June 2021

Designing Targeted Mobile Advertising Campaigns

Kimia Keshanian

University of South Florida

Follow this and additional works at: https://digitalcommons.usf.edu/etd

Part of the Business Administration, Management, and Operations Commons, Databases and Information Systems Commons, and the Library and Information Science Commons

Scholar Commons Citation

This Dissertation is brought to you for free and open access by the USF Graduate Theses and Dissertations at Digital Commons @ University of South Florida. It has been accepted for inclusion in USF Tampa Graduate Theses and Dissertations by an authorized administrator of Digital Commons @ University of South Florida. For more information, please contact scholarcommons@usf.edu.
Designing Targeted Mobile Advertising Campaigns

by

Kimia Keshanian

A dissertation submitted in partial fulfillment of the requirements for the degree of Doctor of Philosophy in Business Administration
School of Information Systems and Management
College of Business
University of South Florida

Major Professor: Kaushik Dutta, Ph.D.
Balaji Padmanabhan, Ph.D.
Manish Agrawal, Ph.D.
He Zhang, Ph.D.
Nasir Ghani, Ph.D.

Date of Approval: May 19, 2021

Keywords: mobile advertisements, feature selection, game theory, machine learning, optimization

Copyright © 2021, Kimia Keshanian
To my parents, Farideh and Farzan, and my lovely husband, Hadi
Acknowledgments

Foremost, I would like to express my sincere appreciation to my supervisor, Prof. Kuashik Dutta, for the continuous support of my Ph.D research, for his patience, motivation, enthusiasm, and immense knowledge. The opportunities that have been available to me as a research higher degree student at the University of South Florida have been endless. Prof. Dutta guidance helped me in all aspects of my research and the writing of this dissertation. I could not imagine having a better advisor for my Ph.D study. My sincere thanks also go to Dr. Daniel Zantedeschi for his support and encouragement during my Ph.D study. I am indebted to all my friends who have supported and encouraged me to strive towards my goal over the last few years. I would specially like to thank my inspiring parents, Farideh and Farzan, and my wonderful brother, Kamyar, for all of the sacrifices that they have made on my behalf. I owe a lot to them since they encouraged and helped me at every stage of my personal and academic life, and longed to see this achievement come to fruition. Finally, I would like to express my appreciation to my beloved husband, Hadi. Words cannot express how grateful I am to him. He supported me in every possible way to see the completion of this work. I am extremely happy that I was able to complete this dissertation in time before giving birth to our first child.
Table of Contents

List of Tables ............................................................................................................... v

List of Figures ............................................................................................................... vi

Abstract ..................................................................................................................... ix

Chapter 1 Introduction ............................................................................................... 1
  1.1 The curse of high dimensionality .............................................................. 2
  1.2 The curse of identification ....................................................................... 6
  1.3 Literature review .................................................................................... 8
    1.3.1 Mobile advertisements and marketing outcomes ................................ 8
    1.3.2 Effectiveness of mobile advertising campaigns ............................... 9
    1.3.3 Feature selection ........................................................................ 12
  1.4 Outline of the thesis................................................................................ 14

Chapter 2 Feature selection for high-dimensional data............................................. 15
  2.1 The proposed game-theoretic approach..................................................... 16
    2.1.1 Phase-I: computing the status quo............................................... 18
    2.1.2 Phase-II: solving $\text{GT-Con}$ ................................................... 20
    2.1.3 Phase-III: selecting the features................................................... 22
  2.2 A case study in online advertising ............................................................ 22
    2.2.1 The online display dataset .......................................................... 24
    2.2.2 Benchmarks ............................................................................... 25
    2.2.3 Results and discussion ................................................................ 25
  2.3 Computational results ............................................................................. 28
    2.3.1 Simulated datasets ..................................................................... 29
      2.3.1.1 Evaluation of simulated datasets ..................................... 29
    2.3.2 The retail dataset ....................................................................... 32
      2.3.2.1 Evaluation of the retail dataset ......................................... 33

Chapter 3 Designing efficient mobile advertising campaigns .................................. 37
  3.1 Baseline propositions ............................................................................... 39
3.2 Iterative design ....................................................................................... 40
3.3 ROI model .............................................................................................. 41
3.4 Solution methods .................................................................................... 48
    3.4.1 Optimal solution ......................................................................... 48
    3.4.2 Heuristic solution ....................................................................... 49
3.5 Case study: boosting in-store visits of a retail store ................................... 52
    3.5.1 Pilot phase ................................................................................ 53
    3.5.2 Production phases ...................................................................... 55
3.6 Managerial insights ................................................................................. 58

Chapter 4 Addressing device identity matching in mobile advertisement ........... 62
4.1 Entity matching ....................................................................................... 63
    4.1.1 Related EM methods .................................................................. 66
4.2 Problem description ................................................................................ 67
4.3 Proposed framework ................................................................................ 71
4.4 Heuristic algorithm ................................................................................. 75
4.5 Case study .............................................................................................. 76
    4.5.1 Data description ........................................................................ 77
    4.5.2 Results ...................................................................................... 79

Chapter 5 Conclusions and future research directions ........................................ 83
5.1 Conclusion of Chapter 2 .......................................................................... 83
5.2 Conclusion of Chapter 3 .......................................................................... 83
5.3 Conclusion of Chapter 4 .......................................................................... 84

Appendix 1: Proof of Theorem 2.1 .................................................................... 104
Appendix 2: Attributes to target users (Step 1) ....................................................... 108
Appendix 3: Data description ............................................................................. 113
    Appendix 3.1: Impact of advertisement deliver ..................................... 114
    Appendix 3.2: Impact of location on effectiveness of advertisement ....... 116
    Appendix 3.3: Heterogeneity due to Additional Attributes .................... 116
Appendix 4: Store Visit Prediction Model (Step 2) ............................................. 118
Appendix 5: Proof of Theorem 3.1 .................................................................... 122
Appendix 6: Pseudocodes ................................................................................. 126
Appendix 7: Performance of the heuristic method on the case study .................. 128
Appendix 8: The performance of the proposed VNS .................................................. 130
List of Tables

Table 2.1 – Summary of feature selection approaches................................. 24
Table 2.2 – SVM results for the online display dataset...................................... 26
Table 2.3 – RF results for the online display dataset ........................................ 26
Table 2.4 – Statistics summary of absolute SHAP values across methods........... 28
Table 2.5 – Minimum, average, and maximum of MSE for each method and α .......... 30
Table 2.6 – SVM results for the retail dataset................................................ 34
Table 2.7 – RF results for the retail dataset.................................................... 34
Table 3.1 – Parameters and variables descriptions ........................................ 43
Table 3.2 – Summary of results in the Pilot Phase........................................... 55
Table 3.3 – Summary results of all phases...................................................... 56
Table 3.4 – The absolute difference between the actual and predicted visit rate ...... 56
Table 3.5 – The costs, benefits, and ROI of different campaign designs.............. 58
Table 3.6 – Importance of attributes based on distance from the BM store .......... 60
Table 4.1 – Notations ..................................................................................... 74
Table 4.2 – Summary statistics of the dataset ............................................... 78
Table 4.3 – Overall performance ................................................................... 79
Table A.1 – Summary statistics .................................................................. 115
Table A.2 – Heterogeneity results ................................................................. 119
Table A.3 – SVM, LR, and RF performance for the optimal subset of features .......... 121
Table A.4 – The optimality gap of VNS for different values of parameters............... 129
Table A.5 – The performance of VNS on instances of Class I ............................. 131
Table A.6 – The performance of VNS on instances of Class II ............................. 131
<table>
<thead>
<tr>
<th>Figure</th>
<th>Description</th>
<th>Page</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.1</td>
<td>An illustration of the Nash solution</td>
<td>18</td>
</tr>
<tr>
<td>3.1</td>
<td>A simplified illustration of the decision-making process of DSPs</td>
<td>37</td>
</tr>
<tr>
<td>3.2</td>
<td>The proposed procedure</td>
<td>38</td>
</tr>
<tr>
<td>3.3</td>
<td>Costs and benefits of a mobile advertising campaign</td>
<td>41</td>
</tr>
<tr>
<td>3.4</td>
<td>Proposed Optimization Framework</td>
<td>42</td>
</tr>
<tr>
<td>3.5</td>
<td>Experimental setting</td>
<td>53</td>
</tr>
<tr>
<td>4.1</td>
<td>Entity Resolution architecture</td>
<td>68</td>
</tr>
<tr>
<td>4.2</td>
<td>Sparsity</td>
<td>70</td>
</tr>
<tr>
<td>4.3</td>
<td>The Proposed Framework</td>
<td>71</td>
</tr>
<tr>
<td>4.4</td>
<td>The Solution Approach for Clustering Stage</td>
<td>72</td>
</tr>
<tr>
<td>4.5</td>
<td>SHAP values in the EM model</td>
<td>82</td>
</tr>
<tr>
<td>A.1</td>
<td>Distribution of users from MyStore</td>
<td>113</td>
</tr>
<tr>
<td>A.2</td>
<td>Distribution of users who visited MyStore</td>
<td>116</td>
</tr>
<tr>
<td>A.3</td>
<td>The SHAP variable importance results</td>
<td>118</td>
</tr>
</tbody>
</table>
Abstract

With the proliferation of smart, handheld devices, there has been a multifold increase in the ability of firms to target and engage with customers through mobile advertising. Therefore, not surprisingly, mobile advertising campaigns have become an integral aspect of firms’ brand building activities, such as improving the awareness and overall visibility of firms’ brands. In addition, retailers are increasingly using mobile advertising for targeted promotional activities that increase in-store visits and eventual sales conversions. However, in recent years, mobile or in general online advertising campaigns have been facing one major challenge and one major threat that can negatively impact the effectiveness of advertising campaigns. The challenge is the curse of high dimensionality while the threat is the curse of identification. We refer to the former as a challenge as it currently exists as a major problem, but we use the term threat for the latter as it is becoming a major problem.

The main focus of this dissertation is on resolving the curse of high dimensionality due to it “current” status. However, this dissertation also includes some solutions for addressing the curse of identification too. The curse of high dimensionality is an issue in online advertising domain that refers to the fact that personalized advertisement is obtained with the help of learning predicative models trained on a large number of users’ specific features to predict a user outcome (e.g., will the user buy our product or not after seeing the ad) for each available decision (e.g., should the ad agency expose the user or not). The curse of high dimensionality is not limited to online advertising datasets, the same issue exists in many other Information Systems (IS) related datasets such as online marketing, healthcare, finance (such as bank marketing), social media as well. A main solution to resolve the curse of high dimensionality is selecting the best subset of features/attributes. This is because identifying
the most relevant features are crucial for knowledge discovery and building generalizable and accurate models.

While selecting the best subset of features sounds like a classical problem, it comes with two unique characteristics in the online advertising domain compared to other IS domains. In other domains, firms do not usually need to buy data because it usually can be collected automatically. However, in online advertising domain, firms have a limited budget and the features are either costly to be purchased or costly to be computed. Therefore, if a firm invests in collecting data to run a campaign but does not gain any benefits, then that investment was unnecessary. With this mind, we propose two solution approaches for selecting the best subset of features in this dissertation. One is suitable for cases where the cost of all features are the same, and therefore the focus is on minimizing the number of futures while keeping the accuracy at the highest possible. The novelty of our first approach is that it employs the concept of Nash bargaining optimization in the field of cooperative game theory to solve the best subset selection problem in advertising domain. Our second approach is suitable for cases where the cost of features are not the same and there are limited budgets. We show that for such cases, it is significantly more efficient if users are first geographically partitioned into groups and then for each group different subsets of attributes to be selected. In order to select the best subset of features in all groups at the same time considering the budget constraint, we propose a novel optimization model and algorithm.

Regarding the curse of identification, it is important to first note that in online advertising ecosystems, collecting raw data about users are dependent on the ability to track them. The main tool for so doing is the so-called Device Identifier (device ID), assigned by Apple and Android to identify every individual smartphone in the world. So, the curse of identification means that the access to this valuable tool of obtaining data is become more and more restricted over time. Therefore, identifying appropriate way to create accurate profiling of users who are the recipients of targeted advertisement seems integral for ad agencies.
Without finding appropriate way to profile users, advertisers cannot send relevant ads to users and consequently it can damage their brand reputation and waste their financial resources. Therefore, due to less accessibility to users device IDs, we propose an optimization framework for identifying unique devices in online advertising ecosystems. The hope is that by applying the framework helps ad agencies to target right customers and help mobile advertising market to continue generate revenues.
Chapter 1: Introduction

There has been a surge in the dimension of data-driven decision support systems. The growing availability of data provides a unique opportunity to model and forecast economic phenomena that was previously difficult to perform [6]. For instance, in online advertising, due to ubiquitous access of customers to mobile devices and the proliferation of mobile apps, companies can virtually reach customers anytime and anywhere. This is further highlighted by an observation that more than 50% of internet advertising originates from mobile apps [100]. Mobile advertising campaigns have become an integral aspect of firms’ brand building activities, such as improving the awareness and overall visibility of firms’ brands [11, 86]. In addition, retailers are also increasingly using mobile advertising for targeted promotional activities that increase in-store visits and eventual sales conversions [117].

In online advertising one of the most significant concepts in recent years is Real Time Bidding (RTB) or the programmatic delivery of advertisements. RTB helps advertisers to run their advertising campaigns in a cost-effective manner and serve ads to the right person and at a right time [1, 87]. Through RTB advertisers can target both context and specific users which can result in higher return on investment [76]. RTB ecosystem consists of three main platforms including a Demand Side Platform (DSP), a Supply Side Platform (SSP), and a RTB ad exchange. In programmatic advertising setup, advertisers send ad impressions to mobile users through DSPs, which manage ad campaigns and bid on impressions on the behalf of advertisers. DSPs bid on impressions based on data sent from RTB ad exchanges, and user data that is collected via Data Management Platforms (DMPs) such as Mapp Digital and Oracle BlueKai [137].
By gathering user-specific data such as the details about the devices and apps, users’ locations, and their demographics information, companies can offer more personalized and directed promotions to customers and achieve higher conversions (number of clicks or in-store visits). Although mobile advertising campaigns have become an integral aspect of firms’ brand building activities or targeted promotional activities, in recent years designing effective campaigns has been facing one major challenge and one major threat:

- **Challenge:** the curse of high dimensionality
- **Threat:** the curse of identification

We refer to the former as a challenge as it currently exists as a major problem, but we use the term threat for the latter as it is becoming a major problem. The main focus of this dissertation is on resolving the curse of dimensionality due to it “current” status. However, this dissertation also includes some solutions for addressing the curse of privacy too.

### 1.1 The curse of high dimensionality

Although the positive impact of features such as users’ locations and users’ demographics information has been tested in various studies [9, 53, 98, 140], the acquisition costs of these features are considerable. In other words, while collecting, gathering, and computing user data from DMPs and RTBs improve chance of targeting the right user and consequently enhance campaign performance, acquiring such data is often expensive for DSPs. Computing features from dynamic streams of advertisements request in the DSP needs robust computational cloud services, which is highly costly. This is because a typical DSP gets 200K-500K bid requests per second on average. Therefore, managing and processing such a large volume of data are expensive for DSP [4].

In light of the above, the curse of high dimensionality refers to the existence of large number of costly users’ specific features. A solution to resolve the curse of high dimensionality is to develop feature selection approaches for determining the most relevant and promising
features, referred to as the Best Subset Selection Problem (BSSP). Given the ever-increasing operational costs and the competitive pressure, a main challenge of DSPs is to select features that can increase conversion rates while considering the costs of computing and collecting those features. Additionally, from the perspective of ad platform’s users, it is not clear when to use which additional data points to earn the appropriate return by the targeted users. Indeed, if a DSP invests in collecting other data to run a campaign but does not gain any additional conversions, then that investment was unnecessary. Since the number of available features is often huge in the mobile advertising platform (e.g., we have worked with DMPs selling 80+ attributes), identifying the most relevant features seems a necessary task for ad platforms. Moreover, DSPs have a limited budget for running advertisement campaigns. Therefore, they should select the most predictive subset of costly features that result in higher conversion rates and, consequently, return on the investment amount.

Note that at a first glance the BSSP is similar to a problem known as Active Feature-Value Acquisition Problem (AFVAP) [127]. However, the BSSP and the AFVAP are quite different. In the AFAP, it is assumed that there are some not-readily available data points. In other words, for each feature, some data points may be missing. So, the AFVAP seeks to determine which missing data points (rather than features) need to be purchased. The chosen (i.e., purchased) data points should increase the prediction accuracy at a low cost [73, 91]. However, in the BSSP, all data points related to each feature are missing unless the feature is purchased. In other words, in the BSSP, by purchasing a feature, all its corresponding data points will be available. Due to this main difference between the BSSP and the AFVAP, it is not trivial how the existing AFVAP approaches can be applied to solve the BSSP as in the existing AFVAP approaches readily available data points of a feature are used to estimate the conditional distribution for each missing value. However, in the BSSP, since there is no beforehand information about data points of a feature, obtaining such distributions are not trivial.
In light of the above, in this dissertation, we develop two novel techniques for resolving the curse of high dimensionality through developing novel algorithms for solving two different variations of the BSSP:

- In Chapter 2, we develop a novel solution approach by using ideas from the field of cooperative game theory for solving a variation of the BSSP that all features have the same acquisition and storage cost for a DSP. Similar to many existing methods, our approach belongs to the class of linear-regression based techniques. This implies that our approach attempts to select the best subset of features by fitting a linear regression score (on a training set). However, in this process, minimizing the amount of error, i.e., MSE, is important, and minimizing the number of features is crucial. So, our method attempts to solve a bi-objective optimization problem that concerns reducing the amount of error and the number of features simultaneously. To solve this bi-objective optimization problem, we employ a well-known concept in game theory, the so-called Nash bargaining solution [114]. A bargaining problem is a cooperative game in which all players agree to create a grand coalition to obtain a higher payoff [129]. Specifically, we assume that each objective function of the BSSP belongs to an imaginary player, i.e., the first player is interested in minimizing the amount of bias and the second player is involved in minimizing the number of predictors. Thus, our method finds a solution by creating a coalition between these two imaginary players using the Nash bargaining solution concept.

Our proposed method supports ad agencies to deliver more accurate and reliable predictions and decrease their cost by only collecting the most relevant data. To examine the proposed method’s effectiveness in the online advertising domain, we use a real rich dataset related to display ads and customer conversion in this section. The dataset is obtained from the Wharton Customer Analytics Initiative sponsored by Annalect, a leading global advertising and marketing communication services company. Through this case study, we show the efficacy of our proposed method compared to six established feature selection methods including Minimum Redundancy Maximum Relevance (mRMR), Backward Selection (BS),
Recursive Feature Elimination (RFE), LASSO, ElasticNet (ENET), and Sure Independence Screening (SIS). These feature selection techniques are selected for comparison because they are well-known methods in data mining and machine learning literature [59, 48, 101]. The case study results show that the proposed method outperforms all other benchmark methods in terms of accuracy metrics.

- In Chapter 3, we develop a novel solution approach a variation of the BSSP that features can have various acquisition and storage cost for a DSP. In many practical mobile advertising campaigns, the various attributes (features) that may be used in targeting an advertisement have different data management and acquisition costs. Each of these attributes would also have different success rates in identifying (targeting) users who will provide the expected return from the advertisement. Leveraging an user’s physical location as the primary and inexpensive attribute to select the bid requests to drive in-store visit to BM store is a common scenario [51, 71]. However, additional attributes such as demographic information, customer preferences, and prior store visits (which are usually more costly to be obtained), may also influence the placement and effectiveness of the mobile advertisement.

While location-based targeting of consumers is recognized as a useful strategy by researchers and practitioners, campaign designs that programatically address the heterogeneous effects of targeting attributes as well as their potential interaction effects with location in influencing in-store visits have not been systematically studied in prior work. Therefore, in Chapter 3, we build on two propositions: (1) the impact of location-based targeted advertisements on the probability of a user’s in-store visit declines as the distance between the store and the focal user’s physical locations increase, and (2) the impact of other targeting attributes, such as demographic information, on the probability of in-store visits depend on the distance between the retail store’s and the user’s physical locations. A key implication of these empirically-validated propositions is that a campaign design strategy that statically utilizes all the available data attributes or an adhoc subset of attributes for targeting would be suboptimal. Rather, the selection of attributes to utilize for targeting should be guided
by an analysis of the interaction effects between location and other data attributes and a careful assessment of the costs and benefits (i.e., a return-on-investment (ROI) perspective). We incorporate such an analysis through an ROI model, offer robust procedures to derive solutions, and illustrate the framework’s application through a real-world case study. The developed framework advocates an iterative design approach wherein the outcome of a campaign stage is utilized to refine the prediction model and the appropriate set of attributes rebuilt and reapplied for continuous improvement of ROI.

We demonstrate the utility of the proposed campaign design framework through a case study of a real-world promotional campaign undertaken by a global retailer for one of its largest stores in Asia. The case study results show strong empirical support for our design propositions and illustrate importance of utilizing the location-contingent effects of attributes for achieving campaign goals. The case study reiterates that location information is of paramount importance for effective campaigns and that other attributes must be carefully selected. In the case of our research site, demographic and customer profiles were important targeting attributes when customers were near to the store, but prior purchasing history and competitor visits were better predictors for customers who were farther from the store. Furthermore, our results support a multi-stage campaign design approach and reveal that the optimal set of attributes to be selected for an effective campaign stabilizes after a few iterations, which limits model retraining costs and maximizes overall ROI.

1.2 The curse of identification

As it was mentioned earlier, in mobile advertising domain one of the most significant concept in recent years is RTB [146]. RTB, as a promising business model of mobile advertisement market, represents autonomous and algorithmic-driven framework which completes a full transaction in milliseconds [1, 147]. RTB uses powerful user-tracking properties [124]. By employing user-tracking properties, the RTB ecosystem provides the opportunity and in-
rastructure for advertisers to target the right person in the right context at the right time [142].

To track users and deliver customized ads, an RTB uses what is called as Device Identifier (device ID), assigned by Apple and Android to identify every individual smartphone in the world [75]. Although trackability of users (by using device IDs) gives the advertisers the opportunity to target users more directly and personally, RBTs do not always have an access to users’ device IDs. One of the main reasons for that is privacy and data protection concerns [14]. Due to privacy risks, privacy advocates believe that appropriate data protection and privacy safeguard for using users’ device IDs must be guaranteed [14]. For this purpose, before tracking users, an RBT requires to have users’ approval, otherwise, it cannot have an access to users’ device IDs.

While the issues around privacy are becoming more evident over time, identifying appropriate way to create accurate profiling of users who are the recipients of targeted advertisement seems integral for ad agencies. The mobile advertising industry is an essential part of digital marketing since it provides the financial incentives for all participants in a RBT ecosystem including advertisers [88]. However, without finding appropriate way to profile users, advertisers cannot send relevant ads to users and consequently it can damage their brand reputation and waste their financial resources. Therefore, due to privacy issues and less accessibility to users device IDs, in this dissertation we also propose a framework for identifying unique devices in RBT ecosystem. In other words, the goal of Chapter 4 of this dissertation to resolve the curse of identification. Applying the proposed framework not only provides a privacy-aware market (by not using devices IDs to track customers) but also it helps ad agencies to target right customers and help mobile advertising market to continue generate revenues.
1.3 Literature review

In this section, we provide an overview of the research literature on mobile advertising and related methods. We first describe the broader studies that have examined the use of mobile advertising for brand building and sales conversion. Then, we present insights from prior studies that focus on the effectiveness of Internet- and mobile-based advertisement campaigns. Finally, we discuss about methods and algorithms that have been introduced to solve challenges in online advertising domain.

1.3.1 Mobile advertisements and marketing outcomes

Leveraging the proliferation of smart, hand-held devices, firms have utilized mobile advertising for both brand building and sales conversion. For example, Barwise and Strong [11] examined post-campaigns and phone interview metrics and reported that advertising based on mobile Short Messaging Services (SMS) created better brand awareness and stronger brand attitudes. Similarly, other studies have focused on conversion goals of promotional activities undertaken by online stores and have investigated the association between mobile display ads and outcomes, such as the click-through rate [58, 105]. Ghose et al. [58] found that a mix of web and mobile display advertising triggers more clicks and purchases, compared to using either form of advertisement exclusively. Molitor et al. [105] examined the underlying effect of context-specific factors such as geographical location and time on consumers’ coupon choice behavior. Their findings showed a co-location effect of consumers on the use of mobile coupons. Due to commonalities in consumers’ preferences, the authors indicated that sending out mobile coupons to consumers who use the same mobile app in similar locations at the same time can provide higher sales conversion.

To provide higher sales conversion, advertisers also use in-app advertising programs which help them to target users directly and deliver higher quality ads [69]. This has motivated several researchers to study this topic in recent years. For instance, Hao et al. [64] extended
the two-sided market model of in-app advertising, with advertisers and users being the two sides of the market, to capture the unique characteristics of the in-app advertising market.

In another study, Ji et al. [69] applied a game theoretical approach to study a joint advertising problem for platform participants including both platform owners and app developers. Using a differential game theoretical model, they addressed important considerations, such as the optimal conditions for the adoption of in-app advertisements by platform participants and adjustments that can be made to the advertising campaign based on the impacts of these ads.

Another outcome metric that has been examined by scholars for online stores is the redemption period for SMS coupons [32, 98]. Luo et al. [98] showed that while same-day mobile coupons were effective for proximal targeting, next-day coupons were more effective when the targeting was non-proximal. Danaher et al. [32] found that location and time of delivery play important roles in the redemption rate of SMS coupons. They also suggested that shorter expiration periods for mobile coupons can increase redemption likelihood by intensifying the time urgency of the promotion.

Our literature review also revealed that a majority of the studies examining the association between mobile ads and sales conversion have focused on online stores. Examining the impact of mobile advertising in a physical BM store setting is often more challenging because of the difficulties involved in tracking customer pathways in the sales funnel. Therefore, in this dissertation we not only consider mobile advertising challenges for online stores but also we address the rising trend in the retail industry to utilize mobile advertising for improving in-store visits and customer engagements in BM stores [117, 85, 52].

1.3.2 Effectiveness of mobile advertising campaigns

As shown in earlier studies (e.g., [82, 62, 81, 49, 34]), the modalities and effectiveness of Internet-based advertising campaigns have been important and productive areas of research
at the interface of information systems and other disciplines such as operation management and marketing.

Earlier studies have used optimization approaches for examining diverse issues and trade-offs related to online advertisements. For example, Mookerjee et al. [110, 109] showed how to optimize the decisions of an Internet advertising firm regarding the decision to display online ads to a user. They also developed a model to manage Internet ads more appropriately so that firm-level revenues are maximized while considering other constraints related to boosting click-through rates. Liu and Mookerjee [93] has developed operational strategy for internet advertising by comparing traffic based and spending based campaigns. In another study, Balseiro et al. [8] examined ad exchanges and developed a multiobjective stochastic control approach for deriving an efficient policy for allocation of online ads. In the online advertising domain, Balseiro and Candogan [7] also studied the role of contracting policies between advertisers and intermediaries who run advertising campaigns on behalf of clients. Specifically, they proposed a mechanism design model to study the optimal contract offered by the intermediary to an advertiser in a setting in which the advertiser’s budget and targeting criteria are private.

Nobibon et al. [118] proposed an optimization model for the selection of sets of customers who receive an offer for one or more products during a promotion campaign. They showed that the problem was NP-hard, and they developed a heuristic approach to derive optimal solutions. Rokach et al. [126] used a different approach to solve a similar problem. Specifically, they presented a new active learning framework, named Pessimistic Active Learning (PAL), for selecting customers using confidence intervals. In the PAL approach, however, the costs related to the acquisition and selection of the effective data attributes are not considered while assessing the effectiveness of an advertising campaign.

With the emergence of smart mobile devices and the enhanced capability of users to easily access Internet content through mobile devices, it is not surprising that mobile advertising has become an important topic of research in the Internet-based advertising domain.
Given users’ ubiquitous access to digital information and mobile apps, companies need to develop new advertising approaches to leverage the emerging patterns of customer consumption [43]. For this purpose, on behalf of advertisers, DSPs need to employ features (such as location, demographics, customers’ profiles) to deliver highly personalized ads. To deliver such deeply personalized marketing materials, DSPs can use a RTB system (also known as the programmatic delivery of advertisements). Using real-time pricing and bidding, a RTB enables advertisers to show their ads on publishers websites or apps through a complex ecosystem. For comprehensive overview of RTBs ecosystem, interested readers may refer to [143] and [3].

Due to the importance of customers related features in RTB system, for some features DSPs need to execute costly computations (typically using cloud computing) and for some others DSPs need to buy them through DMPs. However, the cost of acquiring, processing and operationalizing the data features is often expensive. This is because A typical DSP receives bid request at the rate of 200K-500K requests/seconds. Thus, the computation of data features for a such large volume of data would require considerable storage space, and management which is costly. Moreover, DMPs (such as Lotame) charges usually changes in the range of $1.00 – $1.50 per user [95]. Therefore, by increasing the number of customers profile dimensions, the cost of purchasing features from DMPs increases substantially as well.

Given high operational costs of acquiring and processing data features, a main challenge of a DSP is to select the most predictive subset of costly features that result in higher conversion rates and, consequently, return on the investment amount. Furthermore, the number of available features in the mobile advertising datasets is usually high. Therefore, since DSPs have a limited budget for running advertisement campaigns, identifying the most relevant features among available huge number of candidates can help DSPs to save a lot of money. In the literature, identifying the most the most relevant and promising features from a large set of candidates, referred to as the BSSP. Although different methods have been developed to deal with BSSP they typically choose either a vast number of attributes that
are not necessarily good in terms of acquisition costs or a tiny subset of attributes decreasing
the overall accuracy. In the next section, we provide an overview of some well-known feature
selection method and explain how our method differentiate these methods.

1.3.3 Feature selection

With the ever-increasing amount of data collected from the Internet and many other real-
world sources [74], there has been a surge in the dimension/size of datasets with information
coming from smartphones, cameras, wireless sensory networks, social media, and Internet
search [60]. Data mining and machine learning are concerned with discovering patterns from
these large datasets and making better predictions. Although the availability of massive data
can provide more opportunities to discover new knowledge, most of these datasets have the
high dimensionality problem. In high dimensional datasets, because of the excessive number
of features, the training process of machine learning models is significantly higher [148].
Therefore, due to the high dimensionality problem extracting new patterns and knowledge
creates significant challenges for the development of machine learning models [46].

In the literature, one of the well-known methods to handle the high dimensionality prob-
lem is feature selection which plays pivotal role in utilization of massive datasets along with
building accurate predictive models. Applying the proper feature selection method (1) re-
duces the effects of irrelevant and redundant variables [150], (2) prevents the predictive model
from overfitting, thus increasing the model accuracy on the new (test) datasets [44, 102], (3)
reduces both storage requirement and computing resources [89], (4) reduces computational
cost and increasing learned models performances [145, 149, 150], and (5) reduces the cost
of measuring the features to make operational learned models [21]. Due to these benefits,
developing methodologies for addressing feature selection is still an important research area
[46, 150].

From the benefits mentioned above, two of them, i.e., the first and last ones, are particu-
larly important in the online advertising domain because features are often costly to obtain
and have a large number of them typically resulting in over-fitting issues in practice [12, 130]. Despite the importance of feature selection in the online advertising domain, the literature is suffering from the lack of effective generic techniques for solving BSSP. Existing studies typically employ linear-regression based techniques for solving BSSP, e.g., LASSO [136], in the context of online advertising [12, 122, 135].

Overall, feature selection methods can be classified into four main categories including Filter, Wrapper, Embedded, and screening-based methods [20, 37]. Filter methods, e.g., mRMR [121], are directly linked to statistical methods using thresholds for selection with the independence of the induction algorithm [20, 37]. Wrapper methods, e.g., BS and RFE [63], are iterative procedures and use a learning algorithm to evaluate the accuracy produced by the selected features [83]. Embedded methods, e.g., LASSO and ENET [151], combine both Filter and Wrapper methods and make feature selection by optimizing an objective function during the process of training [66]. Finally, screening-based techniques, e.g., SIS [47], rank features instead of performing model selection itself. By ranking features, the candidate set for building training models will be reduced. For an additional review of feature selection methods, interested readers may refer to [37, 63, 134].

Although above-mentioned methods have been widely used in the literature [84, 101], they typically choose either a vast number of attributes that are not necessarily good in terms of acquisition costs or a tiny subset of attributes decreasing the overall accuracy. For instance, Filter methods may choose redundant subsets of features [27]. Wrapper methods on the other hand need a large amount of computations to obtain the feature subset. For each candidate subset, a new model needs to be trained and tested to obtain the model accuracy. Therefore, by increasing number of features, the number of iterations required to obtain the best feature subset increases exponentially as well [27, 5]. Conversely, embedded methods are less computationally expensive; however, since they are very specific to the learning algorithm, they are limited in terms of generalization [16]. To overcome these challenges, in this dissertation, we develop new feature selection methods for solving BSSP
in online advertising domain. However, our proposed methods can also be applied in other IS-related domains such as online retail, healthcare, marketing, and social media that suffer from high-dimensionality problem.

1.4 Outline of the thesis

The remaining content of this thesis is organized as follows:

• In Chapter 2, feature selection for high-dimensional data is explained.

• In Chapter 3, designing efficient mobile advertising campaigns is discussed.

• In Chapter 4, device fingerprinting approach for addressing device identity matching in mobile advertisement is explained.

• Finally, in Chapter 5, conclusions and future research directions are discussed.
Chapter 2: Feature selection for high-dimensional data

We start this chapter by formally defining the BSSP, i.e., finding the best subset of \( p \) predictors given \( n \) observations, in the context of linear regression. Let \( \mathbf{x} \in \mathbb{R}^p \) be a (column) vector representing the value of independent variables and \( \mathbf{\beta} \in \mathbb{R}^p \) be a (row) vector representing the regression coefficients. We represent a linear regression model by \( \mathbf{\beta} \mathbf{x} \) where we assume that \( x_1 \) is always equal to 1, i.e., the first feature is assumed to be the intercept. In linear regression modeling, \( \mathbf{\beta} \) is assumed to be unknown and should be estimated. We denote by \( \hat{\mathbf{\beta}} \in \mathbb{R}^p \) an estimate of \( \mathbf{\beta} \) that will be obtained based on \( n \) observations. For each observation \( i \in \{1, \ldots, n\} \), the vector of values of independent variables is denoted by \( \mathbf{x}_i \) and its corresponding dependent variable is denoted by \( y_i \). Based on these notations, a BSSP can be stated as the following bi-objective integer optimization model [28],

\[
(\text{BSSP-Int}) \quad \min_{\hat{\mathbf{\beta}} \in \mathbb{R}^p} \{ \sum_{i=1}^{n} (y_i - \sum_{j=1}^{p} x_{ij} \hat{\beta}_j)^2, \sum_{j=1}^{p} r_j \}
\]

such that: \( r_j l_j \leq \hat{\beta}_j \leq r_j u_j \) for \( j = 1, \ldots, p \),

\[
r_j \in \{0, 1\} \quad \text{for} \quad j = 1, \ldots, p,
\]

where \( l_j \in \mathbb{R} \) and \( u_j \in \mathbb{R} \) are given parameters. They are basically a lower bound and an upper bound for \( \hat{\beta}_j \) for each \( j \in \{1, \ldots, p\} \), respectively. Also, \( r_j \) for each \( j \in \{1, \ldots, p\} \) is a binary decision variable that takes the value of one if feature \( j \) is selected/active, i.e., \( \hat{\beta}_j \neq 0 \). The first objective function simply minimizes the bias and the second objective function minimizes the number of features. The constraints ensure that if \( r_j = 0 \) then \( \hat{\beta}_j = 0 \). In practice solving BSSP-Int is challenging because of several reasons: (1) the
existence of binary decision variables in BSSP-Int and (2) finding proper values for \( l_j, u_j \in \mathbb{R} \) is not a trivial task for each \( j \in \{1, \ldots, p\} \). So, in practice, it is typical to use the \( l_1 \)-norm instead of the second objective function and construct the following bi-objective continuous optimization model,

\[
\text{(BSSP-Con)} \quad \min_{\hat{\beta} \in \mathbb{R}^p} \left\{ \sum_{i=1}^n (y_i - \sum_{j=1}^p x_{ij}\hat{\beta}_j)^2, \sum_{j=1}^p |\hat{\beta}_j| \right\}.
\]

This problem has two conflicting objectives, i.e., minimizing the bias and minimizing the number of features, respectively. Therefore, due to conflicting objectives, there may exist many (and possibly an infinite set of) Pareto-optimal solutions as BSSP-Con only involves continuous variables. Note that a solution that none of its objective values can be improved individually without making the value of at least one of its other objectives worse is called a Pareto-optimal solution. With this in mind, the main challenge when solving BSSP-Con is how a desirable Pareto-optimal solution should be computed. One simple way that is used, for example, in LASSO is to add the second objective function with some penalties to the first objective function. However, in this study, we use a novel technique by utilizing some ideas from the field of game theory.

### 2.1 The proposed game-theoretic approach

Our proposed idea in this study is based on a game-theoretic approach. In machine learning, there have been many efforts in using game-theoretic approaches to quantify the importance of features to a model [33, 80, 125]. The importance of a feature can be determined thorough (for example) estimating what is called as the Shapley values [33, 96]. Shapley values have been widely used to measure the influence of individual features on the prediction of a machine learning model [138, 17]. Therefore, game-theoretic approaches have become popular as a way to provide better explanation to machine learning models [80]. In this study, however, instead of focusing on providing more explainable models, we use a game-theoretic approach.
to select the best subset of features. Identifying the best subset of features among all feature candidates is crucial in machine learning since it helps us to build more generalizable and accurate models.

The underlying idea of the proposed technique is based on a well-known game-theoretic concept known as Nash social welfare. Nash [129, 114, 115] introduced this concept for solving the so-called *bargaining problem* which is a cooperative game in which all competing players agree to create a grand coalition to obtain a higher payoff. Since the agreement of all players is necessary in a bargaining problem, it is important to inform players about their payoffs under coalition. Nash showed that to create a coalition, the payoff of players can be obtained by maximizing the Nash social welfare.

In light of the above, to solve the **BSSP-Con**, we assume that there are two independent imaginary players that one is interested in only minimizing the first objective function of **BSSP-Con** and the other attempts to minimize the second objective function of **BSSP-Con**. Suppose, for instance that two players want to create a coalition to obtain higher combined payoffs than by acting independently. We denote by $q_1$ and $q_2$ the payoffs (or objective values) of the players if they do not create the coalition, i.e., the status quo. We later in Section 2.1.1 explain how $q_1$ and $q_2$ can be initialized. In the meantime, by assuming that $q_1$ and $q_2$ are known, the Nash solution for the bargaining problem can be obtained by solving the following optimization model,

\[
\text{max} \quad \sum_{\beta \in \mathbb{R}^p} [q_1 - \sum_{i=1}^n (y_i - \sum_{j=1}^p x_{ij} \hat{\beta}_j)^2] \left[ q_2 - \sum_{j=1}^p |\hat{\beta}_j| \right]
\]

such that:
\[
\sum_{i=1}^n (y_i - \sum_{j=1}^p x_{ij} \hat{\beta}_j)^2 \leq q_1,
\]
\[
\sum_{j=1}^p |\hat{\beta}_j| \leq q_2,
\]

where the objective function in **GT-Con** maximizes the product of the individual gains of the players. The objective function is also referred as *Nash Social Welfare*. The *gain* for
each player is defined as the difference between the objective values of the player under the coalition and the no-coalition scenarios. We note that the payoffs for each player under a coalition cannot be worse/larger than the the objective value of the player under no-coalition. Thus, the incentive to form a coalition is captured by the constraints in $\text{GT-Con}$.

Figure 2.1 – An illustration of the Nash solution

Figure 2.1 offers a pictorial description of the proposed approach. The status quo of the game, i.e., $(q_1, q_2)$, is assumed to be known and is represented by a triangle in the figure. The curve represents the image of all Pareto-optimal solutions of $\text{BSSP-Con}$ in the objectives’ space, the so-called Pareto-optimal frontier. Geometrically speaking, the proposed optimization model, $\text{GT-Con}$, attempts to find a Pareto-optimal solution that the area of the box between its image and the status quo is maximized.

In light of the above, our proposed approach proceeds into three phases to solve $\text{BSSP-Con}$. (1) The first phase is to compute the status quo. (2) The second phase is to solve $\text{GT-Con}$. Finally, the last phase (3) is to select the best subset of features based on the outcome of the second phase. Next, we explain all three phases in detail.

2.1.1 Phase-I: computing the status quo

For the first imaginary player the status quo can be defined as the worst possible value for the first objective function. This can be achieved by assuming that no feature (including the
intercept) is selected. In other words, we can set \( \hat{\beta}_i = 0 \) for all \( i = 1, \ldots, p \). Hence, in that case, we have that,

\[
q_1 = \sum_{i=1}^{n} (y_i - \sum_{j=1}^{p} x_{ij}\hat{\beta}_j)^2 = \sum_{i=1}^{n} y_i^2.
\]

To compute the status of the second imaginary player we can proceed in similar fashion, i.e., computing the worst value for the second objective function. Unfortunately, this cannot be done directly because the second objective function is unbounded when being maximized. However, we observe from Figure 2.1 that the top endpoint of the curve, i.e., Pareto-optimal frontier, has the minimum possible value for the first objective function and at the same time it has the worse possible value for the second objective function (for the points in the frontier). So, we can employ this observation and take two steps to compute \( q_2 \). In the first step, we compute the minimum possible value for the first objective function, denoted by \( m_1^* \), by solving,

\[
m_1^* = \min_{\beta \in \mathbb{R}^p} \sum_{i=1}^{n} (y_i - \sum_{j=1}^{p} x_{ij}\hat{\beta}_j)^2.
\]

We note that this optimization problem is convex and can be solved by commercial solvers such as CPLEX or Gurobi. Thus, in the second step, we compute the value of the second objective function by considering the following optimization problem,

\[
m_2^* = \min_{\beta \in \mathbb{R}^p} \sum_{j=1}^{p} |\hat{\beta}_j|
\]

such that:

\[
\sum_{i=1}^{n} (y_i - \sum_{j=1}^{p} x_{ij}\hat{\beta}_j)^2 \leq m_1^*.
\]

Importantly, we note that although the constraint of the above model is convex, its objective function is not. So, we cannot solve the problem directly using CPLEX or Gurobi. To address
this concern, we consider an alternative equivalent formulation:

\[ m_2^* = \min_{\beta \in \mathbb{R}^p} \sum_{j=1}^{p} \hat{r}_j \]

such that:

\[ \sum_{i=1}^{n} (y_i - \sum_{j=1}^{p} x_{ij} \hat{\beta}_j)^2 \leq m_1^* \]

\[ \hat{\beta}_j \leq \hat{r}_j \quad \forall \ j \in \{1, \ldots, p\}, \]

\[ -\hat{\beta}_j \leq \hat{r}_j \quad \forall \ j \in \{1, \ldots, p\}, \]

\[ \hat{r}_j \geq 0 \quad \forall \ j \in \{1, \ldots, p\}, \]

where \( \hat{r}_j \) is a non-negative continuous decision variable for each \( j \in \{1, \ldots, p\} \). Note that for any feasible solution, the model guarantees that \( \hat{r}_j \geq |\hat{\beta}_j| \) for each \( j \in \{1, \ldots, p\} \). However, the objective function is in the form of minimization and hence for any optimal solution we must have that \( \hat{r}_j = |\hat{\beta}_j| \) for each \( j \in \{1, \ldots, p\} \). Now, we propose to set

\[ q_2 = p \times m_2^*. \]

Note that by construction, we could set \( q_2 = m_2^* \). However, the issue is that our proposed game-theoretic approach, i.e., \( \text{GT-Con} \), is designed for solving \( \text{BSSP-Con} \) and not \( \text{BSSP-Int} \). Therefore, \( m_2^* \) can be too small when being used in \( \text{GT-Con} \) and may possibly remove many feasible solutions. So, we propose to enlarge the value of \( q_2 \). Overall, for larger values of \( p \), it is more likely that \( m_2^* \) not to be sufficiently large when employed as \( q_2 \) in \( \text{GT-Con} \). That is the reason that we propose to set \( q_2 = p \times m_2^* \).

2.1.2 Phase-II: solving \( \text{GT-Con} \)

After computing the status quo, the subsequent phase is to solve \( \text{GT-Con} \). The optimization model is again not convex and needs to be reformulated in order to be solved by commercial solvers. The following theorem is helpful.
Theorem 2.1. GT-Con is equivalent to the following second-order cone program,

\[
\begin{align*}
\text{(GT-Con-SOCP)} & \max_{\hat{\beta} \in \mathbb{R}^p} \Gamma \\
\text{such that:} & \sum_{i=1}^{n} (y_i - \sum_{j=1}^{p} x_{ij} \hat{\beta}_j)^2 \leq q_1, \\
& \sum_{j=1}^{p} \hat{r}_j \leq q_2, \\
& \sum_{i=1}^{n} (y_i - \sum_{j=1}^{p} x_{ij} \hat{\beta}_j)^2 \leq \gamma^2, \\
& f_1 = \sqrt{q_1} - \gamma, \\
& f_2 = \sqrt{q_1} + \gamma, \\
& f_1 = q_2 - \sum_{j=1}^{p} \hat{r}_j, \\
& \Gamma^2 \leq v_1 v_2, \\
& v_1^2 \leq f_1 f_2, \\
& v_2^2 \leq f_3 \Gamma, \\
& \hat{\beta}_j \leq \hat{r}_j \quad \forall j \in \{1, \ldots, p\}, \\
& -\hat{\beta}_j \leq \hat{r}_j \quad \forall j \in \{1, \ldots, p\}, \\
& \hat{r}_j \geq 0 \quad \forall j \in \{1, \ldots, p\}, \\
& v_1, v_2, \Gamma, f_1, f_2, f_3 \geq 0.
\end{align*}
\]

where \( v_1, v_2, \Gamma, f_1, f_2, f_3 \) are some non-negative variables required for creating GT-Con-SOCP.

Proof. In the interest of brevity, the interested reader can refer to Appendix 5.3 for the detailed proof. \( \square \)

Observe that the objective function of GT-Con-SOCP is linear, and all its constrains are linear or (convex) quadratic. Hence, GT-Con-SOCP can be directly solved by a commercial
solver such as CPLEX or Gurobi. So, instead of solving $\textit{GT-Con}$, we simply solve its equivalent problem $\textit{GT-Con-SOCP}$ by a commercial solver in this study.

2.1.3 Phase-III: selecting the features

After solving $\textit{GT-Con-4}$, it is important to select the best subset of features. Let $\hat{\beta}^*$ be an optimal estimation of regression coefficients obtained by solving $\textit{GT-Con-4}$. We use the following procedure to select the best subset of features which is similar to existing literature [148, 94].

- Step 1. We compute a probability for each feature, i.e., $q_j := \frac{|\hat{\beta}_j^*|}{\sum_{k=1}^{p} |\hat{\beta}_k^*|}$ for each $j \in \{1, \ldots, p\}$.

- Step 2. We then sort the probabilities in non-increasing order. For simplicity, we assume that the probabilities are already sorted, i.e.,

$$q_1 \geq q_2 \geq \ldots \geq q_p,$$

in the remaining of this section.

- Step 3. Let $j^* \in \{1, \ldots, p\}$ be the maximum value with $\sum_{j=1}^{j^*} q_j < 1 - \alpha$ where $\alpha$ is a small positive hyperparameter that can be computed by the grid search method.

2.2 A case study in online advertising

To examine the proposed method’s effectiveness in the online advertising domain, we use a real rich dataset related to display ads and customer conversion in this section. The dataset is obtained from the Wharton Customer Analytics Initiative sponsored by Annalect, a leading global advertising and marketing communication services company. This paper’s computational experiments are carried out on an Intel(R) Xeon(R) Gold 6136 CPU with
3.00 GHz and 350GB RAM. For all computations, we use Python 2.6.6. To implement and solve our Game-Theoretic (GT) model, we use Gurobi optimizer version 7.5.1.

We evaluate our proposed method’s performance concerning other well-established feature selection methods in the literature. As noted in Section 1.3.3, there are, broadly speaking, four categories of feature selection approaches, including Filter, Wrapper, and Embedded, screening-based techniques. Among the Filter methods, mRMR [121], which is one of the recognized methods is used in this study. The technique can be used in both categorical and continuous features. It selects features according to the maximal statistical dependency criterion based on mutual information, i.e., selects elements that are maximally dissimilar to each other [20]. This method selects features with the highest relevance with the target class and eliminates features that highly correlated among themselves.

Among the Wrapper methods, BS and RFE [63] are two well-known techniques used in this study. BS is a sequential technique that starts with all features and then removes the unnecessary features step-by-step. RFE is an iterative algorithm where redundant or irrelevant features with the least contribution to the accuracy are eliminated at each iteration. Among embedded approaches, LASSO [136] and ENET [151] are two relevant benchmarks are employed in this study. LASSO is a regression analysis method that uses regularization (penalized loss function) for feature selection. The LASSO penalty expects many feature coefficients to be close to zero and only small subsets of feature coefficients to be nonzero. ENET is an extension of LASSO that is more robust to high correlations among features [54]. In other words, ENET is useful in selecting groups of correlated features when the groups are not known in advance [35].

Finally, among the screening-based techniques, SIS, which is one of the well-known methods, is used. The method proceeds in two stages. In the first, it filters out unimportant features in a model-free manner. Specifically, in this stage, the method just removes the less correlated features from the set of candidates. Then, in the next stage, feature selection and parameter estimation are performed simultaneously through a lower-dimensional penalized
least squares method. Table 2.1 summarizes the benchmark methods and highlights the characteristics of each feature selection method.

Table 2.1 – Summary of feature selection approaches

<table>
<thead>
<tr>
<th>Algorithm</th>
<th>Type</th>
<th>Applicable to task</th>
</tr>
</thead>
<tbody>
<tr>
<td>mRMR</td>
<td>Filter</td>
<td>classification, regression</td>
</tr>
<tr>
<td>BS</td>
<td>Wrapper</td>
<td>classification, regression</td>
</tr>
<tr>
<td>RFE</td>
<td>Wrapper</td>
<td>classification, regression</td>
</tr>
<tr>
<td>LASSO</td>
<td>Embedded</td>
<td>classification, regression</td>
</tr>
<tr>
<td>ENET</td>
<td>Embedded</td>
<td>classification, regression</td>
</tr>
<tr>
<td>SIS</td>
<td>Screening</td>
<td>classification, regression</td>
</tr>
</tbody>
</table>

2.2.1 The online display dataset

As mentioned in the previous section, The data analyzed in this study is provided by a multinational company offering vacation packages. On behalf of the firm, the advertising agency performs multiple ad campaigns using an array of platforms and ad-exchanges (social and traditional media). Each advertisement is associated with a unique campaign and a specific targeting strategy. Every time a user browses a website that belongs to the firm’s advertising network, a cookie embedded in the site places a unique identifier in the user’s browser. The cookie then tracks the user’s viewing and clicking on ads across all websites within the firm’s advertising networks. If the user visits the firm’s website or makes a purchase, the information is also recorded. Precisely, our individual-level data consists of time information of a user’s impression, visit, and purchase (if any) over two months from December 2014 to January 2015. Importantly, each user is associated with textual demographic information provided by MasterCard based on the general user browsing and purchasing behavior in the previous year. Overall, the dataset contains 133,916 observations and 179 features including but not limited to the type of ad, detail about exposure, demographics, targeting strategy, platform type, ad network, and type of publishers for each advertisement and each cookie chain throughout the purchasing funnel. In this case study we assume that all features

1See https://insight.factset.com/resources/mastercard-sector-insights-datafeed for additional details.
have the same acquisition and storage cost. This implies that by selecting less number of features the acquisition cost would be smaller.

2.2.2 Benchmarks

The goal of the empirical analysis is to perform a binary prediction about whether a user would purchase or not after receiving different forms of advertisements. For this purpose, each feature selection method is applied separately. Based on the features selected by each method, a machine learning classification model is subsequently applied. In this study, we use SVM and RF for classification purposes.

It is worth mentioning that one of the challenges of using online display advertisement dataset in this study is the existence of a large number of zeros, i.e., non-purchasers. Specifically, from the total of 133,916 observations, only 1,505 are purchasers. In other words, the dataset is imbalanced. To resolve this issue we rely to the well-established under-sampling methods whose shown to be effective way to deal with minor classes [42, 123]. The underlying idea of applying under-sampling is to randomly remove data points from major classes and create a desirable balance so that minor classes could be detected better. We use the ratio of 1:2, i.e., the number of non-purchasers is twice of the number of purchasers, which provides the best result in our case. So, after applying under-sampling, our dataset will have a total of 4,515 observations which include 1,505 purchasers and 3,010 non-purchasers.

2.2.3 Results and discussion

The prediction results for both classes of customers (purchasers or non-purchasers) are shown in Tables 2.2 and 2.3. Specifically, Table 2.2 shows the classification results of SVM and Table 2.3 provides the classification results of RF based on selected features by each method. The numbers in parenthesis in front of each feature selection method shows the number of features that have been selected by the methods and the highest prediction performance is highlighted in bold face. For all feature selection methods we use the same 80/20 training/test split on
the dataset. Therefore, the same training set is used for all feature selection methods and
the same testing test is utilized for machine learning algorithms to compare the accuracy
performance.

We note that for mRMR, the number of features that should be selected by the method
should be set in advance by the analyst. So, for this method, we have tested different values
and we report only the best case scenario in terms of prediction accuracy. Specifically, for
each dataset, an interval is first created based on the minimum and maximum number of
features selected by other feature selection methods. Then, all numbers in that interval are
tested one by one in mRMR. Afterwards, the value that provides the highest amount of
accuracy metrics is reported as the number of features selected by the mRMR.

As it pertains the regularization methods, for GT, Lasso and ENET, we employ a grid
search procedure with 10-fold cross-validation to tune hyper-parameters. In detail, for GT
the value of $\alpha$ is set between $10^{-3}$ and $10^{-1}$, for LASSO, the value of $\lambda$ is set to be between
$10^{-5}$ and 100 in the grid search. Additionally, for ENET, $\lambda_1$ and $\lambda_2$ are set to be between
$10^{-5}$ and 100 in the grid search. Finally, for RFE we apply 10-fold cross-validation to identify
optimal number of features.

Table 2.2 – SVM results for the online display dataset

<table>
<thead>
<tr>
<th>Evaluation Metric</th>
<th>mRMR (30)</th>
<th>BS (28)</th>
<th>RFE (123)</th>
<th>LASSO (27)</th>
<th>ENET (59)</th>
<th>SIS (33)</th>
<th>GT (24)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>87%</td>
<td>87%</td>
<td>87%</td>
<td>86%</td>
<td>86%</td>
<td>86%</td>
<td>87%</td>
</tr>
<tr>
<td>Precision</td>
<td>90%</td>
<td>90%</td>
<td>87%</td>
<td>86%</td>
<td>86%</td>
<td>86%</td>
<td>87%</td>
</tr>
<tr>
<td>Recall</td>
<td>81%</td>
<td>81%</td>
<td>82%</td>
<td>82%</td>
<td>82%</td>
<td>82%</td>
<td>82%</td>
</tr>
<tr>
<td>F1-score</td>
<td>83%</td>
<td>83%</td>
<td>84%</td>
<td>83%</td>
<td>83%</td>
<td>83%</td>
<td>84%</td>
</tr>
</tbody>
</table>

Table 2.3 – RF results for the online display dataset

<table>
<thead>
<tr>
<th>Evaluation Metric</th>
<th>mRMR (25)</th>
<th>BS (28)</th>
<th>RFE (84)</th>
<th>LASSO (27)</th>
<th>ENET (59)</th>
<th>SIS (33)</th>
<th>GT (24)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>87%</td>
<td>86%</td>
<td>89%</td>
<td>88%</td>
<td>88%</td>
<td>87%</td>
<td>90%</td>
</tr>
<tr>
<td>Precision</td>
<td>87%</td>
<td>87%</td>
<td>91%</td>
<td>88%</td>
<td>90%</td>
<td>86%</td>
<td>92%</td>
</tr>
<tr>
<td>Recall</td>
<td>84%</td>
<td>82%</td>
<td>85%</td>
<td>84%</td>
<td>84%</td>
<td>83%</td>
<td>86%</td>
</tr>
<tr>
<td>F1-score</td>
<td>85%</td>
<td>84%</td>
<td>87%</td>
<td>84%</td>
<td>86%</td>
<td>84%</td>
<td>88%</td>
</tr>
</tbody>
</table>

A comparison between Tables 2.2 and 2.3 indicates that for all feature selection methods,
RF achieves higher performance than SVM. Moreover, as illustrated in Table 2.3, based
on RF results, the proposed GT has better performance comparing to other benchmark methods. Specifically, GT outperforms other methods between 1% and 4% in terms of accuracy, 2% and 6% in terms of precision, 1% and 4% in terms of recall, 2% and 4% in terms of F1-score. It is also worth mentioning that, although based on Table 2.2 most methods have similar performance in terms of accuracy, the proposed GT method carries out at least as good as other methods when using SVM. Hence, it can be concluded that the proposed GT method is more robust when is applied within other algorithms.

Another observation from Tables 2.2 and 2.3 is that GT has selected a smaller number of features, i.e., 24 (out of 179), comparing to other methods. This result implies less total cost given that the cost of features are assumed to be the same. Particularly, the proposed GT method selects about 13% of the available features while other benchmark methods select between approximately 14% and 68% of the features. Overall based on the obtained results, it can be concluded that the proposed GT method creates a desirable balance between reducing the amount of error and the number of features. As it was mentioned in Section 1, collecting and computing users’ data is often expensive for DSPs. Therefore, identifying the most important and relevant features that results in higher accuracy rates can be beneficial for DSPs.

To provide additional interpretation about the selected features based on each method, we consider the approach known as Shapley Addittive exPlanations (SHAP) [96], based on the well-known concept of Shapley values in cooperative game theory. The approach computes the Shapley value as the average marginal contribution of a feature value across all possible combination of features [113]. It is then possible to rank features based on the absolute Shapley values. Therefore, the SHAP values allow us to explore different interpretations about the output of complex prediction models based on the contribution of each feature [113] and [108].

We recall that the RF predictions have higher on average accuracy and consequently, we conduct SHAP-values analysis based on this method. Since RF is a tree-based machine
learning model, we employ a variant of SHAP referred to as TreeSHAP [97, 68], which is specifically designed for tree-based machine learning models. TreeSHAP is a publicly available Python package and we use it for comparing feature selection methods in terms of their SHAP values in this section. Table 2.4 represents statistics summary of the absolute SHAP values across different feature selections methods. In the table, the highest values in each row are highlighted in bold face. Note that since SHAP values represent the contribution of features to the prediction, it can be concluded that the feature selection method with higher absolute SHAP values can result in higher performance comparing to other methods. With this in mind, it is possible to notice from Table 2.4 that the SHAP values appear to be heterogeneous within and between benchmark methods but with the relative impression of being higher for our proposed method. To further investigate this, we perform a formal pairwise comparison test of means of the absolute SHAP values. The results unequivocally reject the null hypothesis of the equality in each comparison (p-value \( \leq 0.01 \)), pointing out that, on average, the absolute SHAP values for the GT method outperforms the competitors. Based on the above discussions, we conclude that our proposed GT model not only offers comparable if not superior predictions to other methods, but it also selects features with higher “value”, in the SHAP-sense.

### 2.3 Computational results

To show the generalizability of the proposed approach, we evaluate the performance of GT method in both regression and classification models. For the first model, i.e., regression, the GT method is tested on 96 regression simulated datasets (see Section 2.3.1). For the
second model, i.e., classification, the GT method is tested on a high dimensional retail
dataset known as sentiment classification obtained from UCI machine learning repository.
The datasets consists of set of online reviews posted on Amozon, IMDb, and Yelp (see
Section 2.3.2). The performance of the proposed method is evaluated in terms of MSE
for regression instances and correct classification metrics for classification instances. For
solving classification instances, SVM and RF are employed to perform classification using
the optimal feature subsets identified by each method.

2.3.1 Simulated datasets

We generate 8 classes of datasets (sometimes referred to as instances) denoted by $C_{\gamma}$
where $\gamma \in \{1, 2, 3, 10, 20, 30, 40, 50\}$ denotes the standard deviation of the normal
distribution that is used to generate errors. Specifically, each class contains 12 instances
each denoted by $(p, n)$ where $p \in \{100, 200, 400, 800\}$ is the number of features and $n$
is the number of observations. In this study, we set $n = \sigma p$ where $\sigma \in \{2, 4, 8\}$. To generate each instance, we set $x_{i1} = 1$ for
all $i = 1, \ldots, n$. However, $x_{ij}$ is randomly drawn from the discrete uniform distribution on
interval $[-50, 50]$ for each $i \in \{1, \ldots, n\}$ and $j \in \{2, \ldots, p\}$. To construct $y_i$ for $i = 1, \ldots, n$
two steps are taken similar to the procedure proposed by Charkhgard and Eshragh [28]: (1) A
vector $\beta$ is generated such that 70% of its components are zero, and the others are randomly
resulted from the uniform distribution on the $(0, 1)$ interval. (2) We set $y_i = \epsilon_i + \sum_{j=1}^{p} x_{ij} \beta_j$
where $\epsilon_i$ is randomly generated from the normal distribution with mean of 0 and standard
deviation of $\alpha$.

2.3.1.1 Evaluation of simulated datasets

In this section, we compare the performance of mRMR, RFE, BS, LASSO, ENET, SIS, and
GT on the 96 simulated regression datasets. The MSE is used for evaluation of each method.
Grid search with 10-fold cross-validation is applied to tune hyper-parameters of GT, LASSO,
and ENET methods. Specifically, for GT the value of $\alpha$ is set between $10^{-3}$ and $10^{-1}$ and
for LASSO, the value of $\lambda$ is set to be between $10^{-5}$ and 100 in the grid search. Also, for ENET, $\lambda_1$ and $\lambda_2$ are set to be between $10^{-5}$ and 100 in the grid search. By construction of the simulated regression datasets, we know that the number of true features that should be selected is $0.3p$. Therefore for mRMR and FRE this information is given as an input. For instance, for a dataset with 200 observations and 100 features, mRMR and RFE select $0.3 \times 100 = 30$ of the features. Similarly, for a dataset with 6400 observations and 800 features, they select $0.3 \times 800 = 240$ of the features.

Table 2.5 – Minimum, average, and maximum of MSE for each method and $\alpha$

<table>
<thead>
<tr>
<th>Standard Deviation</th>
<th>MSE</th>
<th>mRMR</th>
<th>BS</th>
<th>RFE</th>
<th>LASSO</th>
<th>ENET</th>
<th>SIS</th>
<th>GT</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\alpha=1$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Min</td>
<td>1.08</td>
<td>0.97</td>
<td>0.95</td>
<td>1.03</td>
<td>1.03</td>
<td>1.08</td>
<td>0.94</td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>1.87</td>
<td>1.47</td>
<td>1.20</td>
<td>1.48</td>
<td>1.46</td>
<td>1.87</td>
<td>1.19</td>
<td></td>
</tr>
<tr>
<td>Max</td>
<td>3.21</td>
<td>3.34</td>
<td>1.91</td>
<td>2.67</td>
<td>2.61</td>
<td>3.21</td>
<td>1.56</td>
<td></td>
</tr>
<tr>
<td>$\alpha=2$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Min</td>
<td>4.44</td>
<td>3.94</td>
<td>3.87</td>
<td>4.37</td>
<td>4.31</td>
<td>4.44</td>
<td>3.80</td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>7.22</td>
<td>5.23</td>
<td>4.70</td>
<td>6.69</td>
<td>6.28</td>
<td>7.22</td>
<td>4.66</td>
<td></td>
</tr>
<tr>
<td>Max</td>
<td>17.17</td>
<td>7.41</td>
<td>5.97</td>
<td>14.99</td>
<td>12.92</td>
<td>17.17</td>
<td>5.81</td>
<td></td>
</tr>
<tr>
<td>$\alpha=3$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Min</td>
<td>9.81</td>
<td>7.92</td>
<td>8.23</td>
<td>9.82</td>
<td>9.64</td>
<td>9.81</td>
<td>7.87</td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>15.42</td>
<td>11.08</td>
<td>10.28</td>
<td>14.37</td>
<td>13.04</td>
<td>15.42</td>
<td>10.12</td>
<td></td>
</tr>
<tr>
<td>Max</td>
<td>23.83</td>
<td>17.00</td>
<td>13.95</td>
<td>23.01</td>
<td>23.35</td>
<td>23.83</td>
<td>12.84</td>
<td></td>
</tr>
<tr>
<td>$\alpha=10$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Min</td>
<td>117.10</td>
<td>104.78</td>
<td>104.88</td>
<td>116.49</td>
<td>113.45</td>
<td>117.10</td>
<td>104.60</td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>179.52</td>
<td>127.82</td>
<td>122.20</td>
<td>177.15</td>
<td>145.91</td>
<td>179.52</td>
<td>119.86</td>
<td></td>
</tr>
<tr>
<td>Max</td>
<td>327.14</td>
<td>184.93</td>
<td>164.31</td>
<td>324.43</td>
<td>213.28</td>
<td>327.14</td>
<td>155.55</td>
<td></td>
</tr>
<tr>
<td>$\alpha=20$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Min</td>
<td>377.14</td>
<td>347.46</td>
<td>369.45</td>
<td>377.14</td>
<td>365.25</td>
<td>377.14</td>
<td>347.46</td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>669.67</td>
<td>480.81</td>
<td>464.66</td>
<td>665.13</td>
<td>570.63</td>
<td>669.67</td>
<td>440.36</td>
<td></td>
</tr>
<tr>
<td>Max</td>
<td>967.98</td>
<td>632.28</td>
<td>608.42</td>
<td>942.04</td>
<td>717.33</td>
<td>967.98</td>
<td>576.66</td>
<td></td>
</tr>
<tr>
<td>$\alpha=30$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Min</td>
<td>1024.14</td>
<td>945.91</td>
<td>937.13</td>
<td>1023.85</td>
<td>1005.40</td>
<td>1024.14</td>
<td>935.06</td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>1632.28</td>
<td>1234.74</td>
<td>1176.42</td>
<td>1630.17</td>
<td>1465.30</td>
<td>1632.28</td>
<td>1160.58</td>
<td></td>
</tr>
<tr>
<td>Max</td>
<td>2677.36</td>
<td>1698.99</td>
<td>1571.21</td>
<td>2676.75</td>
<td>2230.90</td>
<td>2677.36</td>
<td>1536.92</td>
<td></td>
</tr>
<tr>
<td>$\alpha=40$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Min</td>
<td>1584.17</td>
<td>1497.05</td>
<td>1484.71</td>
<td>1584.17</td>
<td>1542.85</td>
<td>1584.17</td>
<td>1466.75</td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>2585.00</td>
<td>1988.64</td>
<td>1916.30</td>
<td>2584.37</td>
<td>2141.22</td>
<td>2585.00</td>
<td>1853.94</td>
<td></td>
</tr>
<tr>
<td>Max</td>
<td>3798.68</td>
<td>2721.46</td>
<td>2477.82</td>
<td>3783.53</td>
<td>2872.62</td>
<td>3798.68</td>
<td>2363.33</td>
<td></td>
</tr>
<tr>
<td>$\alpha=50$</td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Min</td>
<td>2667.28</td>
<td>2495.50</td>
<td>2331.72</td>
<td>2667.10</td>
<td>2609.03</td>
<td>2667.28</td>
<td>2340.86</td>
<td></td>
</tr>
<tr>
<td>Average</td>
<td>4268.97</td>
<td>3285.14</td>
<td>3239.48</td>
<td>4263.54</td>
<td>3578.16</td>
<td>4268.97</td>
<td>3149.80</td>
<td></td>
</tr>
<tr>
<td>Max</td>
<td>7064.34</td>
<td>5292.02</td>
<td>5292.02</td>
<td>7053.10</td>
<td>5980.27</td>
<td>7064.34</td>
<td>5142.41</td>
<td></td>
</tr>
</tbody>
</table>

The minimum, average, and maximum MSE for each class of instances can be found in Table 2.5. For convenience, the smallest value of each row is shown in bold font. As illustrated in Table 2.5, for the prediction of diverse continuous variables, the proposed GT achieves the smallest MSE in almost all classes of instances in comparison with those of all other methods. Specifically, based on Table 2.5 when considering all classes of instances,
on average the MSE values of GT is between 1% and 58% smaller than other methods. Therefore, GT provides more comparable solution and robust performance among other benchmark methods. It can be also inferred that in all regression instances SIS an mRMR have the same performance, i.e., min, average, and max amounts of MSE are the same.

Please note that, in this section, since the number of features that can be selected are specified in advanced for some methods, we cannot directly compare the feature selection methods in terms of the number of features selected by them. However, since the true linear regression function is known for each instance by construction, we can identify whether each method has actually selected the right subset of features. With this in mind, for a given instance and feature selection method, let \( T \) be the percentage of features selected from the set of true features by the method and \( F \) be the percentage of features selected from the set of false features by the method. Using these notations, the percentage error, denoted by \( E \), in identifying the right subset of features for a given instance and method can be obtained as follows,

\[
E := \frac{(100 - T\%) + F\%}{2}.
\]

Observe that the smaller the \( E \)-value for a given instance and method, the better the performance of the method in terms of identifying the right subset of features for that instance. Using this observation, we first calculate the \( E \)-value for all instances and all methods and then take the average of them for each method. Based on our results, the average \( E \)-value of GT is only 5.79% while the average \( E \)-value of other methods is between 5.95% and 41.16%. This clearly indicates that our method outperforms other methods in terms of selecting the right subset of features.
2.3.2 The retail dataset

Sentiment analysis is one of the important areas of research in IS domain. For instance, in social media, sentiment analysis is frequently applied for analyzing customers’ opinions [29, 119]. Sentiment analysis is also applied in opinion mining for labeling (general) opinion of general public and consumers regarding a product, social event, and marketing campaigns. Labeling such opinions can provide valuable insights and hence help firms to have better understanding of their customers. Firms can also receive real-time feedback from users and monitor response of them to achieve effective marketing campaigns. However, due to the vast information that is posted frequently and the large scale of features, feature selection is necessary in this domain [2]. For this purpose, to evaluate the proposed method, we use Sentiment Labelled Sentences dataset containing online reviews posted on Amazon, IMDb, and Yelp. The dataset can be obtained from UCI machine learning repository [79].

For each website, there exist 1000 raw text sentences (500 positive and 500 negative) which are selected randomly from larger dataset of reviews. For the evaluation, a holdout validation method is used to randomly split the data into a train and test set such that each contains 250 positive and 250 negative reviews. To prepare the data for analysis we used Scikit-learn in python which has feature extraction function out of text [92]. We also have eliminated all special characters and words that appear only once from the set of features. Further details for each website are provided next.

- **Amazon**: This data contains customer reviews and scores of cellphones and accessories on amazon.com. Scores are integer numbers from 1 to 5. In our study, the reviews with a score of 4 and 5 are considered as positive, and scores of 1 and 2 are considered as negative. It is also worth mentioning that reviews with score of 3 are considered as neutral and they are not included. After preparation, the total number of features in this data is 1259.
• **IMDb**: This data is related to IMDb movie reviews that each of them is labeled either positive or negative. After preparation, the total number of features in this data is 2424.

• **Yelp**: This data pertains restaurant reviews. Similar to Amazon, scores are integer numbers from 1 to 5. The reviews that have a score of 4 and 5 are considered as positive, and the reviews that have a score of 1 and 2 are considered as negative. Moreover, reviews with score of 3 are considered as neutral and they are not included. After preparation, the total number of features in this data is 1457.

### 2.3.2.1 Evaluation of the retail dataset

In this section, we compare the performance of mRMR, RFE, BS, LASSO, ENET, SIS, and GT on the retail dataset. For each website, the results are shown in two tables, i.e., Tables 2.6 and 2.7. The first one indicates the performance of SVM on the features selected by each method and the other shows the performance of RF on the features selected by each method.

In the tables, the numbers in parentheses in front of each method represents the number of features selected by each method. We again note that for mRMR, the number of features that should be selected by this method should be given in advance by users. So, for this method, we have tested different values and have reported the best value. Specifically, for each website, an interval is first created based on the minimum and maximum number of features selected by other feature selection methods. Then, all numbers in that interval are tested one by one in mRMR. Afterwards, the value that provides the highest amount of accuracy metrics is reported as the number of features selected by the mRMR. For instance, in the Amazon review data the minimum and maximum number of selected features are 200 and 848. Therefore, all numbers in the interval \([200, 848]\), i.e., 200, 201, 202, 203, \ldots, 848, are used for feature selection using mRMR method. Finally, it is also worth mentioning that, for tuning the hyper-parameter(s) of GT, LASSO, ENET, and RFE the same process described
in Section 2.2.3 is used, i.e., we used 10-fold cross validation and grid search with the same values.

Table 2.6 – SVM results for the retail dataset

<table>
<thead>
<tr>
<th>Evaluation Metric</th>
<th>mRMR (780)</th>
<th>BS (497)</th>
<th>RFE (768)</th>
<th>LASSO (200)</th>
<th>ENET (848)</th>
<th>SIS (302)</th>
<th>GT (736)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>81%</td>
<td>80%</td>
<td>86%</td>
<td>82%</td>
<td>85%</td>
<td>79%</td>
<td>86%</td>
</tr>
<tr>
<td>Precision</td>
<td>81%</td>
<td>80%</td>
<td>86%</td>
<td>82%</td>
<td>85%</td>
<td>79%</td>
<td>86%</td>
</tr>
<tr>
<td>Recall</td>
<td>81%</td>
<td>80%</td>
<td>86%</td>
<td>82%</td>
<td>85%</td>
<td>79%</td>
<td>86%</td>
</tr>
<tr>
<td>F1-score</td>
<td>81%</td>
<td>80%</td>
<td>86%</td>
<td>82%</td>
<td>85%</td>
<td>79%</td>
<td>86%</td>
</tr>
</tbody>
</table>

Table 2.7 – RF results for the retail dataset

<table>
<thead>
<tr>
<th>Evaluation Metric</th>
<th>mRMR (795)</th>
<th>BS (407)</th>
<th>RFE (780)</th>
<th>LASSO (200)</th>
<th>ENET (848)</th>
<th>SIS (302)</th>
<th>GT (736)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Accuracy</td>
<td>81%</td>
<td>80%</td>
<td>81%</td>
<td>76%</td>
<td>81%</td>
<td>80%</td>
<td>82%</td>
</tr>
<tr>
<td>Precision</td>
<td>81%</td>
<td>80%</td>
<td>81%</td>
<td>76%</td>
<td>81%</td>
<td>80%</td>
<td>82%</td>
</tr>
<tr>
<td>Recall</td>
<td>81%</td>
<td>80%</td>
<td>81%</td>
<td>76%</td>
<td>81%</td>
<td>80%</td>
<td>82%</td>
</tr>
<tr>
<td>F1-score</td>
<td>81%</td>
<td>80%</td>
<td>81%</td>
<td>76%</td>
<td>81%</td>
<td>80%</td>
<td>82%</td>
</tr>
</tbody>
</table>

From Table 2.6, i.e, SVM results, we observe that GT performs better or as good as other benchmark methods (in terms of accuracy, precision, recall, and F1-score) for all three websites. In the Amazon data, GT and RFE achieve similar performance with 1% to 7% higher accuracy values comparing to other methods. In the IMDb data, GT obtains the highest accuracy values which is between 6% and 11% higher than other methods. The results also show that IMDb reviews are hardest to be correctly classified because it has
the smallest amount of accuracy obtained by all methods. In the Yelp data, GT performs between 4% and 8% better than other methods in terms of accuracy. Finally, while the cost of features are not necessarily relevant in the context of sentient analysis, we observe that GT has selected only around 55% of features for all three websites. This combined with the accuracy results show that GT has created a good balance between the number of features and accuracy comparing to other methods.

From Table 2.7, i.e., RF results, we again observe that GT outperforms the other benchmark methods. Specifically, in terms of accuracy, GT outperforms between 1% and 6% in the Amazon data, between 1% and 5% in the IMDb data, and between 1% and 4% in the Yelp data. Like SVM results, RF accuracy results in IMDb reviews are lower comparing to other websites which again shows that the IMDb reviews are harder to be correctly classified. Finally, it can be also concluded by comparing the results of SVM and RF that SVM has better classification performance on the retail dataset. As an aside, we note that we have not applied SHAP analysis to the retail dataset, since the number of features is significantly larger, i.e., up to 28 times, than our case study dataset (in Section 2.2). This implies that the total number of subsets of features in the retail dataset is an astronomically large value which can not only negatively impact the performance of SHAP package but also make it significantly slower [56].

In our first study, we developed a methodology for selecting the best subset of features in the context of online advertising and, more broadly in the field of information systems. While in this study we select the most predictive subset of costly features that result in higher accuracy, we have not explicitly accounted the different costs involved in acquiring and utilizing such features. In other words, in the first study, we assumed that all feature have the same acquisition and storage cost for a DSP. However, in mobile advertising campaigns the different features have different data management and acquisition costs. For our second study, we address these issues faced by DSPs and develop a 7-step framework for optimizing the selection and utilization of costly data attributes in mobile ad campaigns; the framework
serves to improve the targeting accuracy of the campaign and to achieve higher levels of in-
store visits and sales conversions for brick-and-mortar (BM) retailers. As it was mentioned
in Section 1.3.1, in our second study we focus on BM stores since examining the impact
of mobile advertising in a physical BM store setting is often more challenging due to the
difficulties involved in tracking customer pathways in the sales funnel.
Chapter 3: Designing efficient mobile advertising campaigns

As shown in Figure 3.1, in the programmatic delivery of mobile ads, the DSPs use a prediction model to decide whether it is worth bidding on the available advertising spaces in mobile apps of users by assessing a variety of attributes, such as the details about the devices and apps, users’ locations, and their demographics information. The accuracy of the DSP’s targeting decisions and the overall performance of the ad campaign, thus, depend on the set of attributes used for enacting the prediction model.

Leveraging an user’s physical location as the primary attribute to select the bid requests to drive in-store visit to BM store is a common scenario [51, 71]. However, additional attributes such as demographic information, customer preferences, and prior store visits, may also influence the placement and effectiveness of the mobile advertisement. Acquiring and utilizing such data attributes is often expensive and contributes to the overall operational expenses of a mobile ad campaign, but the return-on-investments of these expenditures are often uncertain. While location-based targeting of consumers is recognized as a useful strategy by researchers and practitioners, campaign designs that programatically address the heterogeneous effects of targeting attributes as well as their potential interaction effects with location in influencing in-store visits have not been systematically studied in prior work.

Figure 3.1 – A simplified illustration of the decision-making process of DSPs
In this study, we build on two propositions: (1) the impact of location-based targeted advertisements on the probability of a user’s in-store visit declines as the distance between the store and the focal user’s physical locations increase, and (2) the impact of other targeting attributes, such as demographic information, on the probability of in-store visits depend on the distance between the retail store’s and the user’s physical locations. A key implication of these empirically-validated propositions is that a campaign design strategy that statically utilizes all the available data attributes or an adhoc subset of attributes for targeting would be suboptimal. Rather, the selection of attributes to utilize for targeting should be guided by an analysis of the interaction effects between location and other data attributes and a careful assessment of the costs and benefits (i.e., a return-on-investment (ROI) perspective). We incorporate such an analysis through an ROI model, offer robust procedures to derive solutions, and illustrate the framework’s application through a real-world case study. Figure 3.2 summarizes our proposed 7-step framework for mobile advertising campaign design.

In light of the above, in this section, we first present our baseline propositions for developing our campaign design framework and then explain the iterative multi-stage approach for refining the design parameters while executing targeted mobile advertisement campaigns. The empirical validation of these key components of our campaign design framework are presented later in Section 3.5.
3.1 Baseline propositions

Direct effect of location: As we discussed in Section 1.3, the physical location of a user is an important parameter that can be used for targeting mobile advertisements. Customers are more responsive to offers from events that are located nearer to them, and factors such as traveling costs and familiarity may contribute to such preferential behaviors [10, 9, 61, 107]. Based on this, our first proposition for campaign design is:

- \( p1: \) As distance increases from the focal store to a targeted user, the impact of targeted mobile advertisements on the user’s in-store visit decreases.

Interaction effect of location: Prior research has shown that beyond location there may be a host of other factors such as demographics, device usage, day and time, and prior purchasing history that can impact the effectiveness of targeted advertisements [67, 98, 144]. In practice hundreds of such data attributes collected from a variety of sources can be utilized for a firm’s advertisement campaigns. We expect contingent and heterogeneous effects of these data attributes in mobile advertisements. Specifically, we expect a user’s location to influence the effect of other additional attributes in impacting the probability of the user’s in-store visit. For example, the sensitivity of customers to targeted ads may depend on both their current location as well as their gender and/or age. Thus, to optimally balance costs and benefits of acquiring and utilizing these additional data attributes, we propose to select a subset of effective attributes by taking into consideration their location-contingent effects on in-store visits. Hence, our second design proposition is:

- \( p2: \) The impact of additional attributes, such as demographics, customer profiles, and purchasing history, on a user’s in-store visit depends on the user’s current location.
3.2 Iterative design

Reflecting industry practice, we conceptualize the programmatic delivery of a mobile advertising campaign in multiple phases. The first one is the pilot phase and the rest are the production phases of a campaign. In the pilot phase, the campaign is executed on a small and representative sample of the target population with the purpose of identifying the optimal number of distance intervals and collecting the data on the effectiveness of the campaign. The optimal number of distance intervals and the collected data can then be employed to generate an effective prediction model (Figure 3.1) using an optimization procedure (Figure 3.2) in order to be used in the Production Phase I for identifying which users should be exposed. While keeping the number of distance intervals fixed, the data collected in the Production Phase I is then used to further fine-tune the campaign design using the same optimization procedure. The refined prediction model will then be used in Production Phase II for identifying which users should be exposed. This iterative process continues till the campaign design stabilizes and does not require further fine-tuning.

In this iterative campaign design context, the optimization procedure is concerned with making optimal decisions about which subset of data attributes must be acquired and used for the campaign. As illustrated in Figure 3, the selection of data attributes to be used critically impacts both the costs of executing the campaign and the estimations of the revenues resulting from the campaign. The campaign’s cost component is a function of the targeted population size and the cost of acquiring and operationalizing the data attributes. The fixed costs capture the infrastructural overheads related to the underlying costs related to the ad bidding and delivery mechanisms encountered by the DSPs. The total revenue is a predicted value, as the sales conversion rate resulting from the exposure of mobile advertisements is ex ante unknown but can be estimated using the DSP’s prediction model utilized for selecting the target population. We further elaborate on this aspect in Section 3.3.
Thus, the campaign optimization procedure seeks to identify the best collection of data attributes about the target population, so that the ROI or difference between the campaign’s costs and revenues is maximized within a budget constraint. As our review of the related literature indicated, prior studies have emphasized the use of location, temporal, demographics, and purchase related attributes for mobile advertising [67, 98, 144]. In this study, we consider these attributes, and more importantly, account for the location-contingent effectiveness of these attributes on the targeted population. For each targeted segment, the procedure in Figure 3.3 can be applied to compute the costs and revenues of the campaign. Thus, campaign optimization can be executed in a granular and an iterative fashion within an allocated budget, by simultaneously determining the best subset of attributes for all location segments.

### 3.3 ROI model

In this section, we present the model for optimally selecting attributes for mobile advertising campaign for BM stores to boost in-store visits (Step 3 of Figure 3.2). To recall, the objective of the target campaign is to maximize ROI, i.e., the difference between the total expected revenue and cost of campaign shown in Figure 3.3. Based on the selected attributes the campaign would target mobile users, after which, the user is expected to visit the BM store. When an advertisement is shown to a user, the probability that he/she visits the store within the next seven days is termed the visit rate of the customer. As discussed in previous literature [13, 61, 58, 9], the distance of the customer from the BM store is a
critical attribute in determining this probability. Existing literature has shown that as the
distance of the mobile user from the BM store decreases, the visit rate increases.

Using this observation, we divide the target geographic space of our advertisement into
several segments based on the distance from the BM store. Let \( D \) be an integer decision vari-
able reflecting the number of distance intervals. At the time of delivering an advertisement,
each customer falls into one of these distance intervals. We denote the index of distance
intervals by \( k \in \{1, \ldots, D\} \) and based on the earlier discussion in Section 3.1, the goal of our
proposed model is to identify which attributes should be selected in each distance interval
\( k \in \{1, \ldots, D\} \).

![Figure 3.4 – Proposed Optimization Framework](image)

For example, assume that there are four attributes and data collection for each of them
costs \( c_1, c_2, c_3, \) and \( c_4 \), respectively. Assume further that the maximum distance of customers
from a BM store is \( d_{\text{max}} \) and the optimal number of distance intervals is \( \hat{D} \). In that case,
the set of distance intervals is \( k = \{1, \ldots, \hat{D}\} \). Assume further that the BM store has a
limited budget and wants to maximize its ROI by selecting key attributes for each distance
interval that increases the visit rate of the customer to compensate for the increased cost of
advertisement campaign. An illustration of a possible optimal solution for this problem is
shown in Figure 3.4. In Figure 3.4 the gray shapes represent attributes that are selected in each distance interval and white shapes represent the otherwise. Specifically, based on our model, in the first distance intervals \([0 - \frac{d_{\text{max}}}{D})\) attributes \(a_1\) and \(a_2\) are selected, whereas in the last distance interval only \(a_4\) is selected.

Table 3.1 – Parameters and variables descriptions

<table>
<thead>
<tr>
<th>Sets</th>
<th>Parameters</th>
<th>Variables</th>
</tr>
</thead>
<tbody>
<tr>
<td>(S)</td>
<td>(R)</td>
<td>(D)</td>
</tr>
<tr>
<td>(I)</td>
<td>(M_k)</td>
<td>(x_{jk})</td>
</tr>
<tr>
<td>(S_i)</td>
<td>(N_k)</td>
<td>(y_{ik})</td>
</tr>
<tr>
<td>(D)</td>
<td>(c_j)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(F_c)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(B)</td>
<td></td>
</tr>
<tr>
<td></td>
<td>(\sigma_{ik})</td>
<td></td>
</tr>
</tbody>
</table>

With this backdrop, we present our proposed formulation after introducing a few parameters and notations. We denote the index set of attributes by \(S = \{1, \ldots, n\}\), where attribute \(j \in S\). We denote the set of all non-empty subsets of attributes by \(S\), i.e., \(S = \{S \subseteq S : S \neq \phi\}\). Observed that the cardinality of \(S\) is \(2^n - 1\). Hence, we denote the index set corresponding to \(S\) by \(I = \{1, \ldots, 2^n - 1\}\) and for each \(i \in I\) we donate its corresponding subset by \(S_i \in S\). The objective of our model is to select at most one of these subsets for each distance interval to target the mobile users in that distance interval in the advertising campaign. The list of mathematical notation, parameters, and variables used in the proposed formulation are given in Table 3.1.
For a given optimal \( \hat{D} \), the objective function of maximizing ROI to decide which attribute to select in which distance interval is

\[
\text{ROI}(\hat{D}) := \max_k \left( \sum_{i \in I} N_k R \sigma_{ik} y_{ik} - M_k \sum_{j \in S} c_j x_{jk} - N_k F_c \right)
\]  

(3.1)

The objective function of maximizing ROI contains three mathematical expressions. The first term in the objective function, i.e., \( N_k \sum_{i \in I} R \sigma_{ik} y_{ik} \), shows the expected return for a given distance interval \( k \in D \). Parameter \( \sigma_{ik} \) is the visit rate, for subset \( i \in I \) of attributes in distance interval \( k \in \hat{D} \). It indicates the probability that the customer will visit the BM store within a specific time period (in our case we assume one week) after the advertisement has been delivered to him/her. We identify the visit rate based on a machine learning algorithm (for more detailed information about the visit rate computation please refer to Online Appendix 5.3). We assume that for each BM store visit by a potential customer, the BM store gets an average profit of \( R \). This value is well known to each BM store. Hence, \( N_k R \sigma_{ik} \) is the expected return of the store by selling to the target mobile users for distance interval \( k \in \hat{D} \) within one week of the advertisement if subset \( i \in I \) of attributes is selected, i.e., \( y_{ik} = 1 \). The second expression, i.e., \( M_k \sum_{j \in S} c_j x_{jk} \), captures the total cost of collecting data for the selected attributes for the population (in the RTB incoming bid request) at distance interval \( k \in D \). Finally, the last expression, i.e., \( N_k F_c \), captures the total cost of delivering advertisements. The \( F_c \) is the average delivery cost of the advertisement per impression. This is the amount that is paid by the DSP to the RTB exchange.

However, \( \hat{D} \) needs to be determined also. Thus we can write the total formulation with \( D \) as the decision variable as a bi-level integer programming formulation for selecting attributes as follows,

\[
\max_{D \in \mathbb{Z}_+} \text{ROI}(D),
\]  

(3.2)
The proposed bi-level optimization model is a max-max problem where the upper-level model is designed to find the optimal number of distance intervals and the lower-level model generates the objective value of the upper-level model for any given value of \( D \). Specifically, for any value of \( D \), the lower-level model returns \( \text{ROI}(D) \) which is the optimal ROI value obtained by identifying the best subset of attributes in each distance interval. In other words, the objective function of the lower-level model maximizes the difference between the total expected returns of the advertising campaign and the total cost of running the campaign, which includes variable and fixed costs of delivering advertisements, as illustrated in Figure 3.3. Therefore, the objective of the lower-level model is to simultaneously identify the best subset of attributes for all distance intervals, so that the difference between the total returns and total costs is maximized. The complete model along with the budget and structural constraints is given as below.

\[
\begin{align*}
\max \max_{D \in \mathbb{Z}_+} & \quad \sum_{k \in \mathcal{D}} \left( N_k \sum_{i \in \mathcal{I}} R_{\sigma_{ik}} y_{ik} - M_k \sum_{j \in \mathcal{S}} c_j x_{jk} - N_k F_c \right) \\
\sum_{k \in \mathcal{D}} M_k \sum_{j \in \mathcal{S}} c_j x_{jk} + N_k F_c & \leq B \\
\sum_{i \in \mathcal{I}} y_{ik} & \leq 1 \quad \forall k \in \mathcal{D}, \\
y_{ik} = \left(1 - \sum_{j \in \mathcal{I} \setminus \{i\}} y_{jk}\right) \prod_{j \in S_i} x_{jk} & \quad \forall i \in \mathcal{I}, \forall k \in \mathcal{D}, \\
y_{ik} & \in \{0, 1\} \quad \forall i \in \mathcal{I}, \forall k \in \mathcal{D}, \\
x_{jk} & \in \{0, 1\} \quad \forall j \in \mathcal{S}, \forall k \in \mathcal{D}.
\end{align*}
\]  

Constraint (3.4) is the budget constraint and guarantees that the total cost of advertising is not more than the available budget. Constraint (3.5) is a structural constraint that ensures that at most one subset can be selected for each distance interval. This allows the model to not select any subset for a given distance interval if its ROI is expected to be negative.
Finally, Constraint (3.6) guarantees that $y_{ik}$ follows its definition for all $i \in I$ and for each distance interval. If the visit rate is below a certain threshold there is no incentive for DSP to use the attributes. This threshold will be higher than the visit rate obtained by targeting the advertisement based on the distance of the mobile users from the BM store. We next make a few observations about the proposed lower-level model.

- The proposed lower-level model guarantees that for any optimal solution, the optimal objective value will be captured accurately. However, this does not necessarily hold for non-optimal solutions. This is because for each $i \in I$ and $k \in D$, Constraint (3.6) guarantees that if $y_{ki} = 1$ then $x_{kj} = 1$ for all $j \in S_i$ but it does not guarantee the other way around, i.e., if $x_{kj} = 1$ for all $j \in S_i$ then $y_{ki} = 1$. However, the latter will naturally hold for optimal solutions because the objective function is in the form of maximization and all its coefficients are assumed to be strictly positive in this study.

If one wants to ensure that the objective value will be accurately captured even for non-optimal solutions then the following constraint can be added,

$$y_{ik} \leq 1 - x_{jk} \quad \forall i \in I, \forall k \in D, \forall j \in S \setminus S_i.$$ (3.9)

The constraint guarantees that for each $\forall i \in I$ and $\forall k \in D$, if there exists $j \in S \setminus S_i$ such that $x_{jk} = 1$ then $y_{ik} = 0$.

- It is obvious that the lower-level model is a nonlinear integer program because of Constraint (3.6). To transform the non-linear model into a linear model we replace Constraint (3.6) with following additional constraints.

$$y_{ik} \leq x_{jk} \quad \forall i \in I, \forall k \in D, \forall j \in S_i,$$ (3.10)

$$y_{ik} \leq (1 - \sum_{j \in I \setminus \{i\}} y_{jk}) \quad \forall i \in I, \forall k \in D,$$ (3.11)

$$y_{ik} \geq \sum_{j \in S_i} x_{kj} + (1 - \sum_{j \in I \setminus \{i\}} y_{jk}) - |S_i| \quad \forall i \in I, \forall k \in D.$$ (3.12)
Note that because the objective function is in the form of maximization, there is no need to add Constraints (3.11) and (3.12) for linearization. Since Constraints (3.9) and (3.10) are upper bound constraints, they ensure that if \( y_{ki} = 1 \) then \( x_{kj} = 1 \) for all \( j \in S_i \). However, they do not ensure that if \( x_{kj} = 1 \) for all \( j \in S_i \) then \( y_{ki} = 1 \). Since by assumptions the cost of acquiring each attribute is strictly positive, the latter will naturally hold for optimal solutions.

Finally, it is worth mentioning the presented lower-level model is a natural formulation of the ROI problem because the impact of each attribute is directly shown in the model. However, the ROI problem can be rewritten in a more generic form by using only \( y \) variables. Specifically, for each \( \forall i \in I \) and \( \forall k \in D \), let \( \bar{c}_{ik} \) be a parameter representing the cost of employing subset \( S_i \in S \) in the distance interval \( k \). By using this parameter, the ROI problem can be stated as the following integer linear program,

\[
\begin{align*}
\max & \sum_{k \in D} (N_k \sum_{i \in I} R\sigma_{ik} y_{ik} - M_k \sum_{i \in I} \bar{c}_{ik} y_{ik} - N_k F_c) \\
& \sum_{k \in D} M_k \sum_{i \in I} \bar{c}_{ik} y_{ik} + N_k F_c \leq B \\
& (3.5) \text{ and } (3.7).
\end{align*}
\]

The above model does not require the non-linear Constraint (3.6) because \( x \) variables are removed from the model. Also, the above model is generic because the parameter \( \bar{c}_{ik} \) can be calculated in any desirable way in advance. In the context of our research, we should set \( \bar{c}_{ik} = \sum_{j \in S_i} c_j \) for all \( \forall i \in I \) and \( \forall k \in D \). However, if, for example, bundling is allowed then one can set the value of \( \bar{c}_{ik} \) to the cost of bundling all attributes in \( S_i \) for the distance interval \( k \). Note that although data providers can use bundling and sell data with a discount, this discount is not substantial. This implies that the cost is often additive in nature. However, if the discount is substantial such that the
incremental cost of multiple attributes is much less than the additive values, the above model can be used directly.

3.4 Solution methods

In this section, we explain the solution methods for solving our proposed model described in the last section. As the value of $D$ is limited, instead of incorporating complex solution techniques for the max-max problem (Equation 3.3), we solve the upper-level problem by enumerating and solving the lower-level problem for all realistic values of $D$. For example, for all practical purposes, it’s meaningless to solve for less than 1 km distance interval (such fine granular management of attribute is difficult). Thus in this section, we focus on the lower level problem, i.e. for a given value of $D$ what are the different attributes at different distance intervals. Once the lower level problem is solved for different values of $D$, we select the $D$ that is giving us the maximum ROI. It is easy to demonstrate that the lower level problem (also referred to as ROI Problem in the rest of the paper) is $NP$-hard (please see Theorem 3.1). Thus we rely on two solution approaches - optimal solution and heuristic solution. Next we discuss each of these and their applicabilities in practice.

**Theorem 3.1. The ROI Problem is $NP$-hard.**

*Proof.* In the interest of brevity, the interested reader can refer to Appendix 5.3 for the detailed proof. □

3.4.1 Optimal solution

In order to solve the lower-level model, a straightforward approach is to employ high-end optimizers (or exact solvers) such as CPLEX; see Step 5 in Figure 3.2. However, there are three main challenges in using high-end optimizers. Firstly, we demonstrated that ROI Problem is $NP$-hard. So, it is expected that as the size of the model increases, high-end optimizers start to struggle when solving the proposed model. Secondly, the size of the model grows exponentially with the number of attributes. Thus, when the number of
available attributes is large (e.g., more than 15), there is no hope that even the model can be
generated and/or solved using exact solvers. Finally, employing high-end optimizers such as
CPLEX requires robust software and hardware infrastructure to solve the prescribed model.
Such robust software and hardware may not be available to regular DSPs, who may not have elaborated
time and expertise to solve such complex problems.

Next, we present two observations that are the basis for our proposed heuristic method
(given in Section 3.4.2), which is specifically designed to be applied in real-life scenarios
where an elaborate setup to solve the NP-hard problem is not available to DSPs. For this
purpose, we provide a generic result for \( D = 1 \) without making any assumptions about the conversion rates and the number of attributes. The following definition is helpful.

**Definition 1.** Subset \( v \in I \) is called a feasible solution for the ROI problem if it does not violate the budget constraint, i.e., \( M \sum_{j \in S_v} c_j + NF_c \leq B \).

**Observation 3.1.** Selecting Subset \( v \in I \) is an optimal solution for the ROI problem if and only if the following two conditions hold:

- It is a feasible solution.

- \( NR\sigma_v - M \sum_{j \in S_v} c_j - NF_c \geq NR\sigma_w - M \sum_{j \in S_w} c_j - NF_c \) for any other feasible solution \( w \in I \).

**Observation 3.2.** Following Observation 3.1, the ROI problem can be solved in \( O(|S|^2) \) in the worst-case scenario where \( |S| := 2^n - 1 \) is the cardinality of set \( S \).

3.4.2 Heuristic solution

In this section, we propose a heuristic solution approach based on Variable Neighborhood Search (VNS), which is a well-known metaheuristic for solving (discrete) optimization problems [103]. The proposed heuristic is basically Step 6 of the proposed optimization framework (see Figure 3.2), and can be used when a company does not have access to an exact integer programming solver or the size of the proposed ROI model is unmanageable.
The proposed approach iterates for a certain number of iterations predefined by the users and is denoted by Max-Iteration. After reaching the maximum number of iterations, the algorithm returns the best solution obtained during its search and its corresponding ROI. Each iteration of the algorithm starts with the best subset of attributes (discovered in the previous iterations) which is denoted by $S^*$ and has a manageable size of $n'$ (for example $n' = 10$ because $2^{10}$ is reasonably small). The parameter $n'$ is a user-defined parameter and it is evident that $n' \leq n$. The algorithm searches over all neighbors of $S^*$ in each iteration. In order to make sure that the neighbors remain the same during the course of each iteration, the algorithm first makes a copy of $S^*$ and denotes it as $S'$ because $S^*$ may change during the course of each iteration. Neighbors of $S'$ are defined as,

$$\text{Neighborhood}(S', \theta) := \{ \tilde{S} \subseteq S : |\tilde{S}| = |S'|, |\tilde{S}\backslash S'| \leq \theta \},$$

where $\theta$ is a user-defined parameter that defines the neighborhood distance and it is typically a small integer value (e.g., 2). Note that it is evident that $\theta \leq n'$. Intuitively, $\text{Neighborhood}(S', \theta)$ is the set of all same size neighbors of $S'$ that their differences are in at most $\theta$ elements. For example, if $S = \{1, 2, 3, 4\}$, $S' = \{1, 3\}$, and $\theta = 1$ then,

$$\text{Neighborhood}(S', \theta) = \{\{1, 3\}, \{1, 2\}, \{1, 4\}, \{2, 3\}, \{4, 3\}\}.$$

Next the procedure applied for exploring each neighbor of $S'$, denoted by $S''$, is explained. However, before doing so, it is worth mentioning that at the beginning of the algorithm, one need to initialize $S^*$. This initialization can be done by selecting $n'$ number of attributes through any existing attribute selection technique in the literature.

To explore $S''$, the algorithm allocates a proportion of the total budget available, $B$, to each distance interval. The budget allocated to distance interval $k$ is denoted by $B'_k$. At the beginning of the algorithm, we initialize $B'_k = B$ for each $k \in \mathcal{D}$. Observe that $\sum_{k \in \mathcal{D}} B'_k$ does not have to be necessarily equal to $B$. Overall, after exploring $S''$, the budget allocation will
be updated as it will be explained later. For a given budget allocation, we can consider each interval independently and compute its optimal solution easily based on Observations 3.1 and 3.2 for $S''$ and the allocated budget. Note that the obtained solution of each interval may not be (globally) optimal for the entire problem but it will be optimal for that particular interval with respect to its allocated budget and $S''$ (which is a subset of all available features). We denote the operation of computing an optimal solution for each distance interval $k$ based on $B'_k$ and $S''$ by Interval-Optimizer($k, B'_k, S'', n', N_k, M_k, R, c$). Note that in this section, vectors are shown in bold fonts. So, $c$ is $(c_1, c_2, \ldots, c_n)$.

The details of Interval-Optimizer($k, B'_k, S'', n', N_k, M_k, R, c$) can be found in Algorithm 23 in Online Appendix 5.3. Overall, this operation returns three pieces of information: the selected attributes (of $S''$) for the distance interval $k$, denoted by $\bar{x}_k$, the exact budget used in the interval, denoted by $\bar{w}_k$, and the exact ROI obtained in the interval, denoted by $\bar{p}_k$.

After computing $(\bar{x}_k, \bar{w}_k, \bar{p}_k)$ for all $k \in D$, the algorithm solves the following knapsack problem,

$$\max \sum_{k \in D} \bar{p}_k z_k$$

s.t. $\sum_{k \in D} \bar{w}_k z_k \leq B$

$z_k \in \{0, 1\}$  \quad \forall k \in D,$

where $z_k$ is a binary decision variable that takes the value of zero if interval $k$ should become inactive, i.e., solution $\bar{x}_k$ is not acceptable and no attribute should be selected for interval $k$, and one otherwise. Note that there are many effective exact and/or heuristic solution approaches for solving the proposed knapsack problem [99]. In this study, we use a Python’s package\(^2\) to solve any instance of the knapsack problem. Overall, solving the proposed knapsack problem is necessary for three reasons: (1) As mentioned before, we may have $\sum_{k \in D} B'_k > B$ and this may imply that $\sum_{k \in D} \bar{w}_k z_k > B$. So, in this case, some of the

\(^2\)https://pypi.org/project/knapsack/
intervals should obviously become inactive. (2) For some intervals \( k \in D \), we may have \( \bar{p}_k < 0 \) and this clearly indicates that selecting attributes for such intervals is not beneficial. (3) It helps us develop a mechanism for updating \( B'_k \) for all \( k \in D \).

For the latter, let \( \bar{z} \) be an optimal solution of the proposed knapsack problem. If \( \bar{z}_k = 1 \) for some \( k \in D \) then the interval \( k \) is important. This indicates that the allocated budget for the interval \( k \) should be increased (if possible). As a result, the algorithm sets \( B'_k \) equal to \( \min\{B'_k + \lambda, B\} \). The parameter \( \lambda \) is a user-defined parameter (for example one can set \( \lambda = \frac{B}{10D} \)). However, if \( \bar{z}_k = 0 \) for some \( k \in D \) then the interval \( k \) is probably not important. So, the algorithm (if possible) decreases the allocated budget for the interval \( k \) by setting it to \( \max\{B'_k - \lambda, N_k F_c + M_k \min\{c_j|\forall j \in S\}\}\). Note that the lower bound \( \frac{B}{2D} \) is selected to ensure that the allocated budget of an interval is always strictly positive. By doing so, it is expected that if an interval becomes inactive, it will have the chance to be reactivated in future iterations.

Overall, after solving the proposed knapsack problem, the best solution obtained during the course of the algorithm and also \( S^* \) should be updated (if necessary). Specifically, if the optimal objective value of the proposed knapsack problem is better/larger than the best ROI obtained during the search then for each \( k \in D \) the best obtained solution during the search, denoted by \( \bar{x}_k^* \), would be set to \( \bar{x}_k \) if \( \bar{z}_k = 1 \). Otherwise, \( \bar{x}_k \) would be set to \( 0 \) for \( k \in D \). To update \( S^* \), it will be set \( S'' \). The details of the proposed VNS approach can be found in Algorithm 3 in Online Appendix 5.3.

3.5 Case study: boosting in-store visits of a retail store

In this section, we present the application of ROI model using our case study data. The details about the can study dataset can be found in Online Appendix 5.3. Our objective is to emulate the multi-stage application of procedure described in Figure 3.2 to demonstrate the selection of attributes at different locations and impact of selection of subset of attributes on ROI. The execution setup used in our case study is shown in Figure 3.5. In the Pilot
phase, a small subset of the target population (25%) is utilized for collecting data to build a probabilistic predictive model of in-store visits. We use the store visit rate estimates obtained from the pilot phase to develop the ROI model and identify the best subset of attributes at varied distance segments from the store. We do so by identifying the best number of intervals for segmenting the targeted geography and the best subset of attributes for each of the distance segments. The Pilot phase results are then utilized to execute the first production phase of the campaign. In this phase, we identify customers who are selected for advertisement exposure using the attributes selected in the pilot phase. We note the customers’ real-world actions (visit the store or not) to compare the predicted ROI vs. the actual ROI. The data obtained from the first production phase is used to retrain the model and identify the best subset of attributes for the different distance intervals for the next production phase of the campaign. We repeat this iteration for a third production phase to illustrate the convergence of our results.

![Figure 3.5 – Experimental setting](image)

3.5.1 Pilot phase

For the Pilot phase, 25% of the target population is randomly selected, which is further randomly divided into an optimization dataset (80%) and a validation dataset (20%). In the optimization step, we run our ROI model on the optimization dataset multiple times. Specifically, we assume that the number of distance intervals can be from the set \{1, 2, 4, 6, 8, 10, 12, 14, 16\} and in each iteration we pick an element of this distance segment set and solve the optimization model. Note that, as discussed in Section 5.3, the distance range under consideration in our case study is [5km, 21km] because distances below 5km are
not acceptable as there is no locality around the store up to 5km and also distances above 21km are too far outside the geography to be considered. For example, if the number of distance interval is chosen to be 4, then the distance intervals are \([5\text{km},9\text{km})\), \([9\text{km},13\text{km})\), \([13\text{km},17\text{km})\), and \([17\text{km},21\text{km})\).

In our case study, there are 9 attributes including but not limited to customers’ demographic information, customers’ home and work location, and effects of competitors. For more information about the case study attributes and their computational details please see Online Appendix 5.3. We set their associated costs, i.e., \(c_j\), to 2, 5, 5, 2, 4, 1, 3, 3, and 4, respectively. These values are chosen based on the price at which the MDP (mobile data provider) sold data to the retail firm. Based on input from the case study firm, the value of \(R\) is set to 290. The \(F_c\) is set to 1 based on the feedback from the mobile DSP of the firm’s campaign. We note that the proposed design framework is flexible to accommodate any set of values relevant for other scenarios.

In the Pilot phase, we take all attributes into consideration. Whereas in subsequent phases, only the selected attributes based on data in the previous phase are taken into consideration. Since the optimization model should be run after each phase (i.e., once data collection is complete), the population size of each phase and the number of exposed users are known and can be directly obtained from the data collected in that phase. In this case study, we simply ignore the budget constraint as the population size of each phase is the same. Note that ignoring budget constraint is equivalent to setting the budget constraint equal to the total cost incurred in the last phase.

In the Pilot phase, for any given number of distance intervals, the outcome of each optimization model is the set of attributes selected for each distance interval and the prediction model to be used. So, in the validation step, we employ the prediction models to the validation dataset to identify which users have to be exposed and use that to compute the ROI. The prediction model (and the distance interval) with the highest ROI is selected as optimal and is used in the Production Phase I. Table 3.2 summarizes the results of the Pilot phase.
Table 3.2 – Summary of results in the Pilot Phase

<table>
<thead>
<tr>
<th>Number of Intervals</th>
<th>ROI of Optimization</th>
<th>ROI of Validation</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>35354.94</td>
<td>8247.4</td>
</tr>
<tr>
<td>2</td>
<td>37636.9</td>
<td>8261.2</td>
</tr>
<tr>
<td>4</td>
<td>60870.25</td>
<td>10145.2</td>
</tr>
<tr>
<td>6</td>
<td>66004.24</td>
<td>7940.1</td>
</tr>
<tr>
<td>8</td>
<td>52531.8</td>
<td>5195.3</td>
</tr>
<tr>
<td>10</td>
<td>62522.04</td>
<td>4096.3</td>
</tr>
<tr>
<td>12</td>
<td>55859.99</td>
<td>4474.9</td>
</tr>
<tr>
<td>14</td>
<td>67728</td>
<td>7133.2</td>
</tr>
<tr>
<td>16</td>
<td>58088.71</td>
<td>3811.3</td>
</tr>
</tbody>
</table>

Numbers in Table 3.2 are the average out of 10 runs. In each run, the Pilot phase dataset is divided into an optimization dataset (80%) and validation dataset (20%). So, in Table 3.2, for any given number of distance intervals, the column labelled ‘ROI of Validation’ shows the average value among ROI values obtained by running the prediction model on the validation dataset over all 10 runs. Moreover, for any given number of distance intervals, the column labelled ‘ROI of Optimization’ shows the optimal objective value of the optimization model on the optimization dataset (corresponding to the validation dataset used in column ‘ROI of Validation’). From Table 3.2, we observe that the optimal number of distance intervals is 4 as it generates the highest ROI value, i.e., $10,145.2, on the validation dataset.

3.5.2 Production phases

We consider only three production phases in this study as our goal is to show that our attribute selection converges and that will be achieved after three production phases. For each production phase in the case study, 25% of the target population were randomly selected. The results are summarized in Table 3.3. Note that there are only four distance intervals in the table as that was identified to be optimal in the Pilot phase.

The table shows which attributes are selected in each phase and the actual ROI value achieved. We have also reported the predicted ROI values in this table to illustrate the
Table 3.3 – Summary results of all phases

<table>
<thead>
<tr>
<th>Phase</th>
<th>Attribute</th>
<th>Interval</th>
<th>Actual ROI</th>
<th>Predicted ROI</th>
<th>Difference (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pilot</td>
<td>Demographics</td>
<td>✓</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Customer profiles</td>
<td>✓ ✓</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Home distance</td>
<td>✓</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td></td>
<td>Competitors</td>
<td>✓ ✓ ✓</td>
<td>–</td>
<td>–</td>
<td>–</td>
</tr>
<tr>
<td>Production I</td>
<td>Demographics</td>
<td>✓</td>
<td>$46,988.1</td>
<td>$76,087.7</td>
<td>38.24%</td>
</tr>
<tr>
<td></td>
<td>Customer profiles</td>
<td>✓ ✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Home distance</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Competitors</td>
<td>✓ ✓ ✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Production II</td>
<td>Demographics</td>
<td>✓</td>
<td>$45,742.6</td>
<td>$61,642.3</td>
<td>25.79%</td>
</tr>
<tr>
<td></td>
<td>Customer profiles</td>
<td>✓ ✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Home distance</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Competitors</td>
<td>✓ ✓ ✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Production III</td>
<td>Demographics</td>
<td>✓</td>
<td>$49,505.7</td>
<td>$47,819</td>
<td>3.41%</td>
</tr>
<tr>
<td></td>
<td>Customer profiles</td>
<td>✓ ✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Home distance</td>
<td>✓</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>Competitors</td>
<td>✓ ✓ ✓</td>
<td></td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Convergence pattern. The predicted ROI value in each phase reflects the estimated ROI value which was expected to be achieved for that phase before running the campaign. This is computed based on the data in the previous phase, i.e., the optimal objective value of the optimization model. The actual ROI value is computed based on the actual visit data to the BM store out of users to whom the advertisement has been delivered. As we progress through multiple phases in the campaign, we expect the predicted ROI to converge to the actual ROI, which is demonstrated through the reduction in the difference (%) between the predicted ROI and actual ROI in Table 3.3.

Table 3.4 – The absolute difference between the actual and predicted visit rate

<table>
<thead>
<tr>
<th>Phase</th>
<th>Distance Interval</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>1     2     3     4</td>
</tr>
<tr>
<td>Pilot</td>
<td>9%  0%  28%  54%</td>
</tr>
<tr>
<td>Production I</td>
<td>4%  2%  7%  29%</td>
</tr>
<tr>
<td>Production II</td>
<td>6%  4%  3%  5%</td>
</tr>
<tr>
<td>Production III</td>
<td>2%  1%  5%  1%</td>
</tr>
</tbody>
</table>

Overall, from Table 3.3, we observe that the predicted ROI value is getting closer to the actual ROI value after each phase. Specifically, the difference between the actual ROI
and predicted ROI reduces from 38.24% in Pilot phase to 3.41% in Production Phase III. In Production Phases I and II, some attributes have been filtered out. However, in Production Phase III, the attribute selection process is finally converged and no more attributes are filtered out compared to its previous production phase. That justifies why the predicted ROI value of Production Phase III is very close to its actual ROI value. This can be also observed from Table 3.4 which shows the absolute difference between predicted and actual visit rate of each distance interval in different phases based on the attributes selected, i.e., those shown Table 3.3. Note that the actual visit rate is the actual ratio of the users visited the store and being exposed when running a phase. The actual visit rate values can be found in Table A.3 in Online Appendix 5.3. The predicated visit rates are prediction of visit rates for each phase based on the machine learning model results. We again observe that the visit rates have converged as well. For the prediction model, we tested three machine learning methods, i.e., Random Forest (RF), Support Vector Machine (SVM), and Logistic Regression (LR), in this study. We chose SVM for our case study due to its better performance. Interested readers may refer to Online Appendix 5.3 for further details on prediction models and superiority of the SVM approach in our case study. Overall, our proposed campaign design approach is independent of the underlying machine learning, and any suitable prediction method can be utilized to implement our campaign design framework.

We conclude this section by mentioning that our case study involved only 9 attributes and, thus, it was possible for us to use an exact solver (CPLEX) for the optimization models. Specifically, CPLEX was able to solve the model of each phase in at most 7 seconds in our case study. Hence, there was no need to use the heuristic approach in our case study. However, to see the performance of the heuristic approach in our case study, interested readers may refer to Online Appendix 30. It is also worth mentioning that during the course of this study, we also conducted a separate experiment that involved a larger set of attributes. The results of that analysis are reported in Online Appendix 30, which illustrates the practical utility
of the heuristic solution approach when a large set of attributes are considered in real-world campaigns.

### 3.6 Managerial insights

In this section, we provide managerial insights about the relative effectiveness of our campaign design approach, compared to intuitive approaches, i.e., Labels A-I, introduced in Table 3.5. We highlight the importance of optimally selecting attributes for mobile advertising campaigns in order to maximize the benefits. The costs, benefits, and ROI of different campaign designs with different patterns of attribute selections are summarized in Table 3.5.

**Table 3.5 – The costs, benefits, and ROI of different campaign designs**

<table>
<thead>
<tr>
<th>Labels</th>
<th>Category of Attributes</th>
<th>List of Attributes</th>
<th>Cost ($)</th>
<th>Return ($)</th>
<th>ROI ($)</th>
</tr>
</thead>
<tbody>
<tr>
<td>Label A</td>
<td>Demographics-oriented</td>
<td>Demographics</td>
<td>19,936</td>
<td>4,091</td>
<td>-15,845</td>
</tr>
<tr>
<td>Label B</td>
<td>Distance-oriented</td>
<td>Current distance</td>
<td>28,309</td>
<td>51,233</td>
<td>22,925</td>
</tr>
<tr>
<td>Label C</td>
<td>Time-oriented</td>
<td>Time</td>
<td>7,962</td>
<td>0</td>
<td>-7,962</td>
</tr>
<tr>
<td>Label D</td>
<td>Customer profile-oriented</td>
<td>Previous visits</td>
<td>43,854</td>
<td>4,568</td>
<td>-39,287</td>
</tr>
<tr>
<td>Label E</td>
<td>Competitors-oriented</td>
<td>Competitors</td>
<td>16,332</td>
<td>55,610</td>
<td>39,278</td>
</tr>
<tr>
<td>Label F</td>
<td>A + B</td>
<td></td>
<td>48,437</td>
<td>90,594</td>
<td>42,157</td>
</tr>
<tr>
<td>Label G</td>
<td>A + B + C</td>
<td></td>
<td>56,489</td>
<td>102,952</td>
<td>46,463</td>
</tr>
<tr>
<td>Label H</td>
<td>A + B + C + D</td>
<td></td>
<td>100,514</td>
<td>140,584</td>
<td>40,070</td>
</tr>
<tr>
<td>Label I (all attributes)</td>
<td>A + B + C + D + E</td>
<td></td>
<td>116,252</td>
<td>135,065</td>
<td>18,813</td>
</tr>
<tr>
<td><strong>Proposed Approach</strong></td>
<td></td>
<td></td>
<td>24,819</td>
<td>74,325</td>
<td>49,506</td>
</tr>
</tbody>
</table>

Our **first managerial insight** is that incorporating only a single category of attributes including demographics-oriented, time-oriented, and customer profile-oriented in a mobile advertising campaign may result in significant loss, i.e., negative ROI, for BM stores. Labels A, C, and D, in Table 3.5 show that a naive approach for campaign design would result in a loss of between 7,962 USD and 39,287 USD for the BM store. A main reason for the poor performance of incorporating these categories of attributes separately is that they generate small conversion rates. Consequently, the total return of running the campaign will be smaller than its total cost when incorporating demographics-oriented, time-oriented, and customer profile-oriented individually.
Our **second managerial insight** is that although demographics-oriented category of attributes results in loss individually, it generates significant positive ROI values when being considered with distance-oriented category, i.e., Label F. In other words, the interactions of these two categories of attributes can improve the conversion rates significantly. This result is compatible with findings in the literature stating that location and demographic data are important attributes for accessing the right customers in specific locations [90]. Due to customers’ heterogeneity and differences in their needs across demographics, users are more receptive and positive to those ads which are relevant to their individualized needs [72, 26]. Therefore, considering distance attributes with reference to user demographics can result in more effective targeted advertisements.

Our **third managerial insight** from Table 3.5 is that by using designs other than the proposed approach, firms would miss sales opportunities. This can be observed from Table 3.5 in which our proposed campaign design approach has reached to a ROI value which is about 7% better than the best ROI value obtained by other campaign designs, i.e., Label G. In other words, although applying Label G can result in the highest ROI amount compared to other labels, this amount is still less than the ROI value of our proposed campaign design approach. This is mainly because in the intuitive design approaches A-I, the selected subset of attributes is fixed over all distance intervals but their heterogeneity with respect to location is accounted for in our proposed design. Note that Labels H, and I have created slightly higher returns compared to Label G because they have more attributes. However, because the cost of running a campaign by utilizing them is almost additive in nature (but not fully additive due to the fixed cost in Figure 3.3), the total cost is significantly larger than the total return for those labels. Therefore, their ROI values are smaller than the ROI value of Label G. This suggests that time-oriented, competitors-oriented, and customer profile-oriented attributes should not be necessarily considered in all distance intervals. This observation is in fact compatible with the results of our proposed approach explained next.
Table 3.6 – Importance of attributes based on distance from the BM store

<table>
<thead>
<tr>
<th>Category of attributes</th>
<th>List of Attributes</th>
<th>Distance</th>
</tr>
</thead>
<tbody>
<tr>
<td>Demographics-oriented</td>
<td>Demographics</td>
<td>✓</td>
</tr>
<tr>
<td>Customer profile-oriented</td>
<td>Customer profile</td>
<td>✓</td>
</tr>
<tr>
<td>Competitors-oriented</td>
<td>Competition</td>
<td>✓</td>
</tr>
</tbody>
</table>

* In this study, Close means up to 9km, Middle means between 9km and 17km, and Far means between 17km to 21km.

Further managerial insights can be obtained from Table 3.6 that indicates the importance of different categories of attributes based on the findings of our proposed campaign design approach given in Section 3.5. Specifically, our findings show (fourth managerial insight) that customer profile-oriented and demographics-related attributes are important for close distances from the BM store. This in fact is consistent with previous studies. In particular, studies show that in targeted advertisement, segmentation of customers based on both demographics and customer profiles has superior performance comparing to use any of them individually [111]. Demographic and customer profile attributes can help advertisers carry out a precise way with a clear focus on the target group. Identifying mobile device users’ demographics and the most related information can help advertisers target ads based on the interests and preferences of users [57].

Although for close distances, both customer profile-oriented and demographics-related attributes are important, for far distances only customer profile has been selected. One reason for that can be for close distances it is much easier to convince customers to visit the store (fifth managerial insight). Therefore, advertisers can offset costs by higher levels of in-store visits. However, for large distances, due to transportation costs, it is harder for advertisers to encourage customers to in-store visits. Hence, they should be more selective and choose attributes that not only persuade users to visit the store but also compensate the underlying costs.
Finally, our **sixth managerial insight** is that the competitors-oriented attributes are important only for middle distances from the BM store. One reason that competitors’ locations are important only for middle distances in our case study can be because of the unique location of the MyStore. As it is explained in Online Appendix 5.3, MyStore is located completely outside of the city and it is in an unpopulated area with three main competitors which are relatively close to it. Therefore, for far distances, users may not have enough incentive to visit competitors’ stores after receiving the mobile ads.
Chapter 4: Addressing device identity matching in mobile advertisement

In this chapter we propose a framework for identifying unique devices in RBT ecosystem. In other words, the goal of this chapter is to resolve the curse of identification. Applying the proposed framework not only provides a privacy-aware market (by not using devices IDs to track customers) but also it helps ad agencies to target right customers and help mobile advertising market to continue generate revenues.

The main idea of the proposed framework is to find devices that belong to the same entity and put them in one group. This process is called clustering in our framework. Specifically, for clustering devices, only available users’ attributes (except device ID) are used to cluster unique devices. In the framework, the clustering process is done with the help of a well-known concept known as Entity Matching (EM). EM is a learning-based method that does pairwise comparison between data instances (tuples) to predict whether they "match" or "no-match" [112]. For implementing EM, we employ an state-of-art open-source leaning-base solution known as Magellan [77].

Although EM is a powerful method, it has two main challenges. The first challenge is related to time. Since mobile advertising data is usually big, exploring and doing pairwise comparison to find matches is computationally expensive and even impossible. The second challenge is related to inconsistency. As the number of observations increases the inconsistency in matching tuples increase as well. For instance, in the process of EM, a marcher may consider pairs (1,2) and (2,3) as match while consider (1,3) as non-match. These contradictions can create inconsistency in the data and consequently making the clustering process a difficult task.
In order to do clustering based on pairwise comparison results of EM and at the same time deal with inconsistency issue of EM results, we develop a novel optimization model. The proposed model tries to minimize the number of nonempty clusters while keeping the inconsistency error of EM predictions in each cluster below a desirable threshold. The threshold itself can be defined based on the accuracy result of EM. Since in mobile advertising data, the number of pairwise comparisons is usually high and number of instances increases dynamically, solving the proposed model directly using exact methods, i.e., off-the-shelf solvers, is computationally infeasible in practice. Therefore, we propose a heuristic algorithm to quickly find high-quality feasible solutions for the proposed optimization model in practice.

4.1 Entity matching

Entity matching (EM), the problem of identifying data instances referring to the same real-world entity is a long-standing challenge in data cleaning and integration [112]. For instance, EM has been widely used in data heterogeneity problem where organizations try to integrate data from heterogeneous sources in order to support their business needs [24, 133, 39]. EM can also be applied in mobile advertising domain specially when due to privacy risks, users’ device IDs are not available. In that case, advertisers can use EM to identify unique devices that belong to the same entity and send the most relevant ads to them. Sending more relevant ads can result in higher chances of reaching to the target audience and therefore higher chances of increasing clicks.

In the Big Data era, due to high volume, variety and velocity of data, EM has become even more challenging [31]. With growth in amount of data, organizations needs find scalable EM approaches to deal with large data volumes that being collected continuously. Before covering approaches that are developed for solving EM problem we first define classical EM decision model that is developed in the literature [39, 40, 38].

Let $R_1 := \{a_1, a_2, \ldots, a_m\}$ and $R_2 := \{b_1, b_2, \ldots, b_n\}$ be two sets of observations and suppose that we are interested in identifying whether any pair of observations $(a_i, b_j)$ where
$a_i \in R_1$ and $b_j \in R_2$ is a match or not. The set of attributes that are common to both $R_1$ and $R_2$ are denoted by $\{Y_1, Y_2, \ldots, Y_k\}$. Consider two observations $a_i \in R_1$ and $b_j \in R_2$ and let $a_i(Y_k) = y_{aik}$ and $b_j(Y_k) = y_{bjk}$ be their recorded values of attribute $Y_k$ in $R_1$ and $R_2$. The possible match between $a_i$ and $b_j$ is denoted by $p_{ij}$. Specifically, $p_{ij}$ is the conditional probability that $a_i$ and $b_j$ are the same, given their recorded attribute values. Therefore,

$$p_{ij} = Pr[a_i = b_j | a_i(Y_k) = y_{aik} \text{ and } b_j(Y_k) = y_{bjk} \forall k \in \{1,2,\ldots,K\}]$$.

We now examine possible scenarios of matching between $a_i$ and $b_j$.

- If $a_i = b_j$ in reality and are matched, then there is no error.
- If $a_i \neq b_j$ in reality and are non-matched, then there is no error.
- If $a_i = b_j$ in reality but we fail to match them, then a type-I error is occurred. The cost of committing this error by the decision make is denoted by $c_1$.
- If $a_i \neq b_j$ in reality but we considered as matched, then a type-II error is occurred. The cost of committing this error by the decision make is denoted by $c_2$.

Accordingly, the relative cost if type-II error is define by $\alpha = c_2/(c_1 + c_2)$. Therefore, the developed decision model based on minimization of total error cost is as following:

$$\begin{align*}
\text{(P)} \quad & \max_x \sum_{i=1}^{m} \sum_{j=1}^{n} (p_{ij} - \alpha)x_{ij} \\
\text{s.t.} \quad & \sum_{i=1}^{m} x_{ij} \leq 1, \quad \forall j \in \{1,2,\ldots,n\} \\
& \sum_{j=1}^{n} x_{ij} \leq 1, \quad \forall i \in \{1,2,\ldots,m\} \\
& x_{ij} \in \{0,1\} \quad \forall i \in \{1,2,\ldots,m\}, \forall j \in \{1,2,\ldots,n\},
\end{align*}$$ (4.1)

where $x_{ij}$ is a binary decision variable and takes value of 1 if $a_i \in R_1$ is matched with $b_j \in R_2$, and 0 otherwise. The developed optimization model, i.e., Model (P), is the well-known
assignment problem where Constraints (2) and (3) ensure that an entity in one relation is matched with at most one entity in the other relation. Although results from solving Model (P) can be effective to match two entities from two different databases or within a single dataset, there is a need to estimate a large number of probability parameters which is not trivial [40]. Specifically, probabilities can be easily estimated as long as the necessary training data is available. However, calculating these estimations in training data usually needs to be done manually. Hence, by increasing the size of the data, creating unbiased and consistent estimations is a difficult task in practice. To overcome this challenge, Dey et al. [40] propose a distance-based approach. They rewrite Model (P) based on the expected distance between \(a_i\) and \(b_j\) that can be estimated based on the distance between each of the \(a_i\) and \(b_j\) common attributes. In other words, distance between two entities is expressed as a weighted sum of the distances between their attribute values. Although this approach can alleviate the challenge of probability estimation, it requires acquiring weights from users which is not an easy task. Specifically, the weights are obtained based on the ranks of the attributes from users. However, the actual predictive power of an attribute cannot fully and clearly capture in its relative rank. Moreover, the distance function adopted in [40] only captures a linear combination of the distances between their attribute values. In fact, the non-linearity between entity instances is completely ignored in the linear distance function. In another study, Dey [38] proposes an alternative technique based on logistic regression model for estimating probabilities. However, since this approach again relies on input from users knowledge about the application context, for large datasets this approach cannot be applicable [141].

Beside relying on users for estimating the matching decision model, all previous proposed approaches cannot be used for large datasets. For large datasets each entity needs to be compare to all other entities which results in quadratic computational cost, i.e., \(O(n^2)\). To avoid comparing each entity to all other entities blocking methods have been introduced [78]. Blocking is used as a prepossessing step with the goal of grouping similar entities together.
where only records within the same block need to be compared with each other [31, 120]. So, the matcher only performs \((pm^2)\) comparisons, where \(p\) is the number of blocks and \(m \ll n\). Therefore, by employing blocking the computational cost restricts to the comparison of a subset of the input entities. Despite the significant enhancements in efficiency, in the Section 4.2 we explained why blocking cannot be applied to our problem. In the next section we explain different methodologies that have been developed for the matching task.

4.1.1 Related EM methods

EM methods have received much attention [45, 116]. EM approaches generally can be divided into two categories: (i) approaches that rely on domain knowledge or on genetic distance metrics to match entities, (ii) approaches rely on learning-based methods [45].

A different line of work develop methods to leverage expert human domain knowledge to conduct EM. For instance, rule-based methods require a human expert to define rules to decide whether two records belong to the same real-world entity or not [50, 131]. However, since rule-based methods require extremely high manual effort of a domain expert, applying these methods in practice is a difficult task [45]. To alleviate human intervention, distance-based methods can be employed. Distance-based methods use nonprobabilistic similarity measures to identify the best matches [41]. Therefore, in distance-based methods by using a distance metric (similarity measure) and an appropriate matching threshold, it is possible to match similar records. However, since distance-based methods usually are more useful where the training data is not available and identifying the appropriate threshold is a challenging task [38, 45]. Moreover, as it was mentioned in Section 4.1, the human intervention issue cannot be completely resolved by employing distance-based methods.

Unlike rule-based and distance-based methods, learning-based methods focus on learning matching functions [18, 132, 77]. In supervised leaning-based methods, traditional machine learning (ML) models (such as decision trees and Support Vector Machine (SVM)) are used to learn a classification model, i.e., matching model. Using learning-based models in EM
domain gives researchers to implement more sophisticated matching techniques that results in higher performance comparing to rule-based and distance-based approaches [78]. However, the effectiveness of learning-based approaches is depends on the provision of sufficient and balanced training data otherwise they can result in unsatisfactory performance.

In recent years, Deep Learning (DL) methods have also greatly influenced EM research. Mudgal et al. [112] compared DL solutions with state-of-art learning based EM solution (such as Magellan) on structured, textual, and dirty datasets. Their results show that DL does not outperform current learning-based approaches on structured data, but it significantly outperforms them on textual and dirty datasets. Although EM methods have been widely used to identify data instances that refer to the same real-world entity, EM itself belongs to the more generic framework known as Entity Resolution (ER). In the next section we explain ER architecture in more details and describe implementation challenges that can arise in our problem domain.

4.2 Problem description

ER aims to match entities within or across data sources specially when unique entity identifiers are not available. The general ER workflows consist of three main components including: Blocking, Matching, and Clustering (see Figure 4.1) [128, 55]. Blocking refers to grouping similar entity profiles in order to apply pairwise comparisons on only records within the same block (group) rather than the entire data set. In other words, blocking can be viewed as pre-processing step and the goal of blocking algorithms is to reduce the number of possible pair of comparisons by indexing similar entities into blocks. Blocking is a fundamental step for large datasets since it reduces the computational cost of pairwise comparisons significantly. Based on the identified blocks, in the Entity Matching (EM) step, matching functions are applied to all pairs in a set of blocks to distinguish between matching and non-matching entities. Particularly, in the matching task, a block collection is received as an input. For each pair of candidate matches that co-occur in a block, a matching function decides if they refer
to the same real-world entity or not [31]. After applying the matching function, in Entity Clustering step, clustering algorithms are used to produce groups of entities that represent the same entity based on the results of the matching stage.

Since in our study the goal is identifying unique device IDs in large mobile advertising datasets and cluster them, ER can be used as a standard framework for this purpose. However, while the ER architecture looks straightforward, applying it directly to mobile advertising datasets for identifying unique customers from huge, dirty, noisy, and sparse datasets is not a trivial task. This is mainly because of three important characteristics of online advertising datasets explained next.

- **Volume and Sparsity.** One main characteristic of mobile advertising datasets is their large volume and sparsity. As the data volume increases identifying appropriate approaches to block the data becomes harder. Generally in the literature they are two main techniques for blocking large datasets: (i) schema-aware and (ii) schema-agnostic. Schema-aware approaches detect near duplicates based on the noise-free distinctive values of a specific attribute name [120]. These techniques usually rely on structured data and human intervention for defining blocks. In large datasets schema-aware techniques have two main issues: (i) They cannot be used for loosely structured and highly heterogeneous entity descriptions [31]; (ii) They require labeled data to train their internal classification algorithms or domain experts to select best attributes to combine [55].

In contrast to schema-aware approaches, schema-agnostic approaches are more flexible since they can be applied on heterogeneous semi-structured data and they require...
no human intervention. Therefore, in schema-agnostic approach, all attributes values regardless of their associated attribute names can be considered [31]. However, schema-agnostic methods have a very low precision due to the large number of unnecessary comparisons [55].

Although both of these approaches can be useful during blocking stage, it cannot be guaranteed that they are applicable in all datasets. Specifically, in mobile advertising domain, due to huge amount of data in online datasets blocking seems inevitable. However, finding the specific rule or attribute(s) to block devices based on schema-aware methods is not trivial. Most device data are related to users’ location, IP addresses, device specific information (such as device model, device operating system, and etc.), and users demographics. Among these attributes users’ location, IP addresses, and device specific information are so variable (e.g., the same user can be in different location at different time and he/she can use several devices such as cellphone and tablet) that cannot be used as a rule to block users. At first glance demographic information can be considered as a good candidate to perform schema-aware blocking. However, due to the sparsity issue, i.e., most values are NAs, in demographic information using schema-aware methods is impractical.

In general, mobile advertising data sets are so sparse that make blocking even with all attributes a challenging task. This implies that even schema-agnostic methods are not promising for blocking. Figure 4.2 shows the sparsity of a typical mobile advertising dataset, i.e., a sample of our whole dataset. In this figure, horizontal axis shows the number of attributes and vertical axis shows the maximum percentage of NAs. Each point in this figure shows the number of attributes that their percentage of NAs have been at most equal to its corresponding value shown in the vertical axis. For example, point (24,10) shows that the percentage of NA values in 24 attributes is at most 10%. From the figure we observe that in at least half of attributes, i.e., 37 (out of 73), the
number of NAs is larger than 50%. In this study, NA means either no information is available or all observations have the same value for an attribute.

![Figure 4.2 – Sparsity](image)

- **Time sensitivity.** Another main characteristic of mobile advertising datasets is their time sensitivity. Identification of users should be done in a fractional second to allow ad agencies bid on each user. The decision whether two entities match is typically based on the result of a matching function. A matching function usually takes two entities as input and calculates the likelihood that they are matched. For large datasets comparing every pair of entities is $O(n^2)$. Therefore using matching function on the entire large dataset is computationally expensive. However, matching function cannot provide real-time answers for large mobile advertising dataset. One approach to alleviate this complexity is blocking. However, as it was mentioned earlier due to sparsity issue and lack of specific attributes to filter observations based on them, using blocking strategies on mobile advertising data sets is challenging.

- **Dynamic nature with varying data.** Another main characteristic of mobile advertising datasets is that the data come dynamically in large streams. Ad agencies need to cluster the huge amount of new entities (e.g., 500,000 entities) every second. Therefore,
there is a need to employ powerful incremental (dynamic) methods to update entity clusters in an efficient and accurate way. Traditional clustering methods mostly use static approaches with higher runtimes [128]. However, for online advertising domain there is a need to employ fast dynamic clustering methods [31] to update the clusters in less than a second.

In light of the above, we develop a novel custom-built ER framework that can address the above challenges and identify unique device IDs in mobile advertising with high accuracy.

4.3 Proposed framework

Figure 4.3 shows the proposed framework for identifying unique devices in the RTB ecosystem. The goal of the proposed framework is to cluster observations that represent the same device. For this purpose, we develop a framework, shown in Figure 4.3, that consists of two main stages: Entity Matching (EM) and Clustering.

Figure 4.3 – The Proposed Framework

In the EM stage, the goal is developing a highly-accurate prediction model for identifying data instances that refer to the same real-word entity. Specifically, here we seek to train a highly-accurate supervised EM model that can predict whether any pair of observations/tuples is a match or not. For this purpose, a small sample of entire dataset (e.g., 200 observations out of 32,143 in our case study) is applied for obtaining a EM model. Note that for a sample of size \( n \), the number of pairs that can be generated is \( n^2 \). This implies
that if \( n = 200 \), then the total number of pairs of tuples is 40,000 (which is a relatively big number). We use 70% of the pairs for training learning-based matchers (e.g., decision trees, support vector machine, and logistic regression) and 30% of the labeled data for testing to identify a EM model with the highest accuracy. For implementing the EM stage, we employ the Magellan [77]. Magellan is a state-of-the-art open-source learning-based EM solution [112] that has been developed in the form of Python packages.

![Figure 4.4 – The Solution Approach for Clustering Stage](image)

In the Clustering stage, the goal is to cluster unique devices on the remaining dataset with the help of the best EM model that was identified in the EM stage. Since it is expected that the accuracy of EM model does not reach to 100%, inconstancy errors in the EM model results are inevitable. Therefore, in the Clustering stage it is necessary to minimize the inconsistency issue and at the same time minimize the number of nonempty clusters. Since these two objectives, i.e., minimizing inconsistency and minimizing the number of nonempty clusters, conflict with each other, in the clustering stage we are basically dealing with a bi-objective optimization problem. Our approach to solve this bi-objective problem is to
keep the goal of minimizing the number of nonempty clusters as the main objective function while considering an acceptable threshold for the consistency of each cluster as a constraint. The threshold can be determined based on the accuracy of the best EM model identified in the EM stage.

We now explain the solution approach for clustering stage using a simple example. Assume that there are 7 observations (devices) that are needed to be clustered. Imagine there are three possible cases that can be used for clustering devices. Figure 4.4 shows these three possible cases. The rectangles show identified clusters in each case. The gray circles in clusters show the devices that are predicted as matched and white circles show the devices that are predicted as no-match by the EM model. Also, suppose that the acceptable consistency threshold here is 60%. The obtained consistency scores of clusters are shown below of each cluster. The scores are ratios of matched devices to all pairwise comparisons in each cluster. For instance in Case 1 scores of cluster 1, 2, and 3 are $\frac{2}{3} \times 100 = 67\%$, $\frac{1}{3} \times 100 = 33\%$, and $\frac{1}{1} \times 100 = 100\%$ respectively. Note that we assume that the consistency score of any cluster with a single element always satisfies the imposed threshold, i.e., it has an accuracy of 100%. Since our objective is to find the case that minimizes the number of nonempty clusters and at the same time satisfies the imposed acceptable consistency threshold, i.e., scores that are higher than 60%, Case 2 is selected as the best solution. The reason is that in Case 1, Cluster 2 violates the imposed threshold for consistency (Cluster 2 score is 33%) while Case 3 could not minimize the number of nonempty clusters (it has 4 clusters instead of 3 clusters).

In light of the above in order to do clustering, we propose a novel optimization model where its notations are described in Table 4.1. In this optimization model, it is assumed that there are $m$ observations that need to be clustered. For example, in Figure 4.3, $m$ is equal to 31,500. So, the maximum number of clusters that can be generated is $m$, i.e., each cluster contains one observation.
Table 4.1 – Notations

<table>
<thead>
<tr>
<th>Sets:</th>
</tr>
</thead>
<tbody>
<tr>
<td>( M )</td>
</tr>
<tr>
<td>( M_i )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Parameters:</th>
</tr>
</thead>
<tbody>
<tr>
<td>( \alpha )</td>
</tr>
<tr>
<td>( s_{ii'} )</td>
</tr>
</tbody>
</table>

<table>
<thead>
<tr>
<th>Variables:</th>
</tr>
</thead>
<tbody>
<tr>
<td>( x_{ij} )</td>
</tr>
<tr>
<td>( y_{ii'j} )</td>
</tr>
<tr>
<td>( z_j )</td>
</tr>
</tbody>
</table>

\[
\begin{align*}
\min_{j \in M} & \sum_{j \in M} z_j \tag{4.5} \\
\text{s.t.} & \sum_{j \in M} x_{ij} = 1 \quad \forall i \in M \tag{4.6} \\
& y_{ii'j} \leq x_{ij} \quad \forall i, j \in M \text{ and } \forall i' \in M_i \tag{4.7} \\
& y_{ii'j} \leq x_{i'j} \quad \forall i, j \in M \text{ and } \forall i' \in M_i \tag{4.8} \\
& y_{ii'j} \geq x_{ij} + x_{i'j} - 1 \quad \forall i, j \in M \text{ and } \forall i' \in M_i \tag{4.9} \\
& \sum_{i \in M} \sum_{i' \in M_i} s_{ii'y_{ii'j}} \geq \alpha \sum_{i \in M} \sum_{i' \in M_i} y_{ii'j} \quad \forall j \in M \tag{4.10} \\
& x_{ij} \leq z_j \quad \forall i, j \in M \tag{4.11} \\
& x_{ij} \in \{0, 1\} \quad \forall i, j \in M \tag{4.12} \\
& y_{ii'j} \in \{0, 1\} \quad \forall i, j \in M \text{ and } \forall i' \in M_i \tag{4.13} \\
& z_j \in \{0, 1\} \quad \forall j \in M \tag{4.14} 
\end{align*}
\]

The objective function of the proposed integer linear program is to minimize the total number of nonempty clusters. Constraint (4.6) guarantees that each observation is assigned to exactly one cluster. Constraints (4.7)-(4.9) ensure that if observation \( i \) and observation \( i' \)
are included in cluster \( j \) then \( y_{ii'j} \) will take the value of one and zero otherwise. Specifically, Constraint (4.7) ensures that if \( x_{ii'j} = 0 \) then \( y_{ii'j} = 0 \). Constraint (4.8) ensures that if \( x_{ii'j} = 0 \) then \( y_{ii'j} = 0 \). Constraint (4.9) ensures that if \( x_{ij} = 1 \) and \( x_{i'j} = 1 \) then \( y_{ii'j} = 1 \). Constraint (4.10) ensures that the average consistency score of each non-empty cluster with at least two elements is at least \( \alpha \). Note that we assume that non-empty clusters with exactly one element always satisfy the consistency threshold. Finally, Constraint (4.11) ensures that if \( x_{ij} = 1 \) then cluster \( j \) is not empty, i.e., \( z_j = 1 \).

**Remark 4.1.** Although the acceptable threshold, i.e., \( \alpha \), can be set to any value by users, a reasonable choice of it is the error, i.e., \( 1 - \text{accuracy} \), of the EM model developed as that truly captures the inconsistency rate of the EM model.

### 4.4 Heuristic algorithm

Due to high number of pairwise comparisons and dynamic increases in the number of observations, the proposed model cannot be solved with off-the-shelf solvers in practice. To deal with this challenge we develop a heuristic approach named as Dynamic Clustering algorithm to solve the model. The details of Dynamic Clustering algorithm is represented in Algorithm 1.

In the process of Clustering, it is assumed that at iteration \( t \) all observations \( 1, \ldots, t - 1 \) have been processed and divided into clusters. Based on the EM model, at iteration \( t \), a new element/observation \( D_t \) is compared by all previous clustered elements, i.e., \( D_1, \ldots, D_{t-1} \). The results of EM model show matching and no-matching predictions between \( D_t \) and all previous clustered elements (see Line 5). Then, based on the prediction results, for each existing cluster, a score is computed. For instance, if cluster \( B_1 \) has 5 elements and \( D_t \) matches with 3 of them, the \( B_1 \) score could be considered as \( 3/5 \) while if cluster \( B_2 \) has 3 elements and \( D_t \) matches with 2 of them, the \( B_1 \) score could be considered as \( 2/3 \). After computing all clusters scores, the cluster with the highest score is identified (see Lines 7-10). If the highest score is greater than the acceptable threshold, i.e., \( \alpha \) in Table 4.1, \( D_t \) will be
assigned to its corresponding cluster (Line 11-12). However, if the highest score is less than a specified threshold, a new cluster will be created and \( D_t \) will be assigned to it (see Lines 13-15).

**Algorithm 1:** Dynamic clustering heuristic

```plaintext
1 Inputs: Data as list, Entity Matching (EM) as prediction function
2 List.CreateEmpty(Clusters)
3 List.CreateEmpty(ExploredData)
4 for \( D \in Data.Elements \) do
5     Predictions ← EM(D,ExploredData)
6     BestMatchScore ← 0
7     for \( B \in Clusters.Elements \) do
8         if BestMatchScore < Predictions.ComputeScore(B) then
9             BestMatchScore ← Predictions.ComputeScore(B)
10            BestMatch ← B
11     if BestMatchScore ≥ \( \alpha \) then
12         Clusters.BestMatch.Add(D)
13     else
14         Create(NewCluster)
15         Clusters.NewCluster.Add(D)
16     ExploredData.Add(D)
17 return Clusters
```

4.5 Case study

In the next section, we evaluate our proposed framework in Figure 4.3. We use Python programming language to implement the framework. It is worth mentioning that although the proposed framework looks simple, its implementation is not trivial. This is because the implementation contains two important pieces that need to be linked together. The first piece is an effective EM model that needs to be developed. For any given pair of observations, the EM model should predicts whether they represent the same entity or not, i.e., match or no-match. In order to avoid reinventing the wheel, we employ Magelland for developing and training an EM model. The Magelland python package has access to a variety of machine learning models and can automatically generate additional useful attributes (based on the
original set of features in the dataset) proven to be useful in many studies in the relevant literature [112]. The typical machine learning models that can be employed within Magellan are Support Vector Machines (SVM), Logistic Regression (LR), and Random Forests (RF). So, our first goal in this section is to show that Magellan can be indeed accurate for the purpose of EM in the mobile advertising domain.

The second piece is the clustering mechanism. For the purpose of clustering, two approaches can be used including our proposed integer linear programming model and our proposed heuristic approach. The main weakness of the integer programming model is that it is impractical. Its size is a quadratic function of the number of observations that need to be clustered. Therefore, even commercial solvers will struggle to solve the model for a few hundreds of observations. The proposed heuristic, on the other hand, is expected to be very fast but its implementation is not trivial as it needs to be connected to Magellan and also its accuracy is unknown. So, if after overcoming the implementation barrier, the experimental results show that the proposed heuristic approach is effective then one can conclude that the approach is indeed practical. So, our second goal in this section is to show the accuracy of our proposed heuristic approach for the purpose of clustering.

4.5.1 Data description

Our mobile advertising dataset spans over 31 days from May 01, 2020 until May 31, 2020. In total, about 32,000 observations are available in the dataset. Each observation represents a user and its corresponding attributes that have appeared during the advertising campaign at a particular time. While the dataset contains 73 columns, some of them contain only NA values or a given constant. Therefore, they can be removed and by so doing, the total number of columns reduces to 41; one of which is Device ID that will be removed during our experiments as the goal of this section is to assume that Device ID is unknown. We only use the Device ID for measuring the accuracy of the EM model and our proposed clustering
approach. The summary statistics of the dataset can be found in Table 4.2; the dataset is converted to numeric in order to create the table.

Table 4.2 – Summary statistics of the dataset

<table>
<thead>
<tr>
<th>Attributes</th>
<th>Mean</th>
<th>Std</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td>1. asn</td>
<td>94.12</td>
<td>76.66</td>
<td>0</td>
<td>268</td>
</tr>
<tr>
<td>2. Datetime (date &amp; time for the first received signal)</td>
<td>590.6</td>
<td>365.93</td>
<td>0</td>
<td>1,393</td>
</tr>
<tr>
<td>3. Device name</td>
<td>27.43</td>
<td>35.89</td>
<td>0</td>
<td>144</td>
</tr>
<tr>
<td>4. Device vendor</td>
<td>4.045</td>
<td>35.89</td>
<td>0</td>
<td>21</td>
</tr>
<tr>
<td>5. Device manufacturer</td>
<td>3.81</td>
<td>2.76</td>
<td>0</td>
<td>17</td>
</tr>
<tr>
<td>6. Device model</td>
<td>29.82</td>
<td>39.23</td>
<td>0</td>
<td>158</td>
</tr>
<tr>
<td>7. Device year of release</td>
<td>3.34</td>
<td>2.02</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>8. Device hardware</td>
<td>6.34</td>
<td>9.09</td>
<td>0</td>
<td>35</td>
</tr>
<tr>
<td>9. Platform of a device</td>
<td>1.06</td>
<td>0.93</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>10. MajorOS (device operating system)</td>
<td>3.31</td>
<td>2.44</td>
<td>0</td>
<td>12</td>
</tr>
<tr>
<td>11. Device category (e.g. smartphone and tablet)</td>
<td>0.14</td>
<td>0.35</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>12. Connection type (e.g. cellular and wifi)</td>
<td>0.65</td>
<td>0.48</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>13. Carrier (Internet service provider)</td>
<td>4.99</td>
<td>5.59</td>
<td>0</td>
<td>36</td>
</tr>
<tr>
<td>14. Browser language</td>
<td>0</td>
<td>0.02</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>15. IP address</td>
<td>535.9</td>
<td>297.60</td>
<td>0</td>
<td>1,042</td>
</tr>
<tr>
<td>16. User agent</td>
<td>99.5</td>
<td>93.50</td>
<td>0</td>
<td>325</td>
</tr>
<tr>
<td>17. Store Url</td>
<td>94.84</td>
<td>77.29</td>
<td>0</td>
<td>270</td>
</tr>
<tr>
<td>18. Bundle (a platform-specific application identifier)</td>
<td>94.47</td>
<td>76.78</td>
<td>0</td>
<td>268</td>
</tr>
<tr>
<td>19. categoryAsn</td>
<td>11.8</td>
<td>10.98</td>
<td>0</td>
<td>53</td>
</tr>
<tr>
<td>20. mwCarrierName (the DSP Internet Carrier Name)</td>
<td>1.79</td>
<td>2.21</td>
<td>0</td>
<td>24</td>
</tr>
<tr>
<td>21. PublisherId</td>
<td>41.5</td>
<td>47.49</td>
<td>0</td>
<td>173</td>
</tr>
<tr>
<td>22. Publisher name</td>
<td>41.14</td>
<td>46.86</td>
<td>0</td>
<td>171</td>
</tr>
<tr>
<td>23. MaxMind city</td>
<td>0.42</td>
<td>1.55</td>
<td>0</td>
<td>9</td>
</tr>
<tr>
<td>24. MaxMind zip</td>
<td>9.53</td>
<td>10.48</td>
<td>0</td>
<td>54</td>
</tr>
<tr>
<td>25. MaxMind latitude</td>
<td>9.54</td>
<td>10.53</td>
<td>0</td>
<td>53</td>
</tr>
<tr>
<td>26. MaxMind_itude (MaxMind longitude)</td>
<td>9.44</td>
<td>10.34</td>
<td>0</td>
<td>54</td>
</tr>
<tr>
<td>27. MaxMind connectionType</td>
<td>1.29</td>
<td>0.95</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>28. Brq city (The city mobile signals observed)</td>
<td>0.21</td>
<td>0.93</td>
<td>0</td>
<td>10</td>
</tr>
<tr>
<td>29. Brq state (The state mobile signals observed)</td>
<td>0</td>
<td>0.01</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>30. Brq_itude (the longitude mobile signals observed)</td>
<td>139.4</td>
<td>211.67</td>
<td>0</td>
<td>882</td>
</tr>
<tr>
<td>31. Brq_latitude (the latitude mobile signals observed)</td>
<td>174.3</td>
<td>269.52</td>
<td>0</td>
<td>1,116</td>
</tr>
<tr>
<td>32. Brq type</td>
<td>0.33</td>
<td>0.47</td>
<td>0</td>
<td>2</td>
</tr>
<tr>
<td>33. Brq zip (the zip mobile signals observed)</td>
<td>7.09</td>
<td>9.83</td>
<td>0</td>
<td>51</td>
</tr>
<tr>
<td>34. Brq country (the country mobile signals observed)</td>
<td>0</td>
<td>0.01</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>35. User gender</td>
<td>1.04</td>
<td>0.90</td>
<td>0</td>
<td>3</td>
</tr>
<tr>
<td>36. User yob (user year of birth)</td>
<td>1.36</td>
<td>3.36</td>
<td>0</td>
<td>16</td>
</tr>
<tr>
<td>37. User keyword (list of keywords requested by a user)</td>
<td>495.5</td>
<td>983.58</td>
<td>0</td>
<td>3,528</td>
</tr>
<tr>
<td>38. User consent</td>
<td>0</td>
<td>0.10</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>39. Request carrier</td>
<td>2.08</td>
<td>3.10</td>
<td>0</td>
<td>32</td>
</tr>
<tr>
<td>40. Request connectionType</td>
<td>1.49</td>
<td>0.80</td>
<td>0</td>
<td>5</td>
</tr>
</tbody>
</table>
4.5.2 Results

Although our dataset contains around 32,143 observations, many of them represent the same entity/user. Specifically, the number of non-identical users is 685 in our dataset. According to the proposed framework, we randomly select 200 of these observations and use them to train an EM model (through Magellan), i.e., identifying whether a given pair of observations represents the same entity or not. Note that 200 observations creates 40,000 pairs from which 70%, i.e., 28000 pairs, will be selected for the training of the entity matching model and 30%, i.e., 12000 pairs, for testing it. The performance of the EM model obtained by Magellan can be found in Table 4.3. As mentioned earlier the EM can be developed based on SVM, RF, and LR in Megallan. Therefore, in Table 4.3, the performance of EM model for all three settings are reported.

Table 4.3 – Overall performance

<table>
<thead>
<tr>
<th>Setting (in Magellan)</th>
<th>EM Model</th>
<th>Clustering Error</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td>Precision</td>
<td>Recall</td>
</tr>
<tr>
<td>RF</td>
<td>99.1%</td>
<td>99.1%</td>
</tr>
<tr>
<td>SVM</td>
<td>99.6%</td>
<td>99.7%</td>
</tr>
<tr>
<td>LR</td>
<td>99.5%</td>
<td>99.5%</td>
</tr>
</tbody>
</table>

Note that in our testing set (i.e., 12000 pairs) about 8.1% of the pairs have the value of one and the remaining 91.9% pairs have the value of zero. So, the accuracy value of 99.9% obtained by SVM-or-LR-based EM model is around 8% larger than predicting the value of zero for all pairs in the testing set, i.e., 99.9%-91.9%=8%. Moreover, the recall value of 99.7% obtained by the SVM-based EM model is quite impressive as it shows that although only 8.1% of the pairs have the value of one, the SVM-based EM model is able to correctly identify 99.7% of them.

According to the proposed framework, after obtaining the EM model, clustering needs to be done on the remaining dataset, i.e., 31,943 observations. In order to do the clustering, the obtained EM model will be used within our proposed heuristic approach. The error of
clustering using our proposed is also shown in Table 4.3. Note that computing the error of clustering is not a trivial task. In order to obtain the error, we use the following procedure:

- For each non-identical entity \( i \in \{1, \ldots, 685\} \), there can be many observations in the dataset. Let \( C^i_1, \ldots, C^i_K \) be the clusters that they contain at least one of the observations representing entity \( i \). Note that \( K \) is not necessary equal to one.

- Let \( r^i_1, \ldots, r^i_K \) be the number of observations representing entity \( i \) in clusters \( C^i_1, \ldots, C^i_K \), respectively. Moreover, let \( r^i_1, \ldots, r^i_K \) be the number of observations not representing entity \( i \) in clusters \( C^i_1, \ldots, C^i_K \), respectively.

- We denote the number of observations predicted incorrectly for entity \( i \) by \( e_i \) and compute it as follows
  - For each \( k \in \{1, \ldots, K\} \), add \( r^i_k \) to \( e_i \) if \( \frac{r^i_k}{r^i_1} \leq 1 \). This is because in that case the majority of observations in cluster \( k \) does not represent entity \( i \).
  - Let \( K = \{ k \in \{1, \ldots, K\} : \frac{r^i_k}{r^i_1} > 1 \} \) be the indices of clusters that the majority of their observations represent entity \( i \). Only the cluster with the maximum observations from entity \( i \) is correctly identified by the proposed heuristic; the other ones are all incorrect because there cannot exist multiple clusters for a given entity. In other words, add \( \sum_{k \in K} r^i_k - \max_{k \in K} r^i_k \) to \( e_i \).

- In light of the above, the error of the heuristic clustering approach for the case study is computed as follows:
  \[
  \sum_{i=1}^{685} e_i 
  \]
  \[
  31,943.
  \]

Overall, we observe from Table 4.3 that the error of the proposed clustering heuristic with the SVM-based EM model is around 2%. This is quite promising and shows that the proposed heuristic is effective. This effectiveness is partly due to the SVM-based EM model. In order to identify which features were most effective for EM, we conduct a SHAP-value analysis. As mentioned the number of attributes is 40 in our case study. When comparing
any pair of observations in Magellan, the number of attributes will be doubled: 40 attributes for the left observation (referred to as ltable) and 40 observations for the right observation (referred to as rtable). Note that we calculate the SHAP values based on the same training set used for training the EM model, i.e., 28,000 pairs. The SHAP values are illustrated in Figure 4.5. We observe that the five attributes are device-hwv, mwCarrierName, publisher-Name, publisherId, catagoryAsn which denote the hardware version of the device, Internet carrier (e.g."VERIZON), the name of the publisher of the media in which the ad will be displayed, the exchange-specific publisher ID, and the description of the content producer, respectively.
Figure 4.5 – SHAP values in the EM model
Chapter 5: Conclusions and future research directions

In this section we present conclusions and future research directions of each chapter.

5.1 Conclusion of Chapter 2

In Chapter 2 of this dissertation, we developed a methodology for selecting the best subset of features in the context of online advertising and, more broadly in the field of information systems. The proposed method is based on a bi-objective optimization problem that trades off “fit” and “bias”. The proposed method employs the concept of Nash bargaining solution in cooperative game theory to create a good balance between the value of both objectives when selecting the best subset of features. We applied the proposed method on a real-life online advertising case study providing evidence of superior performance both in predictions and in interpretation of the features. Moreover, to investigate that the proposed method is applicable to a broader range of feature selection problems, we also conducted a comprehensive computational study on both simulated regression datasets and other real-life classification datasets widely available in the machine learning domain. The result of these effort indicates that our method is pretty robust in terms of prediction accuracy by outperforming several state-of-the-art techniques. We hope that the provided evidence in support of our approach motivate further research in the interface between game theory and feature selection problems.

5.2 Conclusion of Chapter 3

In Chapter 3 of this dissertation, we have demonstrated that instead of using all available targeting attributes for advertisements to boost in-store visits, a better strategy is to
use a few attributes that have the best impact. The selection of these few effective attributes for maximizing ROI is dependent on the target user’s location. We have proposed a complete end-to-end approach of the selection of these attributes along with a ROI maximization model and a heuristic solution approach. Furthermore, through a real-world case study, we have demonstrated the practical relevance of our proposed campaign design approach.

The application of our approach on the case study has provided several insights for companies and advertisers about effective execution of targeted mobile advertising for increasing in-store visits. One important insight, among others, is that to improve the ROI, companies should consider both demographics and distance-oriented attributes together rather than one of them alone. Although both demographics and distance attributes are marked as two important attributes in the mobile advertisement literature, incorporating only one of them can result in significant reduction in ROI whereas their interactions together can improve the ROI substantially (Table 3.5).

Our study also provides multiple avenues for future research. Future research should consider user-related environmental and device-related technological attributes such as weather, socioeconomic parameters, and delivery mechanisms of mobile advertising for better understanding of ad targeting. There is also a need to consider how the trade-offs between costs to acquire targeting data and the benefits of improved in-store visits vary across different business sectors and cultural contexts. We believe that our study’s methodological approach lays a good foundation for such future research efforts.

5.3 Conclusion of Chapter 4

In chapter 4 of this dissertation, we developed a methodology to resolve the curse of identification. Specifically, for delivering customized ads, a typical RTB needs to use device IDs. However, due to appropriate data protection and privacy safeguard for using device IDs, there is a huge challenge for the RTB to access the right customer at the right time.
Therefore, in this study we proposed a framework that helps ad agencies identify unique customers in the RTB ecosystem.

The proposed framework has two main parts: (1) EM and (2) Clustering. EM is used to employ a learning-based method that does pairwise comparison between data instances to predict whether they “match” or “no-match”. Clustering is based on pairwise comparison results of EM, and it seeks to group the data instances by minimizing the inconsistencies observed in the EM outcomes. Our numerical results on a real-world case study reveals that our framework can identify unique customers pretty accurately and its error is only 2.0%. Our results are quite promising and show that our proposed framework has the potential to help ad agencies to target unique customers in RTB ecosystem without having access to their device IDs.
References


[120] George Papadakis, George Mandilaras, Luca Gagliardelli, Giovanni Simonini, Emmanouil Thanos, George Giannakopoulos, Sonia Bergamaschi, Themis Palpanas, and


Appendix 1: Proof of Theorem 2.1

**Theorem.** GT-Con is equivalent to the GT-Con-SOCP.

**Proof.**

Observe that GT-Con can be simplified by introducing a new non-negative variable, i.e., \( \gamma \), and an additional constraint as follows,

\[
\text{(GT-Con-1)} \quad \max_{\beta \in \mathbb{R}^p} \left[ q_1 - \gamma^2 \right] \left[ q_2 - \sum_{j=1}^{p} |\hat{\beta}_j| \right]
\]

such that:

\[
\sum_{i=1}^{n} (y_i - \sum_{j=1}^{p} x_{ij} \hat{\beta}_j)^2 \leq q_1,
\]

\[
\sum_{j=1}^{p} |\hat{\beta}_j| \leq q_2,
\]

\[
\sum_{i=1}^{n} (y_i - \sum_{j=1}^{p} x_{ij} \hat{\beta}_j)^2 \leq \gamma^2,
\]

\( \gamma \geq 0 \).

Since the objective function is in the form of maximization, the model naturally attempts to minimize \( \gamma^2 \). Therefore, for any optimal solution, we must have that,

\[
\sum_{i=1}^{n} (y_i - \sum_{j=1}^{p} x_{ij} \hat{\beta}_j)^2 = \gamma^2.
\]

Observe that the objective function of GT-Con-1 can be written as follows,

\[
\left[ \sqrt{q_1} - \gamma \right] \left[ \sqrt{q_1} + \gamma \right] \left[ q_2 - \sum_{j=1}^{p} |\hat{\beta}_j| \right].
\]
In this case, the objective function will have three linear pieces, and each one can only take non-negative values for any feasible solution because of the first two constraints of $\text{GT-Con-1}$. For simplicity, we introduce three auxiliary non-negative variables, $f_1$, $f_2$, and $f_3$, to capture each linear piece of the objective function. In that case, $(\text{GT-Con-1})$ can be stated as follows,

$$(\text{GT-Con-2}) \quad \max_{\beta \in \mathbb{R}^p} f_1 f_2 f_3$$

such that:

$$\sum_{i=1}^{n} (y_i - \sum_{j=1}^{p} x_{ij}\hat{\beta}_j)^2 \leq q_1,$$

$$\sum_{j=1}^{p} |\hat{\beta}_j| \leq q_2,$$

$$\sum_{i=1}^{n} (y_i - \sum_{j=1}^{p} x_{ij}\hat{\beta}_j)^2 \leq \gamma^2,$$

$$f_1 = \sqrt{q_1} - \gamma,$$

$$f_2 = \sqrt{q_1} + \gamma,$$

$$f_1 = q_2 - \sum_{j=1}^{p} |\hat{\beta}_j|$$

$$f_1, f_2, f_3 \geq 0.$$

In the next step, we linearize the absolute value function using a similar approach that we used in Section 2.1.1 for computing $q_2$. So, $\text{GT-Con-2}$ can be reformulated as follows,

$$(\text{GT-Con-3}) \quad \max_{\beta \in \mathbb{R}^p} f_1 f_2 f_3$$

such that:

$$\sum_{i=1}^{n} (y_i - \sum_{j=1}^{p} x_{ij}\hat{\beta}_j)^2 \leq q_1,$$

$$\sum_{j=1}^{p} \hat{r}_j \leq q_2,$$

$$\sum_{i=1}^{n} (y_i - \sum_{j=1}^{p} x_{ij}\hat{\beta}_j)^2 \leq \gamma^2,$$

$$f_1 = \sqrt{q_1} - \gamma,$$
\[ f_2 = \sqrt{q_1 + \gamma}, \]
\[ f_1 = q_2 - \sum_{j=1}^{p} \hat{r}_j, \]
\[ \hat{\beta}_j \leq \hat{r}_j \quad \forall \; j \in \{1, \ldots, p\}, \]
\[ -\hat{\beta}_j \leq \hat{r}_j \quad \forall \; j \in \{1, \ldots, p\}, \]
\[ \hat{r}_j \geq 0 \quad \forall \; j \in \{1, \ldots, p\}, \]
\[ f_1, f_2, f_3 \geq 0. \]

Note that since the objective function is in the form of maximization, the model naturally attempts to maximize \( f_1 \) and this itself implies that \( \sum_{j=1}^{p} \hat{r}_j \) will be naturally minimized. Hence, for any optimal solution of \textbf{GT-Con-3}, we must have that, \( \hat{\beta}_j = |\hat{\beta}_j| \) for each \( j \in \{1, \ldots, p\} \). In other words, \textbf{GT-Con-3} is an exact reformulation of \textbf{GT-Con-2}. Next, we attempt to simplify the objective function of \textbf{GT-Con-3}. Observe that, the following problem,

\[ \max \{ f_1 f_2 f_3 : \; f_1, f_2, f_3 \geq 0 \}, \]

is equivalent to,

\[ \max \{ \Gamma : \; \Gamma \leq \sqrt{f_1 f_2 f_3}, \text{ and } \Gamma, f_1, f_2, f_3 \geq 0 \}, \]

where \( \Gamma \) is a non-negative variable that captures the value of \( \sqrt{f_1 f_2 f_3} \) for any optimal solution. Based on the work of Ben-Tal and Nemirovski [15], an equivalent model for the above model can be constructed by introducing two new non-negative variables, i.e., \( v_1 \) and \( v_2 \), and three constraints as follows,

\[ \max \{ \Gamma : \; \Gamma^2 \leq v_1 v_2, \; v_1 \leq f_1 f_2, \; v_2 \leq f_3 \Gamma, \; \text{and} \; v_1, v_2, \Gamma, f_1, f_2, f_3 \geq 0 \}. \]
All the constraints of the above model are (convex) quadratic constraints and the objective function is linear. Based on this observation, \textbf{GT-Con-3} can be reformulated as follows,

\[(\text{GT-Con-SOCP}) \quad \max_{\hat{\beta} \in \mathbb{R}^p} \Gamma \]

such that:

\[\sum_{i=1}^{n} (y_i - \sum_{j=1}^{p} x_{ij} \hat{\beta}_j)^2 \leq q_1,\]

\[\sum_{j=1}^{p} \hat{r}_j \leq q_2,\]

\[\sum_{i=1}^{n} (y_i - \sum_{j=1}^{p} x_{ij} \hat{\beta}_j)^2 \leq \gamma^2,\]

\[f_1 = \sqrt{q_1} - \gamma,\]

\[f_2 = \sqrt{q_1} + \gamma,\]

\[f_1 = q_2 - \sum_{j=1}^{p} \hat{r}_j,\]

\[\Gamma^2 \leq v_1 v_2,\]

\[v_1^2 \leq f_1 f_2,\]

\[v_2^2 \leq f_3 \Gamma,\]

\[\hat{\beta}_j \leq \hat{r}_j \quad \forall \ j \in \{1, \ldots, p\},\]

\[-\hat{\beta}_j \leq \hat{r}_j \quad \forall \ j \in \{1, \ldots, p\},\]

\[\hat{r}_j \geq 0 \quad \forall \ j \in \{1, \ldots, p\},\]

\[v_1, v_2, \Gamma, f_1, f_2, f_3 \geq 0.\]
Appendix 2: Attributes to target users (Step 1)

We now describe a list of attributes that can be derived from the incoming RTB bid requests or through the DMP. Note that in practice DSPs can select attributes from a large array of choices in a data mart for executing mobile campaigns. Our goal is not to exhaustively present all the possible attributes but to present a selected set of attributes that have been identified as important in prior research and those that we encountered in our real-world case.

We consider a set of important attributes that DSPs purchase directly from data management platforms (DMPs) such as Oracle BlueKai [19], Mobilewalla [104], Lotame [70] or utilize their own cloud data storage and computational infrastructure to process them. Below we explain these attributes.

- **Distance of home and workplace of users from the BM store**: The home and workplace locations of users are typically not known. So, we apply a heuristic to derive the home and workplace location of users. The DSP receives a continuous stream of RTB bid requests from the RTB exchange. The device ID can be used to uniquely identify users over time across multiple bid requests. Although the device ID is a virtual ID that can be reset by users, users are seldom aware of this feature and even if they are aware, they do not take the effort to reset it frequently. Therefore, the device ID received by the RTB through the SSP can be considered the unique userid and is passed from the RTB to the DSP as part of the bid requests. In the mobile advertising space, this device ID is used to uniquely identify a user. By analyzing the users’ locations and times, which come as part of the RTB bid requests, we heuristically derive the home and work locations. If we see more
than 3 occurrences of a user’s device in approximately the same location (within 100 meters) between 8 PM and 7 AM, we label that location as the home of the user.

Similarly, if we see more than 3 occurrences of a user’s device in approximately the same location (within 100 meters) between 8 AM and 5 PM we label that location as the work location of the user. For a small group of users (who work at night shift) this assignment may switch (i.e. the home is identified as the work location and vice versa). We compute the distance of the BM store from the user’s home and work locations. As can be seen from the above discussion, the computation of the user’s home and work locations’ distance from the BM store is not trivial. It would require collecting historical data from RTB bid requests and computing on top of it. A typical DSP receives bid requests at the rate of 200K-500K requests/seconds. Thus, the computation of these two distances would require considerable storage space, and management of such a large volume of data is costly. Additionally, the computation of work and home distances from BM stores for consumers would require massive parallel data analytics tasks that involve map-reduce and Spark [36], the costs of which are non-trivial for a DSP. For example, the medium size DSP with whom we have worked pays 200K USD per month to AWS for cloud services. Therefore, we need to optimally decide the group of consumers for which distances need be computed. This is the decision problem that is described in the following section. We denote the cost of computing work and home location distances as $c_w$ and $c_h$.

- **Effects of Competitors**: We use a binary variable to indicate whether the customer has visited any of the competing stores before receiving the advertisement. A customer’s visit to a competitor store is estimated from the RTB bid request by measuring the distance of the customer’s location (obtained from the RTB bid request) from the competing store. If the distance is below a certain threshold ($< 3$ km for our case) we assume that the customer has visited the competing store. We anticipate the customer’s visit to the competitor would have some impact of on the effect of the advertisement for the BM store.
Similar to the computation of the customer’s home and work location distances, the computation of the customer’s visit to the competitor store will require managing the historical RTB bid request data and running analytics on this data. Additionally, for competitor visits this is a continuous process. The RTB bid request is a stream and this is implemented using Spark Streaming [36], which would require additional resources beyond those required for the computation of the customer’s work and home location distance. We denote the cost of computing a customer’s competitor store visit by \( c_{\text{comp}} \). From the above discussion we can conclude \( c_{\text{comp}} \geq c_h = c_w \).

- **Previous visits:** We define a binary variable to indicate whether a mobile user has visited the BM store before or not. From the RTB bid request stream, we compute the distance of the user from the BM store. If the distance is below a certain threshold, we can conclude that the customer has visited the store. Threshold distance would depend on the location of the BM store. If the BM store is in a locality outside the city perimeter with no other retail facilities around it, 3 km distance is logical enough to identify a customer’s visit to the BM store.

To compute this binary variable we need to store and manage the historical RTB bid requests. Additionally, we need to identify the new visit to stores in a continuous manner through Spark stream, similar to the determination of the competitor store visit. Thus, if we denote the cost of determining this binary variable indicating the past visit to the BM store as \( c_{\text{st}} \), we can say \( c_{\text{st}} = c_{\text{comp}} \).

- **Mobile operating systems:** To indicate the affinity of a user to the retail store based on whether he is an Android or iOS device user, we use a binary variable, where 1 indicates the user is an Android user and 2 indicates the user is an iOS user. As the RTB bid requests include the browser agent as a field in the bid request, the process of determining the mobile operating system from the RTB bid request is straightforward. Thus, if we denote the cost of identifying the mobile OS as \( c_{\text{os}} \), we can conclude \( c_{\text{os}} < c_w = c_h < c_{\text{st}} = c_{\text{comp}} \).
• **Demographics and customer profiles:** DSPs need to purchase these from DMPs. For example, Lotame charges in the range of $1.00 – $1.50 per user [95]. As per Lotame’s pricing scheme, the cost of acquiring demographic and customer profiles increases with the increase in the number of profile dimensions. We can conclude that in the case of Lotame, the average price of each demographic and profile attribute per user is $1.25. The other DMPs in the market have a similar pricing scheme. Thus, it becomes important for DSPs to choose the right attributes for the right group of users for targeting purposes. If the addition of the attributes does not improve the ROI of the advertisement, the investment in DMP data is a loss to the DSPs. If we denote $c_{dm}$ as the cost of each demographic attribute, we have, $c_{os} < c_w = c_h < c_{st} = c_{comp} < c_{dm}$.

For the purpose of this research we consider three demographic attributes: age, gender and income. Additionally, we use the user profile that has five dimensions: occupation, favorite product categories (such as books, garden, tools etc.), indicator of fast moving customers and indicator of whether the customer likes sport activities. If we denote the cost of fetching each of these attributes as $c_p$, we have, $c_{os} < c_w = c_h < c_{st} = c_{comp} < c_{dm} = c_p$.

• **Time category:** We divide a day into four different time categories: morning, afternoon, evening and night. The day interval in which a mobile advertisement is delivered by the DSP to the mobile device of users may have an effect on the ROI of the advertisement, which is a BM store visit in our case. The day interval can be derived from the RTB bid request, as it arrives at the DSP. If we consider the cost of deriving the day interval as $c_d$, we have, $c_d = c_{os} < c_w = c_h < c_{st} = c_{comp} < c_{dm} = c_p$.

• **Current location of users:** As customers adopt GPS-enabled devices, more marketers apply location-based advertising [98]. The incoming RTB bid requests to the DSP include current latitude and longitude of the user. This can be used to measure the distance from the target BM store using the haversine formula [30]. If we consider the cost of deriving the day interval as $c_l$, we have, $c_l < c_d = c_{os} < c_w = c_h < c_{st} = c_{comp} < c_{dm} = c_p$. 

111
From the above discussion it is clear that the various attributes that may be used in targeting an advertisement have different data management and acquisition costs. Each of these attributes would also have different success rates in identifying (targeting) users who will provide the expected return from the advertisement.
Appendix 3: Data description

Our data is drawn from MyStore mobile advertising for both exposed and unexposed devices between 31 March 2017 and 18 June 2017. The advertisement was delivered to mobile users through in-app advertisements within 125000 meters of the MyStore and mobile users were targeted randomly in the advertising campaign. So, we have two datasets: exposed dataset and unexposed dataset. Figure A.1 shows the distribution of store visit by users based on users’ distances at the time their presence was picked up by the RTB system in both datasets. As can be seen for both exposed and unexposed users, the number of users who visit MyStore decreases substantially beyond 21000 meters. Also, MyStore is a large store and there is no locality around it up to 5km. Therefore, for the purpose of this case study, we consider only mobile users whose distances from MyStore are in the range of [5000, 21000] meters.

![Graphs showing distribution of users](image)

(a) Distribution of exposed users’ who visit MyStore

(b) Distribution of unexposed users’ who visit MyStore

Figure A.1 – Distribution of users from MyStore

Other than distance, the dataset provided to us by the DSP firm also includes attributes of users such as age, income, and customer favorite products. Additional attributes of mobile
users are computed as explained in Section 5.3. Table A.1 reports the summary statistics of all attributes for exposed and unexposed users. We represent the categorical attributes with dummy variable.

In this table, for non-numeric/dummy variables, we assign integer values to each choice/option. We assign the value of 0 to the first choice, 1 to the second choice, and so on. Note that for some variables, “N/A” is a feasible option. For such a case, we assume that “N/A” is the first choice, i.e., “N/A” always takes the value of 0.

Appendix 3.1: Impact of advertisement delivery

In this study, we mainly use our exposed dataset. However, the unexposed dataset is helpful for showing that the advertisement delivery is indeed useful. Specifically, in order to understand the importance of advertisement delivery and whether the mobile ad increases the probability of a BM store visit, we compare exposed and unexposed customers. We identify the store visits by a mobile user by his/her presence within 3 km of MyStore. The presence of the mobile user is identified by the location information received at the RTB from the mobile user’s device. The location of the BM store is in the outskirts of a densely populated city and there are no other major destinations within 5 km of the BM store. This indirect way of identifying the store visit would suit this case study. As direct attribution to online advertising is difficult to obtain (other than providing value coupons), this is a very common strategy in mobile advertising to identify the user’s visit to stores. In this case, we assume that if a mobile user has visited the MyStore within 1 week of being exposed to the advertisement, then this visit is attributed to the advertisement. MyStore is a very large BM retail store and is known to be a popular place to visit especially during the weekends. Therefore, the 1-week time period for assessing the outcome of the ad exposure ensures that at least one weekend will be within the time horizon.

As can be seen from Figure A.2, the proportion of exposed users who visit MyStore is much higher than the proportion of unexposed users who visit MyStore at same distance
Table A.1 – Summary statistics

<table>
<thead>
<tr>
<th>Variable</th>
<th>Mean</th>
<th>SD</th>
<th>Min</th>
<th>Max</th>
</tr>
</thead>
<tbody>
<tr>
<td><strong>Exposed</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Distance (in meters)</td>
<td>9460.4</td>
<td>5646.382</td>
<td>265.1</td>
<td>21000</td>
</tr>
<tr>
<td>Time (morning, evening, afternoon, night)</td>
<td>1.959</td>
<td>0.923</td>
<td>1</td>
<td>4</td>
</tr>
<tr>
<td>Gender (Male,Female)</td>
<td>1.2</td>
<td>0.732</td>
<td>0*</td>
<td>2</td>
</tr>
<tr>
<td>Age (13-18, 19-24, 25-34, 35-49, Older(50+))</td>
<td>3.378</td>
<td>1.160</td>
<td>0*</td>
<td>5</td>
</tr>
<tr>
<td>Income (low, average, high)</td>
<td>1.574</td>
<td>0.972</td>
<td>0*</td>
<td>3</td>
</tr>
<tr>
<td>Customer’s occupation(FIP†,CIP†,TIP†,MIP†)</td>
<td>2.125</td>
<td>1.352</td>
<td>0*</td>
<td>4</td>
</tr>
<tr>
<td>Fast moving customer ( no,yes)</td>
<td>0.956</td>
<td>0.206</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Customer does sporting activities ( no,yes)</td>
<td>1.569</td>
<td>0.822</td>
<td>0*</td>
<td>2</td>
</tr>
<tr>
<td>Customer favorite products categories (books, garden, tools, etc.)</td>
<td>3.456</td>
<td>3.148</td>
<td>0*</td>
<td>7</td>
</tr>
<tr>
<td>Previous visits to MyStore (no, yes)</td>
<td>0.0149</td>
<td>0.121</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Home distance (in meters)</td>
<td>26530</td>
<td>27930.25</td>
<td>0*</td>
<td>13960000</td>
</tr>
<tr>
<td>Work distance (in meters)</td>
<td>31465</td>
<td>33693.75</td>
<td>0*</td>
<td>19210000</td>
</tr>
<tr>
<td>Mobile operating systems (Android, iOS)</td>
<td>0.992</td>
<td>0.713</td>
<td>0*</td>
<td>2</td>
</tr>
<tr>
<td>Visits competitor1 after receiving Ad (no, yes)</td>
<td>0.073</td>
<td>0.261</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Visits competitor2 after receiving Ad (no, yes)</td>
<td>0.053</td>
<td>0.224</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Visits competitor3 after receiving Ad (no, yes)</td>
<td>0.004</td>
<td>0.068</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Previous visits to competitor1 (no, yes)</td>
<td>0.167</td>
<td>0.373</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Previous visits to competitor2 (no, yes)</td>
<td>0.103</td>
<td>0.304</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>Previous visits to competitor3 (no, yes)</td>
<td>0.015</td>
<td>0.123</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td><strong>Unexposed</strong></td>
<td></td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>Distance (in meters)</td>
<td>7440</td>
<td>5151.398</td>
<td>244.6</td>
<td>20990</td>
</tr>
</tbody>
</table>

* The corresponding information is not available.
†FIP=finance industry professionals; †CIP=computer industry professional; †TIP=technology industry professionals; †MIP=media industry professionals.
intervals. The proportion of mobile users who visit MyStore decreases with the distance from the MyStore location for both exposed and unexposed customers. However, beyond 10,000 meters, that proportion is almost 0 in case of unexposed mobile users, whereas for exposed mobile users between 10,000 and 15,000 meters about 10% visited the store following the advertisement. Thus, we conclude positive impact of advertisement on store visit.

![Figure A.2 – Distribution of users who visited MyStore](image)

Appendix 3.2: Impact of location on effectiveness of advertisement

From Figure A.2, it can be inferred that as the distance increases between MyStore and mobile users at the time of delivery of the advertisement, the proportion of store visits decreases significantly. A similar result can be observed for unexposed users, as seen in Figure A.2 based on the distance of the customers at the time it is observed in the RTB incoming bid. However, it needs to be noted that the slope of the downward trend is sharper in the case of exposed mobile users than for unexposed mobile users. This indicates that the positive impact of mobile advertisements on store visits decreases with distance. We conclude that, location plays an important role in mobile advertising and this is compatible with the findings of multiple previous studies [106, 13, 61, 58].
Appendix 3.3: Heterogeneity due to Additional Attributes

To understand the impact of additional attributes on the campaign performance along with location, we employ the logistic regression. We present the results on the coefficients of the main model in Table A.2. The first three columns show the effect of attributes (e.g. gender, age, home distance, and previous visits of competitors) in determining effectiveness of ads. Consistent with prior evidence on the location effect, the second column of the table shows that the distance is the most important attribute in determining performance of the ad (given that the coefficient of distance is -7.42 and it is higher than coefficients of other attributes). Furthermore, distance has a significant negative impact on the probability of visiting MyStore ($\beta = -7.42, p < 0.0001$). Increasing distance between the store and user’s location are assumed to increase the underlying traveling costs, which can have a negative impact on the users’ preferences towards the store [107].

Besides distance, other attributes also play a significant role. Specifically, the estimate for competitors is negative and statistically significant ($\beta = -4.81, p < 0.0001$ and $\beta = -5.01, p < 0.0001$ respectively). Similarly for demographic-oriented attributes such as gender, female group has negative significant coefficient ($\beta = -2.13, p < 0.0001$) while people with average income have positive significant coefficient ($\beta = 0.86, p < 0.05$). So, comparing to other groups, people with average income have higher tendency to visit MyStore after receiving the ad. Regarding time-oriented attributes, it seems that people have higher tendency to visit MyStore during afternoon ($\beta = 3.27, p < 0.0001$). Finally, for customer profile-oriented attributes, customer occupations, doing sports activities, and customer favorite products play a significant role in determining performance of the ad. These results reflect that other than location of the customers from a store, other attributes can also play important role in identifying probability of store visits after receiving ads.

The interaction between distance and other attributes are shown in the last three columns of Table A.2. Based on Table A.2 results, the interaction between distance and other at-
tributes remains statistically significant for attributes that have significant main effects. In other words, it can be concluded that the effect of other attributes (e.g. gender, competitors, time, and customers occupation) are significantly depend on distance. Therefore, based on our results, it is expected that selection and use of attributes in a mobile advertising campaign are based on the real-time distance between a potential target customer and the focal store.

Figure A.3 – The SHAP variable importance results
Table A.2 – Heterogeneity results

| Attributes                                      | Estimates | Pr(>|z|) | Attributes                                      | Estimates | Pr(>|z|) |
|------------------------------------------------|-----------|---------|------------------------------------------------|-----------|---------|
| **Demographics-oriented**                      |           |         | **Demographics-oriented × Distance**           |           |         |
| Gender                                         |           |         | Gender                                         |           |         |
| Male                                           | -0.66     | 0.31    | Male × Distance                                 | 0.05      | 0.44    |
| Female                                         | -2.13     | 0.000 ***| Female × Distance                               | 0.25      | 0.000 ***|
| **Age**                                        |           |         | **Age**                                        |           |         |
| 13-18                                          | -5.44     | 0.003 **| 13-18 × Distance                                | 0.67      | 0.041 **|
| 19-24                                          | 0.38      | 0.84    | 19-24 × Distance                                | -0.02     | 0.92    |
| 25-3                                           | -1.17     | 0.31    | 25-3 × Distance                                 | 0.13      | 0.30    |
| 35-4                                           | -0.31     | 0.78    | 35-4 × Distance                                 | 0.09      | 0.40    |
| Older(50+)                                      | -1.53     | 0.22    | Older (50+) × Distance                          | 0.24      | 0.070 **|
| **Income**                                     |           |         | **Income**                                     |           |         |
| Low                                            | 0.70      | 0.22    | Low × Distance                                  | -0.09     | 0.15    |
| Average                                        | 0.86      | 0.032 **| Average × Distance                              | -0.11     | 0.007 **|
| High                                           | -0.61     | 0.25    | High × Distance                                 | 0.02      | 0.75    |
| **Distance-oriented**                          |           |         | **Distance-oriented × Distance**                |           |         |
| Distance                                       | -7.42     | 0.000 ***| Home distance × Distance                        | 0.26      | 0.000 ***|
| Home distance                                  | -2.19     | 0.020 ***| Work distance × Distance                        | 0.15      | 0.000 ***|
| Work distance                                  | -1.31     | 0.000 ***| Time-oriented × Distance                        |          |         |
| **Time-oriented**                              |           |         | **Time-oriented × Distance**                    |           |         |
| Evening                                        | 0.48      | 0.18    | Evening × Distance                              | 0.07      | 0.082 **|
| Afternoon                                      | 3.27      | 0.000 ***| Afternoon × Distance                            | 0.38      | 0.000 ***|
| Night                                          | 0.05      | 0.37    | Night × Distance                                | -0.10     | 0.19    |
| **Competitors-oriented**                       |           |         | **Competitors-oriented × Distance**             |           |         |
| Previous visits of competitor1                 | -4.81     | 0.000 ***| Previous visits of competitor1 × Distance       | 0.59      | 0.000 ***|
| Previous visits of competitor2                 | -5.01     | 0.000 ***| Previous visits of competitor2 × Distance       | 0.52      | 0.000 ***|
| Previous visits of competitor3                 | 2.16      | 0.12    | Previous visits of competitor3 × Distance       | -0.07     | 0.59    |
| **Customer profile-oriented**                  |           |         | **Customer profile-oriented × Distance**        |           |         |
| Customers occupation                           |           |         | Customers occupation                            |           |         |
| FIP                                            | -4.06     | 0.001 **| FIP × Distance                                  | 0.35      | 0.013 **|
| CIP                                            | 67.26     | 0.98    | CIP × Distance                                  | -0.42     | 0.98    |
| TIP                                            | -2.46     | 0.000 ***| TIP × Distance                                  | 0.24      | 0.000 ***|
| MIP                                            | 278.20    | 0.92    | MIP × Distance                                  | -31.81    | 0.92    |
| Fast moving customer                           | -0.35     | 0.82    | Fast moving customer × Distance                 | 0.09      | 0.60    |
| Doing sport activities                         | -2.11     | 0.000 ***| Doing sport activities × Distance               | 0.19      | 0.001 **|
| **Customer favorite products categories**      |           |         | **Customer favorite products categories**       |           |         |
| Book                                           | 639.80    | 0.96    | Book × Distance                                 | -75.73    | 0.96    |
| Garden                                         | 2.53      | 0.45    | Garden × Distance                               | -0.35     | 0.36    |
| Toys                                           | -4.67     | 0.000 ***| Toys × Distance                                 | 0.46      | 0.000 ***|
| Beauty and health                              | -3.02     | 0.000 ***| Beauty and health × Distance                    | 0.25      | 0.001 **|
| Clothing                                       | -2.36     | 0.001 **| Clothing × Distance                             | 0.52      | 0.000 ***|
| Electronic tools                                | -4.87     | 0.000 ***| Electronic tools × Distance                     | -2.45     | 0.003 **|
| Movies                                         | 2.38      | 0.003 **| Movies × Distance                               |          |         |
| Previous visits of MyStore                     | 0.39      | 0.76    | Previous visits of MyStore × Distance           | -0.01     | 0.95    |
| **Mobile operating systems**                   |           |         | **Mobile operating systems × Distance**         |           |         |
| Android                                        | -2.82     | 0.079 . | Android × Distance                              | 0.29      | 0.10    |
| iOS                                            | -2.07     | 0.45    | iOS × Distance                                  | 0.14      | 0.64    |

Denotes significant at 0.1, * denotes significant at 0.05, ** denotes significant at 0.01, *** denotes significant at 0.0001.
Appendix 4: Store visit prediction model (Step 2)

We utilize machine-learning approaches for predicting the outcomes of advertisement exposures. Such a prediction model helps us identify the visit rate for any given subset of attributes. In specific, for any given subset of attributes, the percentage of those users truly identified for being exposed, i.e., they indeed visited the store, by the prediction model to the total number of exposed users can be used as a reliable estimate of the visit rate for the campaign design (in Step 3.2 of Figure 3.2 and Figure 3.3).

We compare the performance of three machine learning methods in this study: (1) Random Forest (RF; [23]), which is an ensemble learning method for classification by constructing a combination of tree classifiers, (2) Support Vector Machine (SVM; [139, 22]), which is a supervised machine learning method that can capture the multi-dimensional, non-linear relationships between the dependent variable and the independent variables, and (3) Logistic Regression (LR), which is a statistical model for the analysis and prediction of a categorical outcome [25]. The predicted outcome from the machine learning models is compared with the actual outcome to identify the most appropriate machine learning approach to estimate in-store visits. Once an appropriate machine learning approach is selected, the same approach is used to identify users for targeting the mobile advertisements. One of our key design propositions is that the effectiveness of attributes in predicting desired outcomes would vary with the distance between the focal BM store and user locations. Hence, we applied the machine learning-based prediction for different location categories, by segmenting the geographical space of the targeted users into specific intervals depending on the distance between the users’ locations and the BM store. Throughout, we used 10-fold cross validation
in the prediction models for tuning the hyper parameters of the models for each distance interval.

To show why SVM is used in Section 3.5.2, we now compare the performance of three machine learning techniques including RF, SVM, and LR to estimate in-store visits. In Table A.3, the predicted visit rate for the optimal subset of features is shown for each method in each phase.

Table A.3 – SVM, LR, and RF performance for the optimal subset of features

<table>
<thead>
<tr>
<th>Phase</th>
<th>Interval</th>
<th>Attributes</th>
<th>SVM</th>
<th>LR</th>
<th>RF</th>
</tr>
</thead>
<tbody>
<tr>
<td></td>
<td></td>
<td>Demographics</td>
<td>Customer profiles</td>
<td>Home distance</td>
<td>Competitors</td>
</tr>
<tr>
<td>Pilot</td>
<td>1</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>46%</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>47%</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>35%</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>19%</td>
</tr>
<tr>
<td>Production 1</td>
<td>1</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>44%</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>39%</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>54%</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>28%</td>
</tr>
<tr>
<td>Production 2</td>
<td>1</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>60%</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>42%</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>52%</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>22%</td>
</tr>
<tr>
<td>Production 3</td>
<td>1</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>44%</td>
</tr>
<tr>
<td></td>
<td>2</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>43%</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>52%</td>
</tr>
<tr>
<td></td>
<td>4</td>
<td>✓</td>
<td>✓</td>
<td>✓</td>
<td>14%</td>
</tr>
</tbody>
</table>

Note that in Table A.3, the best performance of each method is reported by tuning its hyper-parameters thorough jointly applying grid search and 10-fold cross validation. Since the hyper-parameters of RF, SVM, and LR are at most two, grid search is an effective tuning method [65]. In RF, hyper-parameters are the number of decision trees and the number of features considered by each tree when splitting. In this study, the sets \{100, 200, 300, 400, 500\} and \{50, 75, 125, 150, 175, 200\} are considered for tuning the first and second hyper-parameters of RF, respectively. For SVM, the radial basis function (RBF) is used as the basic kernel function of SVM (since it performs the best for our application). There are two hyper-parameters associated with SVM including \(C\), i.e., controls the cost of misclassifica-
tion on the training data, and $\gamma$, i.e., controls the width of kernel function, that play a vital role in the performance of SVM.

In this study, the sets

$$\{1, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$$

and

$$\{0.1, 0.2, 0.3, 0.4..., 9.8, 9.9, 10\}$$

in increment of 0.1 up-to 10 are considered for tuning the first and second hyper-parameters of SVM, respectively. Finally, in LR, there is a single regularization hyper-parameter, i.e., $C$, that should be tuned as it controls the complexity of the model. Hence, the set $\{1, 10, 20, 30, 40, 50, 60, 70, 80, 90, 100\}$ is considered for tuning the hyper-parameter of LR.
Appendix 5: Proof of Theorem 3.1

Theorem. The ROI Problem is \( \mathcal{NP} \)-hard.

Proof. We show that the (binary) Knapsack Problem is polynomially reducible to the ROI Problem. First note that any instance of the Knapsack Problem can be presented by

\[(m, p_1, \ldots, p_m, w_1, w_2, \ldots, w_m, W),\]

and refers to the following optimization problem,

\[
\begin{align*}
\max & \sum_{k=1}^{m} p_k x_k \\
\text{s.t.} & \sum_{k=1}^{m} w_k x_k \leq W, \\
& x_k \in \{0, 1\} \quad \forall k \in \{1, \ldots, m\},
\end{align*}
\]

where \( m, p_1, \ldots, p_m, w_1, w_2, \ldots, w_m, W \geq 0 \). The first step for transforming any instance of the Knapsack Problem, i.e., \((m, p_1, \ldots, p_m, w_1, w_2, \ldots, w_m, W)\), to an instance of ROI Problem is setting \( n = 1 \). Note that in this case, we have that \( S = \{1\} \) and \( I = \{1\} \). So, in this case, the ROI problem will be simplified to,

\[
\begin{align*}
\max & \sum_{k \in D} (N_k R \sigma_k y_k - M_k c x_k - N_k F_c) \quad (1) \\
\text{s.t.} & \sum_{k \in D} M_k c x_k + N_k F_c \leq B, \quad (2) \\
& y_k \leq 1 \quad \forall k \in D, \quad (3) \\
& y_k = x_k \quad \forall k \in D, \quad (4)
\end{align*}
\]
Based on Constraint (4) and the fact that all decision variables are binary, the model can be simplified even further to,

$$\begin{align*}
\text{max} & \sum_{k \in \mathcal{D}} (N_k R \sigma_k x_k - M_k c x_k - N_k F) \\
\text{s.t.} & \sum_{k \in \mathcal{D}} M_k c x_k + N_k F \leq B, \\
x_k & \in \{0, 1\} \quad \forall k \in \mathcal{D}.
\end{align*}$$

The second step is setting $D = m$, $M_k = w_k$ for all $k \in \mathcal{D}$, $c = 1$, $R = 1$, $F = 0$, $N_k = 1$ for all $k \in \mathcal{D}$, $B = W$, and $\sigma_k = p_k + w_k$ for all $k \in \mathcal{D}$. By doing this the model will be as follows,

$$\begin{align*}
\text{max} & \sum_{k=1}^{m} ((p_k + w_k)x_k - w_k x_k) \\
\text{s.t.} & \sum_{k=1}^{m} w_k x_k \leq W, \\
x_k & \in \{0, 1\} \quad \forall k \in \{1, \ldots, m\},
\end{align*}$$

which can be simplified further to,

$$\begin{align*}
\text{max} & \sum_{k=1}^{m} p_k x_k \\
\text{s.t.} & \sum_{k=1}^{m} w_k x_k \leq W, \\
x_k & \in \{0, 1\} \quad \forall k \in \{1, \ldots, m\}.
\end{align*}$$
The above formulation is basically the formulation of the knapsack problem. This immediately implies that the (decision form of the) initial knapsack instance is a Yes-instance if and only if the (decision form of the) generated instance of ROI is a Yes-instance.
Appendix 6: Pseudocodes

In this section the pseudocodes of the proposed heuristic algorithm introduced in Section 3.4.2 are provided.

**Algorithm 2:** Interval-Optimizer\((k, B'_k, S'', n', N_k, M_k, R, c)\)

\[
\begin{align*}
\text{input:} & \quad k, B'_k, S'', n', N_k, M_k, R, c & \quad \text{//Vectors are shown in bold fonts//} \\
S' & \leftarrow \{S \subseteq S'' : S \neq \emptyset\} & \quad \text{//} S' \text{ is the set of all non-empty subsets of } S'' \text{//} \\
I' & \leftarrow \{1, \ldots, 2^{n'} - 1\} & \quad \text{//} I' \text{ is the index set corresponding to } S' \text{//} \\
\text{for } i \in I' & \text{ do} \\
\sigma_{ik} & \leftarrow \text{use SVM to compute the conversion rate for subset } S_i' \text{ in the interval } k \\
\bar{x}_k & \leftarrow 0; \bar{y}_k \leftarrow 0 & \quad \text{//Initializing } \bar{x}_k \text{ and } \bar{y}_k \text{ that record the optimal solution in the interval } k\text{//} \\
\text{for } v \in I' & \text{ do} \\
\text{SearchDone} & \leftarrow \text{True} & \quad \text{//At each iteration we suppose that } v \text{ is optimal until proven otherwise//} \\
\text{if } M_k \sum_{j \in S_v'} c_j + N_k F \leq B'_k & \text{ then} \\
\text{for } w \in I' & \text{ do} \\
\text{if } M_k \sum_{j \in S_w'} c_j + N_k F \leq B'_k & \text{ then} \\
N_k R \sigma_{vk} - M_k \sum_{j \in S_v'} c_j < N_k R \sigma_{vk} - M_k \sum_{j \in S_w'} c_j & \text{ then} \\
\text{SearchDone} & \leftarrow \text{False} \\
\text{Break and go to Line 15} \\
\text{if } \text{SearchDone} = \text{True} & \text{ then} \\
\text{for } i \in S_v' & \text{ do} \\
\bar{x}_{ik} & \leftarrow 1 \\
\bar{y}_{vk} & \leftarrow 1 \\
\text{Break and go to Line 21} \\
\bar{w}_k & \leftarrow M_k \sum_{j \in S''} c_j \tilde{x}_{jk} + N_k F_c & \quad \text{//The exact budget used in the interval } k\text{//} \\
\bar{p}_k & \leftarrow N_k \sum_{i \in I'} R \sigma_{ik} \tilde{y}_{ik} - \bar{w}_k & \quad \text{//The exact ROI obtained in the interval } k\text{//} \\
\text{return } (\bar{x}_k, \bar{w}_k, \bar{p}_k)
\end{align*}
\]
Algorithm 3: A Variable Neighborhood Algorithm for solving the ROI Problem

input: \( \lambda, n', \theta, \text{Max-Iteration}, B, \mathcal{D}, S, M, N, R, c \) //Vectors are shown in bold fonts//

1. Best-ROI \( \leftarrow 0 \) //Initializing the best ROI obtained during the search.//

2. for \( k \in \mathcal{D} \) do
   3. \( B'_k \leftarrow B \) //Initializing \( B'_k \) which shows the maximum budget allocated to interval \( k \) //

4. Select a subset of attributes \( S^* \subseteq S \) such that \( |S^*| = n' \) //This can be done using any existing attribute selection techniques (e.g., LASSO, Recursive Feature Elimination, etc) //

5. Iteration \( \leftarrow 1 \)

while Iteration < Max-Iteration do

6. \( S' \leftarrow S^* \)

7. for \( S'' \in \text{Neighborhood}(S', \theta) \) do
   8. //Optimizing each interval independently using Observations 3.1 and 3.2 //
   9. for \( k \in \mathcal{D} \) do
      10. \( (\bar{x}_k, \bar{w}_k, \bar{p}_k) \leftarrow \text{Interval-Optimizer}(k, B'_k, S'', n', N_k, M_k, R, c) \)
   11. //Solving a Knapsack problem to identify which intervals should be active//
      12. \( \bar{z} \leftarrow \text{Knapsack-Optimizer}(\bar{w}, \bar{p}) \)
   13. //Updating the best solution found//
      14. if \( \sum_{k \in \mathcal{D}} \bar{p}_k \bar{z}_k > \text{Best-ROI} \) then
         15. Best-ROI \( \leftarrow \sum_{k \in \mathcal{D}} \bar{p}_k \bar{z}_k \)
         16. for \( k \in \mathcal{D} \) do
             17. \( S^* \leftarrow S'' \)
             18. if \( \bar{z}_k = 1 \) then
                 19. \( \bar{x}^*_k \leftarrow \bar{x}_k \) // \( \bar{x}^*_k \) is a binary vector showing the best solution found for each interval//
                 20. else
                     21. \( \bar{x}^*_k \leftarrow 0 \)
             22. end if
         23. end for
      24. end if

25. for \( k \in \mathcal{D} \) do
   26. if \( \bar{z}_k = 1 \) then
      27. \( B'_k \leftarrow \min\{B'_k + \lambda, B\} \)
   28. else
      29. \( B'_k \leftarrow \max\{B'_k - \lambda, \frac{R}{2D}\} \)
   30. end if
   31. Iteration \( \leftarrow \) Iteration + 1
32. end for

33. return \((\text{Best-ROI}, \bar{x}^*_1, \ldots, \bar{x}^*_D)\)
Appendix 7: Performance of the heuristic method on the case study

In this section we provide some numerical results on the performance of our approach on our case study. We do not report any solution time for our proposed heuristic in this section, as it terminates in a fraction of second. Moreover, since the optimal number of distance intervals is four, we compare the results of CPLEX and different settings of proposed heuristic for each of the four different distance intervals and each phase. Specifically, Table A.4 shows the optimality gap, i.e., the gap between the optimal objective value obtained by CPLEX and the objective value of the proposed heuristic \((\text{Optimal} - \text{Heuristic}) \times 100 / \text{Optimal})\), for different values of \(n'\) and \(\theta\). Note that “Max-Iteration” is set to 5 in order to create this table. Overall, we observe that the performance of the VNS is better when \(\theta\) is set to 2 (which is not surprising because the size of the neighborhood is larger for such cases in the proposed VNS method). Additionally, the optimality gap tends to decrease as \(n'\) increases and eventually reaches to zero, i.e., the VNS finds an optimal solution.
Table A.4 – The optimality gap of VNS for different values of parameters

<table>
<thead>
<tr>
<th>Phase</th>
<th>$n'$</th>
<th>$\theta = 1$</th>
<th>$\theta = 2$</th>
</tr>
</thead>
<tbody>
<tr>
<td>Pilot</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>100%</td>
<td>25%</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>28%</td>
<td>12%</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>25%</td>
<td>0%</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>11%</td>
<td>0%</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>Production I</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>100%</td>
<td>67%</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>45%</td>
<td>11%</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>45%</td>
<td>0%</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>34%</td>
<td>0%</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>Production II</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>100%</td>
<td>55%</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>46%</td>
<td>0%</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>46%</td>
<td>0%</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>46%</td>
<td>0%</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>0%</td>
<td>0%</td>
</tr>
<tr>
<td>Production III</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td></td>
<td>1</td>
<td>100%</td>
<td>61%</td>
</tr>
<tr>
<td></td>
<td>3</td>
<td>40%</td>
<td>0%</td>
</tr>
<tr>
<td></td>
<td>5</td>
<td>40%</td>
<td>0%</td>
</tr>
<tr>
<td></td>
<td>7</td>
<td>40%</td>
<td>0%</td>
</tr>
<tr>
<td></td>
<td>9</td>
<td>0%</td>
<td>0%</td>
</tr>
</tbody>
</table>
Appendix 8: The performance of the proposed VNS

The goal of this section is to provide numerical evidence that our proposed heuristic approach can perform well on even random instances and it is not limited to the dataset of MyStore. Hence, we generate two random classes of instances and compare the performance of the proposed VNS approach with an exact method, i.e., directly solving the ROI problem using CPLEX, on those instances. In each class of instances, there are 6 randomly generated instances. The number of attributes of each instance belongs to the set \(\{5, 7, 9, 11, 13, 15, 17, 19, 20\}\), meaning the first instance has 5 attributes and the last instance contains 20 attributes. To create each instance, we set \(D = 10\), and \(F_c = 1\); We randomly select \(M_k\) from the uniform distribution in the interval \([2000, 3000]\) for each \(k \in \mathcal{D}\); We randomly select \(N_k\) from the uniform distribution in the interval \([1000, 2000]\) for each \(k \in \mathcal{D}\); We randomly select \(c_j\) from the uniform distribution in the interval \([1, 5]\) for each \(j \in \mathcal{S}\); We randomly select \(R\) from the uniform distribution in the interval \([\sum_{j \in \mathcal{S}} c_j, 2 \sum_{j \in \mathcal{S}} c_j]\);

We set \(\sigma_{ik} = \omega + \frac{0.2 |S_i|}{n}\) for each \(i \in \mathcal{I}\) and \(k \in \mathcal{D}\), where \(\omega\) is randomly selected from the uniform distribution in the interval \([0.3, 0.7]\). Note that one can view \(\frac{0.2 |S_i|}{n}\) as a deterministic adjustment that adds up to 0.2 to the randomly generated conversion rates. The underlying idea is that for a larger subset of attributes, i.e., a higher value of \(|S_i|\), a larger adjustment should be added to the corresponding conversion rate. Finally, it is worth mentioning that the main difference between the two different classes of instances is the available budget. Overall, Class I has a smaller budget compared to Class II. Specifically, \(B\) is set to \(0.25(\sum_{k \in \mathcal{D}} M_k \sum_{j \in \mathcal{S}} c_j + N_k F_c)\) for Class I but it is set to \(0.5(\sum_{k \in \mathcal{D}} M_k \sum_{j \in \mathcal{S}} c_j + N_k F_c)\) for Class II.
We impose a time limit of 600 seconds for solving each instance when using the VNS and 7,200 seconds when using the exact method. The results for Class I and Class II can be found in Tables A.5 and A.6. Columns labeled by “OV” and “Time (s.)” show the objective value and the run time of CPLEX in seconds for solving an instance of the ROI problem using the exact method. For instances with \( n \geq 15 \), CPLEX is not able to solve it to optimality within the imposed time limit.

The optimality gap (column labeled “Opt. Gap”) reported by CPLEX is 61.57% and 43.33%, for the instance with \( n = 15 \) in Classes I and II, respectively. For instance, with \( n \geq 17 \), CPLEX is not even able to find a solution with a positive objective value and so the optimality gap of CPLEX for those the instances is over 100%. Overall, instances of Class I are more challenging for the exact method because the budget is tighter for those instances.

We run our proposed VNS heuristic under three different scenarios and for each of them, the parameters “Max-Iteration” and \( \theta \) are set to 10 and 3 respectively (as we observed they provide the best results during the course of our research). The difference among the number of scenarios is the value of \( n' \) which is set to \([0.5n]\), \([0.75n]\), and \([n]\), respectively. For each...
scenario, columns labeled “Gap” show the difference between the objective value obtained by our VNS approach to the objective value of the exact method. Similarly, columns labeled “RTime” show the ratio of the run time of our VNS approach to the run time of the exact method. Evidently, if RTime < 1 then it implies that our proposed VNS approach has been faster. Note that for instances with $n < 15$, the exact method is able to solve them to optimality within the time limit. So, for those instances if Gap = 0% then it implies that our approach is able to generate an optimal solution.

Observe that for the instances with $n = 15$, the exact method is not able to solve them to optimality but is able to find solutions with positive objective values. Therefore, because Gap < 0% for those instances, our proposed heuristic is able to arrive at a solution that is better than the one obtained by the exact method within the imposed time limit. Specifically, for the instances with $n = 15$, we observe that Gap is around -23% and -12% for Classes I and II, respectively. This implies that the quality of the solution obtained by our proposed heuristic is around 23% and 12% better for Classes I and II, respectively. Observe that for the instances with $n > 15$, the exact method is not only able to solve them to optimality but it is also not able to find solutions with positive objective values. Therefore, for those instances, we have that Gap $\ll$ 0%, which implies that our proposed heuristic performs significantly better.

Finally, it is worth mentioning that, our proposed VNS heuristic performs well under all scenarios. For only small instances, the first scenario, i.e., $n' = \lfloor 0.5n \rfloor$, is a little worse than the others in terms of the quality of the solutions. However, for large instances, it is comparable to the other scenarios. Overall, the solution time of the proposed heuristic (under all scenarios) is just a small fraction of the solution time of the exact method. However, the proposed VNS spends more time under the second scenario, i.e., $n' = \lfloor 0.75n \rfloor$. As an aside, one may conclude that based on the results, the third scenario, i.e., $n' = \lfloor n \rfloor$, seems to be the best. Although this is true, one can expect that for larger values of $n$, e.g., 30, the third scenario is not a feasible option. This is because in that case, in each iteration, the
proposed VNS needs to explore up to $2^{30}$ solutions for each distance interval, which is not computationally feasible. However, scenarios 1 and 2 are still viable options for such an instance because based on Tables A.5-A.6, one can expect to obtain similar-quality solutions by setting $n'$ to much smaller values.