason to price merchants lower than the maximum MSF, using the expected revenue constraint is simply not necessary. The tradeoff between expected profit and expected revenue no longer holds. The expected revenue obtained by maximizing the expected profit will be the maximum expected revenue that can be obtained from this model.

- As mentioned, there is no factor that forces the merchants to price themselves lower than their initial MSFs (maximum MSF can be given to merchants since their initial MSF is already smaller or equal to the CBRT limit). That's why actual churn scores will not change after the pricing as well.

After these observations, erp constraints were removed from the model, and results are represented in Table-14:

**Table 14:** Comparison of Dynamic and Static Churn Scores

| Instance Number | 1 | 2 | 3 | 4 | 5 | 6 | 7 | 8 |
|---|---|---|---|---|---|---|---|---|
| Expected Profit-IP | $224,429,556 | $220,371,752 | $225,518,229 | $227,869,484 | $231,478,347 | $229,930,057 | $230,402,008 | $228,245,314 |
| Expected Profit-Static Churn | $193,729,311 | $189,475,217 | $194,243,685 | $196,896,983 | $156,449,980 | $153,729,766 | $153,731,036 | $153,644,869 |
| Difference | $30,700,245 | $30,896,535 | $31,274,543 | $30,972,500 | $75,028,367 | $76,200,290 | $76,670,972 | $74,600,445 |
| Expected Revenue-IP | $404,623,719 | $400,956,744 | $401,496,128 | $405,834,709 | $405,462,483 | $403,960,382 | $402,613,882 | $404,030,764 |
| Expected Revenue-Static Churn | $325,227,188 | $317,316,666 | $323,929,033 | $327,795,789 | $247,496,333 | $243,393,401 | $243,271,838 | $244,588,236 |
| Difference | $79,396,530 | $83,640,078 | $77,567,094 | $78,038,920 | $157,966,150 | $160,566,980 | $159,342,043 | $159,442,527 |

In IP there was a tradeoff between expected revenue and expected profit. If the expected revenue target is lowered, it is expected to obtain higher expected profits but the results show that there will be a significant amount of expected profit and revenue loss if the price sensitivity of the merchants is not considered during the pricing. The reason for observing a significant amount of loss is that usage of static churn diminishes the worry of retention of the merchants. Whether they priced with maximum or minimum MSF, their churn rate stays the same, and that's why they priced to the highest feasible MSF. Achieving the *erp* target becomes impossible since the expected revenues shown in Table-14 are the maximum expected revenues that can be obtained by the static churn model.

## 5.3 Hardware and software details

NLP model is coded in AMPL format using the AMPL Integrated Development Environment (IDE), and as a solver, Octeract Engine (Non-linear global solver) is used. IP model is coded in the Python programming language using PyCharm IDE. The Pyomo library is used to code the IP model, and as a solver, Gurobi is used. Results are obtained on a laptop with an Intel 10870H processor and 2x16 gigabytes of RAM.

# CHAPTER VI

# CONCLUSIONS

As an overview of this study, we initially introduced a problem faced by the banking industry. Where acquiring banks need to optimize their pricing strategy to maximize profitability. The significance of various factors during the pricing stage was explored, such as the price sensitivity of the merchants, industry-specific targets, and restrictions. We also clarify how each of these factors affects each other and represent the problem by a mathematical model.

As an approach to solving this problem, a non-linear programming (NLP) model, with logistic regression integration, was constructed and solved by using a global non-linear solver. However as anticipated, these solvers are only capable of solving problems of small sizes as their current solving techniques require an extensive amount of computational power. Sequentially, an integer programming (IP) model has been developed by discretizing the nonlinearity in the first NLP model built. After a comparison has been done between the two models, it was realized that the IP model can produce solutions with insignificant gaps against the NLP model. The IP model successfully scale-up the NLP model but problem size required more efficient techniques. To further scale up the proposed IP, stratification techniques integrated and required instance size handled successfully.

The introduction of a new variant of the product pricing problem and the usage of logistic regression in the mathematical model environment are the two main contributions of this study. Results can be used as a valuable resource and change the methodology of the pricing, used by the banking industry. A similar methodology can be applied to other pricing problems by adapting the proposed models with new restrictions and targets, as well.

As a future study, the performance of the proposed solution methodology can be enhanced and compared by integrating prediction algorithms other than logistic regression in a similar optimization model.

In conclusion, this study has adapted the world of the payment industry into the optimization field. The significant balance between profitability and merchant satisfaction was explored and justified. Our research revealed the benefits of integrating data-mining techniques into the optimization field.

# REFERENCES


[1] M. Pulina, "Consumer behaviour in the credit card market: a banking case study," *International Journal of Consumer Studies*, vol. 35, no. 1, pp. 86–94, 2011.

[2] C. M. Insights, "Latest global credit card payments market size & share worth usd 286.5 billion by 2032 at a 9.1% cagr - custom market insights analysis, outlook, leaders, report, trends, forecast, segmentation." `https://shorturl.at/mDJL7`, 2023. Accessed on 2023-09-30.

[3] J. Bagnall, D. Bounie, K. P. Huynh, A. Kosse, T. Schmidt, S. D. Schuh, and H. Stix, "Consumer cash usage: A cross-country comparison with payment diary survey data," 2014.

[4] B. Pokora, "Credit card statistics." `https://shorturl.at/xKO57`, 2023. Accessed on 2023-09-30.

[5] N. Ramya, D. Sivasakthi, and M. Nandhini, "Cashless transaction: Modes, advantages and disadvantages," *International Journal of Applied Research*, vol. 3, no. 1, pp. 122–125, 2017.

[6] B. Fung, K. P. Huynh, G. Stuber, *et al.*, "The use of cash in canada," *Bank of Canada review*, vol. 2015, no. Spring, pp. 45–56, 2015.

[7] M. Thirupathi, G. Vinayagamoorthi, and S. Mathiraj, "Effect of cashless payment methods: A case study perspective analysis," *International Journal of scientific & technology research*, vol. 8, no. 8, pp. 394–397, 2019.

[8] N. Li and P. Zhang, "Consumer online shopping attitudes and behavior: An assessment of research," *AMCIS 2002 proceedings*, p. 74, 2002.

[9] S. Rochemont, "Issue 21-environmental sustainability of a cashless society," *Institute and Faculty of Actuaries*, 2018.

[10] J. Wright, "Pricing in debit and credit card schemes," *Economics Letters*, vol. 80, no. 3, pp. 305–309, 2003.

[11] J.-C. Rochet and J. Wright, "Credit card interchange fees," *Journal of Banking & Finance*, vol. 34, no. 8, pp. 1788–1797, 2010.

[12] A. Oyeniyi, A. Adeyemo, A. Oyeniyi, and A. Adeyemo, "Customer churn analysis in banking sector using data mining techniques," *African Journal of Computing ICT*, vol. 8, no. 3, pp. 165–174, 2015.

[13] K. Chitra and B. Subashini, "Customer retention in banking sector using predictive data mining technique," in *ICIT 2011 The 5th International Conference on Information Technology*, 2011.

[14] A. Keramati, H. Ghaneei, and S. M. Mirmohammadi, "Developing a prediction model for customer churn from electronic banking services using data mining," *Financial Innovation*, vol. 2, pp. 1–13, 2016.

[15] A. C. Yeo, K. A. Smith, R. J. Willis, and M. Brooks, "Modeling the effect of premium changes on motor insurance customer retention rates using neural networks," in *Computational Science-ICCS 2001: International Conference San Francisco, CA, USA, May 28—30, 2001 Proceedings, Part II 1*, pp. 390–399, Springer, 2001.

[16] L. Yan, R. H. Wolniewicz, and R. Dodier, "Predicting customer behavior in telecommunications," *IEEE Intelligent Systems*, vol. 19, no. 2, pp. 50–58, 2004.

[17] K. G. M. Karvana, S. Yazid, A. Syalim, and P. Mursanto, "Customer churn analysis and prediction using data mining models in banking industry," in *2019 international workshop on big data and information security (IWBIS)*, pp. 33–38, IEEE, 2019.

[18] S. H. Dolatabadi and F. Keynia, "Designing of customer and employee churn prediction model based on data mining method and neural predictor," in *2017 2nd international conference on computer and communication systems (ICCCS)*, pp. 74–77, IEEE, 2017.

[19] A. Bilal Zorić, "Predicting customer churn in banking industry using neural networks," *Interdisciplinary Description of Complex Systems: INDECS*, vol. 14, no. 2, pp. 116–124, 2016.

[20] J. Han, M. Kamber, and J. Pei, "Data mining concepts and techniques third edition," *University of Illinois at Urbana-Champaign Micheline Kamber Jian Pei Simon Fraser University*, 2012.

[21] T. Vafeiadis, K. I. Diamantaras, G. Sarigiannidis, and K. C. Chatzisavvas, "A comparison of machine learning techniques for customer churn prediction," *Simulation Modelling Practice and Theory*, vol. 55, pp. 1–9, 2015.

[22] A. H. Karp, "Using logistic regression to predict customer retention," in *Proceedings of the Eleventh Northeast SAS Users Group Conference. http://www. lexjansen. com/nesug/nesug98/solu/p095. pdf*, vol. 15, 1998.

[23] J. Hadden, A. Tiwari, R. Roy, and D. Ruta, "Computer assisted customer churn management: State-of-the-art and future trends," *Computers & Operations Research*, vol. 34, no. 10, pp. 2902–2917, 2007.

[24] G. J. van Ryzin and K. T. Talluri, "An introduction to revenue management," in *Emerging theory, methods, and applications*, pp. 142–194, Informs, 2005.

[25] W. Soon, "A review of multi-product pricing models," *Applied mathematics and computation*, vol. 217, no. 21, pp. 8149–8165, 2011.

[26] T. C. Botimer and P. P. Belobaba, “Airline pricing and fare product differentiation: A new theoretical framework,” *Journal of the Operational Research Society*, vol. 50, no. 11, pp. 1085–1097, 1999.

[27] G. Gallego and G. Van Ryzin, “A multiproduct dynamic pricing problem and its applications to network yield management,” *Operations research*, vol. 45, no. 1, pp. 24–41, 1997.

[28] P. Milgrom and J. Roberts, “Rationalizability, learning, and equilibrium in games with strategic complementarities,” *Econometrica: Journal of the Econometric Society*, vol. 58, no. 6, pp. 1255–1277, 1990.

[29] D. M. Topkis, “Equilibrium points in nonzero-sum n-person submodular games,” *Siam Journal on control and optimization*, vol. 17, no. 6, pp. 773–787, 1979.

[30] D. Besanko, S. Gupta, and D. Jain, “Logit demand estimation under competitive pricing behavior: An equilibrium framework,” *Management Science*, vol. 44, no. 11-part-1, pp. 1533–1547, 1998.

[31] A. C. Yeo, K. A. Smith, R. J. Willis, and M. Brooks, “A mathematical programming approach to optimise insurance premium pricing within a data mining framework,” *Journal of the Operational research Society*, vol. 53, pp. 1197–1203, 2002.

[32] H. M. Markowitz, “Portfolio selection,” *The Journal of Finance*, vol. 7, no. 1, pp. 77–91, 1952.

[33] A. C. Yeo, K. A. Smith, R. J. Willis, and M. Brooks, “Clustering technique for risk classification and prediction of claim costs in the automobile insurance industry,” *Intelligent Systems in Accounting, Finance & Management*, vol. 10, no. 1, pp. 39–50, 2001.

[34] A. C. Yeo, K. A. Smith, R. J. Willis, and M. Brooks, “A comparison of soft computing and traditional approaches for risk classification and claim cost prediction in the automobile insurance industry,” *Soft computing in measurement and information acquisition*, pp. 249–261, 2003.

[35] Central Bank of the Republic of Turkey, “Cbrt maximum commission rates for member businesses.” `https://shorturl.at/lvxIR`. Accessed on 2023-08-02.

# VITA

Berk Cihan Deniz completed his secondary education at Adem Tolunay Anatolian High School (ATAL). He holds a Bachelor of Science degree in Industrial Engineering, which he obtained from Özyeğin University in June 2020. Since October 2020, he has been actively pursuing a Master of Science degree in Industrial Engineering at Özyeğin University, with Dr. Enis Kayış serving as his supervisor. Berk's research focus revolves around the intriguing areas of customer churn and profit optimization.