Decision-Aware Segmentation

This file [link] documents my logic of writing for each section.

1 Introduction

Customer segmentation is foundational to marketing and operations strategy, enabling firms to tailor strategic interventions—such as pricing, advertising, and product recommendations—to distinct customer groups. With the rise of customer analytics and data science, the identification of customer segments has become essential in modern marketing (Bergemann et al., 2024; Lu et al., 2025; Simester et al., 2020b; van de Geer and den Boer, 2022; Wedel and Kamakura, 2000; Yang et al., 2016; Zhang and Misra, 2024). The core idea of segmentation is that customers within a segment share relevant characteristics that justify similar treatment, while differences across segments can be leveraged for differential targeting. The effectiveness of downstream decisions depends heavily on the quality of the segmentation that informs them. Managers hope to find the customer segmentation structure that maximize firm profits.

However, this outcome-oriented objective stands in contrast to how most widely adopted segmentation methods are constructed: they remain largely agnostic about decision relevance. For example, popular clustering methods aim either to minimize within-cluster variance in consumer covariates (e.g., K-means), or to maximize model likelihood under a parametric specification (e.g., Gaussian Mixture Models, Latent Class Regression). These methods prioritize statistical objectives—such as compactness or fit—without explicitly considering how the resulting segments will perform when used to guide decisions. As a result, a segmentation may appear statistically sound yet offer little value for improving firm outcomes.

Equally problematic is how the number of segments—often a critical managerial choice—is typically determined. Standard approaches rely on penalized likelihood criteria such as AIC or BIC (Akaike, 1974; Fraley and Raftery, 1998; Schwarz, 1978), or on subjective heuristics such as the "elbow method" (Sugar and James, 2003). These criteria optimize the trade-off between goodness-of-fit and model complexity, but fail to answer the more relevant question: what level of segmentation granularity leads to the best business outcomes?

A related question for researchers is, how should the quality of a given segmentation be evaluated, in simulation studies where the ground truth is known? Common metrics based on clustering agreement (e.g., Adjusted Rand Index) may not adequately reflect a segmentation's effectiveness in supporting high-quality decisions.

The misalignment between the segmentation process and decision impact can lead firms to deploy ineffective or suboptimal strategies. To bridge this gap, we center our research around three core questions:

- *How can we construct a segmentation that directly optimizes downstream business outcomes from the decisions induced by this segmentation?*
- *How should we determine the number of segments that optimizes downstream business outcomes?*
- *How should researchers assess the quality of a segmentation in simulation studies in which the ground truth segmentation is known?*

To address these questions, we propose **D**ecision-**A**ware **S**egmentation (DAS), a unified segmentation framework explicitly designed to optimize the value of downstream managerial decisions. DAS consists of two key components: (1) Decision-Aware Segmentation Tree (DAST): a decision-aware segmentation algorithm, (2) Decision-Aware Model Selection (DAMS): a model selection method for choosing the number of segments based on policy performance. Importantly, DAMS is agnostic to the specific segmentation algorithm employed and can be applied to any segmentation method, not limited to DAST. In addition, we propose a comprehensive segmentation evaluation framework. In summary, our paper makes the following key contributions:

- First, we propose a novel, decision-aware, tree-based segmentation algorithm designed to maximize the total expected outcome under a policy that is learned from the segmentation. Unlike standard methods, our approach explicitly evaluates splits based on estimated downstream policy value, not just covariate similarity or likelihood. The goal is to create segments that enable high-profit policies, rather than segments that satisfy statistical properties.
- Second, we introduce a managerially implementable model selection method based on out-of-sample evaluation of the policy induced by each candidate segmentation. This approach allows firms to choose the segmentation granularity in a way that maximizes expected business outcomes, rather than statistical fit.

- Third, we propose a comprehensive, oracle-based evaluation framework to assess the quality of a segmentation in terms of three dimensions: segment structure recovery, estimation accuracy, and policy effectiveness. We believe this evaluation framework will help researchers systematically assess and compare segmentation methods in simulation-based studies, and advance our understanding of what makes a segmentation effective, and why.
- Forth, we demonstrate the effectiveness of our algorithm and evaluation framework using both simulated and real-world datasets.

Together, this work aims to reorient the segmentation literature from traditional fit-driven approaches toward an outcome-driven, decision-aware paradigm. The Decision-Aware Segmentation (DAS) framework is broadly applicable across marketing and operations contexts where targeting and personalization rely on customer segmentation. We provide an open-source implementation of our algorithm and evaluation framework in Python, available at [link]. The code is modular and designed to be easily integrated into different personalized decision-making applications.

The remainder of this paper is structured as follows. In Section 2, [Placeholder]

2 Literature Review

Our paper contributes to multiple streams of literature.

2.1 Consumer Segmentation

Market segmentation is the foundation of consumer analytics and marketing strategy. Individual-level targeting is challenging due to limited resources (e.g., data, budget, infrastructure), motivating the need for segment-based policies. Researchers have developed a wide range of methods to derive customer segments, which can be broadly categorized into three classical paradigms: centroid-based clustering, hierarchical clustering, and model-based clustering. The first class, centroid-based clustering, assigns each observation to the nearest cluster center based on distance metrics. The most well-known example is K-means (Hartigan and Wong, 1979; MacQueen, 1967), with extensions including fuzzy C-means for soft assignments (Bezdek et al., 1984), K-medoids for robustness to outliers (Kaufman and Rousseeuw, 2009; Park and Jun, 2009), and scalable variants (Ng and Han, 2002; Sculley, 2010). A second classical approach is hierarchical clustering, which constructs a nested sequence of segments by iteratively merging or splitting data based on similarity metrics (Müllner, 2011; Murtagh and Legendre, 2014). A third influential approach is model-based clustering, which treats segment membership as latent variables under a probabilistic generative model and seeks to recover segments by maximizing the likelihood of observed data under the assumed generative process. Prominent examples include Gaussian Mixture Models (GMMs) (Dempster et al., 1977) and Finite Mixture of Regressions, which are also referred to as Latent Class Regression or Clusterwise Linear Regression (DeSarbo and Cron, 1988; Vermunt and Magidson, 2002; Wedel and DeSarbo, 1994; Wedel and Kamakura, 2000). These methods remain central to academic research and commercial segmentation software due to their interpretability and statistical rigor.

Advances in machine learning have enabled more effective approaches to customer segmentation by learning rich, task-relevant representations from high-dimensional data. For example, Xie et al. (2016) and Ronen et al. (2022) use deep neural networks to jointly learn feature representations and cluster assignments via optimizing reconstruction loss or cluster compactness, and achieve more meaningful groupings than clustering on raw features. In addition, segmentation using decision tree methods has also been applied in marketing analysis, particularly when interpretability is a priority (Aouad et al., 2023; Liu et al., 2000, 2010). While effective at uncovering structural patterns in the data, these approaches above are inherently descriptive. They are designed to describe the data, not to optimize any firm-specific objective.

Despite decades of development in marketing analytics, the decision-agnostic segmentation paradigm remain dominant in academic research and industry practice, implemented in their standard form or with slight modifications (Bapna et al., 2011; Gallego and Berbeglia, 2024; Govind et al., 2018; Li et al., 2021; Park et al., 2017). A recent review examined segmentation methods used in e-commerce applications in hundreds of papers from 2000 to 2022, and found that K-means remains the most widely adopted approach (Alves Gomes and Meisen, 2023).

The recent work by Aouad et al. (2023) has begun to align segmentation more closely with consumer response behavior. They proposed Market Segmentation Trees (MST) framework, which is an interpretable, tree-based method for learning market segments explicitly driven by identifying differences in user response patterns. Rather than segmenting on demographic or behavioral similarity, MST recursively partitions the population by identifying best splits that lead to improvements in response prediction accuracy. Aouad et al.'s method represents a good step toward aligning segmentation with predicting consumer responses; however, it stops short of incorporating the downstream targeting or decision-making objectives into the segmentation process. Also, it doesn't provide a mechanism for determining the optimal tree depth.

Another challenge in segmentation is is determining the appropriate number of segments. One common class of approaches relies on penalized likelihood criteria, which balance model fit with complexity, such as Akaike Information Criterion (AIC) (Akaike, 1974), Bayesian Information Criterion (BIC) (Schwarz, 1978), and Integrated Completed Likelihood (ICL) (Biernacki et al., 2002). Alternatively, heuristic techniques such as the "elbow method" (Sugar and James, 2003) or silhouette analysis (Rousseeuw, 1987) are popular in practice because they are computationally simple and visually intuitive. More recently, advances in machine learning have introduced Bayesian nonparametric models, such as Dirichlet Process Mixture Models (DPMM) proposed by Antoniak (1974), to infer the number of segments through dynamic split-and-merge strategies during training process (Ronen et al., 2022). Also, these model selection methods are fundamentally decision-agnostic: they aim to find the segmentation granularity that best represents the data structure rather than the one that enables the most effective business decisions.

The disconnect between descriptive fit and prescriptive utility is increasingly problematic as the growing adoption of data-driven marketing analytics has elevated the role of segmentation from an exploratory tool to a key input for decision-making. There is an emerging need for decision-aware segmentation algorithms and decision-aware model selection methods that embed business goals directly into the model training process. Our Decision-Aware Segmentation (DAS) framework consists of a segmentation algorithm that directly optimize the total expected outcome of segmentation-induced policies, and a decision-aware model selection approach that chooses the number of segments based on out-of-sample policy performance.

2.2 Decision-Aware Learning Paradigm

Many downstream decisions—such as targeting, pricing, and assortment—rely on predictive models as inputs to an optimization problem. In the standard *predict-then-optimize* pipeline, one first trains a model to predict unknown parameters (e.g. customer responses) as accurately as possible, and then plugs those predictions into a decision model (e.g. a targeting policy). This two-stage approach ignores the fact that a model can be "accurate" by typical measures yet suboptimal for the decisions it informs.

Recognizing this misalignment between predictive loss and decision outcomes, recent work has advocated an outcome-oriented, decision-aware shift, arguing that models should be trained and selected with the ultimate decision task in mind. Elmachtoub and Grigas (2022) formalize this important shift through their Smart Predict-then-Optimize (SPO) framework. Rather than minimizing standard mean squared error, Elmachtoub and Grigas propose training predictive models using a decision-aware loss that directly reflects the downstream optimization objective. Specifically, they introduce SPO loss, defined as the difference between the true cost of the decision induced by predicted parameters and optimal cost under the true parameters. Their results show that decision-aware models can lead to substantially better decisions than those optimized for predictive accuracy alone, especially when the predictive model is mis-specified. Building on this, Elmachtoub et al. (2020) propose a tractable method that trains decision trees directly under SPO loss. El Balghiti et al. (2019) characterize theoretical bounds on how well the performance of a prediction model generalizes out of sample when trained using SPO loss. Vanderschueren et al. (2022) provide an empirical comparison between *predict-then-optimize* and *predict-and-optimize* in the context of cost-sensitive classification, concluding that prediction models trained with decision costs in mind can enable more effective policies. Similarly, Huang and Gupta (2024) propose Perturbation Gradient (PG) losses, which is a family of decision-aware surrogate losses which directly approximate the downstream decision loss.

Researchers have since applied this decision-aware learning paradigm in various contexts, including resource allocation (Stratman et al., 2024), healthcare supply chain (Chung et al., 2022), automated action planning (Mandi et al., 2024), mathematical optimization (Chan et al., 2025; Mandi et al., 2020), learning-to-rank problem (Huang and Gupta, 2024), portfolio optimization (Sarkar et al., 2025), demonstrating the broad applicability of this paradigm across diverse decision-making scenarios. While decision-aware learning has gained attention in various domains, it remains largely unexplored in the context of segmentation. Our work fills this gap by extending the decision-aware paradigm to segmentation, proposing an algorithm that learns segments to maximize the total value of downstream policies.

2.3 Targeting & Personalization

Targeting is central to marketing strategy. Early direct-marketing work showed how purchase histories and customer value can guide profitable targeting strategies (Bult and Wansbeek, 1995; Rossi et al., 1996; Venkatesan and Kumar, 2004). As marketing moved online and mobile, research examined how digital targeting works across diverse domains through methods including personalization, seeding strategies, advertising, pricing, machine learning, and reinforcement learning, among others (Ansari and Mela, 2003; Dubé et al., 2017; Dubé and Misra, 2023; Fong, 2017; Fong et al., 2015; Ghose et al., 2019; Goldfarb and Tucker, 2011; Gordon et al., 2019; Hinz et al., 2011; Iyer and Ke, 2024; Iyer et al., 2005; Karle and Reisinger, 2025; Lemmens et al., 2025; Luo et al., 2014; Rafieian and Yoganarasimhan, 2021; Smith et al., 2023; Valenti et al., 2025; Wang et al., 2023)

Marketing targeting have traditionally relied on predictive models to target customers. Over time, researchers realized that prediction is not the same as prescription. For example, high-propensity customers may purchase regardless of being targeted by the campaign, while moderately likely ones may be more persuadable only if targeted. Our work contributes to the ongoing shift from prediction to prescription in marketing analytics (Bertsimas and Kallus, 2020). Much work has advanced this line of research, often known as *uplift modeling*, or *incremental modeling* (Ascarza, 2018; Athey and Imbens, 2016; Chickering and Heckerman, 2000; Hitsch et al., 2024; Lemmens and Gupta, 2020; Rößler and Schoder, 2022; Rzepakowski and Jaroszewicz, 2012; Wager and Athey, 2018; Wan et al., 2022). This prescription-based targeting has been shown to substantially improve campaign profitability. Ascarza (2018) demonstrates the most profitable strategy focuses on "persuadables," whose behavior is most responsive to treatment. In same spirit, Lemmens and Gupta (2020) develop profit-based targeting policies that explicitly account for cost and incremental effectiveness, showing that policies optimized for expected profit lift significantly outperform those based on churn likelihood alone. These work emphasize the importance of targeting customers based on their incremental impact, that is, how much their behavior is causally influenced by an intervention. Enabled by advances in heterogeneous treatment effect estimation, such as causal forests (Wager and Athey, 2018), meta-learners (Künzel et al., 2019), and counterfactual prediction (Johansson et al., 2016), managers can estimate individual-level treatment effects and allocate interventions where they are expected to have the greatest impact.

As firms increasingly deploy data-driven targeting policies, a growing stream of complementary research has been focused on evaluation, optimization, and experimental design of targeting policies. On the evaluation side, Simester et al. (2020b) examine the robustness of various machine learning methods under realistic data issues, such as class imbalance and missing covariates, while Simester et al. (2020a) propose a method to compare targeting policies without running new experiments. On the optimization side, Hitsch et al. (2024) develop a framework that directly learns targeting rules aligned with firm profit objectives by leveraging heterogeneous treatment effect estimation. Lu et al. (2025) incorporate managerial and fairness constraints into the policy optimization process. A more recent stream focuses on experimental design, particularly how data collection strategies can improve downstream policy quality. Simester et al. (2025) study the sample size requirements for training and certifying targeting policies, offering guidance on how large an experiment must be to ensure reliable policy deployment. Chen et al. (2024) propose a policy-aware experimentation framework that strategically allocates samples based on their value for optimizing targeting policies, contributing to a broader literature on experimental design with online and sequential learning.

A parallel literature develops formal frameworks for policy learning. Foundational work on statistical treatment choice (Manski, 2004; Tetenov, 2012) and empirical welfare maximization (Kitagawa and Tetenov, 2018) establishes regret bounds and decision-theoretic guarantees for policies. Numerous algorithmic innovations have been proposed to improve robustness, efficiency, and identification of policy learning under various forms of data and modeling constraints (Athey and Wager, 2021; Dudík et al., 2011; Huang and Ascarza, 2024; Kallus, 2018; Kallus and Zhou, 2018, 2021; Sahoo and Wager, 2025; Si et al., 2023; Yang et al., 2024; Zhan et al., 2024; Zhou et al., 2023). On the applied side, Sverdrup et al. (2020) provide an open-source implementation of policy trees that learns optimal treatment assignment policies based on doubly robust scores.

Despite this rich literature, the vast majority of targeting studies implicitly or explicitly assume one of two extremes: (i) that customer segments are pre-specified, or (ii) that interventions are targeted at the individual level via estimated customer-specific responses, such as heterogeneous treatment effect. What fewer works explore is how the segmentation structure itself can be optimized to improve targeting performance. Our work builds on this literature by proposing a segmentation-based policy class that is learned explicitly to maximize downstream decision value, combining the interpretability of segments with the outcome-driven precision of modern policy learning.

3 Model and Problem Formulation

3.1 Ground-truth segments and response model

Consider a population of N customers, indexed by the set $\mathcal{I} = \{1, 2, \ldots, N\}$. Each customer i is described by a covariate vector $\boldsymbol{x}_i \in \mathbb{R}^d$, representing observed features (e.g., demographics, browsing behavior, past purchases). At an unobserved *ground-truth* level, the population is assumed to consist of K latent segments, indexed by $k = 1, \ldots, K$. Each customer i belongs to exactly one latent segment, denoted by a segment label $Z_i \in \{1, \ldots, K\}$. Customers within the same segment are assumed to be behaviorally or structurally similar.

Each segment k is characterized by a baseline outcome function $f_k : \mathbb{R}^d \to \mathbb{R}$ and a net treatment effect $\tau_k \in \mathbb{R}$. The net profit outcome y_i for customer i in segment k is generated according to:

$$y_i = f_k(\boldsymbol{x}_i) + \tau_k \cdot D_i + \varepsilon_i \tag{1}$$

where:

- $D_i \in \{0, 1\}$ is a binary treatment indicator (i.e., whether customer i received a marketing intervention),
- τ_k is the net treatment effect for segment k,

- $f_k(\boldsymbol{x}_i)$ represents the baseline outcome function for segment k, [1]
- ε_i is idiosyncratic noise, $\varepsilon_i \overset{\text{i.i.d.}}{\sim} \mathcal{N}(0, \sigma^2)$, with σ^2 denoting the noise variance, which does not need to be known to the manager.

This formulation captures heterogeneous baseline outcomes through $f_k(\boldsymbol{x}_i)$, and heterogeneous treatment effects across segments through τ_k, allowing each segment to exhibit distinct patterns in both covariate influence and treatment response.

In practice, the baseline outcome function $f_k(\cdot)$ can take various parametric or nonparametric forms depending on modeling assumptions and data complexity. Table 1 provides several examples of the forms of the baseline outcome function $f_k(\cdot)$ in applied settings.

Form	**Example Function**	**Description**
Linear	$f_k(\boldsymbol{x}_i) = \alpha_k + \boldsymbol{\beta}_k^\top \boldsymbol{x}_i$	Most common; quite interpretable
Polynomial	$f_k(\boldsymbol{x}_i) = \alpha_k + \boldsymbol{\beta}_k^\top \boldsymbol{x}_i + \boldsymbol{x}_i^\top \Gamma_k \boldsymbol{x}_i$	Includes interactions/quadratics
Generalized Linear Model	$f_k(\boldsymbol{x}_i) = g^{-1}(\boldsymbol{x}_i)$	$g^{-1}(\cdot)$ is a link function (e.g., logistic, exponential)
Kernel Regression	$f_k(\boldsymbol{x}_i) = \sum_{\ell=1}^{L} \omega_{k,\ell} \cdot K(\boldsymbol{x}_i, \boldsymbol{c}_\ell)$	Kernel regression (e.g., with RBF kernel)
Neural Network	$f_k(\boldsymbol{x}_i) = \mathrm{ReLU}(\boldsymbol{W}_k \boldsymbol{x}_i + \boldsymbol{b}_k)$	Feedforward neural network (e.g., 1 hidden layer)
Tree-based	$f_k(\boldsymbol{x}_i) = \mathrm{Tree}(\boldsymbol{x}_i)$	Tree-based models (e.g., CART or random forests)

Table 1: Examples of baseline outcome function forms for $f_k(\boldsymbol{x}_i)$

For quick reference, we summarize the notation in Table 2.

3.2 Sequence of events

We describe the sequence of events, illustrated in Figure 1.

1. **Pilot experiment.** The manager conducts a pilot experiment on a subset of customers indexed by the set $\mathcal{I}^{\text{pilot}} = \{1, \ldots, N^{\text{pilot}}\}$, where N^{pilot} is fixed and $0 < N^{\text{pilot}} < N$. Each customer $i \in \mathcal{I}^{\text{pilot}}$ is independently assigned to the control condition ($D_i = 0$) or the treatment condition ($D_i = 1$) with equal probability. This produces a randomized A/B testing dataset of the form:

$$\{(\boldsymbol{x}_i, D_i, y_i)\}_{i \in \mathcal{I}^{\text{pilot}}}.$$

[1] The functional form $f_k(\boldsymbol{x}_i)$ is sufficiently general to include cases where the outcome depends only on a subset of covariates. In our simulation design, we adopt this setting to reflect practical business contexts in which managers observe a high-dimensional feature vector but do not know which covariates truly influence customer response.

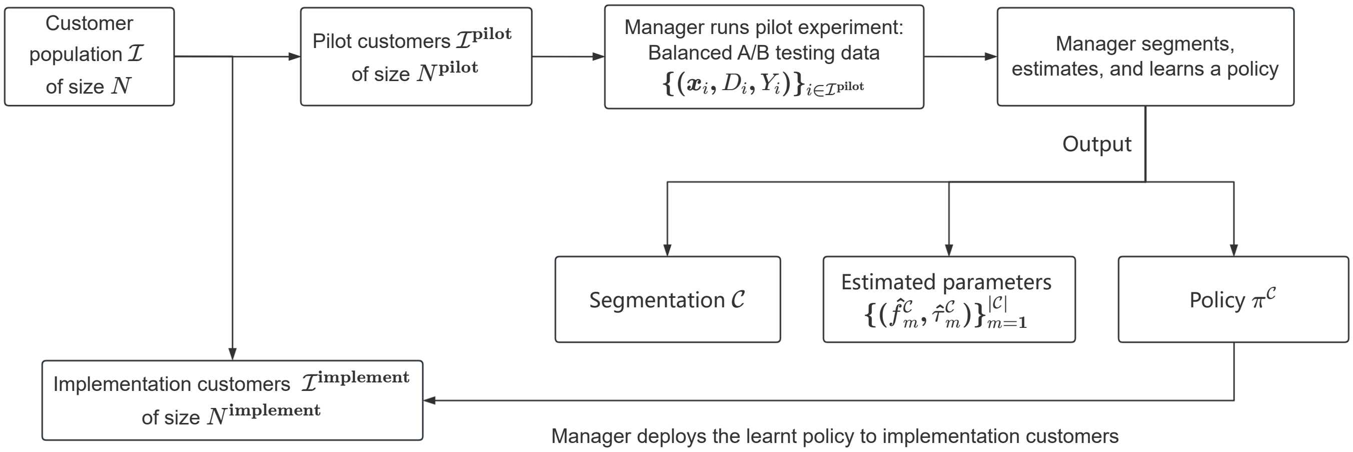
Customer population $\mathcal{I}$ of size N
Pilot customers $\mathcal{I}^{\mathbf{pilot}}$ of size $N^{\mathbf{pilot}}$
Manager runs pilot experiment: Balanced A/B testing data $\{(\boldsymbol{x}_i, D_i, Y_i)\}_{i\in\mathcal{I}^{\mathbf{pilot}}}$
Manager segments, estimates, and learns a policy
Output
Segmentation $\mathcal{C}$
Estimated parameters $\{(\hat{f}_m^{\mathcal{C}}, \hat{\tau}_m^{\mathcal{C}})\}_{m=1}^{|\mathcal{C}|}$
Policy $\pi^{\mathcal{C}}$
Implementation customers $\mathcal{I}^{\mathbf{implement}}$ of size $N^{\mathbf{implement}}$
Manager deploys the learnt policy to implementation customers

Figure 1: Sequence of events: Pilot experiment; segmentation and estimation; final deployment on implementation customers.

2. **Segmentation and segment-level estimation.** The manager fits a segmentation model on the pilot dataset and obtains segment-specific estimates of baseline outcome functions and treatment effects. We define a segmentation as a partition of the pilot customers into a finite number of (disjoint) segments. Let $\mathcal{C}$ denote such a segmentation. Then we have

$$\mathcal{C} = \left\{\mathcal{S}_1, \ldots, \mathcal{S}_{|\mathcal{C}|}\right\}, \qquad \bigcup_{m=1}^{|\mathcal{C}|} \mathcal{S}_m = \mathcal{I}^{\text{pilot}}, \quad \mathcal{S}_m \cap \mathcal{S}_{m'} = \emptyset \text{ for } m \neq m',$$

where set $\mathcal{S}_m \subset \mathcal{I}^{\text{pilot}}$ contains the indices of pilot customers assigned to segment m, and $|\mathcal{C}|$ denotes the number of segments in the segmentation. We use $C_i \in \{1, \ldots, |\mathcal{C}|\}$ to denote the segment index assigned to pilot customer i under segmentation $\mathcal{C}$, so that $C_i = m$ if and only if $i \in \mathcal{S}_m$.

In particular, the number of segments $|\mathcal{C}|$ in the chosen segmentation $\mathcal{C}$ is an important choice at the manager's discretion. It may reflect existing business knowledge or be selected via model selection techniques. In contexts where it is necessary to specify the number of segments, we denote a segmentation with M segments by $\mathcal{C}_M$.

The manager also obtains estimates of segment-specific baseline outcome functions and treatment effects from segmentation $\mathcal{C}$:

$$\{(\hat{f}_m^{\mathcal{C}}, \hat{\tau}_m^{\mathcal{C}})\}_{m=1}^{|\mathcal{C}|},$$

where $\hat{f}_m^{\mathcal{C}} : \mathbb{R}^d \to \mathbb{R}$ is the estimated baseline outcome function for segment m, and $\hat{\tau}_m^{\mathcal{C}} \in \mathbb{R}$ is the estimated treatment effect. These estimates may be obtained either via a two-stage procedure—first fitting a segmentation $\mathcal{C}$, and then estimating segment-specific models within each segment—or via joint methods that estimate simultaneously both a segmentation and model parameters.

The best practice for estimating $\{(\hat{f}^{\mathcal{C}}_m, \hat{\tau}^{\mathcal{C}}_m)\}_{m=1}^{|\mathcal{C}|}$ depends on the assumed functional form (see Table 1). Once the functional form is specified, we assume that a corresponding estimation procedure is fixed. That is, we do not treat the estimation method itself as part of the manager's decision. Our formulation is general and agnostic to the choice of functional form and estimation method. In our simulation analysis, we experiment with multiple functional forms and use standard estimation techniques aligned with each form.

3. **Implementation.** In this stage, the manager decides whether to treat or not the customers in the implementation set, indexed by the set $\mathcal{I}^{\text{implement}} = \{N^{\text{pilot}} + 1, \dots, N\}$. Given a segmentation $\mathcal{C}$, the manager first assigns each implementation customer to a learned segment based on the customer's covariates:

$$A_{\mathcal{C}}(j) = \phi_{\mathcal{C}}(\boldsymbol{x}_j),$$

where we use $A_{\mathcal{C}}(j)$ to denote the segment index to which customer j in the implementation set is assigned, and where $\phi_{\mathcal{C}} : \mathbb{R}^d \to \{1, \dots, |\mathcal{C}|\}$ is a fixed assignment procedure, such as nearest-centroid classification for k-means clustering, maximizing a posterior probability for a Gaussian mixture model, or a decision-tree-based rule. The assignment mechanism depends on, and follows naturally from, the used segmentation algorithm, as we discuss in Section XXX.

The manager then decides whether to treat or not the implementation customers in each learnt segment. In our setting, we consider targeting policies that operate at the segment level and apply the same action to all customers within an estimated segment.[2] We restrict attention to the targeting policy $\pi^{\mathcal{C}}$ that only treats customers in estimated segments with a non-negative estimated treatment effect. In particular, for each implementation customer $j \in \mathcal{I}^{\text{implement}}$, the assigned action $\pi^{\mathcal{C}}(j)$ is given by:

$$\pi^{\mathcal{C}}(j) = \begin{cases} 1 & \text{if } \hat{\tau}^{\mathcal{C}}_{A_{\mathcal{C}}(j)} \geq 0, \\ 0 & \text{otherwise}, \end{cases} \tag{2}$$

where $\pi^{\mathcal{C}}(j) = 1$ (0) indicates that the policy assigns the treatment (control) to customer j. This policy maximizes the expected profit in the implementation stage conditioned on the available treatment effect estimates.

[2] Our conversations with industry collaborators suggest that this aligns with the practical constraints that retailers face when implementing targeting strategies.

3.3 Problem formulation

We formalize the manager's decision problem as choosing a segmentation that maximizes expected total profit in the implementation set. Specifically, the manager selects a segmentation $\mathcal{C}$ over the pilot customers, which induces estimated segment-level baseline outcome models and treatment effects, and in turn defines a targeting policy to be deployed to the implementation customers.

The space of feasible segmentations is constrained in practice. Not all partitions of the pilot data are admissible. The admissible space depends on the model class and the statistical requirements needed for reliable estimation of $\hat{f}_m$ and $\hat{\tau}_m$. For example, if segment-level outcomes are modeled using linear regression and estimated via ordinary least squares (OLS), each segment must contain a sufficient number of observations with variation in covariates and treatment assignment to ensure identifiability and avoid degeneracy. We denote the set of all such admissible segmentations with $\mathscr{C}$.

The expected profit for implementation customer j under the targeting policy in (2) is:

$$\mathbb{E}\left[f_{Z_j}(\boldsymbol{x}_j) + \tau_{Z_j} \cdot \pi^{\mathcal{C}}(j)\right],$$

where Z_j is the latent ground-truth segment label for customer j, and the expectation is taken over the randomness in the pilot-stage [3]. The manager seeks the segmentation $\mathcal{C} \in \mathscr{C}$ that maximizes the total expected ground-truth profit across all customers in the implementation set:

$$\max_{\mathcal{C} \in \mathscr{C}} \quad \sum_{j \in \mathcal{I}^{\text{implement}}} \mathbb{E}\left[f_{Z_j}(\boldsymbol{x}_j) + \tau_{Z_j} \cdot \pi^{\mathcal{C}}(j)\right]. \tag{3}$$

The first term represents the baseline profit under no treatment and the second term captures the incremental effect of applying treatment according to the policy $\pi^{\mathcal{C}}$.

Remark. Our scope is not to solve the optimization problem in (3) to optimality, as the space of admissible segmentations is combinatorially large and the problem may be intractable. Instead, we propose a practically implementable segmentation methodology designed specifically for targeting settings, focused on improving downstream outcomes. Empirically, our approach reliably outperforms mainstream segmentation methods in terms of downstream performance, highlighting the value of designing segmentations

[3] The policy $\pi^{\mathcal{C}}$ is deterministic once estimated, but it is defined based on segment-level treatment effect estimates, which depend on noisy pilot data. Pilot-stage outcomes are generated according to:

$$y_i = f_{Z_i}(\boldsymbol{x}_i) + \tau_{Z_i} \cdot D_i + \varepsilon_i, \qquad \varepsilon_i \sim \mathcal{N}(0, \sigma^2),\ D_i \sim \text{Bernoulli}(0.5).$$

Thus, both the additive noise ε_i and random treatment assignment D_i affect the estimation of treatment effects, and therefore the downstream policy $\pi^{\mathcal{C}}$. As a result, the realized implementation-stage profit under $\pi^{\mathcal{C}}$ is a random variable.

that are aligned with business objectives.

Symbol	**Description**
N	Total number of customers in the population
N^{pilot}	Number of customers in the pilot study
$N^{\text{implement}}$	Number of implementation customers, $N^{\text{implement}} = N - N^{\text{pilot}}$
$\mathcal{I}$	Index set of all customers, $\mathcal{I} = \{1, \dots, N\}$
$\mathcal{I}^{\text{pilot}}$	Index set of customers in the pilot study, $\mathcal{I}^{\text{pilot}} = \{1, \dots, N^{\text{pilot}}\}$
$\mathcal{I}^{\text{implement}}$	Index set of implementation customers, $\mathcal{I}^{\text{implement}} = \{N^{\text{pilot}} + 1, \dots, N\}$
$k \in \{1, \dots, K\}$	Ground-truth segment index (latent)
$Z_i \in \{1, \dots, K\}$	Ground-truth segment label for customer i
$m \in \{1, \dots, M\}$	Manager-chosen segment index (observable)
$C_i \in \{1, \dots, M\}$	Estimated segment index for pilot customer i
f_k, τ_k	Baseline outcome and treatment effect for true segment k
$D_i \in \{0, 1\}$	Marketing intervention to customer i (e.g., *no offer* vs. *offer*)
$\boldsymbol{x_i} \in \mathbb{R}^d$	Observed feature vector for customer i
y_i	The response from customer i in pilot study
$\bar{y}_m^{(a)}$	The averaged response under action a in segment m
$\mathcal{C}$	A partition of pilot customers into disjoint segments: $\mathcal{C} = \{\mathcal{S}_1, \dots, \mathcal{S}_{\lvert\mathcal{C}\rvert}\}$
$\mathcal{C}_M$	Specifically denote a partition with M number of segments.
$\mathcal{S}_m$	The set of pilot customer indices assigned to segment m in segmentation $\mathcal{C}$
$\mathscr{C}$	The set of all admissible segmentations (e.g., satisfying estimation constraints)
$A_{\mathcal{C}}$	Segment assignment function for implementation customers: $\mathcal{I}^{\text{implement}} \to \{1, \dots, M\}$
$A_{\mathcal{C}}(j)$	Assigned segment index for implementation customer j, defined as $A_{\mathcal{C}}(j) = \phi_{\mathcal{C}}(\boldsymbol{x}_j)$
ϕ	Assignment rule mapping covariates and learned segmentation to a segment index
$\hat{f}_m^{\mathcal{C}}, \hat{\tau}_m^{\mathcal{C}}$	Estimated segment-level model from segment m induced by segmentation $\mathcal{C}$
$\pi^{\mathcal{C}}(j)$	Manager action: action assigned to customer
$\pi^\star(j)$	Oracle action: best action for customer j under full knowledge of ground truth models
$\mathcal{L}$	A node in the tree
$\widehat{v}_i$	Estimated individual profit after applying the learned policy to customer i
$\widehat{V}_{\mathcal{L}}$	Estimated profits of tree node $\mathcal{L}$
$V_{\mathcal{C}_M}^{\text{val}}$	The validation score of segmentation $\mathcal{C}$ with M segments
$R(\pi^{\mathcal{C}})$	Expected realized profit under policy $\pi^{\mathcal{C}}$
$R(\pi^\star)$	Oracle profit under policy $\pi^\star$

Table 2: Notations

4 Decision-aware segmentation tree for segmentation and estimation

We propose the **D**ecision-**A**ware **S**egmentation (DAS) framework, a unified approach explicitly designed to optimize the value of downstream managerial decisions. DAS consists of two key components:

- *Decision-Aware Segmentation Tree* (DAST), a decision-aware segmentation algorithm that constructs customer segments by directly maximizing expected decision value;
- *Decision-Aware Model Selection* (DAMS), a model-selection method that determines the optimal number of segments based on out-of-sample policy performance. Importantly, DAMS is agnostic to the specific segmentation algorithm employed.

In this section, we focuse on the first component, DAST, which performs segmentation and segment-level estimation. We begin by randomly splitting the pilot data $\{(\boldsymbol{x}_i, D_i, y_i)\}$ into two subsets: a *training set* $\mathcal{I}^{\text{train}}$ and a *validation set* $\mathcal{I}^{\text{val}}$. The training data are used to construct a full segmentation tree, as described in Section 4.1. Managers can subsequently prune this tree to any desired number of leaves, as described in Section 4.2, thereby controlling the level of segmentation granularity. The validation data are later used to conduct model selection, i.e., to determine the number of segments that maximizes out-of-sample performance, as described in Section 5.

Table 3 summarizes the three stages of the Decision-Aware Segmentation (DAS) framework, including tree construction, pruning, and model selection, along with their respective decision maker, data source, and algorithmic component.

Stage	**What happens**	**Who decides**	**Dataset**	**Algorithm**
1. Tree construction	Build the full tree	Algorithm	$\mathcal{I}^{\text{train}}$	Algorithm 1
2. Pruning	Collapse the tree to a given number of leaves M.	Manager's choice	$\mathcal{I}^{\text{train}}$	Algorithm 2
3. Model selection	Evaluate trees pruned to different M on validation data, and select M^*	Model-selection criterion	$\mathcal{I}^{\text{val}}$	Algorithm 3

Table 3: Conceptual hierarchy of the Decision-Aware Segmentation (DAS) procedure

4.1 Algorithm 1: Constructing a tree

The goal is to build a segmentation tree that partitions the customers into subgroups in a way that maximizes the estimated profits, as evaluated by (Athey and Wager, 2021; Dudík et al., 2011; Sverdrup et al., 2020). A DR score is a per-customer, per-action estimate of counterfactual profit. It combines outcome modeling with inverse-propensity weighting, yielding robustness against model misspecification. Specifically, for each customer $i \in \{1, \ldots, N^{\text{pilot}}\}$, we compute synthetic outcomes under both actions — one for receiving the offer ($a = 1$, (4)) and one for not receiving it ($a = 0$, (5)):

$$\Gamma_{i1} = \widehat{\mu}_1(\boldsymbol{x}_i) + \frac{D_i}{e_i} \left(y_i - \widehat{\mu}_1(\boldsymbol{x}_i)\right), \tag{4}$$

$$\Gamma_{i0} = \widehat{\mu}_0(\boldsymbol{x}_i) + \frac{1 - D_i}{1 - e_i} \left(y_i - \widehat{\mu}_0(\boldsymbol{x}_i)\right). \tag{5}$$

Here, $\widehat{\mu}_a(\boldsymbol{x}_i)$ is the predicted outcome for customer i under action $a \in \{0, 1\}$, which can be estimated using any supervised learning method, such as linear regression, causal forests, or neural networks. $D_i \in \{0, 1\}$ is the observed action, and $e_i = \mathbb{P}(D_i = 1 \mid \boldsymbol{x}_i)$ is the propensity score. In our experimental setting, as described in Section 3.2, treatments are assigned randomly with equal probability, so $e_i = 0.5$. The doubly robust estimates in (4)–(5) are unbiased for the potential outcomes when either the outcome model $\widehat{\mu}_a$ or the propensity model $e(\cdot)$ is correctly specified, hence the term *doubly-robust*.

We collect all synthetic outcomes into a DR score matrix $\boldsymbol{\Gamma} \in \mathbb{R}^{N^{\text{pilot}} \times 2}$, where each row $(\Gamma_{i1}, \Gamma_{i0})$ summarizes the potential outcomes under treatment ($a = 1$) and control ($a = 0$) for customer i.

Tree-growing procedure. We adopt a CART-style (Breiman et al., 1984) greedy decision tree algorithm. Unlike standard CART, which optimizes impurity or prediction error, our Decision-Aware Segmentation Tree uses the DR-based estimated profit to evaluate candidate splits.

At any node in the tree, let $\mathcal{L} \subseteq \{1, \ldots, N^{\text{train}}\}$ denote the set of customers contained in that node. When evaluating a candidate split of node $\mathcal{L}$ on feature j at threshold t, the customers are divided into two disjoint child nodes:

$$\mathcal{L}_L = \{i \in \mathcal{L} : x_{ij} \leq t\}, \qquad \mathcal{L}_R = \mathcal{L} \setminus \mathcal{L}_L.$$

For a given node $\mathcal{L}$, the algorithm computes its estimated total profit $\widehat{V}_{\mathcal{L}}$ using the function *ComputeNodeValue* in Algorithm 1. Specifically, let n_1 and n_0 denote the number of treated and control customers in $\mathcal{L}$, respectively. The estimated treatment effect within the node is

$$\widehat{\tau}_{\mathcal{L}} = \frac{1}{n_1} \sum_{i \in \mathcal{L} : D_i = 1} y_i - \frac{1}{n_0} \sum_{i \in \mathcal{L} : D_i = 0} y_i.$$

If the estimated effect is non-negative ($\widehat{\tau}_{\mathcal{L}} \geq 0$), the policy recommends treatment ($a_i = 1$) for all customers in the node; otherwise, no treatment ($a_i = 0$).

The expected profit for each customer $i \in \mathcal{L}$ is then

$$\widehat{v}_i = \begin{cases} y_i, & \text{if } D_i = a_i, \\ \Gamma_{i,a_i}, & \text{otherwise,} \end{cases}$$

where y_i is the observed outcome and Γ_{i,a_i} is the counterfactual DR score under the assigned action a_i. Summing over all customers in the node yields the total expected profit:

$$\widehat{V}_{\mathcal{L}} = \sum_{i \in \mathcal{L}} \widehat{v}_i.$$

The gain from a potential split is defined as the improvement in the DR-based value relative to the parent node:

$$\Delta V = \widehat{V}_{\mathcal{L}_L} + \widehat{V}_{\mathcal{L}_R} - \widehat{V}_{\mathcal{L}},$$

where $\widehat{V}_{\mathcal{L}_L}$ and $\widehat{V}_{\mathcal{L}_R}$ are the estimated profits of the left and right child nodes, respectively.

Among all feasible feature–threshold pairs (j, t) at node $\mathcal{L}$, the algorithm selects the split with the highest positive gain ΔV. A split is only considered if both child nodes satisfy the statistical-feasibility criterion, ensuring that each segment contains sufficient observations and treatment variation for reliable estimation.

The process repeats recursively for each child node until either the maximum tree depth $q_{\max}$ is reached or no candidate split yields a gain exceeding the tolerance ϵ. The detailed procedure for constructing the Decision-Aware Segmentation Tree is summarized in Algorithm 1.

Algorithm 1: Decision-Aware Segmentation Tree (entry point: *BuildDecisionAwareTree*)

```
Input: Training set of pilot data $\mathcal{D}_{\text{train}} = \{(\boldsymbol{x}_i, D_i, y_i)\}_{i \in \mathcal{I}^{\text{train}}}$; doubly robust score matrix $\mathbf{\Gamma}$; tolerance
       $\epsilon > 0$; maximum tree depth $q_{\max} > 0$; candidate threshold sets $\mathcal{H} = \{\mathcal{H}_j\}_{j=1}^{d}$; admissibility check
       function StatisticallyAdmissible$(\cdot)$ [a]
Output: A segmentation tree $\mathcal{T}$ of depth $q_{\max}$, where each internal node stores an optimal split $(j, t)$
Function GrowNode($\mathcal{L}$, $q$):
    if $q = q_{\max}$ then
        return TerminateNode;
    $bestGain \leftarrow -\infty$;
    $bestSplit \leftarrow \varnothing$;
    $\widehat{V}_{\mathcal{L}} \leftarrow$ ComputeNodeValue($\mathcal{L}$);
    for feature $j = 1$ to $d$ do
        for threshold $t \in \mathcal{H}_j$ do
            $\mathcal{L}_L \leftarrow \{i \in \mathcal{L} : x_{ij} \leq t\}$;
            $\mathcal{L}_R \leftarrow \mathcal{L} \setminus \mathcal{L}_L$;
            if not StatisticallyAdmissible($\mathcal{L}_L$) or not StatisticallyAdmissible($\mathcal{L}_R$) then
                continue;
            $\widehat{V}_{\mathcal{L}_L} \leftarrow$ ComputeNodeValue($\mathcal{L}_L$);
            $\widehat{V}_{\mathcal{L}_R} \leftarrow$ ComputeNodeValue($\mathcal{L}_R$);
            $gain \leftarrow \widehat{V}_{\mathcal{L}_L} + \widehat{V}_{\mathcal{L}_R} - \widehat{V}_{\mathcal{L}}$;
            if $gain > bestGain$ then
                $bestGain \leftarrow gain$ ;
                $bestSplit \leftarrow (j, t, \mathcal{L}_L, \mathcal{L}_R)$;
    if $bestGain \leq \epsilon$ then
        return TerminateNode;
    Split $\mathcal{L}$ at bestSplit into $\mathcal{L}_L$ and $\mathcal{L}_R$, and update global tree object $\mathcal{T}$;
    GrowNode($\mathcal{L}_L$, $q + 1$);
    GrowNode($\mathcal{L}_R$, $q + 1$);
Function ComputeNodeValue($\mathcal{L}$):
    $n_1 \leftarrow |\{i \in \mathcal{L} : D_i = 1\}|$;
    $n_0 \leftarrow |\{i \in \mathcal{L} : D_i = 0\}|$;
    $\widehat{\tau}_{\mathcal{L}} \leftarrow \frac{1}{n_1} \sum_{i \in \mathcal{L}: D_i = 1} y_i - \frac{1}{n_0} \sum_{i \in \mathcal{L}: D_i = 0} y_i$;
    $a_i \leftarrow \mathbf{1}\{\widehat{\tau}_{\mathcal{L}} \geq 0\},$ for all $i \in \mathcal{L}$;
    $\widehat{v}_i \leftarrow \begin{cases} y_i, & \text{if } D_i = a_i \\ \Gamma_{i,a_i}, & \text{otherwise} \end{cases}$ for all $i \in \mathcal{L}$;
    $\widehat{V}_{\mathcal{L}} \leftarrow \sum_{i \in \mathcal{L}} \widehat{v}_i$;
    return $\widehat{V}_{\mathcal{L}}$;
Function BuildDecisionAwareTree($\mathcal{D}$):
    Initialize $q \leftarrow 0$;
    Initialize empty tree $\mathcal{T}$;
    Initialize root $\mathcal{L} \leftarrow \mathcal{I}^{\text{train}}$;
    GrowNode($\mathcal{L}, q$);
    return $\mathcal{T}$;
```

[a] The function StatisticallyAdmissible$(\cdot)$ encodes the requirements that each segment must satisfy for reliable estimation of the segment-level parameters. These requirements depend on the estimation method and may include, for example, sufficient sample size, and treatment variation.

4.2 Algorithm 2: Post-pruning the segmentation tree and estimating parameters

After constructing a fully grown segmentation tree of depth $q_{\max}$, we apply a decision-aware post-pruning procedure to reduce the number of leaf nodes to the desired value M. We implement a bottom-up pruning algorithm that iteratively removes the least beneficial splits. At each iteration, we identify an internal node whose two children are both leaves, and we evaluate the gain from keeping the split versus collapsing it. This process continues until exactly M leaf segments remain. Once pruning is complete, we estimate response models within each segment. We present the procedure in Algorithm 2.

Algorithm 2: Post-Pruning Tree and Estimation

```
Input: Fully grown tree $\mathcal{T}$; Training data $\mathcal{D}_{\text{train}} = \{(\boldsymbol{x}_i, D_i, y_i)\}$ used to build the tree; DR score matrix
    $\boldsymbol{\Gamma}$; target number of segments $M$; parameter estimator $\widehat{\Theta}(\cdot)$
Output: Pruned tree $\mathcal{T}_{\text{pruned}}$ with $M$ leaves, with leaf nodes storing estimated parameters $\{(\hat{f}_m, \hat{\tau}_m)\}_{m=1}^{M}$
Function PostPruneTree($\mathcal{T}, M$):
    while LeafCount($\mathcal{T}$) $> M$ do
        $\Delta^* \leftarrow \infty$;
        $\mathcal{L}^* \leftarrow$ null;
        for internal node $\mathcal{L}$ whose children are both leaves do
            Let $\mathcal{L}_L, \mathcal{L}_R$ be $\mathcal{L}$'s left child and right child;
            // Call the ComputeNodeValue function in algorithm 1
            $\widehat{V}_{\mathcal{L}_L} \leftarrow$ ComputeNodeValue($\mathcal{L}_L$);
            $\widehat{V}_{\mathcal{L}_R} \leftarrow$ ComputeNodeValue($\mathcal{L}_R$);
            $\widehat{V}_{\mathcal{L}} \leftarrow$ ComputeNodeValue($\mathcal{L}$);
            $\Delta V_{\mathcal{L}} \leftarrow (\widehat{V}_{\mathcal{L}_L} + \widehat{V}_{\mathcal{L}_R}) - \widehat{V}_{\mathcal{L}}$;
            if $\Delta V_{\mathcal{L}} < \Delta^*$ then
                $\Delta^* \leftarrow \Delta V_{\mathcal{L}}$;
                $\mathcal{L}^* \leftarrow \mathcal{L}$;
        // Prune the split at $\mathcal{L}^*$
        Collapse children of $\mathcal{L}^*$ and reassign it as a leaf;
        // Estimate parameters in each leaf after pruning
        for leaf node $\mathcal{L}_m$ in $\mathcal{T}$ do
            $(\hat{f}_m, \hat{\tau}_m) \leftarrow \widehat{\Theta}(\mathcal{L}_m)$;
    return ($\mathcal{T}_{\text{pruned}}, \{(\hat{f}_m, \hat{\tau}_m)\}_{m=1}^{M}$
```

5 Decision-aware model selection (DAMS): choosing the number of segments

The second component of the Decision-Aware Segmentation (DAS) framework is the *Decision-Aware Model Selection* (DAMS) procedure. Importantly, our DAMS is agnostic to the specific segmentation algorithm employed: it can be applied to any segmentation method that defines segment-level treatment policies, not limited to DAST.

In Section 7, we conduct ablation studies to disentangle the contribution of segmentation and model selection. Specifically, we combine DAMS with alternative segmentation algorithms (e.g., GMM, k-means) to isolate and quantify the incremental value of decision-aware model selection.

Overview. Given a sequence of candidate segmentation models $\{\mathcal{C}_M\}$ with different numbers of segments $M \in \{1, \ldots, M_{\max}\}$, DAMS evaluates each model using the validation dataset $\mathcal{I}^{\text{val}}$. For each customer $i \in \mathcal{I}^{\text{val}}$, the procedure assigns the customer to a segment according to the segmentation model, applies the corresponding decision rule in that segment, and estimates the expected outcome using observed or counterfactual information. Specifically, if the assigned action matches the customer's observed action in the pilot data, the realized outcome y_i is used; otherwise, the corresponding DR score Γ_{i,a_i} is used as the counterfactual outcome estimate. The validation score of segmentation with M segments, denoted $V^{\text{val}}_{\mathcal{C}_M}$, is the average of the estimated outcomes across all validation customers:

$$V^{\text{val}}_{\mathcal{C}_M} = \frac{1}{N_{\text{val}}} \sum_{i \in \mathcal{I}^{\text{val}}} \Big[\mathbf{1}\{D_i = a_i\}\, y_i + \mathbf{1}\{D_i \neq a_i\}\, \Gamma_{i,a_i} \Big].$$

The optimal number of segments M^* is selected as the model achieving the highest validation value:

$$M^* = \arg \max_{M \in \{1, \ldots, M_{\max}\}} V^{\text{val}}_{\mathcal{C}_M}.$$

This decision-aware model selection method ensures that the chosen segmentation delivers the best out-of-sample policy performance. We present the method in Algorithm 3.

After choosing the number of segments, we obtain the final segmentation structure by retraining on the entire pilot dataset to maximize data efficiency.

Algorithm 3: Decision-Aware Model Selection (DAMS)

```
Input: Validation data $\mathcal{D}_{\text{val}} = \{(\boldsymbol{x}_i, D_i, y_i)\}_{i \in \mathcal{I}^{\text{val}}}$; DR score matrix $\boldsymbol{\Gamma}$; Set of candidate segmentation
       models $\{\mathcal{C}_M\}$ for $M \in \{1, \ldots, M_{\max}\}$; Policy function $\pi^{\mathcal{C}_M}(i)$ induced by each segmentation $\mathcal{C}_M$.
Output: Optimal number of segments $M^*$
Function EvaluateModel($\mathcal{C}_M$):
    for each customer $i \in \mathcal{I}^{val}$ do
        Assign $i$ to segment $m$ according to $\mathcal{C}_M$;
        $a_i \leftarrow \pi^{\mathcal{C}_M}(i)$ // assigned action under policy $\pi^{\mathcal{C}_M}$
        if $D_i = a_i$ then
            $\widehat{v}_i \leftarrow y_i$;
        else
            $\widehat{v}_i \leftarrow \Gamma_{i,a_i}$;
    $V^{\text{val}}_{\mathcal{C}_M} \leftarrow \frac{1}{N_{\text{val}}} \sum_{i \in \mathcal{I}^{\text{val}}} \widehat{v}_i$;
    return $V^{\text{val}}_{\mathcal{C}_M}$;
Function DecisionAwareModelSelection($\{\mathcal{C}_M\}, \mathcal{D}_{val}$):
    for each $M \in \{1, \ldots, M_{\max}\}$ do
        $V^{\text{val}}_{\mathcal{C}_M} \leftarrow$ EvaluateModel($\mathcal{C}_M$);
    $M^* \leftarrow \arg\max_M V^{\text{val}}_{\mathcal{C}_M}$;
    return $M^*$;
```

6 Assessment standards

One of our central research questions is: *How should we assess whether a particular segmentation is "good"?* To evaluate performance in simulation studies, we propose a set of oracle-based assessment criteria along three complementary dimensions: (1) **structure recovery**, (2) **estimation accuracy**, and (3) **policy effectiveness**.

While these metrics are not computable in practice due to dependence on unobserved ground-truth information, they serve as idealized evaluation standards in synthetic settings, enabling rigorous benchmarking and methodological comparisons across segmentation approaches.

6.1 Structure oracle

A *structure oracle* evaluates how well the estimated segmentation recovers the ground-truth latent structure. Specifically, it compares the true latent segment labels $\{Z_i\}_{i \in \mathcal{I}^{\text{pilot}}}$ to the estimated segmentation

Table 4: Oracle evaluation metrics for a candidate segmentation of size M

Dimension	Oracle Inputs	Evaluation Set	Metrics
Structure Recovery	$\{\{Z_i\},\{C_i\}\}$	$\mathcal{I}^{\text{pilot}}$	Adjusted Rand Index (ARI) Normalized Mutual Information (NMI)
Estimation Accuracy	$\{\{f_k,\tau_k\}_{k=1}^K,\{Z_i\},\{C_i\}\}$	$\mathcal{I}^{\text{pilot}}$	Parameter-level MSE Outcome-level MSE
Policy Effectiveness	$\{\{f_k,\tau_k\}_{k=1}^K,\{Z_i\},\{C_i\},\pi^{\mathcal{C}}\}$	$\mathcal{I}^{\text{implement}}$	Regret Mis-treatment rate

$\mathcal{C}$. Since segment labels are inherently non-identifiable up to permutation, these metrics assess clustering *agreement*, not exact label match. We use the following commonly adopted measures:

- **Adjusted Rand Index (ARI):** Quantifies the similarity between the true segmentation and the estimated segmentation, adjusted for chance agreement. ARI takes values in $[-1, 1]$, with 1 indicating perfect agreement, 0 representing chance-level agreement, and negative values implying less agreement than expected by chance.
- **Normalized Mutual Information (NMI):** Measures the mutual dependence between the true segmentation and the estimated segmentation, normalized by the entropies of each clustering. NMI ranges from 0 (no shared information) to 1 (perfect agreement), and is invariant to permutations of cluster labels. It is particularly useful for comparing clusterings with different numbers of clusters

6.2 Estimation Oracle

An *estimation oracle* evaluates how accurately the manager's estimated segment-level models recover the ground-truth segment parameters. Each pilot customer $i \in \mathcal{I}^{\text{pilot}}$ is assigned to a segment $C_i \in \{1, \ldots, |\mathcal{C}|\}$ under the manager's segmentation, and has a latent ground-truth segment label $Z_i \in \{1, \ldots, K\}$. We consider two evaluation metrics:

- **Parameter-level MSE.** For parametric baseline models $f_k(\cdot)$, We assume they belongs to a parametric family, such that it is fully specified by a finite-dimensional parameter vector $\psi_k \in \mathbb{R}^q$. We then define the full segment-level parameter vector as:

$$\theta_k = (\psi_k, \tau_k) \in \mathbb{R}^{q+1}, \qquad \hat{\theta}_m = (\hat{\psi}_m, \hat{\tau}_m) \in \mathbb{R}^{q+1}.$$

Then the *parameter-level mean squared error* compares the estimated parameters from the assigned segment to the ground-truth segment:

$$\mathrm{MSE}_{\mathrm{param}} = \frac{1}{N^{\mathrm{pilot}}} \sum_{i \in \mathcal{I}^{\mathrm{pilot}}} \left\| \hat{\theta}_{C_i} - \theta_{Z_i} \right\|_2^2, \tag{6}$$

where $\| \cdot \|_2$ is Euclidean norm.

- **Outcome-level MSE.** This metric applies to any model class, including nonparametric functions, and evaluates predictive accuracy of counterfactual profit outcomes:

  $$\hat{y}_i(a) = \hat{f}^{\mathcal{C}}_{C_i}(\boldsymbol{x}_i) + \hat{\tau}^{\mathcal{C}}_{C_i} \cdot \mathbf{1}\{a = 1\}, \qquad y_i^{\mathrm{true}}(a) = f_{Z_i}(\boldsymbol{x}_i) + \tau_{Z_i} \cdot \mathbf{1}\{a = 1\}.$$

  The outcome-level mean squared error is:

  $$\mathrm{MSE}_{\mathrm{outcome}} = \frac{1}{N^{\mathrm{pilot}}} \sum_{i \in \mathcal{I}^{\mathrm{pilot}}} \sum_{a \in \{0,1\}} \Big(\hat{y}_i(a) - y_i^{\mathrm{true}}(a) \Big)^2. \tag{7}$$

6.3 Policy oracle

A *policy oracle* evaluates how effectively the manager's targeting policy performs when deployed to new customers in the implementation set. In particular, we compare the expected profit of a policy $\pi^{\mathcal{C}}$ constructed from segmentation $\mathcal{C}$, against the best possible policy under full knowledge, over implementation customers.

We denote the expected profit (or expected return) of a policy π on the implementation set by $R(\pi)$. In particular, the expected realized profit generated under the manager's policy $\pi^{\mathcal{C}}$ is:

$$R(\pi^{\mathcal{C}}) = \frac{1}{N^{\mathrm{implement}}} \sum_{j \in \mathcal{I}^{\mathrm{implement}}} \Big[f_{Z_j}(\boldsymbol{x}_j) + \tau_{Z_j} \cdot \mathbf{1}\{\pi^{\mathcal{C}}(j) = 1\} \Big], \tag{8}$$

where Z_j is the latent ground-truth segment index for implementation customer j, $\pi^{\mathcal{C}}(j)$ is the assigned treatment decision.

The *oracle policy* $\pi^\star$ chooses the action that maximizes each customer's true expected profit:

$$\begin{aligned} \pi^\star(j) &= \arg\max_{a \in \{0,1\}} \Big[f_{Z_j}(\boldsymbol{x}_j) + \tau_{Z_j} \cdot \mathbf{1}\{a = 1\} \Big] && (9) \\ &= \mathbf{1}\{\tau_{Z_j} > 0\}. && (10) \end{aligned}$$

The corresponding *oracle profit* is denoted by $R(\pi^\star)$, and is given by:

$$R(\pi^\star) = \frac{1}{N^{\text{implement}}} \sum_{j \in \mathcal{I}^{\text{implement}}} \max\left\{ f_{Z_j}(\boldsymbol{x}_j),\ f_{Z_j}(\boldsymbol{x}_j) + \tau_{Z_j} \right\} \tag{11}$$

$$= \frac{1}{N^{\text{implement}}} \sum_{j \in \mathcal{I}^{\text{implement}}} \left(f_{Z_j}(\boldsymbol{x}_j) + \max\{0,\ \tau_{Z_j}\} \right). \tag{12}$$

We define two evaluation metrics:

- **Regret.** The regret of the manager's policy is the loss in realized profit relative to the oracle profit:

$$\text{Regret}_{\mathcal{C}} = R(\pi^\star) - R(\pi^{\mathcal{C}}). \tag{13}$$

- **Mis-treatment rate.** This metric computes the fraction of implementation customers who are mistreated relative to their optimal action:

$$\text{MisTreat}_{\mathcal{C}} = \frac{1}{N^{\text{implement}}} \sum_{j \in \mathcal{I}^{\text{implement}}} \mathbf{1}\left\{ \pi^{\mathcal{C}}(j) \neq \pi^\star(j) \right\}. \tag{14}$$

7 Expriment

...

References

Akaike, H. (1974). A new look at the statistical model identification. *IEEE Transactions on Automatic Control*, 19(6):716–723.

Alves Gomes, M. and Meisen, T. (2023). A review on customer segmentation methods for personalized customer targeting in e-commerce use cases. *Information Systems and e-Business Management*, 21(3):527–570.

Ansari, A. and Mela, C. F. (2003). E-customization. *Journal of marketing research*, 40(2):131–145.

Antoniak, C. E. (1974). Mixtures of dirichlet processes with applications to bayesian nonparametric problems. *The annals of statistics*, pages 1152–1174.

Aouad, A., Elmachtoub, A. N., Ferreira, K. J., and McNellis, R. (2023). Market segmentation trees. *Manufacturing & Service Operations Management*, 25(2):648–667.

Ascarza, E. (2018). Retention futility: Targeting high-risk customers might be ineffective. *Journal of Marketing Research*, 55(1):80–98.

Athey, S. and Imbens, G. (2016). Recursive partitioning for heterogeneous causal effects. *Proceedings of the National Academy of Sciences*, 113(27):7353–7360.

Athey, S. and Wager, S. (2021). Policy learning with observational data. *Econometrica*, 89(1):133–161.

Bapna, R., Goes, P., Wei, K. K., and Zhang, Z. (2011). A finite mixture logit model to segment and predict electronic payments system adoption. *Information Systems Research*, 22(1):118–133.

Bergemann, D., Heumann, T., and Wang, M. C. (2024). Consumer-optimal segmentation in multi-product markets. Cowles Foundation Discussion Papers, Yale University.

Bertsimas, D. and Kallus, N. (2020). From predictive to prescriptive analytics. *Management Science*, 66(3):1025–1044.

Bezdek, J. C., Ehrlich, R., and Full, W. (1984). Fcm: The fuzzy c-means clustering algorithm. *Computers & Geosciences*, 10(2-3):191–203.

Biernacki, C., Celeux, G., and Govaert, G. (2002). Assessing a mixture model for clustering with the integrated completed likelihood. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 22(7):719–725.

Breiman, L., Friedman, J., Olshen, R. A., and Stone, C. J. (1984). *Classification and regression trees*. Chapman and Hall/CRC.

Bult, J. R. and Wansbeek, T. (1995). Optimal selection for direct mail. *Marketing Science*, 14(4):378–394.

Chan, T. C., Mahmood, R., and Zhu, I. Y. (2025). Inverse optimization: Theory and applications. *Operations Research*, 73(2):1046–1074.

Chen, Y.-W., Ascarza, E., and Netzer, O. (2024). Policy-aware experimentation: Strategic sampling for optimized targeting policies. *Columbia Business School Research Paper*, (5044549).

Chickering, D. M. and Heckerman, D. (2000). A decision theoretic approach to targeted advertising. *Proceedings of the Sixteenth Conference on Uncertainty in Artificial Intelligence.*

Chung, T.-H., Rostami, V., Bastani, H., and Bastani, O. (2022). Decision-aware learning for optimizing health supply chains. *arXiv preprint arXiv:2211.08507*.

Dempster, A. P., Laird, N. M., and Rubin, D. B. (1977). Maximum likelihood from incomplete data via the em algorithm. *Journal of the Royal Statistical Ssociety: Series B (methodological)*, 39(1):1–22.

DeSarbo, W. S. and Cron, W. L. (1988). A maximum likelihood methodology for clusterwise linear regression. *Journal of Classification*, 5:249–282.

Dubé, J.-P., Fang, Z., Fong, N., and Luo, X. (2017). Competitive price targeting with smartphone coupons. *Marketing Science*, 36(6):944–975.

Dubé, J.-P. and Misra, S. (2023). Personalized pricing and consumer welfare. *Journal of Political Economy*, 131(1):131–189.

Dudík, M., Langford, J., and Li, L. (2011). Doubly robust policy evaluation and learning. *Proceedings of the 28th International Conference on Machine Learning.*

El Balghiti, O., Elmachtoub, A. N., Grigas, P., and Tewari, A. (2019). Generalization bounds in the predict-then-optimize framework. *Advances in Neural Information Processing Systems*, 32.

Elmachtoub, A. N. and Grigas, P. (2022). Smart "predict, then optimize". *Management Science*, 68(1):9–26.

Elmachtoub, A. N., Liang, J. C. N., and McNellis, R. (2020). Decision trees for decision-making under the predict-then-optimize framework. In *International Conference on Machine Learning*, pages 2858–2867. PMLR.

Fong, N. M. (2017). How targeting affects customer search: A field experiment. *Management Science*, 63(7):2353–2364.

Fong, N. M., Fang, Z., and Luo, X. (2015). Geo-conquesting: Competitive locational targeting of mobile promotions. *Journal of Marketing Research*, 52(5):726–735.

Fraley, C. and Raftery, A. E. (1998). How many clusters? which clustering method? answers via model-based cluster analysis. *The Computer Journal*, 41(8):578–588.

Gallego, G. and Berbeglia, G. (2024). Bounds and heuristics for multiproduct pricing. *Management Science*, 70(6):4132–4144.

Ghose, A., Li, B., and Liu, S. (2019). Mobile targeting using customer trajectory patterns. *Management Science*, 65(11):5027–5049.

Goldfarb, A. and Tucker, C. (2011). Online display advertising: Targeting and obtrusiveness. *Marketing Science*, 30(3):389–404.

Gordon, B. R., Zettelmeyer, F., Bhargava, N., and Chapsky, D. (2019). A comparison of approaches to advertising measurement: Evidence from big field experiments at facebook. *Marketing Science*, 38(2):193–225.

Govind, R., Chatterjee, R., and Mittal, V. (2018). Segmentation of spatially dependent geographical units: Model and application. *Management Science*, 64(4):1941–1956.

Hartigan, J. A. and Wong, M. A. (1979). Algorithm as 136: A k-means clustering algorithm. *Journal of the Royal Statistical Society: Series C*, 28(1):100–108.

Hinz, O., Skiera, B., Barrot, C., and Becker, J. U. (2011). Seeding strategies for viral marketing: An empirical comparison. *Journal of Marketing*, 75(6):55–71.

Hitsch, G. J., Misra, S., and Zhang, W. W. (2024). Heterogeneous treatment effects and optimal targeting policy evaluation. *Quantitative Marketing and Economics*, 22(2):115–168.

Huang, M. and Gupta, V. (2024). Decision-focused learning with directional gradients. *Advances in Neural Information Processing Systems*, 37:79194–79220.

Huang, T.-W. and Ascarza, E. (2024). Doing more with less: Overcoming ineffective long-term targeting using short-term signals. *Marketing Science*, 43(4):863–884.

Iyer, G. and Ke, T. T. (2024). Competitive model selection in algorithmic targeting. *Marketing Science*, 43(6):1226–1241.

Iyer, G., Soberman, D., and Villas-Boas, J. M. (2005). The targeting of advertising. *Marketing Science*, 24(3):461–476.

Johansson, F., Shalit, U., and Sontag, D. (2016). Learning representations for counterfactual inference. In *International Conference on Machine Learning (ICML)*.

Kallus, N. (2018). Balanced policy evaluation and learning. *Advances in neural information processing systems*, 31.

Kallus, N. and Zhou, A. (2018). Confounding-robust policy improvement. *Advances in neural information processing systems*, 31.

Kallus, N. and Zhou, A. (2021). Minimax-optimal policy learning under unobserved confounding. *Management Science*, 67(5):2870–2890.

Karle, H. and Reisinger, M. (2025). Imperfect targeting and advertising strategies. *Management Science*, 71(6):5106–5122.

Kaufman, L. and Rousseeuw, P. J. (2009). *Finding groups in data: an introduction to cluster analysis*. John Wiley & Sons.

Kitagawa, T. and Tetenov, A. (2018). Who should be treated? empirical welfare maximization methods for treatment choice. *Econometrica*, 86(2):591–616.

Künzel, S. R., Sekhon, J. S., Bickel, P. J., and Yu, B. (2019). Metalearners for estimating heterogeneous treatment effects using machine learning. *Proceedings of the National Academy of Sciences*, 116(10):4156–4165.

Lemmens, A. and Gupta, S. (2020). Managing churn to maximize profits. *Marketing Science*, 39(5):956–973.

Lemmens, A., Roos, J. M., Gabel, S., Ascarza, E., Bruno, H., Gordon, B. R., Israeli, A., Feit, E. M., Mela, C. F., and Netzer, O. (2025). Advancing personalization: How to experiment, learn, & optimize. *Columbia Business School Research Paper*, (4878819).

Li, T., Song, X., Zhang, Y., Zhu, H., and Zhu, Z. (2021). Clusterwise functional linear regression models. *Computational Statistics & Data Analysis*, 158:107192.

Liu, B., Xia, Y., and Yu, P. S. (2000). Clustering through decision tree construction. In *Proceedings of the ninth international conference on Information and knowledge management*, pages 20–29.

Liu, Y., Ram, S., Lusch, R. F., and Brusco, M. (2010). Multicriterion market segmentation: a new model, implementation, and evaluation. *Marketing Science*, 29(5):880–894.

Lu, H., Simester, D., and Zhu, Y. (2025). Optimizing scalable targeted marketing policies with constraints. *Marketing Science*.

Luo, X., Andrews, M., Fang, Z., and Phang, C. W. (2014). Mobile targeting. *Management Science*, 60(7):1738–1756.

MacQueen, J. (1967). Multivariate observations. In *Proceedings ofthe 5th Berkeley Symposium on Mathematical Statisticsand Probability*, volume 1, pages 281–297.

Mandi, J., Foschini, M., Höller, D., Thiébaux, S., Hoffmann, J., and Guns, T. (2024). Decision-focused learning to predict action costs for planning. In *ECAI 2024*, pages 4060–4067. IOS Press.

Mandi, J., Stuckey, P. J., Guns, T., et al. (2020). Smart predict-and-optimize for hard combinatorial optimization problems. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 1603–1610.

Manski, C. F. (2004). Statistical treatment rules for heterogeneous populations. *Econometrica*, 72(4):1221–1246.

Müllner, D. (2011). Modern hierarchical, agglomerative clustering algorithms. *arXiv preprint arXiv:1109.2378*.

Murtagh, F. and Legendre, P. (2014). Ward's hierarchical agglomerative clustering method: which algorithms implement ward's criterion? *Journal of classification*, 31(3):274–295.

Ng, R. T. and Han, J. (2002). Clarans: A method for clustering objects for spatial data mining. *IEEE transactions on knowledge and data engineering*, 14(5):1003–1016.

Park, H.-S. and Jun, C.-H. (2009). A simple and fast algorithm for k-medoids clustering. *Expert Systems with Applications*, 36(2):3336–3341.

Park, Y. W., Jiang, Y., Klabjan, D., and Williams, L. (2017). Algorithms for generalized clusterwise linear regression. *INFORMS Journal on Computing*, 29(2):301–317.

Rafieian, O. and Yoganarasimhan, H. (2021). Targeting and privacy in mobile advertising. *Marketing Science*, 40(2):193–218.

Ronen, M., Finder, S. E., and Freifeld, O. (2022). Deepdpm: Deep clustering with an unknown number of clusters. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9861–9870.

Rossi, P. E., McCulloch, R. E., and Allenby, G. M. (1996). The value of purchase history data in target marketing. *Marketing Science*, 15(4):321–340.

Rößler, J. and Schoder, D. (2022). Bridging the gap: A systematic benchmarking of uplift modeling and heterogeneous treatment effects methods. *Journal of Interactive Marketing*, 57(4):629–650.

Rousseeuw, P. J. (1987). Silhouettes: a graphical aid to the interpretation and validation of cluster analysis. *Journal of computational and applied mathematics*, 20:53–65.

Rzepakowski, P. and Jaroszewicz, S. (2012). Decision trees for uplift modeling with single and multiple treatments. *Knowledge and Information Systems*, 32(2):303–327.

Sahoo, R. and Wager, S. (2025). Policy learning with competing agents. *Operations Research*.

Sarkar, P., Khanapuri, V. B., and Tiwari, M. K. (2025). Integration of prediction and optimization for smart stock portfolio selection. *European Journal of Operational Research*, 321(1):243–256.

Schwarz, G. (1978). Estimating the dimension of a model. *The Annals of Statistics*, pages 461–464.

Sculley, D. (2010). Web-scale k-means clustering. In *Proceedings of the 19th international conference on World wide web*, pages 1177–1178.

Si, N., Zhang, F., Zhou, Z., and Blanchet, J. (2023). Distributionally robust batch contextual bandits. *Management Science*, 69(10):5772–5793.

Simester, D., Timoshenko, A., and Zoumpoulis, S. I. (2020a). Efficiently evaluating targeting policies: Improving on champion vs. challenger experiments. *Management Science*, 66(8):3412–3424.

Simester, D., Timoshenko, A., and Zoumpoulis, S. I. (2020b). Targeting prospective customers: Robustness of machine-learning methods to typical data challenges. *Management Science*, 66(6):2495–2522.

Simester, D., Timoshenko, A., and Zoumpoulis, S. I. (2025). A sample size calculation for training and certifying targeting policies. *Management Science*.

Smith, A. N., Seiler, S., and Aggarwal, I. (2023). Optimal price targeting. *Marketing Science*, 42(3):476–499.

Stratman, E. G., Boutilier, J. J., and Albert, L. A. (2024). Decision-aware predictive model selection for workforce allocation. *arXiv preprint arXiv:2410.07932*.

Sugar, C. A. and James, G. M. (2003). Finding the number of clusters in a dataset: An information-theoretic approach. *Journal of the American Statistical Association*, 98(463):750–763.

Sverdrup, E., Kanodia, A., Zhou, Z., Athey, S., and Wager, S. (2020). policytree: Policy learning via doubly robust empirical welfare maximization over trees. *Journal of Open Source Software*, 5(50):2232.

Tetenov, A. (2012). Statistical treatment choice based on asymmetric minimax regret criteria. *Journal of Econometrics*, 166(1):157–165.

Valenti, A., Miller, C. J., and Tucker, C. E. (2025). Combining ad targeting techniques: Evidence from a field experiment in the auto industry. *Management Science*.

van de Geer, R. and den Boer, A. V. (2022). Price optimization under the finite-mixture logit model. *Management Science*, 68(10):7480–7496.

Vanderschueren, T., Verdonck, T., Baesens, B., and Verbeke, W. (2022). Predict-then-optimize or predict-and-optimize? an empirical evaluation of cost-sensitive learning strategies. *Information Sciences*, 594:400–415.

Venkatesan, R. and Kumar, V. (2004). A customer lifetime value framework for customer selection and resource allocation strategy. *Journal of marketing*, 68(4):106–125.

Vermunt, J. K. and Magidson, J. (2002). Latent class cluster analysis. *Applied Latent Class Analysis*, 11(89-106):60.

Wager, S. and Athey, S. (2018). Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association*, 113(523):1228–1242.

Wan, S., Zheng, C., Sun, Z., Xu, M., Yang, X., Zhu, H., and Guo, J. (2022). Gcf: Generalized causal forest for heterogeneous treatment effect estimation in online marketplace. *arXiv preprint arXiv:2203.10975*.

Wang, W., Li, B., Luo, X., and Wang, X. (2023). Deep reinforcement learning for sequential targeting. *Management Science*, 69(9):5439–5460.

Wedel, M. and DeSarbo, W. S. (1994). A review of recent developments in latent class regression models. *Advanced methods of marketing research*, pages 352–388.

Wedel, M. and Kamakura, W. A. (2000). *Market segmentation: Conceptual and methodological foundations*. Springer Science & Business Media.

Xie, J., Girshick, R., and Farhadi, A. (2016). Unsupervised deep embedding for clustering analysis. In *International Conference on Machine Learning (ICML)*, pages 478–487. PMLR.

Yang, J., Eckles, D., Dhillon, P., and Aral, S. (2024). Targeting for long-term outcomes. *Management Science*, 70(6):3841–3855.

Yang, J., Liu, C., Teng, M., Liao, M., and Xiong, H. (2016). Buyer targeting optimization: A unified customer segmentation perspective. In *2016 IEEE International Conference on Big Data (Big Data)*. IEEE.

Zhan, R., Ren, Z., Athey, S., and Zhou, Z. (2024). Policy learning with adaptively collected data. *Management Science*, 70(8):5270–5297.

Zhang, W. and Misra, S. (2024). Coarse personalization. In *Proceedings of the 25th ACM Conference on Economics and Computation*.

Zhou, Z., Athey, S., and Wager, S. (2023). Offline multi-action policy learning: Generalization and optimization. *Operations Research*, 71(1):148–183.