MACHINE LEARNING-DRIVEN OPTIMIZATION: APPLICATIONS IN RETENTION MANAGEMENT AND RANK AGGREGATION

A Dissertation

by

Ahmet Şahin

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

Doctor of Philosophy

in the
Department of Industrial Engineering

Özyeğin University
June 2024

MACHINE LEARNING-DRIVEN OPTIMIZATION: APPLICATIONS IN RETENTION MANAGEMENT AND RANK AGGREGATION

Approved by:

Asst. Prof. Erinç Albey, Advisor
Department of Artificial Intelligence and Data Engineering
Özyeğin University

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

Asst. Prof. Enis Kayış
Department of Industrial Engineering
Özyeğin University

Asst. Prof. Görkem Yılmaz
Department of Industrial Engineering
Izmir University of Economics

Date Approved: 22 May 2024

Asst. Prof. Mehmet Önal
Department of Industrial Engineering
Özyeğin University

To my dear wife, Sena, and to my beloved sons, Mehmet Efe and Ali Kerem, whose love has continually inspired and motivated me...

ABSTRACT

This thesis examines the synergy between Machine Learning (ML) and Operations Research (OR) to tackle complex optimization challenges in digital marketing (through proposing a genuine retention management framework) and sharing economy (via introducing a rank aggregation based matching approach for ride pooling problem). It aims to develop innovative models that combine ML's predictive accuracy with OR's decision-making strategies to enhance customer retention and optimize ride-sharing operations. The research follows a structured progression from theoretical concepts to practical applications, leading to creation of sophisticated algorithms that demonstrate significant improvements in operational efficiency.

The introduction outlines the synergistic potential between ML and OR, emphasizing the need for a cross-disciplinary approach to solve real-world problems. It focuses on the application areas of customer retention and ride pooling. The reasoning behind these application areas is that these fields posses digital data, which makes possible to deploy advanced analytical techniques. Developed applications reaffirms the successful integration of ML and OR, as evidenced by enhanced decision-making processes and operational efficiencies in targeted applications. The results show notable advancements in predictive models for customer retention and a data-driven ride-matching algorithm, both of which significantly outperform the existing methods.

Future research is directed towards developing more advanced algorithms that can handle larger datasets and deliver real-time analytics. The thesis also encourages further interdisciplinary studies to explore complex systemic challenges, promising to extend the scope and efficacy of the proposed models.

ÖZETÇE

Bu tez kapsamında, dijital pazarlama ve paylaşım ekonomisi alanlarındaki karmaşık optimizasyon problemlerinin zorlukları ele alınmış ve çözüm için Makine Öğrenmesi (MÖ) ve Yöneylem Araştırması (YA) teknikleri arasındaki sinerji incelenmiştir. Dijital pazarlama konusunda gerçek bir müşteri sadakat yönetimi çerçevesi önerilmiş, paylaşım ekonomisi alanında ise yolculuk paylaşımı problemi için sıra toplulaştırma (agregasyon) temelli bir eşleştirme yaklaşımı sunulmuştur. Çalışma, müşteri sadakat oranını artırmak ve yolculuk paylaşım operasyonlarını optimize etmek için MÖ'nün tahmin doğruluğunu YA'nın karar verme stratejileri ile birleştiren yenilikçi modeller geliştirmeyi amaçlamaktadır. Araştırmada teorik kavramlar pratik uygulamalarla birlikte incelenmiş ve bu sayede operasyonel verimlilikte önemli iyileştirmeler sağlayan sofistike algoritmaların oluşturulması hedeflenmiştir.

Giriş bölümünde, MÖ ve YA arasındaki sinerji potansiyeli özetlenmekte ve gerçek dünya sorunlarını çözmek için disiplinler arası bir yaklaşıma duyulan ihtiyaç vurgulanmaktadır. Uygulama tarafında, dijitalleşmiş verinin kolay erişilebilir olmasından dolayı müşteri sadakati ve yolculuk paylaşımı problemlerine odaklanılmıştır. Bu sayede problemin çözümü için gelişmiş analitik tekniklerin kullanılması mümkün olmuştur. MÖ ve YA tekniklerinin entegrasyonu ile geliştirilen uygulamalar, hedeflenen problemlerdeki karar verme süreçlerini iyileştirmiş ve operasyonel verimliliği artırmıştır. Çalışmada, müşteri sadakatini kontrol etmek için kullanılan kestirimci modeller ve geliştirilen veriye dayalı yolculuk paylaşımı eşleştirme algoritması mevcut modellere göre kayda değer seviyede daha iyi performans göstermiştir. Bu sonuç MÖ ve YA teknikleri arasındaki sinerjinin fayda sağlayabildiğini teyit etmektedir.

İleride yapılacak araştırmalar, daha büyük veri kümelerini işleyebilen ve gerçek

zamanlı analizler sunabilen daha gelişmiş algoritmalar geliştirmeye yönelik olabilir. Tez ayrıca, karmaşık sistemsel zorlukları keşfetmek için daha fazla disiplinler arası çalışmayı teşvik etmekte ve önerilen modellerin kapsamını ve etkinliğini genişletmeyi vaat etmektedir.

ACKNOWLEDGEMENTS

I would like to express my deepest gratitude to my advisor, Asst. Prof. Erinç Albey, for his unwavering support, guidance, and encouragement throughout this research. His expertise and insightful feedback have been invaluable to the completion of this thesis. More than an academic advisor, he has been a mentor and guide in all aspects of my life, providing wisdom and inspiration that have profoundly influenced my personal and professional growth.

I am also deeply thankful to Prof. Mehmet Güray Güler and Asst. Prof. Enis Kayış for their contributions during the progress phase of my research. Their advice and suggestions have significantly enriched my work. Special thanks to Asst. Prof. Mehmet Önal and Asst. Prof. Görkem Yılmaz for their valuable input and support.

I extend my thanks to my peer, İsmail Sevim, for his collaboration and the stimulating discussions that have greatly enhanced my understanding and perspective.

I would also like to thank my best friends, Gökalp Erbeyoğlu and Cem Demirel, for their unwavering friendship and support throughout this journey. Their companionship has been a source of strength and motivation.

Lastly, my appreciation goes to my wife, Sena, and my sons, Mehmet Efe and Ali Kerem, for their love, patience, and constant support. Their understanding and sacrifices have been fundamental to the successful completion of this work. I am also deeply grateful to my entire family for their continuous encouragement and support. Thank you all for your invaluable contributions and for being part of this journey.

TABLE OF CONTENTS

DEDICATION . . . . . **iii**

ABSTRACT . . . . . **iv**

ÖZETÇE . . . . . **v**

ACKNOWLEDGEMENTS . . . . . **vii**

LIST OF TABLES . . . . . **x**

LIST OF FIGURES . . . . . **xi**

I INTRODUCTION **1**

II THEORETICAL BACKGROUND **4**

III A MACHINE LEARNING-DRIVEN RETENTION MANAGEMENT FRAMEWORK **9**

3.1 Background and Literature Review 10

3.2 Retention Management Framework 19

3.3 Take Actions: Retention Management Model 21

3.4 Retention as a Multidimensional Function 27

3.4.1 Multidimensional curve fit 33

3.4.2 Representation of a curve fit with a tree 34

3.5 Results and Discussion 39

IV A MACHINE LEARNING-DRIVEN MATCHING ALGORITHM FOR RIDE POOLING PROBLEM **45**

4.1 Background and Literature Review 45

4.2 Current Ride Matching Procedure: The Benchmark 52

4.3 Ranking the Drivers: BP Based Weighted Rank Aggregation 54

4.3.1 Mathematical Model 55

4.3.2 Calculating Superiorities 56

4.4 Machine Learning Embedded Ride Matching: LBRanker 58

4.4.1 The Main Building Blocks 60

4.4.2 Learning the Weights: Nonlinear Bilevel Programming Model 62
4.5 Numerical Experiments and Results 67
V CONCLUSIONS 76
REFERENCES 79
VITA 86

LIST OF TABLES

1 An overview of selected literature of customer retention. 13

2 An overview of selected literature of customer churn. 14

3 An overview of selected literature of CLV. 17

4 Snapshot from customer transaction data. 28

5 Snapshot from customer profile data. 28

6 Solution statistics. 42

7 Set definitions for current ride matching procedure. 54

8 Parameters and variables of RATBLP. 64

9 Details of training and test sets. 71

10 Comparison of errors for the test set with respect to number of drivers in the time-wise split scenario. 72

11 Comparison of errors on the time-independent scenario. 73

12 Aggregated performance results. 75

LIST OF FIGURES

1 The life-cycle of customers as a stochastic process. 11

2 Managing customer retention framework [1]. 15

3 Proposed retention management framework. 20

4 Schematic Representation of the Integrated Analytical Approach Combining Curve Fitting, Predictive Modeling, and Optimization Techniques. 25

5 Retention curve for different parameters. 27

6 Monthly retention rates for various cohorts. 29

7 Comprehensive analysis of bonus influence on customer retention. . . 30

8 Visualization of survival rates by each cohort and bonus group over a six-month period (Bonus 30). 31

9 Aggregated survival rate over time by bonus category. 32

10 3D surface fill of survival rate by tenure and bonus. 33

11 Beta-Geometric fitting for bonus group 30 ($R^2 = 91.95\%$). 34

12 Illustration of the fitted curve using BGFit: Survival rate by tenure and bonus. 35

13 Illustration of the decision tree. 36

14 Illustration of an example path from root to leaf in the tree. 36

15 Comparison of RPFit-Raw and RPFit-BG models. 43

16 Comparison of average survival rates of all customer over tenure for the all scenarios. 44

17 Flowchart of TAG's ride matching procedure. 54

18 The flowchart of LBRanker. 59

19 Comparison of prediction accuracy breakdown for the test set with respect to number of drivers in the time-wise split scenario. 73

20 Comparison of prediction accuracy breakdown with respect to number of drivers in the time-independent scenario. 74

CHAPTER I

INTRODUCTION

In the dynamic worlds of Machine Learning (ML) and Operations Research (OR), there exist a deep connection and synergy, when these techniques are collaboratively applied to real world problems. This thesis aims to explore and harness this synergy by focusing on concrete problems such as customer retention management and ride pooling. Our goal is to provide innovative and effective solutions that leverage the combined strengths of ML and OR, addressing both the theoretical frameworks and real-world applications.

Machine learning, with its robust data analysis and predictive capabilities, excels at identifying patterns and deriving insights from expansive datasets—capabilities often beyond the reach of traditional analytical methods. Operations Research, especially the literature in optimization and mathematical modeling, specializes in the optimization of decisions and resource management under constraints. This thesis proposes a novel approach by integrating ML's predictive power with OR's optimization strategies to address complex, real-world challenges more effectively and efficiently. The choice of retention management and ride pooling as focal areas is driven by their significant implications on resource allocation (hence their economical impact covering form financial to environmental aspects) and the pressing need for sophisticated analytical approaches.

The necessity for integrating ML with OR is escalating as real-world challenges grow in complexity. ML can refine OR models by providing more dynamic and accurate data inputs, while OR can enhance ML outputs by incorporating them into structured decision frameworks. This interdisciplinary approach promises not only

more scalable and adaptable solutions but also the potential to discover insights that would be unattainable through isolated disciplinary methods, paving the way for transformative bussiness strategies.

Following a foundational overview of machine learning and operations research in the introductory chapter, the thesis progresses to a detailed theoretical exploration in Chapter 2, where it reviews past integration efforts and identifies promising avenues for future research.

In Chapter 3, we delve into retention management, illustrating how ML can enhance OR to develop predictive and prescriptive strategies. This chapter covers theoretical aspects and presents an in-depth analysis of customer lifetime value, retention, and survival analysis. It discusses the simplification of tree-based ML models for integration with OR techniques and introduces an innovative retention management optimization model. The model is designed to convert theoretical insights into actionable, data-driven strategies, aiming to significantly influence business practices by demonstrating the synergy between ML and OR.

Chapter 4 introduces a cutting-edge, data-driven algorithm for solving the ride pooling challenge, conceptualized as a complex matching problem. The algorithm, which employs rank aggregation and optimization of feature weights using historical data, is formulated as a single-level mixed-integer nonlinear optimization problem. Using data from a ride pooling startup, the algorithm demonstrates considerable improvements over existing methods, enhancing the accuracy of weight predictions of the matching criteria and the efficacy of driver recommendations.

The thesis concludes with Chapter 5, which synthesizes the research findings in relation to the initial objectives and hypotheses. This chapter provides valuable insights for both practitioners and researchers, acknowledges the study's limitations, and outlines directions for future research in the field.

Ultimately, this thesis is an effort to combine theoretical insights with practical applications and to foster new research directions at the intersection of machine learning and operations research. By pioneering innovative approaches to retention management and ride-sharing issues, it seeks to inspire subsequent research efforts, thereby catalyzing advancements in these vital sectors.

CHAPTER II

THEORETICAL BACKGROUND

Machine learning-driven optimization combines the predictive power of machine learning (ML) with the strategic problem-solving abilities of operations research (OR). This powerful blend is ideal for improving decision-making processes under uncertain and changing conditions. In practice, ML techniques such as regression, classification, and clustering analyze large datasets to predict trends and identify patterns. These insights are then integrated into OR models like linear programming and stochastic optimization, helping to formulate the best possible strategies.

The concept of merging ML with OR has evolved significantly with the rise of big data and improved computational technologies. Historically, OR concentrated on creating detailed models based on precise, often oversimplified data. ML, however, excels in processing vast amounts of complex data and learning from it to predict outcomes or group data into useful categories. By combining the strengths of both fields, this integrated approach overcomes individual limitations, enhancing the quality of inputs for OR models and resulting in more practical and effective decisions in real-world applications.

Significant innovations have emerged from integrating ML and OR, particularly in decision-making under uncertainty. References like [2] highlight the increasing use of data analytics in operations management, emphasizing the combined role of predictive analytics (ML) and prescriptive analytics (OR) in industries like supply chain management and healthcare. The "Smart Predict, then Optimize" (SPO) framework introduced by [3] focuses on improving predictive models used in optimization tasks by incorporating the structure of the optimization problems themselves, aiming to

reduce decision errors.

Studies like the one by [4] address integrating operational statistics with traditional inventory management methods, tackling the complexities of managing stock under uncertain demand. They propose a method where estimation and optimization are performed simultaneously, which not only improves decision-making but also significantly reduces losses in expected profit.

[5] develop a framework that uses machine learning to transition from predictive to prescriptive analytics in decision-making, demonstrating the effectiveness of combining diverse data sources in inventory management to make better operational decisions and improve performance.

The survey by [6] discusses data-driven methods that enhance the predictability and reliability of decision-making under uncertainty by incorporating contextual information. These methods provide a unified framework for understanding contextual stochastic optimization and illustrate how ML and OR can work together to improve both predictive and prescriptive capabilities in decision-making frameworks.

[6] categorize the approaches that illustrate the synergistic potential of OR and ML.

1. **Decision Rule Optimization:** Decision Rule Optimization focuses on deriving decision rules from data-driven insights. Traditional methods within this framework utilize parameterized decision rules optimized based on historical data [4, 7]. This approach is common in applications such as inventory management and dynamic pricing, where decision rules can directly map observable features to decisions. The strength of DRO lies in its simplicity and direct application, making it highly effective in environments where decisions need to be made rapidly and robustly against variability in data. Researchers have extended these methods to include complex models like deep neural networks to handle more intricate decision surfaces [8].

2. **Sequential Learning and Optimization (SLO):** Sequential Learning and Optimization involves a two-stage process where the first stage utilizes ML models to predict the conditional distribution of uncertain parameters based on covariates, and the second stage involves solving an optimization problem using these predictions [5]. This approach is particularly suited to fields such as supply chain management and healthcare, where decisions such as stock levels and treatment plans significantly depend on accurate predictions of uncertain outcomes. SLO has been robustified to reduce post-decision disappointment through regularization and other techniques, which help mitigate risks associated with model overfitting or misspecification.

3. **Integrated Learning and Optimization (ILO):** Integrated Learning and Optimization represents a more recent paradigm that seeks to fuse prediction and optimization into a cohesive framework. ILO aims to optimize the decision-making process directly, often using end-to-end training methodologies where the model is trained based on the quality of the decisions rather than the accuracy of the predictions alone [9]. This approach is beneficial in complex decision-making environments like logistics and energy management, where the direct impact of decisions on operational performance is critical. ILO methodologies have demonstrated their capability to yield superior decisions by focusing on the final decision quality, even at the expense of less accurate intermediate predictions.

These methodologies underscore a growing trend where OR and ML are not just parallel tools but are integrated to enhance both the predictive and prescriptive capabilities of decision making frameworks. This integration promises to refine how decisions are made in the face of uncertainty, leveraging the strengths of both fields to achieve more nuanced and effective solutions.

By focusing on machine learning-driven optimization, this integrated approach promises to refine decision-making processes, leveraging the strengths of both ML and OR to achieve more nuanced and effective solutions in fields such as retention management and ride pooling. Both embedding ML models into OR frameworks and using OR techniques during the ML training phases exemplify the synergistic potential of this integration, which is essential for developing sophisticated, dynamic systems that effectively address the complexities of modern operational environments. This integration is applied through two main strategies: 1) embedding trained ML models into OR frameworks, and 2) applying OR models to enhance the training and testing of ML models.

The first strategy embeds a trained ML model within an OR framework to boost its decision-making abilities. This approach uses the predictive strength of ML to inform and improve the traditional optimization algorithms used in OR. For example, ML models can predict uncertain outcomes or variables, which are then fed into OR models to refine decisions under uncertain conditions. This method is highly effective in scenarios where decisions must be made quickly and consider complex data or conditions.

The second strategy involves using OR techniques to enhance the training and testing phases of ML models. Here, the optimization methods of OR are used to fine-tune how ML models are trained, aiming to achieve specific goals like reducing prediction errors or increasing reliability under changing conditions. This approach not only improves the precision of ML models but also customizes them to perform best within particular limits and needs.

Both strategies showcase the complementary strengths of ML and OR. ML boosts the adaptability and predictive power of OR models, while OR adds robust, goal-focused methods to the training of ML systems. This collaboration is crucial for creating more advanced, dynamic systems that can adeptly handle the complexities of

contemporary operational challenges. By leveraging these approaches, researchers and practitioners can expand the limits of what's possible at the intersection of machine learning and operations research.

CHAPTER III

A MACHINE LEARNING-DRIVEN RETENTION MANAGEMENT FRAMEWORK

In this chapter, our primary objective is to delve into the effectiveness and practical applicability of the integrated learning and optimization framework within the context of retention management. This framework merges predictive machine learning models, specifically those designed to estimate survival functions, with strategic optimization models that are crucial for determining retention strategies. Our aim is to construct a comprehensive approach that significantly enhances the efforts of businesses to retain their customers.

To achieve this, we first introduce the theoretical underpinnings of survival analysis in machine learning, explaining how these models can accurately predict the duration that customers will remain with a business. We discuss how these predictions can be integrated into optimization models. These models use the predictions to make data-driven decisions about which retention strategies might be most effective for different customer segments. We also explore the dual benefits and limitations inherent in this integrated framework. On one hand, it offers the potential for highly tailored retention strategies that are responsive to the predicted behaviors and preferences of individual customers, potentially leading to enhanced customer loyalty and reduced churn. On the other hand, the complexity of integrating two sophisticated models poses challenges, including the need for data and the computational demands of running complex simulations.

We evaluate the performance of this framework through empirical analysis, using

real-world data from businesses that have implemented similar strategies. This includes a detailed examination of the models' accuracy in predicting customer behavior and their effectiveness in improving retention rates.

Finally, we provide practical insights for businesses considering the adoption of this framework. Our discussion concludes with recommendations on how businesses can tailor the integrated learning and optimization framework to their unique contexts to maximize customer equity and drive business success.

3.1 Background and Literature Review

Customer relationship management (CRM) is an approach for strategically managing interaction between a company and the company's current and potential customers. Basically, it can be defined as the selection of the valuable customers strategically and maximize the present and future value of customers for the company. Today, CRM refers to an indispensable strategy, set of tactics and technology for the modern economy [10].

Today's competitive market has forced companies to manage customers' profitably in the long run and companies have started to design their processes strategically according to carry out customer-oriented actions with the understanding of the importance of CRM. The general purpose is to increase profits from all of the transactions instead of trying to make profit from individual transactions.

[11] suggest that the CRM process has three primary components, which are:

- **Relationship initiation**: Customer evaluation (initiation), acquisition and recovery.
- **Relationship maintenance**: Customer evaluation (maintenance), retention, up-selling/cross-selling, referrals management.
- **Relationship termination**: Customer evaluation (termination), customer

exit.

In CRM, the process of analyzing customer data and extracting meaningful insights from the data is called customer analytics. Customer analytics aims to analyze data sets generated from different activities, in other words, behavioral features, (such as online and/or brick and mortar transactions of customers, contact logs such as call center and/or clicks from the web site), demographic features (such as age, gender, location etc.) and to build data driven decision making (or decision support) systems for that answer various business questions such as segmentation/targeting, churn management, offer management, loyalty management, CLV calculation etc.

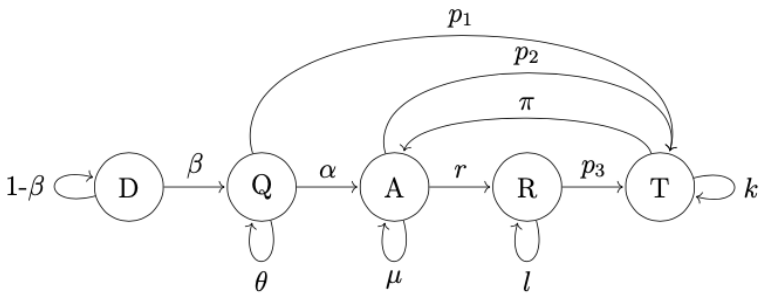


Figure 1: The life-cycle of customers as a stochastic process.

For a better understanding of building intelligent decision making/support systems, the life-cycle of customers can be examined as a Stochastic Process as shown in Figure 1. The life-cycle of a customer can be assumed to consist of acquisition (state Q), activation (state A),retention/development (state R), and termination (state T) phases. The customer can switch between states (or remain in the same state) with a set of transition probabilities. Each node can be examined in itself as a separate stochastic process. For example, in the retention phase, the processes are the examination of the customers churn behavior with churn rate P_{RT} and being loyal activities with loyalty rate P_{RR}.

The ultimate motivation behind building intelligent decision making/support systems can be summarized as follows:

- acquiring correct customers through most suitable channels with minimum cost,
- retaining the current customers as long as possible while maximizing the profit generated from those customers via correct offer/campaign management.

In this thesis, we focus on retention phase of the process. For this phase, it comes first to identify valuable customers. Once valued customers are identified, their relationship with the firm needs to be strengthened. Also, the customers who are likely to churn should be prevented and keep in touch with the firm. It is necessary that making strategic offers for strengthening relationships and preventing churn.

Customer retention is characterized as the ongoing engagement of a customer with a company, as delineated by [1]. The forms of customer-company interactions are diverse, encompassing both monetary and non-monetary exchanges, such as the frequency of utilization of complimentary services.

According to [12], the dynamics of customer-firm relationships can be categorized into two primary settings: contractual and non-contractual. A pivotal distinction within these settings pertains to the cessation of the relationship. In contractual environments, customers are required to explicitly indicate their desire to discontinue the service, whereas in non-contractual environments, such formal declarations are absent.

The simplest method for calculating the retention rate involves dividing the number of active customers who maintain their subscriptions by the total count of active customers from the preceding period.

$$r(t) = \frac{c_a(t)}{c_a(t-1)}$$

where r(t) is retention rate and $c_a(t)$ is the number of active customers at period t.

Enterprises dedicate substantial resources to customer retention initiatives. The scholarly discourse surrounding this subject addresses various questions, including the likelihood of repeat purchases by newly acquired customers, the subsequent purchasing volume, the longevity of the customer's relationship with the enterprise or product, and the customer's preferred product categories. Table 1 presents a selection of studies that have explored these aspects of customer retention.

Table 1: An overview of selected literature of customer retention.

Studies	Model	Notes
[13]	Discrete-hazard, log-normal	Estimate CLV by modeling purchase timing, purchase amount, and customer churn.
[14]	Proportional hazard	Customer retention model considers duration, promotion cohort, and seasonality.
[15]	Shifted beta geometric	Estimate customer tenure by using an alternative approach to survivor analysis.
[16]	Logistic Regression	Analysis of the effect on customer loyalty and cross-purchasing of companies.
[17]	Poisson, linear regression	Effects of price promotion on new and existing customers purchasing.

Retention serves as a critical metric for firms, providing insights into organizational health, profitability, and value, thus underscoring the importance of its measurement. [1] discuss various metrics to gauge customer retention, drawing a distinction between contractual and non-contractual business frameworks. For individual customer level metrics in contractual businesses, they focus on binary indicators that show whether a customer is still under contract or has made transactions in a period, along with analyzing the recency and the ratio of recency to inter-purchase time. These metrics are pivotal for assessing customer loyalty and engagement. In the realm of non-contractual businesses, attention is given to the latent attrition rate or the probability of a customer being active, deduced from statistical models, along with transactional activities and recency measures similar to those in contractual settings. On an aggregate level, contractual businesses examine retention rates and engagement statistics through the distribution and summary statistics of recency and the recency/inter-purchase-time ratio. Non-contractual businesses, meanwhile, look at the distribution

of latent attrition and customer transaction patterns to get a sense of overall customer base health and engagement trends. These metrics, varied and complex, are crucial for developing a nuanced understanding of customer retention and provide a significant foundation for the integration of machine learning in predicting customer behavior.

The primary goal of retention efforts is to mitigate customer churn, defined as the customer's decision to cease transactions with the firm. Indicators of potential churn include reduced transaction frequency or extended intervals between purchases. Identifying customers at risk of churn is crucial, as acquiring new customers is more costly than retaining existing ones. While retention and churn are interconnected concepts, the former is strategically implemented to prevent the latter. The literature encompasses a diverse range of churn management strategies, with selected studies highlighted in Table 2.

Table 2: An overview of selected literature of customer churn.

Studies	Model	Notes
[18]	Clustering, decision trees	Hybridization of unsupervised learning techniques with supervised learning techniques for utilizing the results of clustering using decision trees.
[19]	Logistic regression, neural network, decision tree	A framework that considers a profit-sensitive loss function to select best classification techniques for the expected profit.
[20]	Weibull hazard	Study the empirical link between customer churn and factors such as customer service experience, failure recovery, and payment equity.
[21]	Decision tree	Identify churners in future using the results of decision tree in banking sector.
[22]	Support vector machine, naive Bayes, tree rules	Present a hybrid approach to extract rules from SVM for customer relationship management purposes. The approach is composed of three phases where: 1) SVM-recursive feature elimination is applied to reduce the feature set; 2) the obtained dataset is used to build the SVM model; and 3) using NB, tree rules are generated.
[23]	logistic regression, decision trees	Compare decision trees, logistic regression, and ensemble algorithms to a new hybrid classification algorithm that is based on logistic regression and decision trees.

[1] propose a structured framework for managing customer retention, depicted in

Figure 2. The process begins with identifying customers who are at risk of not being retained. Subsequently, the reasons for each customer's risk are diagnosed. Decisions are then made regarding which customers to target in the campaign, when they should be targeted, and what incentives or actions should be used. Finally, the campaign is implemented and its outcomes are evaluated.

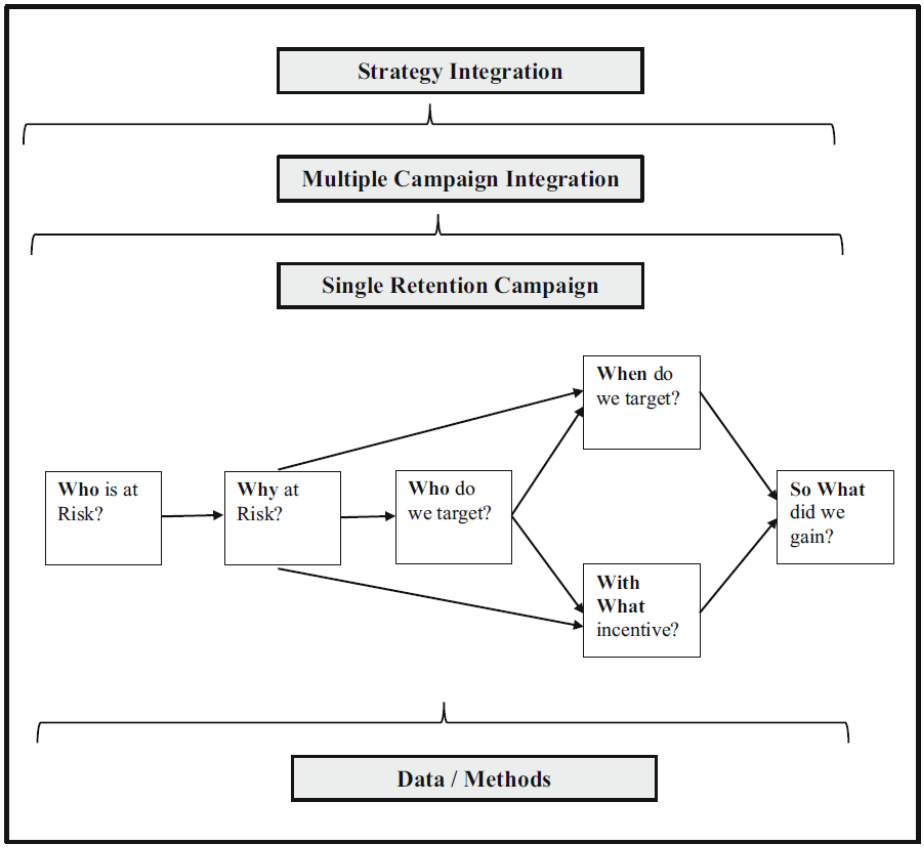


Figure 2: Managing customer retention framework [1].

The process of managing customer relationships commences with the identification of customers, utilizing traditional metrics such as Recency–Frequency–Monetary Value (RFM), share of wallet, and tenure/duration. However, these metrics predominantly assess past activities of a customer. In contrast, a forward-looking metric that incorporates past activities and predicts future behaviors can be more advantageous for accurately identifying valuable customers. Customer Lifetime Value (CLV) is such

a metric, defined as "the present value of the future cash flows attributed to the customer relationship" [24]. CLV is inherently forward-looking; however, it necessitates assumptions about uncertain future variables such as the customer's longevity, activity status, and purchasing timings. Therefore, [25] suggest calculating the expected Customer Lifetime Value, $E(CLV)$, considering these variables as random factors within the specified equation.

$$E(CLV) = \int_0^\infty E[V(t)]S(t)d(t)dt$$

where, for $t > 0$ (with $t = 0$ representing the "birth" of the customer), $E[V(t)]$ is the expected net cash flow of the customer at time t (assuming the customer is alive at that time), $S(t)$ is the probability that the customer has remained alive to at least time t, and $d(t)$ is a discount factor that reflects the present value of money received at time t.

The marketing literature presents a diverse array of methods for calculating Customer Lifetime Value (CLV), all of which aim to identify, maintain, and nurture valuable customers. These methods are broadly categorized into two primary groups: aggregate methods and individual methods. Aggregate methods derive an average lifetime value for customers within a specific cohort or segment, providing a metric that assists companies in evaluating the effectiveness of their marketing strategies. Conversely, individual methods calculate the CLV for each customer throughout their association with the company, furnishing insights that support strategic planning tailored to individual customer needs. Table 3 provides an overview of selected literature on CLV methodologies.

In contractual settings, where the churn of a customer is observable, it is feasible to employ a survival model to the dataset. This approach facilitates the estimation of the survivor function, denoted as $S(t)$. Additionally, retention rates can be directly computed from the dataset or derived from the survivor function by applying the

Table 3: An overview of selected literature of CLV.

Studies	Model	Notes
[26]	Shifted beta geometric	Build a model of relationship duration for predicting retention rates beyond the observed retention rates.
[27]	Logistic Regression	Relation of market capitilization and customer equity that determined by using CLV.
[28]	Seemingly unrelated regression	Show IBM's CLV deployment and impact on profitability and resource allocation.
[29]	Pareto/NBD	Demonstrate using Pareto/NBD models for repeat buying while computing CLV.
[30]	Autoregressive (first-order)	Predict the total CLV of a Netflix customer with using company report data.
[31]	Monte Carlo simulation	Predict customer purchase propensity, profit, and firm marketing actions with using Monte Carlo simulation algorithm..
[32]	Generalized gamma distribution, genetic algorithms	A dynamic framework to maintain customer relationships since allocating of marketing resources and maximizing CLV.
[33]	BG/NBD model	Identify customers' expected repeat visits to determine the customer lifetime value (CLV).

formula:

$$r(t) = \frac{S(t)}{S(t-1)}.$$

proposed that using the beta-geometric (BG) distribution to project customer retention and survival into the future:

To model customer retention using the Beta-Geometric distribution, [15] start by defining the probability that a customer will not renew their contract at the end of period t and the probability of surviving beyond period t. Given the unobserved nature of the customer's value of θ, we use the expectation over the Beta distribution to account for the cross-sectional heterogeneity in θ. The resulting expressions are given by:

$$P(T = t \mid \alpha, \beta) = \frac{B(\alpha + 1, \beta + t - 1)}{B(\alpha, \beta)}, \quad t = 1, 2, \ldots \tag{1}$$

$$S(t \mid \alpha, \beta) = \frac{B(\alpha, \beta + t)}{B(\alpha, \beta)}, \quad t = 1, 2, \ldots \tag{2}$$

where $B(\cdot, \cdot)$ is the Beta function. Equation (1) represents the probability of not renewing the contract at period t, while Equation (2) gives the probability of surviving beyond period t. These expressions provide a robust framework for understanding retention patterns by integrating the Beta distribution to capture individual heterogeneity. This model is referred to as the shifted-beta-geometric (sBG) distribution, and it finds applications beyond business contexts, such as modeling the number of menstrual cycles. For detailed derivations, one can refer to the relevant appendices in the associated literature [15]. Substituting and simplifying gives the following expression for the (aggregate) retention rate associated with the sBG model:

$$r(t|\gamma, \delta) = \frac{\gamma + t - 1}{\gamma + \delta + t - 1}$$

γ and δ are the two parameters of this distribution.

The inherent challenge in non-contractual settings is the unobserved nature of customer churn, necessitating certain assumptions for deriving the survivor function, $S(t)$. This complexity makes it difficult to distinguish between customers exhibiting low transaction frequency and those who have terminated their relationships with the firm. While these two customer types may never be completely distinguishable, statistical models can be employed to make probabilistic assessments of such scenarios [34].

[12] posit that 1) the purchasing behavior of "alive" customers can be described by the Negative Binomial Distribution (NBD) model, which is a gamma mixture of Poissons, and 2) the unobserved customer lifetimes follow the Pareto Type II distribution, a gamma mixture of exponentials. This combination underpins the Pareto/NBD model, a framework for analyzing buyer behavior in non-contractual settings.

Subsequent modifications and extensions of the basic Pareto/NBD model have been explored extensively. A significant line of research has produced variants that

enhance practical implementation, exemplified by the BG/NBD and BG/BB models introduced by [35] and [26], respectively.

3.2 Retention Management Framework

The challenges and objectives of customer retention management are diverse, reflecting the intricate nature of fostering and enhancing customer relationships. Customer retention stands as a pivotal component of business success, fundamentally reliant on strategies that promote customer loyalty and prolonged engagement. Key challenges include the under utilization of customer feedback, substandard customer service, a lack of personalization, the oversight of loyal customers, and lagging behind in technological and competitive advancements. These factors can severely impede a business's capability to maintain its customer base if not properly addressed.

Conversely, the objectives of customer retention encompass maximizing Customer Lifetime Value (CLV), cultivating brand loyalty, and minimizing churn rates. Strategies to achieve these goals involve improving service quality, leveraging data analytics for tailored customer experiences, and implementing robust loyalty programs. Overall, effective management of customer retention necessitates a holistic approach that prioritizes enhancing customer satisfaction and loyalty, thereby augmenting the overall value extracted from each customer relationship.

This study aims to propose a retention management framework for non-contractual settings, as depicted in Figure 3. It begins with the identification of customer. Following this, it predicts several key customer behavior parameters including retention rate, survivor rate, churn probability, loyalty rate, purchase amount, and reaction rate on interactions. Based on these predictions, the framework suggests taking specific actions tailored to campaign variables, model parameters, customer segments, and budget or business constraints. The final step involves evaluating the outcomes of these actions and updating the predictive models based on the results to refine

future strategies. Each step in the framework is sequential, relying on the outputs of the previous step to inform the next, ensuring a systematic approach to enhancing customer retention.

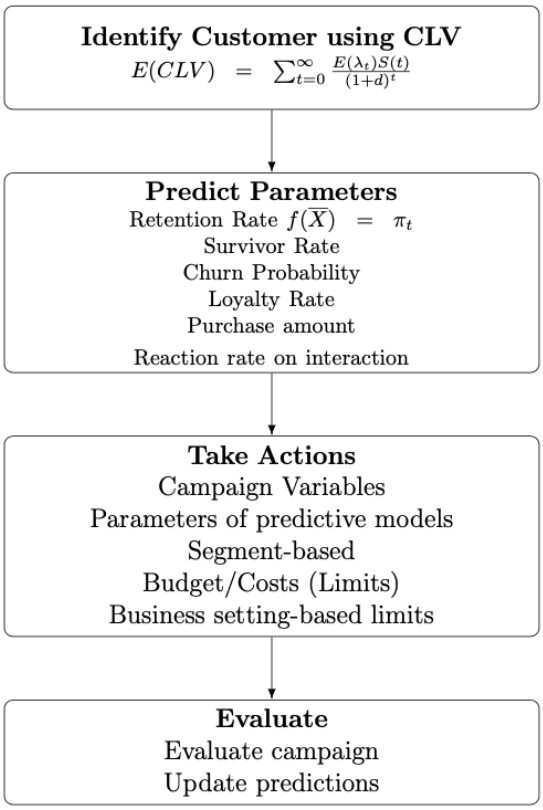


Figure 3: Proposed retention management framework.

To initiate the framework, customer identification is recognized as a pivotal step. Previous research has highlighted Customer Lifetime Value (CLV) as an essential metric for customer identification. However, the direct expression of its parameters, denoted as $r(t)$, cannot be directly derived from available data. Therefore, we propose constructing multi-dimensional curves for these parameters, represented mathematically as:

$$f(\vec{X(t)}) = r(t)$$

where $\vec{X(t)}$ symbolizes the independent variables, and $r(t)$ represents the retention rate function.

To demonstrate the efficacy of constructing parameter functions, two distinct environments are analyzed. The first scenario investigates customer homogeneity by assuming a single segment exists, thereby constructing a curve that represents all customers within this segment. The second scenario examines customer heterogeneity, presupposing the existence of multiple segments, with a separate curve developed for each cluster.

Following the construction of these functions, a mathematical model for retention management is formulated. This model aims to maximize the total expected Customer Lifetime Value ($E(CLV)$) across all customers, while accommodating constraints related to business settings and campaign budgets. The decision variables within the model are categorized into two types: (1) exogenous parameters, which include behavioral and demographic features, and (2) independent variables, which encompass aggregated campaign parameters.

3.3 Take Actions: Retention Management Model

Following the prediction of customer identification parameters, it becomes imperative to implement targeted actions aimed at enhancing customer retention. In this phase, we introduce a mathematical model for the decision-making process within the retention management framework. This model is designed to maximize the customer equity, the total expected customer lifetime value of all customers, while adhering to various business and campaign constraints.

The decision variables in this model are divided into two categories: exogenous parameters and independent variables. Exogenous parameters include customer features, such as the retention rate, whereas independent variables involve aspects of campaign management, such as the campaign budget. The integration of these variables facilitates the development of a multi-period retention management model,

which effectively manages interactions between customers and the firm, enabling informed decision-making throughout the customer lifecycle.

Contrary to traditional practices in this field, our approach utilizes optimization tools and methods specifically tailored to the needs of particular sectors or companies. This strategic application allows for enhanced control over campaigns, improved customer retention, and effective churn prevention. However, the complexity of the model, characterized by its nondeterministic, non-convex, and potentially discrete (combinatorial) nature, presents significant challenges. Therefore, developing efficient methodologies is crucial for effectively solving this integrated problem.

We have conceptualized the retention management challenge as an optimization problem. The mathematical model for retention management aims to maximize customer equity and increase retention rates by delivering the most suitable offers. The development of this model involves several assumptions:

- The model is applicable primarily in non-contractual settings.
- All customers are assumed to have the same tenure, with each acquisition occurring at the start of the planning period.
- The retention rate of a customer is influenced by multiple variables.
- Any offer made to a customer remains valid until the end of the planning period.
- Each customer is eligible to receive only one offer per period.
- The profit generated from each customer per period is considered to be constant.
- All customer-directed actions are assumed to be accepted.

Further details on the notation used in the mathematical model, including sets, parameters, decision variables, and the structure of the proposed model, are elaborated below:

Sets

- i = customer index, $i \in I$
- j = action index, $j \in J$
- t = time index, $t \in T$

Parameters

- c_j = cost of action j,
- p_i = the expected profit per period from customer i,
- B_t = total budget of action at period t,
- d = interest rate.

Decision Variables

- $$x_{ijt} = \begin{cases} 1, & \text{if customer } i \text{ gets an action } j \text{ at period } t \\ 0, & \text{o/w} \end{cases},$$
- r_{it} = retention rate of customer i at period t,
- Δ_{it} = total cumulative cost of actions for customer i at period t

Mathematical Model

$$\max \quad \sum_{i} \sum_{t=1}^{\infty} \frac{p_i \prod_{t'=1}^{t} r_{it'}}{(1+d)^t} \qquad \forall i \in I \tag{3}$$

$$\text{s.t.} \quad \sum_{j} x_{ijt} \leq 1 \qquad \forall i \in I, \forall t \in T \tag{4}$$

$$\sum_{i} \sum_{j} \sum_{t'=1}^{t-6} c_j x_{ijt} \leq B_t \qquad \forall t \in T \tag{5}$$

$$\sum_{j \in J} \sum_{t'=t-6}^{t} c_j x_{ijt'} = \Delta_{it} \qquad \forall i \in I, \forall t \in T \tag{6}$$

$$r_{it} = f(t, \Delta c_{it}) \qquad \forall i \in I, \forall t \in T \tag{7}$$

$$x_{ijt} \in \{0, 1\} \qquad \forall i \in I, \forall j \in J, \forall t \in T \tag{8}$$

$$r_{it}, \Delta c_{it} \geq 0 \qquad \forall i \in I, \forall t \in T \tag{9}$$

The objective of the model aims to maximize customer equity. Constraint 4 ensures that each customer gets at most one action in each period. Constraint 5 states that total action costs cannot exceed the budget limit in each period. Constraint 6 calculates the total cost of actions for each customer in period. Constraint 7 indicates retention rate is a function of the customer features, i.e. tenure and total bonus in last six months. The rest of the constraints are non-negativity constraints for variables.

[36] developed a targeted offer management model for prepaid customers of a telecommunication company. The proposed framework selects the most suitable offer for each customer by employing a mathematical model that integrates customer lifetime value (CLV) and churn risk. This framework operates on three levels: the initial level calculates the churn probability of customers; the second level estimates the CLVs; and the third level solves an offer management optimization model using the churn probabilities and CLVs obtained from the previous steps.

In this approach, crucial parameters such as CLV are incorporated externally into the model. Unlike this method, the retention management mathematical model proposed in this thesis seeks to manage customer behaviors or reactions and the control

parameters influencing these reactions in a holistic manner. The model might embody a nonlinear structure or represent a set of constraints derived from an externally learned decision tree model.

As previously emphasized, CLV is a critical parameter for customer identification. Therefore, this study focuses on representing CLV. Calculating CLV necessitates learning the survivor or retention function; however, in non-contractual settings, this function cannot be directly derived from the data. Consequently, we plan to construct multidimensional curves to represent the retention function, expressed as:

$$r(t) = f(\vec{X(t)})$$

This formulation aims to model retention dynamics effectively, considering the challenges posed by the indirect observability of customer behaviors in non-contractual settings.

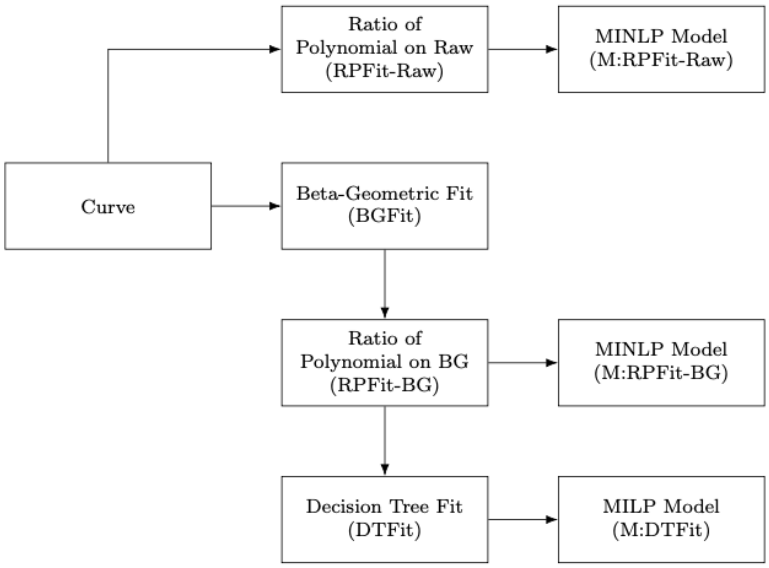


Figure 4: Schematic Representation of the Integrated Analytical Approach Combining Curve Fitting, Predictive Modeling, and Optimization Techniques.

The flow of the experimental framework is depicted in Figure 4, providing a

schematic overview of the methodological sequence adopted in this study. The following section constructs a multi-faceted analysis pipeline, blending the strengths of machine learning and operations research to investigate intricate data patterns and optimize decision-making processes within retention management framework. Initially, the research extracts a fundamental curve from the dataset, representing the core trajectory of the observed phenomena.

This curve is then subject to two distinct but complementary optimization techniques. The first is the *RPFit-Raw* and *RPFit-BG*, ratio of polynomials fit, techniques leveraging polynomial expressions to encapsulate complex interrelations inherent in the data. Concurrently, the Beta-Geometric Fit *BGFit* molds the data in congruence with the beta distribution, an approach that excels in modeling customer retention over time.

From the juncture of the BGFit, the investigational path bifurcates. One avenue advances into a mixed integer nonlinear programming model (MINLP Model), *M:RPFit-Raw* and *M:RPFit-BG*, adept at navigating the intricacies of problems where the objective function or constraints exhibit nonlinearity. This model's application stands as a testament to the adaptability and precision required to unravel and tackle real-world problems with layered complexities.

In a simultaneous and synergistic exploration, a Decision Tree Fit, *DTFit*, is employed. This machine learning algorithm unfolds a hierarchically-structured model of decisions, serving as a predictive mechanism based on a cascade of explanatory variables.

The insights gleaned from the DTFit are funneled into the formulation of a mixed integer linear programming (MILP Model). This model is an exemplar of operations research, where it incorporates both linear programming and integer variables to address optimization problems characterized by discrete decision-making parameters.

This experimental design combines statistical methods with mathematical programming to form a comprehensive strategy that helps analyze and understand complex customer behaviors. By integrating these approaches, we aim to improve retention management strategies, providing an analysis that is more insightful than if these methods are used separately.

3.4 Retention as a Multidimensional Function

[37] suggest a function for calculating retention rates over time:

$$r(t) = r_\infty(1 - e^{-\lambda t})$$

where $r(t)$ is the retention rate for period t, r_∞ is the retention rate ceiling, and λ determines how quickly retention rates converge over time to the retention ceiling. The change of retention rate over time is shown in Figure 5.

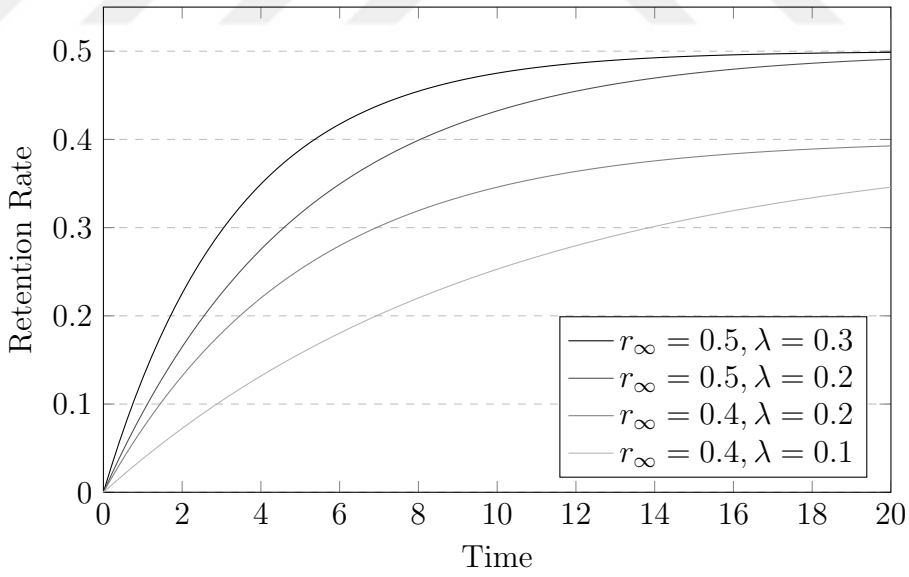


Figure 5: Retention curve for different parameters.

In the quest to develop a more nuanced understanding of customer retention, it becomes crucial to transcend the simplistic portrayal of retention merely as a proportion over time. Acknowledging the influence of various factors on retention dynamics,

there is a compelling need to establish a multidimensional function that encapsulates these additional variables. This section introduces the initial step of developing a retention rate function that integrates two specific variables. Pertinent research by [38] indicates that the application of price discounts or incentives can positively affect customer behavior and, consequently, impact retention patterns.

To empirically validate these assertions and provide real-world insights, we employ a non-contractual dataset sourced from a telecommunication company, Turkcell, previously utilized in [36]. This dataset comprises data from 6,480 customers who were onboarded during the period from March 2015 to February 2016. These customers initiated their first transaction within this specified timeframe. The dataset includes 451 profile variables, which represent a snapshot of the available customer information on the last day of data collection. It also captures transaction details through refill dates and amounts for all included customers. For illustrative purposes, a subset of this dataset is showcased in Tables 4 and 5.

SUBSCRIBER_ID	REFILL_DATE	REFILL_AMOUNT
2559188	20160124	25
2559188	20160715	25
...	...	...
13664295	20160203	89
13664295	20160209	30

Table 4: Snapshot from customer transaction data.

SUBSCRIBER_ID	TGM_TAX_COST	TREASURE_COST	...	TENURE	FRE_BONUSAMTSPENT
143222139	1.21	1.64	...	6	204.63
143357992	1.21	0.64	...	6	6.52
...	...	...	...	...	...
143761121	1.21	2.64	...	17	0.00
139548772	1.21	0.00	...	19	29.91

Table 5: Snapshot from customer profile data.

In the data preparation phase, traditional methods are employed to refine the dataset. Initially, the profile data undergoes a cleaning process which includes max-min elimination and the removal of columns exhibiting a high incidence of null values. This step effectively reduces the dataset to 341 columns. To categorize customers as active or churned, a rule based on transaction activity is applied, identifying approximately 5,251 active customers. The determination of these categories aids in the

Figure 6: Monthly retention rates for various cohorts.

subsequent feature selection process.

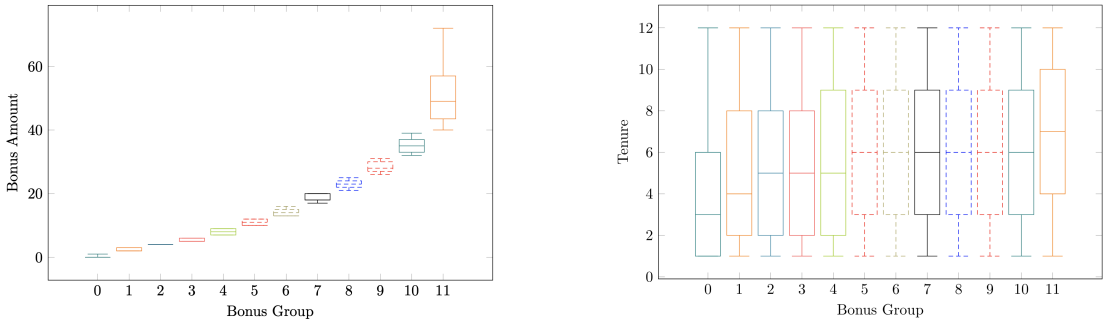
In the telecommunications sector, it is a common practice to offer additional data plans as gifts to customers. The independent variables related to these gifts, denoted as $X(\vec{t})$, are pinpointed by selecting columns containing pertinent keywords and performing a correlation analysis. Features exhibiting high correlations are discarded, narrowing down to 11 gift-related features.

A similar correlation analysis is conducted for the remaining columns, leading to the exclusion of features with high correlation. This refinement process results in 198 columns for general profile data and 11 columns specifically related to gift features.

Feature selection is carried out iteratively using a decision tree classifier. In each iteration, the three most important features are identified and removed, progressively narrowing down the dataset. This iterative procedure ultimately identifies a set of 30 significant features. The process is then repeated on this reduced set to select the top 10 most impactful features.

Ultimately, crucial features such as *average bonus in the last six months*, *ARPU in the last six months*, *average call revenue in the last six months*, and *tenure of customers* are retained. These columns are instrumental for the model, as they inform decisions regarding the allocation of bonuses to customers.

In the application of survival analysis to customer retention, we commence by grouping customers according to the month they were acquired. This grouping facilitates the construction of the survival function, which is derived based on the tenure of customers. In alignment with findings from existing literature, our analysis reveals a negative correlation between customer tenure and survival rate, suggesting that an increase in tenure correlates with an elevated likelihood of churn. Notably, we define the first instance when a customer fails to engage in a transaction as churn.



(a) Distribution of bonus amount across Bonus groups.

(b) Distribution of tenure across Bonus groups.

Figure 7: Comprehensive analysis of bonus influence on customer retention.

To further analyze the influence of bonuses on customer retention, we segment customers into deciles based on the cumulative bonus information available at the end of the dataset. This stratification allows for a detailed examination of how bonuses impact customer behaviors and retention outcomes. Figure 7(a) displays the bonus allocation for each group, providing a granular view of how bonuses are distributed among different customer segments. Meanwhile, Figure 7(b) explores the relationship between the bonus groups and customer tenure, illustrating that bonus groups tend to exhibit consistent patterns across varying tenure lengths, indicating a uniform impact of bonuses across different customer segments.

It is crucial to note that the bonus information utilized in this analysis is restricted to the last six months and is presented as cumulative data. Figure 8 delineates the survival function by tenure, assuming an average bonus level of 30. This allows for specific insights into the bonus effects during different tenure periods; for example, we have data regarding bonuses for tenure intervals of 17-22 for Cohort 23, and for tenure intervals of 1-5 for Cohort 5.

The survival rate is calculated as the average of the survival rates for each tenure, taking into account the specific cohort and bonus group. The dashed lines depicted in Figure 8 illustrate the survival rates for the last six months for each cohort group. Following this, we compute the mean survival rate for each tenure, using data from a

maximum of six cohorts. This method provides a comprehensive view of the survival trends across different customer groups, underpinned by their respective bonus levels.

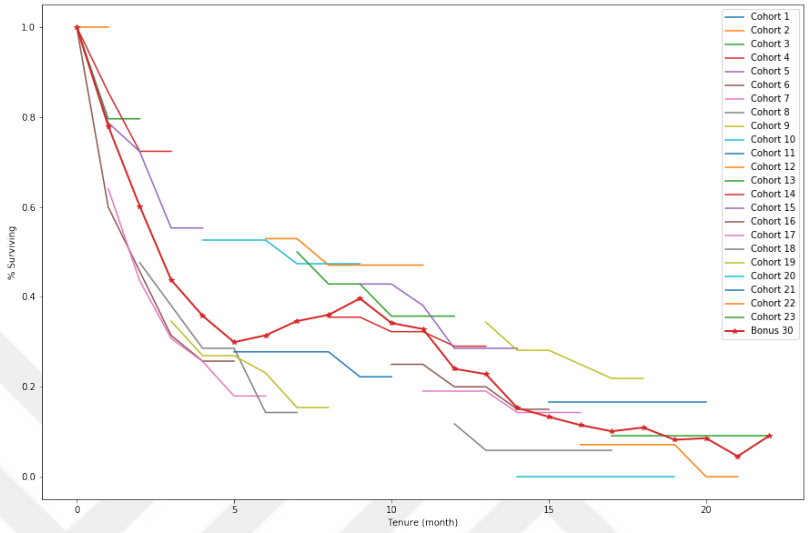


Figure 8: Visualization of survival rates by each cohort and bonus group over a six-month period (Bonus 30).

Figure 8 illustrates the average survival rate across various cohorts and tenures, represented by a red line marked with a star. This symbol facilitates the visualization of the mean survival trajectory in the dataset. Moreover, we compute the overall survival rate, taking into account information on bonuses and tenure for each bonus category. This detailed examination is presented in Figure 9, showcasing the subtle effects of bonuses on the survival rate, or customer retention, over varying tenures.

The multi-dimensional relationships of survival are visualized through 3D surface plot, providing a comprehensive view of how factors, i.e., bonus and tenure, influence survival rates as shown in Figure 10.

However, it is essential to recognize that within the limitations of the data at hand, increases in the survival function are not feasible. To confirm this, beta-geometric fitting is applied to each bonus function. The primary metric for evaluating the fit quality is the coefficient of determination, denoted as R^2.

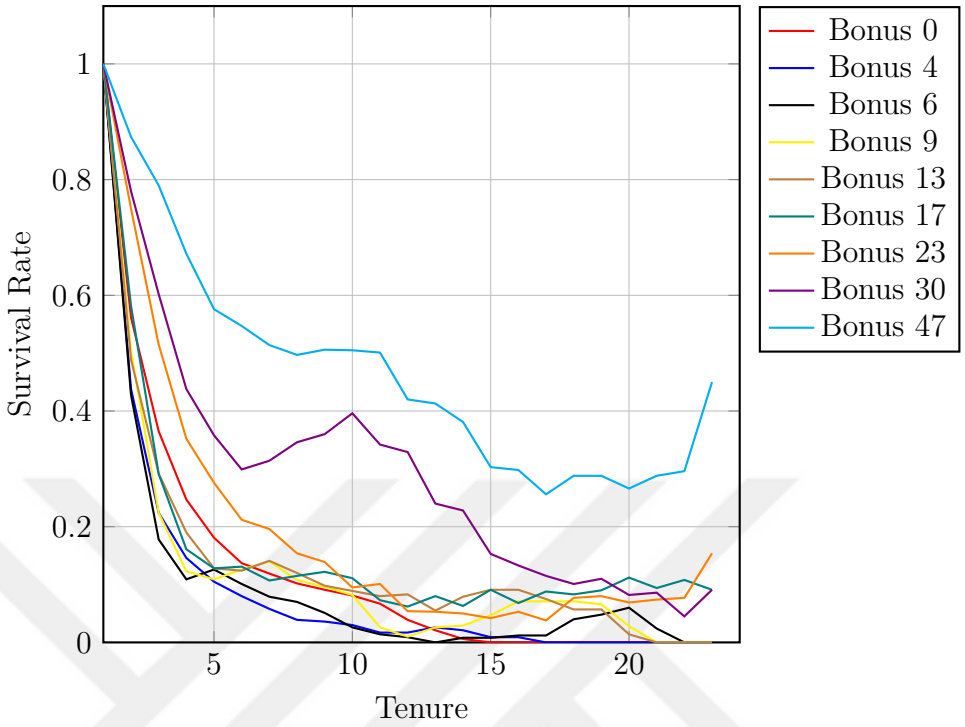


Figure 9: Aggregated survival rate over time by bonus category.

Figure 11 illustrates the survival rate over tenure for a specific group, the customers in bonus group 30, using a survival analysis model known as the Beta-Geometric model. The graph depicts two lines: the solid blue line represents the actual observed survival rates, while the dashed red line represents the survival rates predicted by the Beta-Geometric model.

The fit of the model is quantitatively evaluated by the coefficient of determination, R^2, which in this case is 91.95%. By using all customers in dataset, the effectiveness of this approach is supported by an average R^2 value of 95.54%, which reflects a high level of accuracy in the model fit. Additionally, a 3D surface plot is utilized to offer a comprehensive view of how tenure and bonuses influence the survival rate, as depicted in Figure 12

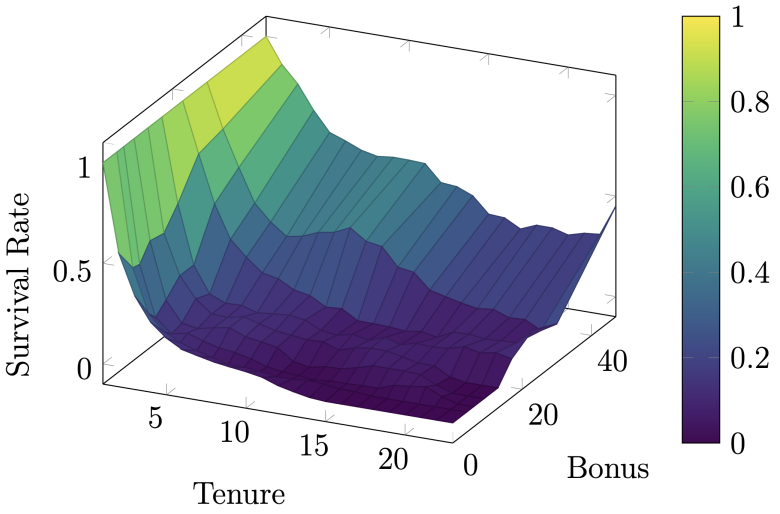


Figure 10: 3D surface fill of survival rate by tenure and bonus.

3.4.1 Multidimensional curve fit

In the initial phase of the analysis, we undertake empirical fits to derive a closed-form retention curve and to estimate its parameters. The parameters are optimized by minimizing the sum of squared errors (SSE) using the Levenberg-Marquardt algorithm [39].

The functional form applied for the fit is given by:

$$survival = \frac{a \times tenure + b \times bonus + c}{d \times tenure + e \times bonus + f}.$$

Firstly, we applied fitting on raw survival rates, denoted as *RPFit-Raw*. The figure depicts the curve that has been adapted to conform to the specified functional form. The coefficient of determination, R^2, associated with this adjustment stands at 90.50%.

Subsequently, this functional form is employed in conjunction with beta-geometric predictions, delineated as *RPFit-BG*. The rationale for this action is the difficulty in deriving a closed-form solution for beta-geometric function with bonus. Therefore, we opted for an alternative methodology, aiming to express this using a ratio of polynomials to represent beta-geometic survival rates with bonus and tenure. The

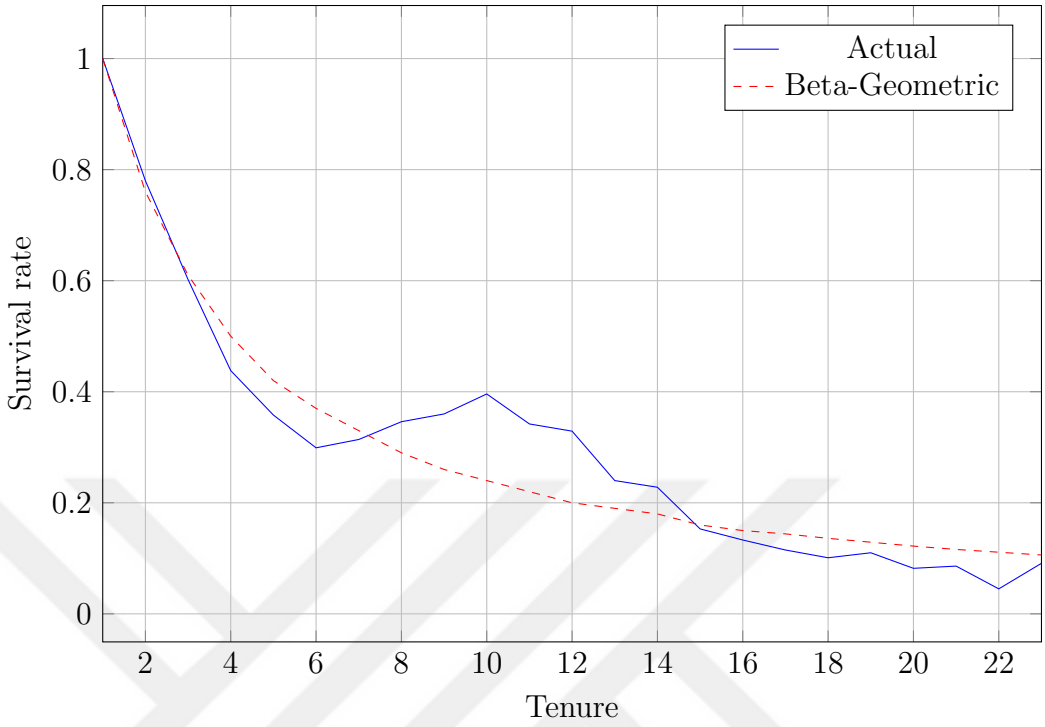


Figure 11: Beta-Geometric fitting for bonus group 30 ($R^2 = 91.95\%$).

findings from this implementation are presented in Figure 12. This model, specifically designed to accommodate heterogeneity, achieves an R^2 value of 92.47% on BGFit.

3.4.2 Representation of a curve fit with a tree

As the number of variables increases, the complexity of deriving closed-form functions from data intensifies. The interaction among multiple variables adds complexity, making it difficult to establish a simple closed-form equation that accurately reflects the data relationships. This challenge is further compounded when dealing with non-linear relationships, which are common in real-world phenomena. These relationships, where dependencies between variables are neither proportional nor additive, require advanced mathematical approaches and numerical methods for approximation.

During the second phase of our analysis, we focus on identifying an appropriate set of constraints that accurately represent the relevant surface. To achieve this, we employ decision tree regression. This method differs from traditional curve fitting as it

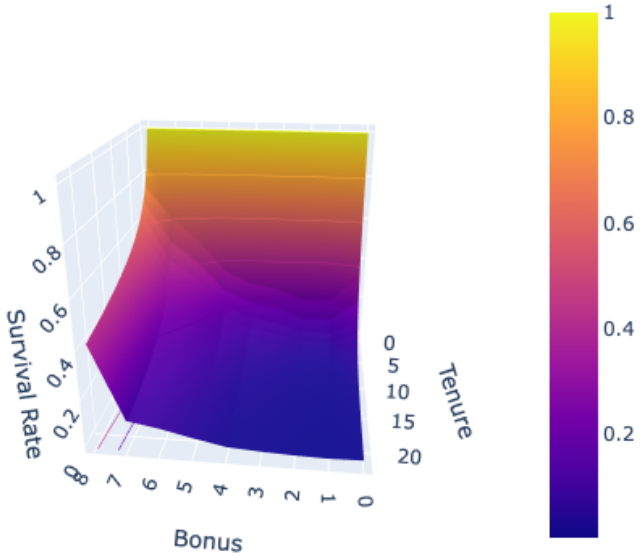


Figure 12: Illustration of the fitted curve using BGFit: Survival rate by tenure and bonus.

uses decision rules to approximate the curve by learning local linear regressions. These decision rules are then translated into a set of constraints that effectively capture the data's underlying relationships.

In decision tree regression, the model observes the features of an object and constructs a tree structure. We utilize regression trees since our dependent variable is continuous. The values at the terminal nodes of the regression tree are the means of the observations within that region. To optimize the model, we set the tree depth parameter to 6, yielding a well-fitted curve with an R-squared value of 99.8%. Here, we are not concerned with overfitting; rather, we are attempting to represent the data accurately. Additionally, we have limited the depth of the tree to prevent excessive complexity when linearizing in future analyses.

The decision tree, as visualized in Figure 13, provides a clear illustration of the tree

structure and the resulting decision rules. This allows us to gain insights into the relationship between the features and the predicted values, aiding in our understanding of the underlying patterns within the data.

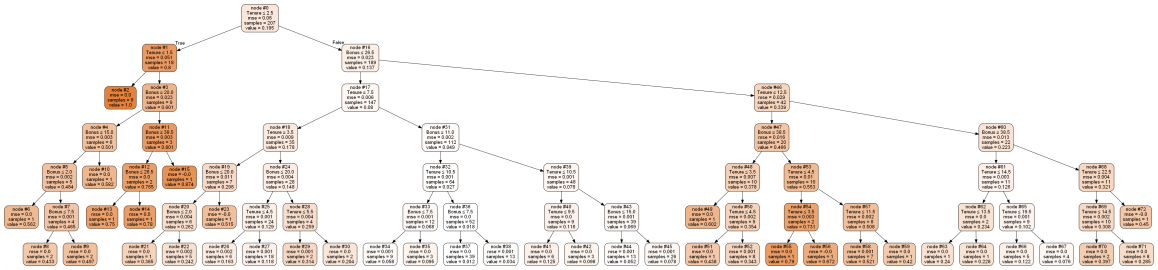


Figure 13: Illustration of the decision tree.

In the illustration, each node has two branches that go to the appropriate states: left, if the condition on the node is true and right, if it is false. For a better understanding of the rule generation process, a branch starting from the root node to a leaf node is shown in Figure 14 as an example.

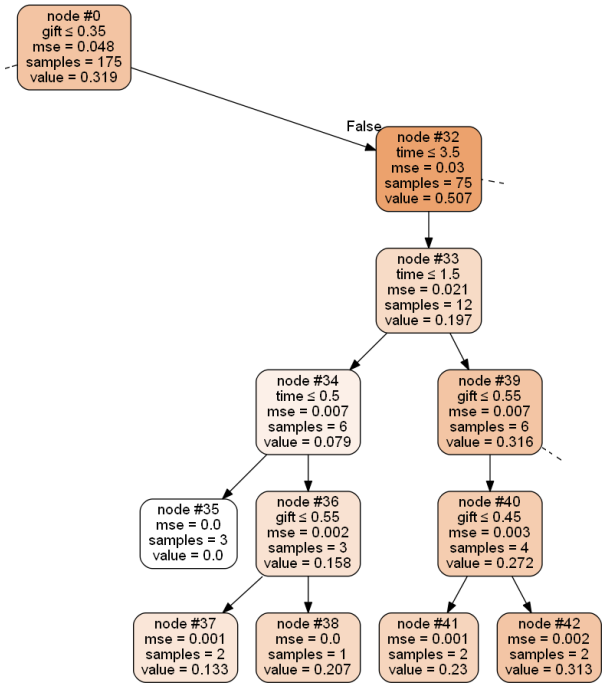


Figure 14: Illustration of an example path from root to leaf in the tree.

During the implementation phase, we consider two features: time and gift. To illustrate this, let's examine node 37 in Figure 14. The decision process starts by applying the rule at the root node (node 0), which states that the gift value must be greater than 0.35. Moving to node 32, 33, and 34, additional restrictions are imposed on the time feature, with values of being smaller than 3.5, 1.5, and 0.5, respectively. Node 37 is created as a result of applying the condition that the gift value must be greater than 0.55 on node 36.

Based on this rule set, we can infer that if the gift value falls between the lower bound of 0.35 and the upper bound of 0.55, and if the time value is less than 0.5, the corresponding retention value is estimated to be 0.133, which is the mean value associated with this set of conditions.

This detailed analysis of the decision tree provides insights into the specific rules and conditions that contribute to the prediction of survival rates. By examining the tree structure and the decision paths, we can gain a deeper understanding of how different combinations of tenure and bonus influence the estimated retention rates.

As mentioned before, the application of decision trees in regression analysis has proven to be an efficient approach for obtaining accurate fittings without the need for complex closed-form functions. Decision trees offer a flexible and intuitive framework for modeling relationships between input variables and their corresponding output values. However, to incorporate decision trees into mathematical formulations and enable their utilization in mathematical models, a suitable mathematical transformation is required. In this part, we propose a mathematical model that facilitates the representation of decision trees within a mathematical context, thereby enabling the integration of decision tree regressors into optimization models.

The notation used for proposed linearization of a decision-tree method, sets, parameters, decision variables, and contraints are provided below:

Sets:

- i = row, $i \in I$,
- f = feature, $f \in F$,
- k = leaf node, $k \in K$.

Parameters:

- u_{fk} = upper bound of feature f in leaf node k,
- l_{fk} = lower bound of feature f in leaf node k,
- v_{if} = value of feature f for row i,
- m_k = mean value of leaf node k.

Decision Variables:

- y_i =prediction value for row i,
- a_{ik} =1, if the node k is active for row i; 0, otherwise.

Constraints:

$$\sum_k a_{ik} = 1 \qquad \forall i \in I \tag{10}$$

$$y_i = \sum_k m_k a_{ik} \qquad \forall i \in I \tag{11}$$

$$v_{if} \leq \sum_k u_{fk} a_{ik} \qquad \forall i \in I, \forall f \in F \tag{12}$$

$$v_{if} \geq \sum_k l_{fk} a_{ik} \qquad \forall i \in I, \forall f \in F \tag{13}$$

$$a_{ik} \in \{0, 1\} \qquad \forall i \in I, \forall k \in K \tag{14}$$

Samples and features in a dataset are denoted as i and f, respectively. After obtaining the fitting, as previously described in example, upper(u_{fk}) and lower(l_{fk})

bounds are determined for the leaf node of the decision tree. Likewise, m_k, the average value of each leaf node, is fed into the system as a parameter. In light of these information, constraint 10 indicates that only one leaf node will be active for each row. Constraint 11 expresses that the predicted value, which is the output of the fitted model, corresponds to the average value of the active leaf node. Constraints 12 and 13 guarantee that each feature value in the upper and lower bounds of the active leaf node.

The utilization of decision trees in regression analysis has demonstrated its effectiveness in achieving accurate fittings, surpassing the need for intricate closed-form functions. The proposed approach also aims to facilitate the integration of decision trees within the context of optimization. By leveraging this approach, decision trees can be seamlessly incorporated into optimization models, enabling enhanced decision making and problem-solving capabilities. This integration allows for the utilization of decision trees as valuable components within the optimization framework, enabling the exploration and utilization of the rich information contained within decision tree models. By incorporating decision trees into the optimization context, the proposed approach enhances the flexibility and versatility of optimization models, providing a powerful tool for solving complex problems and making optimal decisions based on the insights and predictions derived from decision tree analysis.

3.5 Results and Discussion

Prior to presenting the numerical analysis, it is important to highlight the changes that have been introduced through the developed approaches in the retention management model. These modifications are implemented with the aim of reducing and simplifying the complexity of the model while preserving its effectiveness.

One of the key modifications involves objective function, 4, where the previously infinite time set has been limited to the planning period. This adjustment allows for a

more realistic representation of the retention dynamics within a defined time-frame. Additionally, to achieve greater computational efficiency and handle non-linearity, the retention rate and survivor relationship discussed earlier have been utilized in the establishment of this constraint:

$T = \{1, ..., T^n\}$, $T_\infty = \{1, \ldots, T^n, \ldots, M\}$, $t =$ time index, $t \in T_\infty$.

$$clv_i = \sum_{t \in T_\infty} \frac{p_i \prod_{t'=1}^{t} r_{it'}}{(1+d)^t} = \sum_{t \in T_\infty} \frac{p_i S_{it}}{(1+d)^t} \quad \forall i \in I \tag{15}$$

$$r_{it} = r_{i,T^n} \quad \forall i \in I, \forall t \in T_\infty \setminus T \tag{16}$$

Furthermore, Constraint 7 has also updated within Constraint 5 has also updated by changing retention rate to survivor function. In the revised formulation, the survivor calculation is influenced by both the customer's tenure and the total cost of actions taken by the customer during that period.

$$S_{it} = f(t, \Delta_{it}) \qquad \forall i \in I, \forall t \in T \tag{17}$$

In the preceding sections, we introduced and demonstrated *RPFit-Raw* approaches for computing the survivor function (S_{it}). The first approach involved constructing a closed-form function using raw survival rates. In second approach, *RPFit-BG* , the closed-form of the survival function is expressed over *BGFit*. Alternatively, we presented the decision tree method for representing a surface and generating a survivor function based on the data. This approach introduces new sets, and parameters and constraints that are integral to the relevant model as follows:

Sets:

- $f =$ feature, $f \in F = \{1 : \Delta c_{it}, 2 : t\}$,
- $k =$ leaf node, $k \in K$.

Parameters:

- u_{fk} = upper bound of feature f in leaf node k,
- l_{fk} = lower bound of feature f in leaf node k,
- m_k = mean value of leaf node k.

Decision Variables:

- a_{itk} =1, if the node k is active for customer i at period t ; 0, otherwise.

Constraints:

$$\sum_k a_{itk} = 1 \qquad \forall i \in I, \forall t \in T \tag{18}$$

$$S_{it} = \sum_k m_k a_{itk} \qquad \forall i \in I, \forall t \in T \tag{19}$$

$$\Delta c_{it} \leq \sum_k u_{1k} a_{itk} \qquad \forall i \in I, \forall t \in T \tag{20}$$

$$\Delta c_{it} \geq \sum_k l_{1k} a_{itk} \qquad \forall i \in I, \forall t \in T \tag{21}$$

$$t \leq \sum_k u_{2k} a_{itk} \qquad \forall i \in I, \forall t \in T \tag{22}$$

$$t \geq \sum_k l_{2k} a_{itk} \qquad \forall i \in I, \forall t \in T \tag{23}$$

$$y_{itk} \in \{0, 1\} \qquad \forall i \in I, \forall t \in T, \forall k \in K \tag{24}$$

In the evaluation phase, we examine three scenarios: *M:RPFit-Raw*, *M:RPFit-BG*, and *M:DTFit*. *M:RPFit-Raw* and *M:RPFit-BG* incorporate the use of a non-linear survival curve integrated into a refined retention management model. This model is solved using the BARON solver [40]. Conversely, *M:DTFit* adopts a decision tree methodology to generate the survival curve, as detailed in the preceding section. The resulting model is then solved using the CPLEX solver [41].

In the analyses, Cohort 23 has been selected as a reference, which includes customers with tenures extending up to 23 months. The median of the monthly payments, p_i, has been utilized for the customers. The base customer equity, *Baseline*,

is calculated utilizing the survival rates of the customers.

All scenarios are assessed with time interval in $T = 18$. The findings are presented in terms of customer equity and the computational time in seconds required for each approach. Additionally, the baseline method, serving as a benchmark, demonstrates a constant customer equity value. All experiments are conducted on an Intel Core i7-8500U machine equipped with 16GB of RAM.

Model	**Customer Equity**	**Time (sec)**
Baseline	30,233	-
M:RPFit-BG	33,367	174.23
M:RPFit-Raw	34,706	152.56
M:DTFit	33,367	35.49

Table 6: Solution statistics.

The results depicted in Table 6 provide a detailed comparison of customer equity, defined as the objective function, and computational efficiency across various scenarios. Each model achieves convergence to optimality, characterized by an optimality gap of zero.

Regarding customer equity, all scenarios surpass the Baseline scenario, with customer equity values exceeding 31,533. Additionally, the solutions for *M:RPFit-BG* and *M:DTFit* are congruent. This congruence arises due to the neglect of potential overfitting in the DTFit procedure. The benefit of this approach is attributed to employing a mixed integer linear model rather than a mixed integer nonlinear model, which reduces the solution time by approximately a factor of thirteen.

Referring to the solutions presented in Table 6, it is observed that the time required to reach an optimal solution differs significantly between the models. Specifically, *M:RPFit-BG* completes the process in 174.23 seconds, whereas *M:RPFit-Raw* achieves optimality more swiftly, requiring only 152.56 seconds. This highlights a more efficient computation despite potential differences in their methodologies or the

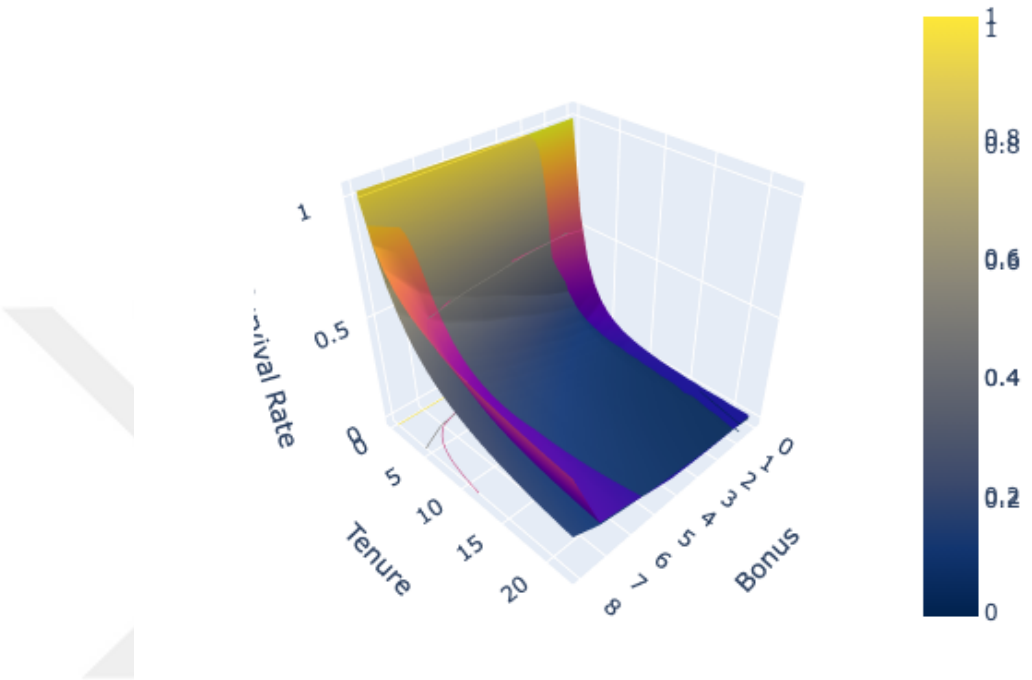


Figure 15: Comparison of RPFit-Raw and RPFit-BG models.

complexity of data handled. Notably, *M:DTFit* demonstrates the shortest computation time at 35.49 seconds, suggesting an even more streamlined process. The variation in computational efficiency and customer equity outcomes across the models underscores the influence of the underlying algorithms and their handling of the dataset complexities. Furthermore, an estimation of the area under the curve is performed using the Trapezoidal method [42]. This estimation involved calculating the sum of the areas under the curves for each bonus group. The area under the curve for *RPFit-Raw* is 38.7, compared to 36.3 for *RPFit-BG*, indicating an overestimation.

Moreover, as depicted in Figure 16, a survival curve is constructed based on the average survival rates at each tenure level for *M:DTFit*, *M:RPFit-Raw*, and *Baseline*.

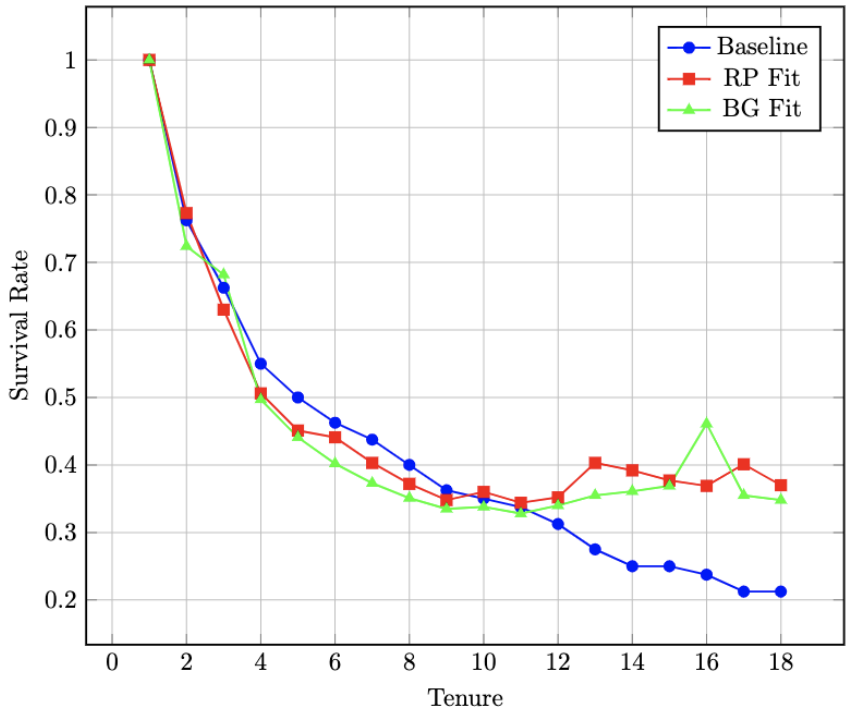


Figure 16: Comparison of average survival rates of all customer over tenure for the all scenarios.

Notably, *M:RPFit-BG* and *M:DTFit* yield identical solutions in this analysis, hence *M:RPFit-BG* is not separately analyzed. The findings demonstrate that the average survival rates for both models significantly exceed those of *Baseline*.

In conclusion, the analysis presented underscores the efficacy of the models in enhancing customer equity and optimizing computational efficiency relative to the Baseline. The models not only demonstrate a notable improvement in customer equity but also offer substantial reductions in solution times through the use of linear formulations.

CHAPTER IV

A MACHINE LEARNING-DRIVEN MATCHING ALGORITHM FOR RIDE POOLING PROBLEM

Ride pooling has emerged as a key solution for sustainable transportation, offering significant benefits such as reduced traffic congestion, lower pollution levels, and decreased transportation costs. With the growing complexity brought about by technological advancements and changing user behaviors, an innovative approach that combines ML with OR is increasingly being utilized.

This chapter explores how machine learning-driven optimization can transform driver and rider matching efficiency. Specifically, we integrate OR techniques not only to manage but also to enhance the ML training and testing phases. By applying OR's strategic optimization methods during these phases, we aim to fine-tune ML models to meet specific performance goals. These goals include reducing prediction errors and boosting reliability across varying conditions—a critical requirement for the dynamic environment of ride pooling services. We delve into the mechanics of how OR can be leveraged to enhance the predictive capabilities of ML models, examining case studies where this methodology has led to tangible improvements in service efficiency. Additionally, we discuss the challenges associated with implementing this integrated approach, such as the need for extensive data.

4.1 Background and Literature Review

In 2019, direct and indirect traffic congestion cost is estimated to reach more than $88 billion in USA [43]. This huge loss is attributed to several factors, such as cost of wasted fuel, lost productivity of individuals stuck in traffic and increased cost of

transporting goods through congested areas [44]. Public transportation is one of the most effective means of reducing congestion caused by commuters, since it significantly reduces the number of private vehicles required to respond to transportation needs of individuals. However, for some commuters and for many individuals, public transportation is not an option. For such cases, one possible way to reduce traffic is to increase the utilization of private vehicles. Several reports indicate that, on the average, 1.5 people travel in a single vehicle [45, 46]. Considering the available capacity of vehicles, merging transportation needs -through efficient matching strategies, that consider commuters and riders preferences- could decrease traffic congestion considerably.

Merging transportation needs of individuals is not a new problem, however, with the changing technology and user habits, its structure has completely changed. It is now possible, more than ever, to track availability of cars and transportation needs of individuals online. This transformation gives rise to new ecosystems, where individuals looking for a ride can be matched with others, who are willing to pool their ride. These ecosystems, mostly mobile applications, allow users to engage in a ride pooling activity under certain terms and conditions. For such ecosystems, ride pooling is referred as a transportation mode, in which individuals share a vehicle for a trip [47, 48]. The typical structure of *ride pooling* as a way of transportation is as follows: On one side, there are drivers, who have predetermined routes and departure times. On the other side, there are riders, who are willing to travel from an origin to a destination and looking for a mean of transportation. In between, there is a matching agency. The agency's main objective is collecting relevant trip information, such as routes and departure time preferences of drivers and riders, and matching them considering the available information. Once, the agency matches a driver and a rider, it offers them the matching and expect them to accept the offer. This process is called **ride matching** and addressed in the literature [47, 48, 49]. Note that

matching a rider and a driver is more complex than assigning a driver to a rider in UBER-like setting [50], since in the ride matching problem the rider chooses a ride-pal rather than picking up a cab with an anonymous driver. Although ridepool/carpool has great potential for commuting, it is reported that the ratio of people that prefers these services reduces every year [51, 52]. This shows that people (riders and drivers) cannot benefit from ride pooling well enough. Therefore, ride pooling agencies should dedicate their relevant resources to popularize their applications.

A ride pooling agency that aims to promote its application must provide flexible, convenient, and reliable services. Moreover, it must devise methods to motivate riders and drivers [53]. Flexibility is defined as the capability of a ride pooling agency to readjust the time schedules and itineraries with respect to unexpected changes. Convenience refers to the idea of incorporating drivers' and riders' preferences into ride matching processes and a reliable ride pooling agency is characterized by its ability to provide a service whenever it is requested. Matching rate, as a performance measure of reliability, is given as the ratio of accepted matching offers to all matching offers [54, 53] and it is concluded that increasing the matching rate is a crucial task for a ride pooling agency [55]. To increase the matching rate, the agency ensures the usage of an effective ride matching procedure.

An UBER-like application offers crowd sourced transportation and matches available drivers with people, who need rides. In UBER, drivers are contractors of UBER, and similar to taxi drivers, they take people from point A to B for a fee. In ridepooling, however, drivers are neither employees nor contractors, and they have their own routes. Drivers share their ride only if a rider has a similar route. Moreover, satisfaction of a set of criteria is also needed for matching. Such decentralization complicates the matching in ridepooling, and results in algorithmic challenges. From a methodological perspective of the peer-to-peer ride matching algorithms, the literature is mainly divided into three categories: (i) one-to-one ride matching, (ii) one-to-many

ride matching, and (iii) multihop ride matching [56]. In one-to-one ride matching, a driver/rider pair shares a single ride without any transfers, and in one-to-many ride matching, a driver shares her ride with multiple riders. The last category is actually an umbrella term includes both many-to-one and many-to-many ride matching. Intermediate transfers between drivers is allowed (e.g., a rider can have multiple cars in her route) in multihop ride matching, and this property distinguishes it from the first two categories. Various modelling approaches, and exact/heuristic solution methods are proposed for all ride matching categories. In the following paragraph, we briefly review the one-to-one ride matching literature, since the ride matching algorithm in this study belongs to this line of research.

In [57], authors propose a ride matching algorithm based on the requirement that the shared route percentage of driver/rider pair exceeds a predefined amount. As the first step of the ride matching procedure, possible driver/rider pairs do not satisfy the aforementioned requirement are pruned. Then, a weighted bigraph matching problem is optimally solved to obtain driver/rider pairs as one-to-one ride matches. A more elaborated model incorporating time-uncertainty into ride matching is proposed in [58]. The ride matching procedure starts with spatiotemporal pruning for all drivers and riders to define the feasible matches. Then a static ride matching problem with stochastic travel times is solved by Monte Carlo simulation for all drivers and riders such that the total travel costs are minimized, and the number of matches is maximized. Authors in [59] propose a ride matching procedure consists of three core components. In the preprocessing component, the targeted region is divided into clusters. Following the deployment of the system, when a ride matching request is provoked, all possible matches are pruned by geographical (e.g., drivers in remote clusters are not preferred) and temporal considerations. Then, the remaining set of matches are submitted to the rider in form of a list. The last component, namely in-memory indexing, deals with storing the data related to matches. However, authors

do not give any details about how to exploit this stored data. In a more recent study, [60], a ride matching algorithm integrating ride pooling and schedule-based transit system is proposed. The algorithm is a composition of the concept of space-time prism and a shortest path labeling algorithm. Space-time prism deals with spatiotemporal pruning as a preprocess to ride matching. Then, a procedure based on shortest path labelling and Belmann's principle of optimality is provoked to obtain a set of possible matches and this set of matches is submitted to the rider in the form of a list. For an up-to-date comprehensive review of matching algorithms, readers are referred to [56].

In [61, 62], it is discussed that the matching can be done through geographical and temporal pruning. Since, the agency have the information of drivers' and riders' departure time and locations, it can match drivers and riders that are close to each other in terms of location and departure time by using rule based methods. On the other hand, it can be argued that mathematical model based approaches find more acceptable matching offers. In [49, 63, 64, 65], authors model the ride matching problem as variants of a dial-a-ride problem, a well-studied problem in the operations research literature [66, 67, 68], which are also based on geographical and temporal pruning. [69] allows riders to choose any preferred driver, instead of pruning and matching the participants. Authors discussed that such a method is inefficient, because the riders tend to call their friends. A different approach is given in [70], where the authors proposed the idea of tag-based matching. In tag-based matching, a driver put tags on her trip and that trip is offered only to the riders who are interested in one or more of these tags. However, the performance of the approach is not presented in the study.

In this study, we propose a ride matching algorithm by respecting to a set of principles of decision theory literature [71]: (i) agents make their decisions based on a set of criteria, and (ii) weights of each criterion may differ from each other. Applying

these principles to ride pooling, we claim that a rider bases her/his decision on a set of criteria, and each criterion has its own weight. For each criterion, a rider has a ranking of matching offers and tends to accept an offer with higher ranks. However, since, the rankings of offers in each criterion may differ, a rider has to combine them into a consensus ranking. This process is known as **rank aggregation** in the literature [72, 73]. The rank aggregation problem focuses on combining multiple rankings into a single ranking. Several studies are presented in different areas such as ranking results of web engines [74], aggregating user ratings [75], and the determination of the winner in sports competitions [76]. In literature, the algorithms are proposed for the rank aggregation problem in three main categories: positional methods, comparison sort methods, and local search methods. For instance, Borda's method [77] is a positional method that finds a ranking that is minimum of the distances between each items' positions and their mean positions in the input rankings. Comparison sort algorithms compare each items' positions in rankings then sort the elements [74, 78]. Local search methods start with an initial ranking and try to improve a fitness function by moving items in each position [79]. In addition to these three main categories, a binary programming (BP) model is also proposed for the rank aggregation problem [80].

In ride pooling context, rank aggregation can be illustrated as follows. Assume that a rider has to decide one out of three drivers d_1, d_2, and d_3 to ride with, and she/he placed her/his request at 1:10pm. Rider's route matches 100% of d_1's route, 80% of d_2's route, and 60% of d_3's route. Hence, a ranking based on route match is given as $d_1 > d_2 > d_3$. Let starting points of all drivers be same with the rider's starting point. Since, all are the same, this information returns a ranking where all drivers share the same rank: $d_1 = d_2 = d_3$. Assume that the departure times are 01:15 PM, 01:25 PM, and 01:25 PM respectively for d_1, d_2, and d_3. Then, a ranking based on departure times is given as $d_1 > d_2 = d_3$. If, route match, starting point, and departure time are the three criteria for ride matching, then a rank aggregation can

be provoked to obtain the consensus ranking $d_1 > d_2 > d_3$. Hence, the rider matches with d_1. Now, consider the following example: There are two criteria and the rankings are $d_1 > d_2 > d_3$, and $d_2 > d_1 > d_3$. In the consensus ranking, d_1 and d_2 are in tie and this tie must be broken arbitrarily. It is possible to resolve this arbitrariness issue by assigning different weights to different rankings. If the weight of the first ranking is higher than the second one's, then the consensus ranking is $d_1 > d_2 > d_3$. Such a rank aggregation is called weighted rank aggregation and addressed in the literature [81]. However, there is no work incorporating the weighted rank aggregation problem into ride matching procedures in the literature.

The contribution of this study is threefold. First, it introduces the weighted rank aggregation into ride matching literature of ride pooling agencies. To do so, we update a Binary Program (BP) from the rank aggregation literature [80] by incorporating weights into the program. Second, we propose a machine learning method to learn these weights such that the matching rate is maximized. This is ensured by using a Nonlinear Bilevel Problem (NLBLP) model. Third, we proposed a ride matching framework, LBRanker, that contains two parts (online and offline) be used as a ride matching procedure. In the offline part of the procedure, the machine learning method learns the weights for each criterion. Once the weights are learned, they are transferred to the online part to provoke a rank aggregation to obtain a consensus ranking whenever a trip request is placed. A static approach is considered here in the sense that when a trip request is placed, we assume that the list of available drivers is fixed. Dynamic version of the problem is out of scope of this study, since dynamicity is an operational issue rather than an algorithmic one. Online part of the procedure is periodically improved by relearning the weights by provoking the offline part with additional data collected during the execution of the online part.

We develop our algorithm considering the needs of Lojika, an Istanbul based start-up company. Lojika developed its ride pooling application, TAG, with funding

received from the national science foundation of Turkey and European Union. TAG reached highest number of active users in city of Istanbul. Istanbul, with a population of more than 15 million, has a huge traffic congestion problem, such that it was a runner-up city in the traffic index announced by TomTom in 2013 and stays in top 10 since then [82]. TAG has a rule based ride matching algorithm, details of which will be provided in Section 4.2. Taking the existing matching algorithm in TAG, we develop our ride matching algorithm and test the performance of our method on the historical data from TAG. Results show that our algorithm performs significantly better than the current ride matching algorithm used in TAG, such that it correctly predicts the first choice of riders 17% to 28% better compared to the benchmark. Moreover, proposed approach offers recommendation lists in which the preferred driver is ranked 0.38 to 1.12 person closer (to the rider's actual choice) compared to the benchmark.

The rest of the paper is organized as follows. Details of the current ride matching approach used in the ride pooling company is presented in Section 4.2. We discuss the idea of using machine learning and optimization in ride matching problem and present our approach in Section 4.4, which is followed by the last section, numerical experiments and result discussions in Section 4.5.

4.2 Current Ride Matching Procedure: The Benchmark

The proposed optimization based approach, LBRanker, is developed to replace the current rule-based approach of the company, which is also used as a benchmark during the numerical experiments in Section 4.5. Therefore, for the sake of following a top-down approach, we first present the current ride matching procedure used in TAG.

The procedure of current approach in TAG follows a decentralized, peer-to-peer approach, i.e., it does not consider global matching of all driver/rider pairs. The procedure is provoked once a rider provides trip information (*origin-destination* pair), and preferred *departure time*. In this sense, it works in an online fashion. Then,

the candidate driver list is prepared by selecting a ranked-subset from the set of all available drivers. The procedure prunes some of the available drivers based on a set of rules, and ranks the remaining ones to prepare the candidate driver list. In particular, it applies a three steps approach: (i) spatiotemporal pruning, (ii) critera-based pruning, and (iii) ranking the drivers via *sorting*. In spatiotemporal pruning [70], the drivers out of the walking vicinity of the rider, and the drivers whose time windows do not intersect with the time window of the rider's trip are pruned. The procedure further prunes the remaining drivers with respect to a set of criteria in criteria-based pruning, e.g., percentage of route match, being co-members of an alumni association etc. The final step, ranking the drivers via *sorting*, ranks the drivers that survive through the pruning steps via a sorting algorithm, and eventually, it provides the candidate driver list. Once the list is ready, it is displayed to the rider and the rider is supposed to choose one or none of the drivers. If a driver is chosen, then the application informs the preferred driver and completes the ride matching with the acceptance of the driver. Otherwise, the procedure ends without a matching.

Flowchart of the ride matching procedure used in TAG is given in Figure 17. The nomenclature used in pseudocode of the algorithm and the pseudocode are provided in Table 7 and in Algorithm 1, respectively. Let D be the set of all available drivers and p be the rider. Also, let o_p and a_p be the origin-destination points of rider p and r_d be the route of driver $d \in D$. ν_1 and ν_2 are the threshold parameters for the distance of origin-destination points. Please note that $\{e_p, l_p\}$ and $\{e_d, l_d\}$ are the pairs of earliest and latest departure times of rider $p \in P$ and driver $d \in D$, respectively. Also note that the operator $m(x, y)$ in Algorithm 1 calculates the minimum distance between point x and route y, and $sortandselect(x, y, z)$ sorts the members of set x by values in set y in direction determined by z where z is either increasing or decreasing and returns the top $min\{\mid x \mid, N_{max}\}$ members where N_{max} is the number of maximum candidate drivers shown to the rider on the mobile app.

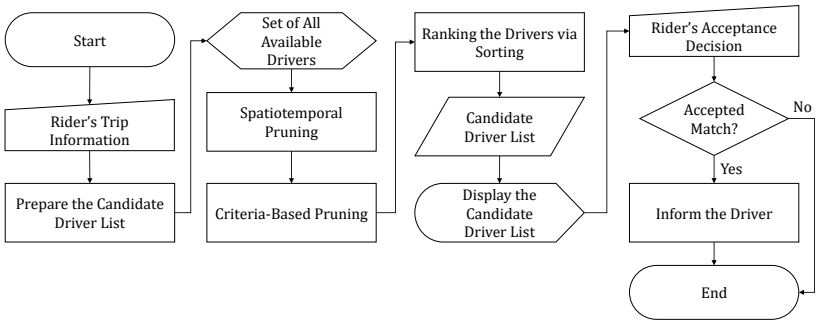


Figure 17: Flowchart of TAG's ride matching procedure.

Table 7: Set definitions for current ride matching procedure.

Set	Members	Size	Description	Notes
D	$\{d_1, d_2, \ldots, d_k, \ldots, d_s\}$	s	Set of All Available Drivers	
R	$\{r_1, r_2, \ldots, r_k, \ldots, r_s\}$	s	Routes of All Available Drivers	
P	$\{p_1, p_2, \ldots, p_k, \ldots, p_q\}$	q	Set of All Riders	
O	$\{o_1, o_2, \ldots, o_k, \ldots, o_q\}$	q	Origins of All Riders	
A	$\{a_1, a_2, \ldots, a_k, \ldots, a_q\}$	q	Destinations of All Riders	
C	$\{c_1, c_2, \ldots, c_k, \ldots, c_n\}$	n	Criteria Set	
V^d	$\{\nu_1^d, \nu_2^d, \ldots, \nu_k^d, \ldots, \nu_n^d\}$	n	Criteria Scores for each $d \in D$	$\nu_k^d \in [\nu_k^{dl}, \nu_k^{du}]$ holds where $\nu_k^d, \nu_k^{dl}, \nu_k^{du} \in R$
Θ	$\{\theta_1, \theta_2, \ldots, \theta_k, \ldots, \theta_m\}$	m	Subset of All Criteria for Candidate Selection	$\Theta \subset C, m \leq n$
Φ	$\{\phi_1, \phi_2, \ldots, \phi_k, \ldots, \phi_m\}$	m	Scores of Criteria Subset for Candidate Selection for driver $a \in A$	Θ and Φ are bijective, ϕ_k^a respects the boundary rules for its counterpart in H.
Λ	$\{\lambda_1, \lambda_2, \ldots, \lambda_k, \ldots, \lambda_m\}$	m	Acceptance Limits for Candidate Selection	Λ and Φ are bijective and λ_k respects the boundary rules for its counterpart in Φ.

In this study, we adopt the framework of the existing ride matching procedure and replace the simple *sorting* operation of *Ranking the Drivers via Sorting* with a BP based weighted rank aggregation model. Furthermore, we propose an NLBLP based machine learning method to learn the parameters of this model in an offline fashion.

4.3 Ranking the Drivers: BP Based Weighted Rank Aggregation

In this study, we propose a ride matching algorithm consists of two parts: (i) online part works with a BP based weighted rank aggregation method to prepare the candidate driver list, and (ii) offline part uses a machine learning algorithm to determine the weights to be used in the weighted rank aggregation. In this section we discuss the details of the BP based weighted rank aggregation method. To do so, the mathematical model, and various approaches of calculating its parameters are

Algorithm 1 Pseudocode of the ride matching procedure used in TAG.

$CD \leftarrow \emptyset$
$D_1 \leftarrow \emptyset$
$D_2 \leftarrow \emptyset$
for $d \in D$:
    if $m(o_p, r_d) \leq v_1$ and $m(a_p, r_d) \leq v_2$:
        $D_1 \leftarrow D_1 \cup d$
for $d \in D_1$:
    if $(e_p, l_p) \cap (e_d, l_d) \neq \emptyset$:
        $D_2 \leftarrow D_2 \cup d$
for $d \in D_2$:
    if $\{\nu_k^d \geq \lambda_k \quad \forall \theta_k \in \Theta\}$:
        $CD \leftarrow CD \cup d$
$\nu_{Total} \leftarrow \emptyset$
for $d \in CD$:
    $\nu_{Total}^d \leftarrow 0$
        for $c_k \in C$:
            $\nu_{Total}^d \leftarrow \nu_{Total}^d + \nu_k^d$
        $\nu_{Total} \leftarrow \nu_{Total} \cup \{\nu_{Total}^d\}$
$Ranking \leftarrow sortandselect(CD, \nu_{Total}, decreasing)$
return $Ranking$

discussed.

4.3.1 Mathematical Model

Assume that spatiotemporal and criteria-based pruning are applied in response to a rider's request and a set of drivers, D, is obtained. The members of set D should be ranked before displaying the list to the rider as candidate drivers list. This step must also be done in an online manner. Following the context of rank aggregation, the BP model *Rank Aggregation with Single Trip (RAST)*, adapted from [80], can be used for ranking the drivers of a single trip.

RAST:

$$\min \quad \sum_{i \in D} \sum_{j \in D | j > i} w_{ji} x_{ij} + w_{ij} x_{ji} \tag{25}$$

$$s.t. \quad x_{ij} + x_{ji} = 1 \qquad \forall i, j \in D, i < j \tag{26}$$

$$x_{ij} + x_{jk} + x_{ki} \geq 1 \qquad \forall i, j, k \in D, i \neq j \neq k \tag{27}$$

$$x_{ij} \in \{0, 1\} \qquad \forall i, j \in D, i \neq j \tag{28}$$

In *RAST*, x_{ij} is equal to 1 if driver i precedes driver j, and 0 otherwise in the final ranking. Parameter w_{ij} represents the *pairwise superiority* of driver i to driver j, and satisfies the following conditions: (i) $w_{ij} \in [0, 1]$, and (ii) $w_{ij} = 1 - w_{ji}$. Basically, w_{ij} represents the fraction of criteria that driver i precedes driver j, and we addressed the issue of calculating w_{ij} parameters in detail in the following section. From now on, we suppress the word *pairwise* and only use *superiority* while referring to parameter set w_{ij}. In *RAST*, Constraints (26) guarantees that either driver i or driver j precedes the other one. Constraints (27) satisfies the triangular inequality between drivers i, j, and k, e.g., if driver i precedes driver j and driver j precedes driver k, then driver i must also precede driver k. Constraints (28) are the binary constraints on x_{ij} variables. Please note that, *RAST* is for rank aggregation for a single trip of a rider. Later, we will extend the model to incorporate several trips of riders.

4.3.2 Calculating Superiorities

Let Π be a set of criteria taken into account while ranking the drivers such that $\pi_1, \pi_2, \ldots, \pi_k, \ldots, \pi_n \in \Pi$ where $\mid \Pi \mid = n$. Here π_k is a permutation, i.e., it is an ordered set of drivers with respect to criteria k. Hence, drivers have precedence relations for each π_k. Precedence matrix W, adopted from [83], is basically a matrix whose diagonal entries are zero and W_{ij} value represents the fraction of criteria that driver i precedes driver j, i.e., superiority value w_{ij}. By calculating the precedence matrix, one obtains w_{ij} values by using the following formula:

$$w_{ij} = \frac{1}{n} \sum_{k=1}^{n} \mathbf{1}(i <_k j), \tag{29}$$

where $i, j \in D$, $\mathbf{1}(\cdot)$ is an indicator function, and $<_k$ is an operator shows that left hand side driver precedes the other one in π_k. Each $w_{ij} \in W$ can be calculated using *Equation 29* for each $i, j \in D$ such that $i \neq j$, and transferred to *RAST* for rank aggregation. However, this calculation is only valid when π_k has full ranking, i.e., it is a permutation and for given two drivers there is exactly one winner. In driver ranking context, two drivers can have a tie in one or more criteria. For instance, rider's route may have an exact match with two distinct drivers' routes. Therefore, a modification of precedence matrix calculation is needed to calculate the superiorities. Let $s_{ij} \in S$ be the *auxiliary superiority point* of driver i over driver j for all $i, j \in D$ such that $i \neq j$. These points are unitless, not necessarily be in the closed set $[0, 1]$, and will be used for calculating the superiorities. An auxiliary superiority point is calculated by the following formula:

$$s_{ij} = \sum_{k=1}^{n} \mathbf{1}(i \leq_k j), \tag{30}$$

where $i, j \in D$, $\mathbf{1}(\cdot)$ is an indicator function, and $\leq_k$ is an operator shows that left hand side driver precedes the other one or they are in tie in π_k. One may calculate w_{ij} values by using the formula:

$$w_{ij} = \frac{s_{ij}}{s_{ij} + s_{ji}}. \tag{31}$$

Equation 31 ensures that $w_{ij} = 1 - w_{ji}$ and $w_{ij} \in [0, 1]$ for all $i, j \in D$ such that $i \neq j$. Hence, these w_{ij} values can be transferred to *RAST* as the full-ranking case.

It is possible to resolve the problem of ties in individual rankings with calculating w_{ij} values via *Equations 30-31*. However, it is not possible to resolve the problem

of breaking ties in a consensus ranking with such a calculation. Recall the rank aggregation example from Section 4.1, where d_1 and d_2 are in tie in the consensus ranking. Following the example, we denoted that this problem can be solved by using weighted rank aggregation by referring to the related literature. Here, we propose a formula that includes a set of weight coefficients ω_k for each criterion $\pi_k \in \Pi$ to calculate the auxiliary superiority points, and hence the superiorities w_{ij}. By such a calculation of w_{ij} values and feeding them to *RAST*, we introduce the model *Weighted Rank Aggregation with Single Trip (WRAST)*. The model is exactly the same with *RAST* with an exception in calculating the w_{ij} parameters. In *WRAST*, s_{ij} values are calculated by using the formula in *Equation 32*, and superiorities w_{ij} is calculated by feeding these values to *Equation 31*. The BP based weighted rank aggregation refers to the idea of obtaining a consensus ranking through the model *WRAST*.

$$s_{ij} = \sum_{k=1}^{n} \omega_k \times \mathbf{1}(i \leq_k j). \tag{32}$$

The weight coefficients, ω_k in *Equation 32*, can be determined by using various methods. For example, as a naïve method, the values of all ω_k coefficients would be considered as equal, i.e., $\omega_1 = \omega_2 = \cdots = \omega_k = \cdots = \omega_n$. On the other hand, one may (i) refer to domain expert opinions to choose ω_k values or (ii) use a machine learning approach to learn them by referring to historical data. Naturally, we assume that there is at least one pair of distinct coefficients such that $\omega_l \neq \omega_m$ for both approaches.

4.4 Machine Learning Embedded Ride Matching: LBRanker

In this study, we propose a ride matching procedure, namely LBRanker, that consists of two parts: (i) Online part: Ranking the drivers with *WRAST* when a rider places a trip request, and (ii) Offline part: Periodically learning ω_k values with a machine learning algorithm. As previously described in Section 4.2, TAG's current

ride matching procedure follows a three steps approach for preparing the candidate driver list: (i) spatiotemporal pruning, (ii) criteria-based pruning, and (iii) ranking the drivers via *sorting*. Ranking the members of a pruned list and sharing this ranked version with users is a common approach in search-based applications. For instance, Google uses PageRank of the relevant web pages and rank them accordingly before displaying the search result [84]. Following this custom in the literature, for the online part of LBRanker, we adopt the first two steps of the current procedure, and replace the third step, i.e., ranking the drivers via *sorting*, with a BP based weighted rank aggregation model, i.e, *WRAST*, to increase the quality of the ranked lists. As the offline part of the proposed ride matching procedure, we introduce an NLBLP model to learn the ω_k values. Following sections are dedicated to the NLBLP model and its main building blocks. One may see the flowchart of LBRanker in Figure 18.

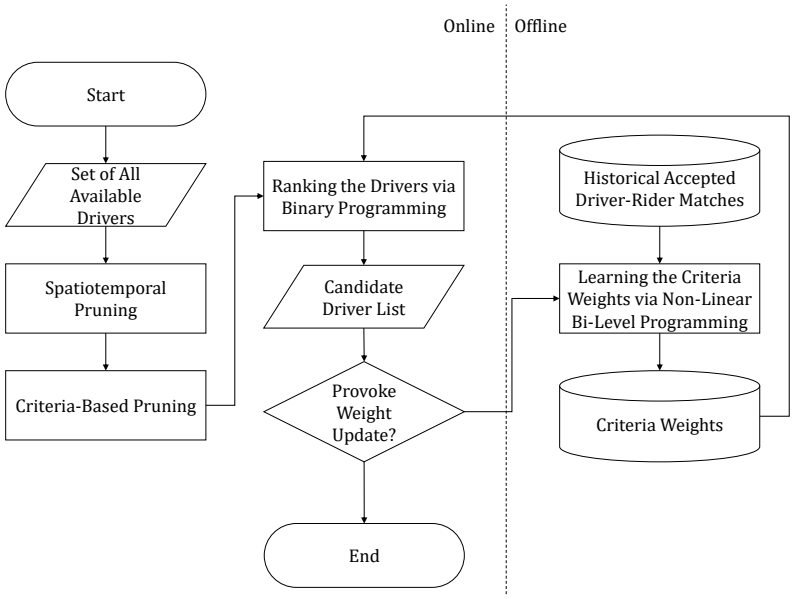


Figure 18: The flowchart of LBRanker.

4.4.1 The Main Building Blocks

There are two multipliers in the summation of *Equation 32* for criterion π_k: (i) Criteria weight ω_k, and (ii) indicator function $\mathbf{1}(i \leq_k j)$. Output of the indicator function k for drivers i and j is directly calculated by using the ranking of criterion π_k. Hence, values of all indicator functions for criterion π_k are computed *online* once the ride matching procedure is provoked. On the other hand, calculating criteria weight ω_k is not a straightforward task and it needs closer inspection. In LBRanker, we propose that ω_k value can be calculated via a machine learning algorithm for each criterion π_k in an *offline* fashion. By offline, we mean that calculation of ω_k values are handled beforehand and they remain constant once ω_k values are calculated, unless a relearning is provoked. For this purpose, we introduce a novel NLBLP model as a machine learning algorithm. Before going into the details of the NLBLP model, main building blocks of the model are shared in this section. These blocks are the followings: (i) A BP for simultaneous rank aggregation, and (ii) a performance measure. In this section, we also give a basic information about bilevel programming, and a solution approach for the NLBLP model.

4.4.1.1 A BP for Simultaneous Rank Aggregation

Ranking of drivers for a *single* rider's trip request is obtained through solving *WRAST*. However, we need a way to prepare rankings for *multiple* riders' trip requests simultaneously to be used in our machine learning algorithm. A binary program to serve such a purpose can be obtained with a slight modification to *WRAST*. Let T be the set of all riders' trip requests. Then, *Weighted Rank Aggregation with Multiple Trips (WRAMT)* is a BP to solve multiple weighted rank aggregation problems simultaneously.

WRAMT

$$\min \quad \sum_{t\in T}\sum_{i\in D_t}\sum_{j\in D_t|j>i} w_{ji}^t x_{ij}^t + w_{ij}^t x_{ji}^t \tag{33}$$

$$s.t. \quad x_{ij}^t + x_{ji}^t = 1 \qquad \forall i,j \in D_t, i<j, \forall t \in T \tag{34}$$

$$x_{ij}^t + x_{jk}^t + x_{ki}^t \geq 1 \qquad \forall i,j,k \in D_t, i\neq j\neq k, \forall t \in T \tag{35}$$

$$x_{ij}^t \in \{0,1\} \qquad \forall i,j \in D_t, i\neq j, \forall t \in T \tag{36}$$

In *WRAMT*, x_{ji}^t is equal to 1 if driver i precedes driver j in trip request t, and 0 otherwise in the final ranking. Parameter w_{ij}^t represents the superiority of driver i to driver j in trip request t, and calculated with *Equations 31-32*.

4.4.1.2 Performance Measure

Assume that the candidate drivers are ranked by *WRAST* and displayed at the screen of the rider. If the rider accepts one of the drivers from the candidate driver list, the error of the given ranking can be calculated with the formula given below:

$$\phi = n - \sum_{j\in D|j\neq f} x_{fj}^* - 1 \tag{37}$$

In *Equation 37*, ϕ is the error, n is the number of drivers in the candidate driver list, $f \in D$ is the index of the accepted driver where D is the set of all drivers in the candidate list, and x_{ij}^* is the optimal variable values. For example, if there are five drivers in the candidate driver list and the rider accepts the second driver in the ranking, then $\sum_{j\in D|j\neq f} x_{fj}^* = 3$ and $\phi = 5-3-1 = 1$. Similarly, if the rider accepts the first driver in the ranking, then $\sum_{j\in D|j\neq f} x_{fj}^* = 4$ and $\phi = 5-4-1 = 0$, meaning that the best possible scenario is realized and the resulting error is zero.

Individual errors for each trip request in case of simultaneous rank aggregation can be calculated by using *Equation 38* where ϕ_t is the error of $t \in T$, n_t is the number of drivers in the candidate list of trip request t, $f_t \in D_t$ is the index of the accepted

driver where D_t is the set of all drivers in the candidate driver list of trip request t, and ${x_{ij}^t}^*$ is the optimal variable values of *WRAMT*.

$$\phi_t = n_t - \sum_{j \in D_t | j \neq f_t} x^*_{f_t j} - 1 \tag{38}$$

The NLBLP model proposed in this study aims to learn the criteria weights ω_k such that the sum of the errors ($\sum_{t \in T} \phi_t$) is minimized, since a global-based approach is inherited.

4.4.2 Learning the Weights: Nonlinear Bilevel Programming Model

In this section we give the details of the offline part of our ride matching procedure which learns the weights of each criteria using an NLBLP.

4.4.2.1 Bilevel Programming

Bilevel optimization problem is an optimization problem that includes another optimization problem. The problem contains two levels of optimization tasks where main optimization task is upper-level optimization and sub optimization task is low-level that is nested in the main problem [85]. The general formulation of bilevel optimization problems is as follows:

$$\min_{x_u \in X_u, x_l \in X_l} \quad F(x_u, x_l) \tag{39}$$

$$s.t. \quad G_k(x_u, x_l) \leq 0 \qquad \forall k = 1, \ldots, K \tag{40}$$

$$x_l \in argmin\{f(x_u, x_l) : g_j(x_u, x_l) \leq 0, j = 1, \ldots, J\} \tag{41}$$

where $F : R^n \times R^m \rightarrow R$ represents the upper level objective function, $G_k : X_u \times X_l \rightarrow R, k = 1, \ldots, K$ denote the upper level constraints, $f : R^n \times R^m \rightarrow R$ represents low level objective function and $g_j : X_u \times X_l \rightarrow R, j = 1, \ldots, J$ represent the lower level constraints.

The usual approaches of solving a bilevel programming problem is based on reformulation of the problem as a single level problem. In the literature, various methods are suggested to reformulate a bilevel programming problem to a single level problem such as primal Karush-Kuhn-Tucker (KKT) transformation, classical KKT transformation, and optimal value transformation [86]. These methods obtain a single level model such that the optimality of lower level problem is guaranteed. The details of the reduction methods are not discussed in this paper, interested readers could refer to [86].

4.4.2.2 Mathematical Model

The mathematical formulation of the machine learning method is based on the idea of finding ω_k values for all $\pi_k \in \Pi$ such that the sum of the errors of each ranking is minimized. Let $\mathbf{1}(i \leq_k^t j)$ be an indicator function equals to 1 if driver i precedes or is in tie with driver j in criteria $k \in K$ for trip request $t \in T$ in current procedure of TAG. Then, one may found the NLBLP model below as *Rank Aggregation Training with Bilevel Programming Model (RATBLP)*. Please note that all the sets used in the formulation is defined in earlier sections with an exception. The set of criteria Π is changed with $K = \{1, 2, \ldots, k, \ldots, n\}$. The parameters, and decision variables are used in *RATBLP* is shown in Table 8:

Table 8: Parameters and variables of RATBLP.

Parameter	Description
n_t	Number of candidate drivers for trip request $t \in D_t$
f_t	The order of the accepted driver in trip request $t \in D_t$

Variable	Description
ϕ_t	Error of the ranking for trip request $t \in D_t$ such that $\phi_t \in \mathbb{R}$
w_{ij}^t	Superiority between driver i and driver j in trip request $t \in D_t$ such that $w_{ij}^t \in \mathbb{R}^{\oplus}$
ω_k	Weight of criteria $k \in K$ such that $\omega_k \in \mathbb{R}^{\oplus}$
x_{ij}^t	Binary variable equals to 1 if driver i precedes driver j in trip request $t \in D_t$, and 0 elsewhere

RATBLP:

$$\min \quad \sum_{t \in T} \phi_t \tag{42}$$

$$s.t. \quad w_{ij}^t = 0 \quad \forall i, j \in D | i = j, \forall t \in T \tag{43}$$

$$w_{ij}^t + w_{ji}^t = 1 \quad \forall i, j \in D | i < j, \forall t \in T \tag{44}$$

$$w_{ij}^t + w_{jm}^t - w_{im}^t \geq 0 \quad \forall i, j, m \in D | i \neq j \neq m, \forall t \in T \tag{45}$$

$$w_{ij}^t - \sum_{k \in K} \mathbf{1}(i \leq_k^t j)\omega_k = 0 \quad \forall i, j \in D | i \neq j, \forall t \in T \tag{46}$$

$$\sum_{k \in K} \omega_k = 1 \tag{47}$$

$$\phi_t - n_t - \sum_{j \in D | j = f_t} x_{ij}^t - 1 \geq 0 \quad \forall t \in T \tag{48}$$

$$w_{ij}^t \geq 0 \quad \forall i, j \in D, i \neq j, \forall t \in T \tag{49}$$

$$\min \quad \sum_{t \in T} \sum_{i \in D} \sum_{j \in D | j > i} w_{ji}^t x_{ij}^t + w_{ij}^t x_{ji}^t \tag{50}$$

$$s.t. \quad x_{ij}^t + x_{ji}^t = 1 \quad \forall i, j \in D, i < j, \forall t \in T \tag{51}$$

$$x_{ij}^t + x_{jm}^t + x_{mi}^t \geq 1 \quad \forall i, j, m \in D, i \neq j \neq m, \forall t \in T \tag{52}$$

$$x_{ij}^t \in \{0, 1\} \quad \forall i, j \in D, i \neq j, \forall t \in T \tag{53}$$

In the above formulation, constraints (43)–(49), together with the objective function at (42), represent the upper-level problem, and *Equations (50)–(53)* represent the lower-level problem which is actually *WRAMT* and ensures that a valid rank aggregation is obtained for each trip request. While the entire model aims to find criteria weights such that the total error of rankings is minimized, lower-level problem ensures that each ranking is obtained through weigthed rank aggregation. In the upper-level problem, constraints (43–45) and (49) satisfy the conditions on superiorities as discussed in earlier parts of this section. Constraints (46) calculate the superiorities by using *Equation (32)*. Constraint (47) ensures that sum of the weight coefficients for all criteria add up to 1, and constraints (48) calculate the errors for individual rankings for each trip request separately.

4.4.2.3 Solution of RATBLP: A Reformulation

It is a common approach to solve bilevel programs by adding the optimality conditions of lower-level problem as constraints to upper-level problem. However, such an approach necessitates that the lower-level problem is convex [87]. Since, the lower-level problem of *RATBLP* is a BP, i.e., it is nonconvex, we cannot exploit such an approach. Therefore, we reformulate the model as a single-level MINLP. One may find the reformulation below as *Rank Aggregation Training with Single-level Programming Model (RATSLP)* and it is obtained by replacing the lower-level problem of *RATBLP* with constraints (55)-(58) and (66).

RATSLP

$$\min \quad \sum_{t \in T} \phi_t \tag{54}$$

$$s.t. \quad x_{ij}^t + x_{ji}^t = 1 \quad \forall i, j \in D, i < j, \forall t \in T \tag{55}$$

$$x_{ij}^t + x_{jm}^t + x_{mi}^t \geq 1 \quad \forall i, j, m \in D, i \neq j \neq m, \forall t \in T \tag{56}$$

$$w_{ji}^t x_{ij}^t + w_{ij}^t x_{ji}^t \leq w_{ij}^t \quad \forall i, j \in D, i < j, \forall t \in T \tag{57}$$

$$w_{ji}^t x_{ij}^t + w_{ij}^t x_{ji}^t \leq w_{ji}^t \quad \forall i, j \in D, i < j, \forall t \in T \tag{58}$$

$$w_{ij}^t = 0 \quad \forall i, j \in D | i = j, \forall t \in T \tag{59}$$

$$w_{ij}^t + w_{ji}^t = 1 \quad \forall i, j \in D | i < j, \forall t \in T \tag{60}$$

$$w_{ij}^t + w_{jm}^t - w_{im}^t \geq 0 \quad \forall i, j, m \in D | i \neq j \neq m, \forall t \in T \tag{61}$$

$$w_{ij}^t - \sum_{k \in K} \omega_k p_{ij}^{tk} = 0 \quad \forall i, j \in D | i \neq j, \forall t \in T \tag{62}$$

$$\sum_{k \in K} \omega_k = 1 \tag{63}$$

$$\phi_t - n_t - \sum_{j \in D | j = f_t} x_{ij}^t - 1 \geq 0 \quad \forall t \in T \tag{64}$$

$$w_{ij}^t \geq 0 \quad \forall i, j \in D, i \neq j, \forall t \in T \tag{65}$$

$$x_{ij}^t \in \{0, 1\} \quad \forall i, j \in D, i \neq j, \forall t \in T \tag{66}$$

$$\omega_k \geq 0 \quad \forall k \in K \tag{67}$$

We present an important analytical result in the following proposition which indicates that the reformulation of *RATBLP*, given as *RATSLP*, is equivalent to *RATBLP* given that Assumption 1 holds, and hence allows us to solve the NLBLP as a single level MINLP:

Assumption 1. *For each $i, j \in D$, $w_{ij} \neq w_{ji}$ holds.*

Proposition 1. *The optimum ranking of RATSLP, is also optimal for RATBLP.*

Proof. By contradiction. Suppose a ranking ρ, that is not optimum for *RATSLP*, is optimal for *RATBLP*. In this ranking, there must be at least two adjacent drivers,

let us say driver k precedes driver l, such that

$$w^*_{kl} < w^*_{lk} \tag{68}$$

where w^*_{ij} are optimal weights obtained at *RATBLP* for all $(i, j) \in D$. *Equation 68* holds thanks to the constraints (57) and (58) in *RATSLP*.

Perform an adjacent pairwise interchange on drivers k and l and name the new ranking ρ'. In ρ, driver k precedes driver l, while in ranking ρ' the opposite is true. All other drivers retain their actual ranks. Since k precedes l in ρ, $x_{kl} = 1$ and $x_{lk} = 0$. Also, for ranking ρ', $x_{kl} = 0$ and $x_{lk} = 1$. Let $f(\rho)$ and $f(\rho')$ be the objective function values of *RATBLP* respectively for rankings ρ and ρ'. Then,

$$\begin{aligned} f(\rho) &= \sum_{i,j \in D'} w^*_{ij} x_{ji} + w^*_{ji} x_{ij} + w^*_{kl} x_{lk} + w^*_{lk} x_{kl} \\ &= \sum_{i,j \in D'} w^*_{ij} x_{ji} + w^*_{ji} x_{ij} + w^*_{lk} \\ f(\rho') &= \sum_{i,j \in D'} w^*_{ij} x_{ji} + w^*_{ji} x_{ij} + w^*_{kl} x_{lk} + w^*_{lk} x_{kl} \\ &= \sum_{i,j \in D'} w^*_{ij} x_{ji} + w^*_{ji} x_{ij} + w^*_{kl} \end{aligned}$$

where $D' = D \setminus \{(k, l)\}$.

Thus the difference of $f(\rho)$ and $f(\rho')$ is due only to driver k and l. It is easy to conclude that

$$f(\rho) > f(\rho')$$

holds. This contradicts the optimality of ρ and completes the proof of the proposition.

□

4.5 Numerical Experiments and Results

In this section, we present performance analysis of the proposed data-driven matching algorithm using a real-life data set obtained from the company Lojika. The data is

retrieved from the application called TAG, which is the ride pooling application of Lojika, listed in both Play Store and App Store since 2017. We work on a partial data set, where the preferred drivers accept ride matching offers. Studied data set contains 1,338 rows, one row for each trip request; and each trip request has 14 columns: (i) candidate driver list, (ii) criteria scores for each criterion $1, 2, \ldots, k, \ldots N$ where $N = 12$, and (iii) the accepted driver. The criteria used in the proposed procedure can be categorized as: criteria based on route (e.g. length of the route), criteria based on community (e.g. school/work information), criteria based on demographics (e.g. age), and criteria based on application usage (e.g. frequency of application usage). Please note that further details about the criterion set cannot be shared due to confidentiality. Details of these columns are given in the following list:

- ***Candidate Driver List:*** List of drivers shared in the screen of the rider e.g. $\{A, B, C\}$.
- ***Criteria Scores for Criteria*** k***:*** A list of real numbers. The values are elements of a closed set $x \in \mathbb{R}$, bijective with the candidate driver list e.g. $\{A = 20, B = 10, C = 30\}$.
- ***Accepted Driver:*** Categorical variable indicating the accepted driver from the candidate driver list e.g. $\{B\}$.

The details of the compared solution approaches, metrics used in the comparisons and scenario analyses conducted are presented next.

The company's current approach, named "Benchmark", is considered as benchmark for the proposed solution approach. In our solution approach, one of the key components is the determination of criteria weights. These weights, ω_k in Equation 32, can be determined by using various methods. In this study, we determine the weights by using two different methods, hence we end up with two variants of our ranking algorithm. In the first variant, we calculate the weights by using domain

knowledge of Lojika experts and name this version fixed weights ranker, FWRanker. Second variant is called learning based ranker, LBRanker, which uses historical data and proposed NLBLP model to find the optimal ω_k values. The reason for proposing two variants, FWRanker and LBRanker, is to clearly show the marginal contribution of learning the criteria weights from historical data. In other words, comparing FWRanker to Benchmark reveals the marginal improvement of using rank aggregation idea through optimization; and comparing LBRanker to FWRanker shows the marginal improvement promised by learning of criteria weights using historical data.

We use *Equation 69* to calculate the weight coefficient values of FWRanker.

$$\omega'_k = \frac{\eta^u_k}{\sum_{i \in C} \eta^u_i}, \qquad \forall k \in C \tag{69}$$

The equation is based on η^u_k values (from Table 7) which are determined by company experts. Once we have the weight coefficients, we construct *WRAMT* by using Equation 32 with the calculated coefficients. Then, the candidate driver list is prepared by solving the model.

We share the details of the second approach, LBRanker, in Section 4.4, where we introduce a NLBLP model to learn the weight coefficients. In the training phase, *RATSLP* is used and the optimal weight coefficients ω^*_k are determined for all $k \in C$. In the testing phase, *WRAMT* is prepared for test set using these weight coefficients. We implement all models by using the GAMS IDE [88]; use BARON solver [40] for *RATSLP* and CPLEX solver [41] for *WRAMT*. We perform all experiments on an Intel Core i7-8500U machine with 16GB RAM.

We compare the algorithms by using two metrics. First metric is based on the error term defined in *Equation 38*. The second metric is an accuracy based performance measure, r, which is calculated as follows.

$$r = \frac{1}{|T|} \sum_{t \in T} \delta_t. \tag{70}$$

where δ_t is equal to 1 if rider's preference is ranked first in the algorithm's recommendation list for trip t, 0 otherwise. In other words, r indicates what percent of the riders' first choice is predicted correctly by the algorithm.

Before providing the results of the proposed method, we prepare the driver rankings for each trip request by using the company's methodology and calculate the errors for each trip request by following *Equation 38*. We finalize the calculation by summing all errors, i.e., $\sum_{t \in T} \phi_t$. Then, we transform the criteria scores to rankings as follows: For each criterion, we sort the scores in decreasing order and replace them by their ranks. In case of tie between a set of drivers, the same rank is assigned to those drivers. For example, if the scores of each driver for a criterion are $\{A = 30, B = 30, C = 40\}$, then the ranks would be $\{A = 2, B = 2, C = 1\}$.

We use two different scenarios to test the performances of the algorithms: i) time-wise split, which means train and test dataset are created by considering time and ii) time-independent split, which utilizes cross validation approach. All in all, we share the performance of ranking algorithms by using two metrics, error and accuracy, under two different split scenarios, the time-wise and the time-independent.

In the first part of analysis, the time-wise split scenario, we divide the dataset into two subsets: (i) training set and (ii) test set. The total number of trip requests used for training set is 937. This set covers data from the interval between 06/12/2018 and 08/17/2018. The test set spans fifteen-days following the ending of training dataset period. The stratified sampling method is used to create the test sets based on the length of *candidate driver list*. We consider lists of size ranging from two to six. Details of all sets are given in Table 9. First column of Table 9 gives the number of drivers in a *candidate driver list*. The following columns give the number of trip requests respectively for each length in the training and test datasets. For example,

test set contains 74 trip requests with five drivers in the *candidate driver list*.

Table 9: Details of training and test sets.

# of Drivers	Training Set	Test Set
2	255	110
3	195	83
4	174	74
5	174	74
6	139	60
Total	**937**	**401**

In the training phase, *RATSLP* is solved using BARON Solver [40]. It takes 665.78 seconds to solve the problem and the resulting error with optimal ω_k^* is found as 0.632, whereas the benchmark's error is 1.287. Note that in our context the error, ϕ, means that the driver is displayed $(\phi+1)^{\text{th}}$ position at the screen of the rider. It can be said that criteria based on route and community turn out to be more important compared to criteria based on application usage and demographics.

The performances of different ranking approaches is compared using the test dataset. Benchmark approach lists the riders' actual choice, on the average, at 2.773^{th} position (1.773 person away from the first position, i.e., the error is 1.773), whereas FWRanker locates rider's first choice, on the average, 0.905 person away from the first position. On the other hand, the error of machine learning embedded ride matching method, LBRanker, is is 0.656. This means that, LBRanker is capable of positioning rider's actual choice 1.117 and 0.249 person closer to the top recommended driver than benchmark and FWRanker, respectively.

To further investigate the performance analysis of the three ranking approaches, we choose to breakdown the error performances with respect to candidate drivers list attribute of rides. For this purpose, we categorize the candidate driver lists with respect to the number of drivers and present the error value of each approach for each case. Error value breakdown of the ranking approaches on the test set can be tracked in Table 10. First column of the table gives the number of drivers

Table 10: Comparison of errors for the test set with respect to number of drivers in the time-wise split scenario.

# of Drivers	Benchmark	FWRanker	LBRanker
2	0.609	0.418	0.418
3	1.205	0.855	0.880
4	2.095	0.838	0.811
5	3.243	0.757	0.581
6	2.483	2.133	0.683
Overall	1.773	0.905	0.656

in the candidate driver list. Remaining columns show the average error values of Benchmark, FWRanker, and LBRanker approaches. The overall average errors can be seen at the last row of the table. Even though FWRanker ties with LBRanker in 2-driver case and beats LBRanker in 3-driver cases; the results reveal that LBRanker is superior to FWRanker as the number of drivers increases. LBRanker shows a significant performance for 6-driver cases, with a difference of 1.45 to its closest rival, FWRanker. We know that as the number of candidate drivers increases, the ranking problem becomes more complex due to increasing number of choices. Therefore, it can be concluded that LBRanker is more suitable than FWRanker for complex cases with larger number of candidate drivers. Both FWRanker and LBRanker dominate the Benchmark approach for all cases listed in Table 10.

We also compare ranking approaches using the second metric, accuracy. Comparison of prediction accuracy for the test set is shown in Figure 19. Please note that, the accuracy values are calculated with respect to number of drivers in a candidate driver list. The number of drivers is presented on the x axis, where the y axis shows attained accuracy values. It is observed in Figure 19 that prediction accuracy of LBRanker on the test set is superior to that of current methodology of the company for all number of driver cases. In comparison with FWRanker, LBRanker yields higher or at worst equal accuracy values. It is observed that, on the average, LBRanker correctly predicts the first choice of riders around 51% of the time, whereas Benchmark and

FWRanker are accurate in 22% and 46% of the predictions, respectively.

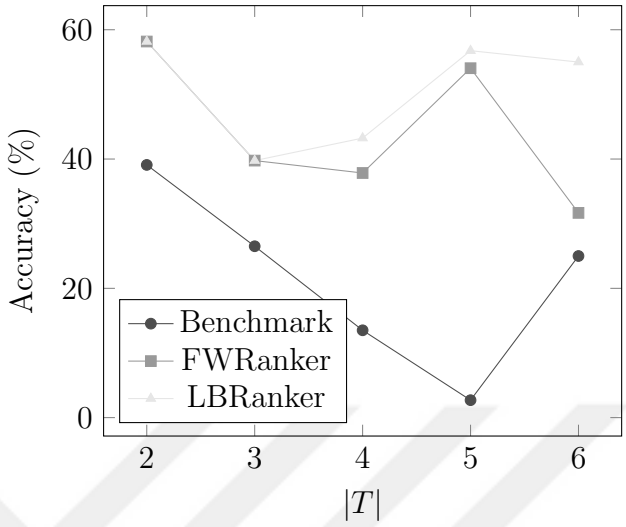


Figure 19: Comparison of prediction accuracy breakdown for the test set with respect to number of drivers in the time-wise split scenario.

In the second (i.e., the time-independent) scenario, we use a cross validation procedure to assess the robustness of the proposed approaches. We divide the data into five equal size subsets (in order to create a 5-fold setting), regardless of time information. While creating the folds, we use a stratified-k-fold sampling method that maintains the same distribution for the number of candidate drivers in every fold.

Table 11: Comparison of errors on the time-independent scenario.

	Fold 1	**Fold 2**	**Fold 3**	**Fold 4**	**Fold 5**	**Overall**
Benchmark	1.380	1.442	1.424	1.446	1.468	1.432
FWRanker	1.271	1.273	1.249	1.251	1.199	1.249
LBRanker	0.794	1.035	0.996	1.177	1.260	1.052

Table 11 shows that rider's actual choice is listed, on the average, at 2.432^{th} position (1.432 person away from the first position) and 1.249^{th} position (2.249 person away from the first position) in the recommendation list provided by Benchmark and FWRanker, respectively. On the other hand, LBRanker provides recommendation lists where the actual choice of a rider is located on average 2.052^{th} position (1.052 person away from the first position).

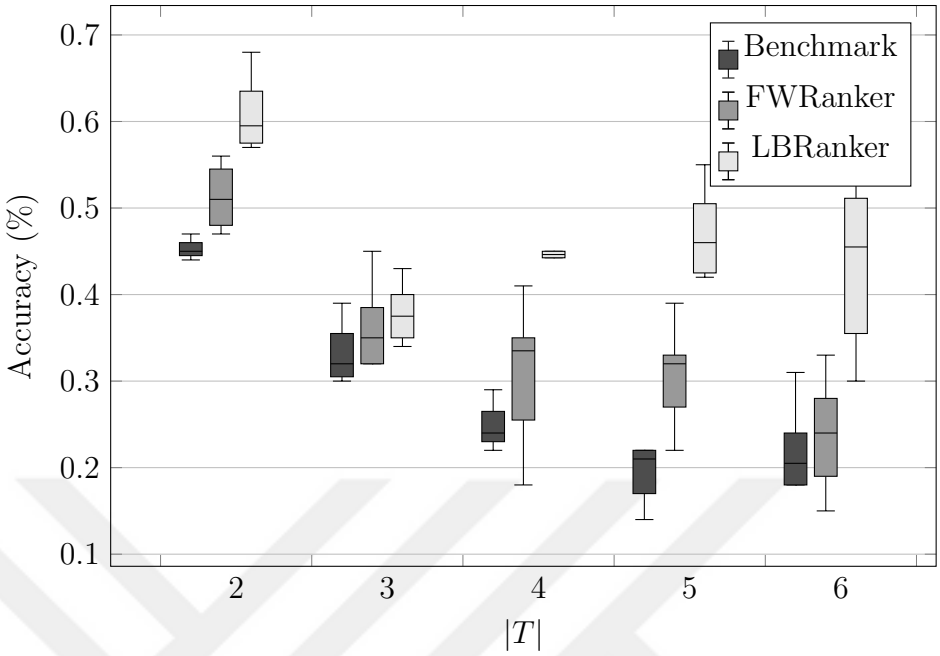


Figure 20: Comparison of prediction accuracy breakdown with respect to number of drivers in the time-independent scenario.

For the time-independent scenario, accuracy values of the approaches are also shared using box-plots, to clearly show the distribution of the performance among folds. Figure 20 shows the accuracy distribution of each approach for each driver case. LBRanker predicts the first choice of riders in 2-driver cases with average accuracy of 61%, where average accuracy of predictions are 37%, 46%, 49%, and 45% for 3-driver, 4-driver, 5-driver, and 6-driver cases, respectively. Moreover, LBRanker's average first choice accuracy is higher than Benchmark for each fold in each driver case. Similarly, LBRanker attains higher average first choice accuracy, in all cases, when compared to FWRanker. Moreover, for 2-driver, 4-driver, and 5-driver cases LBRanker dominates FWRanker in all five-folds.

A closer inspection of Figure 20 reveals that we obtain consistent results with earlier time-wise split scenario analyses: a similar pattern for average prediction accuracy and error performance is observed with respect to number of drivers in both scenarios. This shows that LBRanker is a valid ranking approach, promising high accuracy

and low error results compared to the Benchmark approach. Moreover, comparison of FWRanker and LBRanker reveals that learning weights through computationally expensive machine learning and optimization algorithms pays off as LBRanker yields better results in almost all comparative analyses.

As a final analysis, we aggregated the results over the number of drivers and present an overall performance analysis using the weighted averages of accuracy and error values attained by all algorithms on both split scenarios. Results of the aggregated analysis is presented in Table 12. It is seen that LBRanker correctly predicts the first choice of riders 17% and 28% better compared to the benchmark for time-independent and time-wise split scenarios, respectively. Moreover, LBRanker offers recommendation lists in which the preferred driver is ranked 0.38 person closer to the rider's actual choice compared to the benchmark when time-independent split scenario is considered, and this gap widens to 1.12 person in the time-wise split scenario.

Table 12: Aggregated performance results.

		Performance Metric	
Approach	Split Scenario	Accuracy	Error
Benchmark	Time-wise	22.94%	1.773
FWRanker	Time-wise	45.89%	0.905
LBRanker	Time-wise	50.87%	0.656
Benchmark	Time-independent	32.17%	1.432
FWRanker	Time-independent	37.65%	1.249
LBRanker	Time-independent	48.99%	1.052

CHAPTER V

CONCLUSIONS

This dissertation has explored the synergy between machine learning (ML) and operations research (OR) to address complex optimization issues within the realms of digital marketing and the sharing economy. This chapter synthesizes the findings in relation to the initial objectives, offers insights for both academics and practitioners, and outlines potential avenues for further research.

The work has demonstrated the dynamic interaction between ML and OR in solving real-world problems. The progression from theoretical underpinnings to practical applications in enhancing customer retention and optimizing ride-sharing operations, culminating in algorithmic improvements, has collectively established a framework that strengthens decision-making processes and operational efficiency.

The primary objective was to develop models that integrate ML and OR to enhance decision-making within digital marketing and the sharing economy. The initial hypotheses posited that ML could enhance the predictive accuracy of traditional OR models, thereby enabling more effective strategies and outcomes. The results confirmed these hypotheses, revealing significant improvements in customer retention management and ride-sharing operation optimization through the implemented algorithms.

In the first case study, the focus is on developing a predictive model that effectively utilizes customer behavioral data to predict retention probabilities and a prescriptive model to recommend optimal retention strategies. The predictive model evaluates the likelihood of customer retention following specific incentives, such as gifts or rewards. Subsequently, the prescriptive model considers the total Customer Lifetime

Value (CLV) across the customer base, identifying key targets for retention efforts by integrating the predictive model into prescriptive model. The models has demonstrated an improvement of 10% in customer equity. Moreover, the transformation of the decision tree model into a linear framework offers a significant advantage in terms of solution time when compared to the nonlinear model.

In the second case study, a ride-matching algorithm is developed that significantly outperforms existing methods by leveraging a data-driven approach to enhance driver-rider pairings. This algorithm, developed in collaboration with a startup, utilizes historical data of accepted driver-rider matches to build a recommendation system for future requests. It employs a learning-based approach to determine the importance of various criteria, which it then aggregates into a consolidated ranking of drivers for each request. This approach is shown to surpass current methodology used by the company by 17% to 28% in predicting riders' first choices and improving the accuracy of driver rankings.

This research highlights the practical advantages of combining ML with OR, notably in enhancing operational efficiency and effectiveness. Firms in digital marketing and the sharing economy can leverage these insights to refine their operational strategies, thereby boosting customer satisfaction and operational performance. Researchers may find the methodological innovations and the cross-disciplinary approach of this study particularly beneficial. The successful application of these integrated models opens avenues for further investigation in various sectors requiring complex decision-making.

Future research may focus on developing more sophisticated algorithms capable of handling larger datasets and delivering real-time analytics. Extension of models to different contexts within and beyond the targeted industries to evaluate proposed frameworks' generalizability and robustness also emerges as an interesting and important direction. Additionally, this line of research may encourage more interdisciplinary

studies combining ML, OR, and other relevant fields to tackle complex systemic challenges. Further direction may include the adoption of a math-heuristic approach that combines metaheuristic search techniques with mathematical modeling, specifically an easily solvable linear relaxation, to address large-scale instances efficiently. Furthermore, the implementation of other learning-based algorithms, such as simulation optimization, may be considered to potentially enhance the predictive accuracy.

In summary, this thesis significantly contributes to the fields of machine learning and operations research by illustrating the effective fusion of these disciplines in solving pressing challenges in the digital and sharing economies. It lays the foundation for future research that could broaden the scope of applications and refine existing models, thereby pushing the boundaries of what can be achieved in these sectors.

REFERENCES

[1] E. Ascarza, S. A. Neslin, O. Netzer, Z. Anderson, P. S. Fader, S. Gupta, B. G. Hardie, A. Lemmens, B. Libai, D. Neal, *et al.*, "In pursuit of enhanced customer retention management: Review, key issues, and future directions," *Customer Needs and Solutions*, vol. 5, no. 1-2, pp. 65–81, 2018.

[2] V. V. Mišić and G. Perakis, "Data analytics in operations management: A review," *Manufacturing & Service Operations Management*, vol. 22, no. 1, pp. 158–169, 2020.

[3] A. N. Elmachtoub and P. Grigas, "Smart "predict, then optimize"," *Management Science*, vol. 68, no. 1, pp. 9–26, 2022.

[4] L. H. Liyanage and J. G. Shanthikumar, "A practical inventory control policy using operational statistics," *Operations research letters*, vol. 33, no. 4, pp. 341–348, 2005.

[5] D. Bertsimas and N. Kallus, "From predictive to prescriptive analytics," *Management Science*, vol. 66, no. 3, pp. 1025–1044, 2020.

[6] U. Sadana, A. Chenreddy, E. Delage, A. Forel, E. Frejinger, and T. Vidal, "A survey of contextual optimization methods for decision-making under uncertainty," *European Journal of Operational Research*, 2024.

[7] G.-Y. Ban and C. Rudin, "The big data newsvendor: Practical insights from machine learning," *Operations Research*, vol. 67, no. 1, pp. 90–108, 2019.

[8] A. Oroojlooyjadid, L. V. Snyder, and M. Takáč, "Applying deep learning to the newsvendor problem," *IISE Transactions*, vol. 52, no. 4, pp. 444–463, 2020.

[9] P. L. Donti, B. Amos, and J. Z. Kolter, "Task-based end-to-end model learning in stochastic optimization," in *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, (Red Hook, NY, USA), p. 5490–5500, Curran Associates Inc., 2017.

[10] V. Kumar and W. Reinartz, "Strategic customer relationship management today," in *Customer Relationship Management*, pp. 3–20, Springer, 2012.

[11] W. Reinartz, M. Krafft, and W. D. Hoyer, "The customer relationship management process: Its measurement and impact on performance," *Journal of marketing research*, vol. 41, no. 3, pp. 293–305, 2004.

[12] D. C. Schmittlein, D. G. Morrison, and R. Colombo, "Counting your customers: Who-are they and what will they do next?," *Management science*, vol. 33, no. 1, pp. 1–24, 1987.

[13] S. Borle, S. S. Singh, and D. C. Jain, "Customer lifetime value measurement," *Management science*, vol. 54, no. 1, pp. 100–112, 2008.

[14] D. A. Schweidel, P. S. Fader, and E. T. Bradlow, "Understanding service retention within and across cohorts using limited information," *Journal of Marketing*, vol. 72, no. 1, pp. 82–94, 2008.

[15] P. S. Fader and B. G. Hardie, "How to project customer retention," *Journal of Interactive Marketing*, vol. 21, no. 1, pp. 76–90, 2007.

[16] P. C. Verhoef and B. Donkers, "The effect of acquisition channels on customer loyalty and cross-buying," *Journal of Interactive Marketing*, vol. 19, no. 2, pp. 31–43, 2005.

[17] E. T. Anderson and D. I. Simester, "Long-run effects of promotion depth on new versus established customers: Three field studies," *Marketing Science*, vol. 23, no. 1, pp. 4–20, 2004.

[18] I. Bose and X. Chen, "Hybrid models using unsupervised clustering for prediction of customer churn," *Journal of Organizational Computing and Electronic Commerce*, vol. 19, no. 2, pp. 133–151, 2009.

[19] N. Glady, B. Baesens, and C. Croux, "Modeling churn using customer lifetime value," *European Journal of Operational Research*, vol. 197, no. 1, pp. 402–411, 2009.

[20] Z. Jamal and R. E. Bucklin, "Improving the diagnosis and prediction of customer churn: A heterogeneous hazard modeling approach," *Journal of Interactive Marketing*, vol. 20, no. 3-4, pp. 16–29, 2006.

[21] 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, no. 1, p. 10, 2016.

[22] M. A. H. Farquad, V. Ravi, and S. B. Raju, "Churn prediction using comprehensible support vector machine: An analytical crm application," *Applied Soft Computing*, vol. 19, pp. 31–40, 2014.

[23] A. De Caigny, K. Coussement, and K. W. De Bock, "A new hybrid classification algorithm for customer churn prediction based on logistic regression and decision trees," *European Journal of Operational Research*, vol. 269, no. 2, pp. 760–772, 2018.

[24] P. E. Pfeifer, M. E. Haskins, and R. M. Conroy, "Customer lifetime value, customer profitability, and the treatment of acquisition spending," *Journal of Managerial Issues*, vol. 17, no. 1, pp. 11–25, 2005.

[25] S. Rosset, E. Neumann, U. Eick, and N. Vatnik, "Customer lifetime value models for decision support," *Data mining and knowledge discovery*, vol. 7, no. 3, pp. 321–339, 2003.

[26] P. S. Fader and B. G. Hardie, "Customer-base valuation in a contractual setting: The perils of ignoring heterogeneity," *Marketing Science*, vol. 29, no. 1, pp. 85–93, 2010.

[27] V. Kumar and D. Shah, "Expanding the role of marketing: from customer equity to market capitalization," *Journal of Marketing*, vol. 73, no. 6, pp. 119–136, 2009.

[28] V. Kumar, M. George, and J. Pancras, "Cross-buying in retailing: Drivers and consequences," *Journal of Retailing*, vol. 84, no. 1, pp. 15–27, 2008.

[29] P. S. Fader, B. G. Hardie, and K. Jerath, "Estimating clv using aggregated data: the tuscan lifestyles case revisited," *Journal of Interactive Marketing*, vol. 21, no. 3, pp. 55–71, 2007.

[30] P. E. Pfeifer, "On estimating current-customer equity using company summary data," *Journal of Interactive Marketing*, vol. 25, no. 1, pp. 1–14, 2011.

[31] R. T. Rust, V. Kumar, and R. Venkatesan, "Will the frog change into a prince? predicting future customer profitability," *International Journal of Research in Marketing*, vol. 28, no. 4, pp. 281–294, 2011.

[32] R. Venkatesan and V. Kumar, "A customer lifetime value framework for customer selection and resource allocation strategy," *Journal of marketing*, vol. 68, no. 4, pp. 106–125, 2004.

[33] P. S. Fader, B. G. Hardie, and K. L. Lee, ""counting your customers" the easy way: An alternative to the pareto/nbd model," *Marketing science*, vol. 24, no. 2, pp. 275–284, 2005.

[34] E. Ascarza, P. S. Fader, and B. G. Hardie, "Marketing models for the customer-centric firm," in *Handbook of marketing decision models*, pp. 297–329, Springer, 2017.

[35] P. S. Fader, B. G. Hardie, and K. L. Lee, ""counting your customers" the easy way: An alternative to the pareto/nbd model," *Marketing science*, vol. 24, no. 2, pp. 275–284, 2005.

[36] A. Şahin, Z. Can, and E. Albey, "A mathematical model for customer lifetime value based offer management," in *Data Management Technologies and Applications* (J. Filipe, J. Bernardino, and C. Quix, eds.), (Cham), pp. 27–45, Springer International Publishing, 2018.

[37] V. Kumar and W. J. Reinartz, *Customer relationship management: A databased approach*. Wiley Hoboken, NJ, 2006.

[38] E. Ascarza, "Retention futility: Targeting high-risk customers might be ineffective," *Journal of Marketing Research*, vol. 55, no. 1, pp. 80–98, 2018.

[39] D. W. Marquardt, "An algorithm for least-squares estimation of nonlinear parameters," *Journal of the society for Industrial and Applied Mathematics*, vol. 11, no. 2, pp. 431–441, 1963.

[40] M. Tawarmalani and N. V. Sahinidis, "A polyhedral branch-and-cut approach to global optimization," *Mathematical Programming*, vol. 103, pp. 225–249, 2005.

[41] "Ibm cplex optimizer." `https://www.ibm.com/analytics/cplex-optimizer`. Accessed: 2018-12-25.

[42] A. Iserles, *A first course in the numerical analysis of differential equations*. Cambridge university press, 2009.

[43] "Inrix global traffic scorecard." `http://inrix.com/scorecard/`. Accessed: 2020-05-10.

[44] "Traffic's mind-boggling economic toll." `https://www.citylab.com/transportation/2018/02/traffics-mind-boggling-economic-toll/552488/`. Accessed: 2018-12-25.

[45] "Occupancy of cars and vans in england 2002-2017." `https://www.statista.com/statistics/314719/average-car-and-van-occupancy-in-england/`. Accessed: 2018-12-25.

[46] "Occupancy rates of passenger vehicles." `https://www.eea.europa.eu/data-and-maps/indicators/occupancy-rates-of-passenger-vehicles/occupancy-rates-of-passenger-vehicles/`. Accessed: 2018-12-25.

[47] N. Agatz, A. Erera, M. Savelsbergh, and X. Wang, "Optimization for dynamic ride-sharing: A review," *European Journal of Operational Research*, vol. 223, no. 2, pp. 295–303, 2012.

[48] M. Furuhata, M. Dessouky, F. Ordóñez, M.-E. Brunet, X. Wang, and S. Koenig, "Ridesharing: The state-of-the-art and future directions," *Transportation Research Part B: Methodological*, vol. 57, pp. 28–46, 2013.

[49] L. Hou, D. Li, and D. Zhang, "Ride-matching and routing optimisation: Models and a large neighbourhood search heuristic," *Transportation Research Part E: Logistics and Transportation Review*, vol. 118, pp. 143–162, 2018.

[50] J. Cramer and A. B. Krueger, "Disruptive change in the taxi business: The case of uber," *American Economic Review*, vol. 106, no. 5, pp. 177–82, 2016.

[51] "Traffic's mind-boggling economic toll." `https://www.citylab.com/transportation/2018/02/traffics-mind-boggling-economic-toll/552488/`. Accessed: 2018-12-25.

[52] B. McKenzie, *Who Drives to Work?: Commuting by Automobile in the United States: 2013*. US Department of Commerce, Economics and Statistics Administration, US ..., 2015.

[53] N. Agatz, A. Erera, M. Savelsbergh, and X. Wang, "Optimization for dynamic ride-sharing: A review," *European Journal of Operational Research*, vol. 223, no. 2, pp. 295–303, 2012.

[54] M. Nourinejad and M. J. Roorda, "Agent based model for dynamic ridesharing," *Transportation Research Part C: Emerging Technologies*, vol. 64, pp. 117–132, 2016.

[55] J. Jacob and R. Roet-Green, "Ride solo or pool: Designing price-service menus for a ride-sharing platform." Simon Business School Working Paper No: 3008136, 2018.

[56] A. Tafreshian, N. Masoud, and Y. Yin, "Frontiers in service science: Ride matching for peer-to-peer ride sharing: A review and future directions," *Service Science*, vol. 12, no. 2-3, pp. 44–60, 2020.

[57] N. Ta, G. Li, T. Zhao, J. Feng, H. Ma, and Z. Gong, "An efficient ride-sharing framework for maximizing shared route," *IEEE Transactions on Knowledge and Data Engineering*, vol. 30, no. 2, pp. 219–233, 2017.

[58] J. Long, W. Tan, W. Szeto, and Y. Li, "Ride-sharing with travel time uncertainty," *Transportation Research Part B: Methodological*, vol. 118, pp. 143–171, 2018.

[59] R. S. Thangaraj, K. Mukherjee, G. Raravi, A. Metrewar, N. Annamaneni, and K. Chattopadhyay, "Xhare-a-ride: A search optimized dynamic ride sharing system with approximation guarantee," in *2017 IEEE 33rd International Conference on Data Engineering (ICDE)*, pp. 1117–1128, IEEE, 2017.

[60] P. Kumar and A. Khani, "An algorithm for integrating peer-to-peer ridesharing and schedule-based transit system for first mile/last mile access," *Transportation Research Part C: Emerging Technologies*, vol. 122, p. 102891, 2021.

[61] J. Fan, J. Xu, C. Hou, B. Cao, T. Dong, and S. Cheng, "Uroad: An efficient algorithm for large-scale dynamic ridesharing service," in *2018 IEEE International Conference on Web Services (ICWS)*, pp. 9–16, IEEE, 2018.

[62] N. V. Bozdog, M. X. Makkes, A. Van Halteren, and H. Bal, "Ridematcher: peer-to-peer matching of passengers for efficient ridesharing," in *Proceedings of the 18th IEEE/ACM International Symposium on Cluster, Cloud and Grid Computing*, pp. 263–272, IEEE Press, 2018.

[63] C. Tian, Y. Huang, Z. Liu, F. Bastani, and R. Jin, "Noah: a dynamic ridesharing system," in *Proceedings of the 2013 ACM SIGMOD International Conference on Management of Data*, pp. 985–988, ACM, 2013.

[64] W. Herbawi and M. Weber, "The ridematching problem with time windows in dynamic ridesharing: A model and a genetic algorithm," in *2012 IEEE Congress on Evolutionary Computation*, pp. 1–8, IEEE, 2012.

[65] B. Li, D. Krushinsky, H. A. Reijers, and T. Van Woensel, "The share-a-ride problem: People and parcels sharing taxis," *European Journal of Operational Research*, vol. 238, no. 1, pp. 31–40, 2014.

[66] M. Riedler and G. Raidl, "Solving a selective dial-a-ride problem with logic-based benders decomposition," *Computers & Operations Research*, vol. 96, pp. 30–54, 2018.

[67] M. A. Masmoudi, K. Braekers, M. Masmoudi, and A. Dammak, "A hybrid genetic algorithm for the heterogeneous dial-a-ride problem," *Computers & operations research*, vol. 81, pp. 1–13, 2017.

[68] J.-F. Cordeau and G. Laporte, "The dial-a-ride problem: models and algorithms," *Annals of operations research*, vol. 153, no. 1, pp. 29–46, 2007.

[69] Z. Siddiqi and R. Buliung, "Dynamic ridesharing and information and communications technology: past, present and future prospects," *Transportation Planning and Technology*, vol. 36, no. 6, pp. 479–498, 2013.

[70] D. Sánchez, S. Martínez, and J. Domingo-Ferrer, "Co-utile p2p ridesharing via decentralization and reputation management," *Transportation Research Part C: Emerging Technologies*, vol. 73, pp. 147–166, 2016.

[71] T. L. Saaty, "What is the analytic hierarchy process?," in *Mathematical models for decision support*, pp. 109–121, Springer, 1988.

[72] J. A. Aledo, J. A. Gámez, and A. Rosete, "Partial evaluation in rank aggregation problems," *Computers & Operations Research*, vol. 78, pp. 299–304, 2017.

[73] P. S. Badal and A. Das, "Efficient algorithms using subiterative convergence for kemeny ranking problem," *Computers & Operations Research*, vol. 98, pp. 198–210, 2018.

[74] C. Dwork, R. Kumar, M. Naor, and D. Sivakumar, "Rank aggregation methods for the web," in *Proceedings of the 10th international conference on World Wide Web*, pp. 613–622, ACM, 2001.

[75] J. P. Baskin and S. Krishnamurthi, "Preference aggregation in group recommender systems for committee decision-making," in *Proceedings of the third ACM conference on Recommender systems*, pp. 337–340, ACM, 2009.

[76] N. Betzler, R. Bredereck, and R. Niedermeier, "Theoretical and empirical evaluation of data reduction for exact kemeny rank aggregation," *Autonomous Agents and Multi-Agent Systems*, vol. 28, no. 5, pp. 721–748, 2014.

[77] D. Coppersmith, L. K. Fleischer, and A. Rurda, "Ordering by weighted number of wins gives a good ranking for weighted tournaments," *ACM Transactions on Algorithms (TALG)*, vol. 6, no. 3, p. 55, 2010.

[78] N. Ailon, M. Charikar, and A. Newman, "Aggregating inconsistent information: ranking and clustering," *Journal of the ACM (JACM)*, vol. 55, no. 5, p. 23, 2008.

[79] F. Schalekamp and A. v. Zuylen, "Rank aggregation: Together we're strong," in *2009 Proceedings of the Eleventh Workshop on Algorithm Engineering and Experiments (ALENEX)*, pp. 38–51, SIAM, 2009.

[80] V. Conitzer, A. Davenport, and J. Kalagnanam, "Improved bounds for computing kemeny rankings," in *AAAI*, vol. 6, pp. 620–626, 2006.

[81] V. Pihur, S. Datta, and S. Datta, "Weighted rank aggregation of cluster validation measures: a monte carlo cross-entropy approach," *Bioinformatics*, vol. 23, no. 13, pp. 1607–1615, 2007.

[82] "Drivers around the world lose an average of 8 working days per year due to traffic congestion." `https://corporate.tomtom.com/news-releases/news-release-details/drivers-around-world-lose-average-8-working-days-year-due?ReleaseID=804498`. Accessed: 2018-12-25.

[83] J. A. Aledo, J. A. Gámez, and D. Molina, "Tackling the rank aggregation problem with evolutionary algorithms," *Applied Mathematics and Computation*, vol. 222, pp. 632–644, 2013.

[84] S. Brin and L. Page, "The anatomy of a large-scale hypertextual web search engine," *Computer networks and ISDN systems*, vol. 30, no. 1-7, pp. 107–117, 1998.

[85] A. Sinha, P. Malo, and K. Deb, "A review on bilevel optimization: From classical to evolutionary approaches and applications," *IEEE Transactions on Evolutionary Computation*, vol. 22, no. 2, pp. 276–295, 2017.

[86] V. V. Kalashnikov, S. Dempe, G. A. Pérez-Valdés, N. I. Kalashnykova, and J.-F. Camacho-Vallejo, "Bilevel programming and applications," *Mathematical Problems in Engineering*, vol. 2015, p. 310301, 2015. Sergii V. Kavun (ed.).

[87] J. C. Smith and Y. Song, "A survey of network interdiction models and algorithms," *European Journal of Operational Research*, vol. 283, no. 3, pp. 797–811, 2020.

[88] G. D. Corporation, "General Algebraic Modeling System (GAMS) Release 24.2.1." Washington, DC, USA, 2013.

VITA

Ahmet Şahin earned M.Sc. in Industrial and Systems Engineering from Istanbul Şehir University and B.Sc. in Industrial Engineering from Işık University. He is currently the Operations Manager at OzuBEX Digital Transformation Center. He has extensive experience in data science by leading projects and teams at Turkish Radio and Television Corporation (TRT), Vodafone, and ICATLabs. His expertise includes machine learning, artificial intelligence, and mathematical modelling applications in manufacturing systems and customer analytics.