Design and Evaluation of Personalized Targeting Policies: Application to Free Trials

Hema Yoganarasimhan *
University of Washington

Ebrahim Barzegary*
University of Washington

Abhishek Pani*
Adobe Systems Incorporated

November 21, 2019

*We are grateful to an anonymous firm for providing the data and to UW-Foster High-Performance Computing Lab for providing us with computing resources. We thank the participants of the 2018 Marketing Science conference and the Triennial Choice Symposium. Thanks are also due to seminar audiences at the John Hopkins University, University of California Berkeley, and University of Maryland for their feedback. Please address all correspondence to: hemay@uw.edu, ebar@uw.edu.

Abstract

Effectively targeting at scale is one of the most important problems that today's firms face. We provide a three-pronged framework to design and evaluate personalized targeting policies that is compatible with a high-dimensional covariate space. First, we define the optimal policy design problem and describe two solution approaches – (a) policy design using outcome estimates, and (b) policy design using CATE estimates. Second, we consider five outcome estimators (linear regression, lasso, CART, random forest, and XGBoost) and two CATE estimators (causal tree and causal forest) for this task. Third, we use the Inverse Propensity Score (IPS) estimator to evaluate the reward from any targeting policy offline. We apply our framework to data from a large-scale field experiment on free trials conducted by a leading SaaS firm, where new users were randomly assigned to 7, 14, or 30 days of free trial. Among the uniform targeting policies, the 7-days-for-all policy maximizes the subscription rate. Next, we design personalized targeting policies to optimize subscriptions. We find that policies based on the outcome estimators – lasso and XGBoost – offer the best performance. In contrast, policies based on the two CATE estimators – causal tree and causal forest – perform poorly because they are unable to personalize the policy sufficiently. We then link a method's effectiveness in designing a policy with its ability to personalize the treatment sufficiently without overfitting (i.e., capture spurious heterogeneity). Finally, we show that policies designed to maximize short-run conversions also perform well on long-run outcomes such as consumer loyalty and profitability.

Keywords: Targeting, Personalization, counterfactual policy evaluation, field experiment, machine learning, policy design, digital marketing, free trials

1 Introduction

1.1 Targeting and Personalization

Which consumers to target with what promotions is one of the most important marketing decisions that a firm makes. Historically, firms based their targeting decisions on simple rules or cut-offs determined by a few demographic or behavioral variables. However, over the years, both managers and researchers have come to recognize the limitations of such simplistic targeting policies. A large stream of marketing literature has therefore focused on developing better targeting models that exploit user-level heterogeneity to personalize targeting; see Rossi et al. (1996) and Ansari and Mela (2003) for a couple of representative examples. While these papers, and others in this vein, represent some of the most significant advances in targeting models in the marketing literature, they nevertheless suffer from a key drawback. They require the researcher to impose strong parametric assumptions on users' response behavior. This works well in settings where we have a good theoretical understanding of user behavior and a few variables on which marketers can base their targeting decisions. However, in many settings, we do not have good theoretical foundations of consumer behavior and the dimensionality of the variables on which the firm can target is very high (e.g., click or conversion in digital settings). In such cases, it is hard for the firm to come up with the correct user-level response function to effectively personalize targeting policies.

To overcome these limitations, researchers are now turning to machine-learning models that use data-driven approaches to "learn" users' response function and come up with personalized targeting policies; see Rafieian and Yoganarasimhan (2018) and Simester et al. (2019b) for a couple of recent examples. However, no research has looked at the question of how to empirically design the *optimal* targeting policy that maximizes some outcome of interest for the firm (e.g., conversion, revenues) based on *offline* evaluation (i.e., evaluation without actually deploying the policy in the field).

1.2 Research Agenda and Challenges

In this paper, our goal is to provide a general framework to design and evaluate personalized targeting policies without deploying any policies in the field. In the process, we are interested in understanding the suitability of two broad classes of estimators in designing policy – (1) outcome estimators and (2) Conditional Average Treatment Effect (CATE) estimators. Finally, we want to examine how well targeting policies designed to maximize a short term outcome (e.g., conversion) perform on long-term outcomes (e.g., retention or revenues).

We face three main challenges in this task. First, from a theoretical perspective, searching for the optimal policy is a non-trivial task because the cardinality of the policy space (Π) can be quite large. In a setting where the firm can take W potential actions (or treatments) and we have D variables

on which the firm can base its targeting decisions (such that the d-th variable can take c_d different values), the total number of possible policies is $|\Pi| = W^{\prod_{d=1}^{D} c_d}$. This translates to $3^{987,840}$ policies in our application setting. So an un-directed search over the policy space is not feasible.

Second, the covariate space is high dimensional in most settings. So we will not have sufficient data in each sub-region of the covariate space to learn how well a policy performs there. We, therefore, require methods that can automatically pool together observations from sub-regions that show similar responsiveness to the firm's actions and learn the optimal policy over groups of similar sub-regions. This task is complicated by the low signal-to-noise ratio in most marketing settings, i.e., the variation in the firm's marketing action often only explains a small fraction of the variation in the outcome of interest (Lewis et al., 2015). Therefore, we need methods that can make data-driven bias-variance trade-offs and empirically learn the optimal policy at the right level of granularity.

Third, to learn the optimal policy, we need to be able to evaluate the performance of any policy offline (without deploying it in the field). This is essential because deploying a policy in the field to estimate its effectiveness is costly in time and money. Moreover, given the size of the policy space, it is simply not feasible to test each policy in the field. Therefore, we need an appropriate policy evaluation metric that can give us a *consistent* estimate of the reward from a given policy if it were to be implemented on the full population.

1.3 Our Approach

We present a framework for personalized policy design and evaluation that overcomes the challenges discussed above. It consists of three components.

First, we define the optimal policy design problem for a general case and show that there are two distinct approaches available to empirically learn the optimal personalized policy:

- *Policy design using outcome estimates:* In this approach, we first learn a flexible model of users' response (or outcome) as a function of her/his pre-treatment demographic variables and the firm's action/treatment. Then, in the second step, we obtain the optimal policy for each user by assigning the action that maximizes the expected outcome for that user.
- *Policy design using CATE estimates:* In this approach, we first learn consistent estimates of CATEs for each pair of treatments/actions that the firm can take. Then, in the second step, we use the estimated treatment effects to derive the optimal policy for each user.

In both approaches, we first learn a function (for outcomes or CATEs) in the first step and then use this function to optimally assign policy in the second step. This circumvents an unstructured search for the optimal policy over the high-dimensional policy space. Both approaches have been used in the recent literature; for example, Rafieian and Yoganarasimhan (2018) develop personalized mobile ad targeting policies using outcome predictions, whereas Hitsch and Misra (2018) derive

CATEs and then use it design targeting policies for promotions. However, we are the first to consider both outcome and CATE estimators and compare their performance in policy design.

In the second component of our framework, we design personalized policies based on five outcome estimators (linear regression, lasso, CART, random forest, and boosted regression trees) and two CATE estimators (causal tree and causal forest). All these methods impose different parametric or semi-parametric assumptions on users' responses or CATEs. As such, while they should all give the same optimal policy when response functions are relatively simple and data are infinite, neither of these conditions are likely to be true in most application settings. So the optimal policies generated by these models in an empirical context are likely to be different. Our goal is to identify the estimators that are ideal for policy design and pin down their source of advantage.

Finally, the third component of our framework consists of offline policy evaluation. For this task, we use the Inverse Propensity Score (IPS) reward estimator, which has been used in reinforcement learning and counterfactual policy evaluation literature in computer science (Horvitz and Thompson, 1952; Dudík et al., 2011). Intuitively, for *any* given policy, the estimator takes all the users who happened to receive the policy-prescribed treatment, and scales them up by their propensity of receiving that treatment. This scaling gives us a pseudo-population that has received the policy being considered. The observed reward of this pseudo-population is therefore a consistent estimate of the reward that we can expect from the policy if we were to implement it. The IPS estimator is valid as long as we have sufficient randomization in the data.

1.4 Application Setting: Large Scale Field Experiment on Free Trials for SaaS Software

We apply our framework to data from a large-scale field experiment on free trials conducted by a leading Software as a Service (SaaS) firm. Over the last few years, software firms have been steadily migrating from the perpetual licensing model to the SaaS model. Global revenues for the SaaS industry now exceed 94 million USD (Gartner, 2019). This shift has led to some fundamental changes in software marketing. In particular, a new type of customer acquisition strategy has gained popularity – *free trial* promotions, where users are given a limited time to try the software for free.

While most SaaS firms offer free trial promotions, there is no consensus on what the *optimal* length of these trials should be. There are pros and cons to both long and short trials, and we observe a substantial variation in the length of free trials offered in the industry. For example, Microsoft 365 offers a 30-day free trial whereas YouTube TV offers a 14-day free trial. Moreover, the optimal length can vary by consumers' attributes. For example, beginners may benefit from longer free trials since it gives them sufficient time to learn the features of the software, while a shorter trial period may be better for students because it prevents them from free-riding.

To understand the effectiveness of different free trial lengths, our data partner conducted a

large-scale field experiment involving over $337,724$ consumers and six geographic markets. The firm exogenously manipulated the length of free trials by assigning new users to one of 7, 14, or 30-day trial length conditions. It then followed these users' subscription and retention decisions for two years. The firm also collected data on users' pre-treatment characteristics (e.g., skill level and job) and post-treatment usage behavior to examine the heterogeneity in users' responses to trial lengths.

1.5 Findings and Contribution

We now discuss the main findings from the analysis.

At the outset, we find that the firm can do significantly better by simply assigning the 7-day trial to all consumers (best uniform policy). In the test data, this leads to a 5.59% gain over the baseline of 30 days for all policy. In contrast, there is no significant effect of the 14-day treatment.

Next, we design seven personalized targeting policies based on five outcome and two CATE estimators. We find that the treatment allocations in these policies vary a good amount, which suggests that the exact estimator used to design policies can have a substantial impact on outcomes. The policy based on lasso performs the best, with a 6.81% gain over baseline of 30 days for all, followed by the policy based on XGBoost (6.17% improvement). However, policies based on other outcome estimation methods such as regression and random forest perform poorly, even worse than the best uniform policy (7 days for all). Both causal tree and causal forest also perform poorly when it comes to personalized policy design. Causal tree is unable to personalize the policy at all. Causal forest personalizes policy by a small amount, but the gains from doing so are marginal compared to simply implementing the best uniform policy (7 days for all). This is an important finding since these methods are gaining traction in the marketing literature for CATE estimation (e.g., Guo et al. (2017)). However, both in marketing research and practice, the end goal is often policy design and not CATE estimation (e.g., who should be targeted with advertising and who should not be). In such cases, relying on these CATE estimators can be sub-optimal.

We then investigate the source of the difference in the performance of these policies. We find that CATE estimates based on CART, causal tree, and casual forest show very little heterogeneity. So they are unable to personalize the policy sufficiently. In contrast, CATE estimates based on linear regression and random forest seem to infer excessive spurious heterogeneity, which is problematic. The CATEs based on lasso and XGBoost lie somewhere in between these two groups. So policies based on these methods are able to personalize the treatment sufficiently while avoiding overfitting.

We also examine why some users respond well to longer trials while others respond better to shorter trials by segmenting consumers into three groups based on their optimal trial length. We find that consumers for whom the 30-day trial is optimal are more likely to be *legacy users* (e.g.,

are more experienced and less likely to be students or hobbyists), consumers for whom the 14-day trial is optimal are more likely to be *learners* (e.g., have higher usage and subscription rate), and consumers for whom the 7-day trial is optimal are more likely to be *free-riders* (e.g., more likely to be beginners and students and have the lowest subscription rate).

Finally, we examine how personalized targeting policies designed to maximize short-run conversions (e.g., subscriptions) perform on long-run outcomes such as consumer loyalty and profitability. This is important because these two objectives can lead to starkly different policies. For example, a policy designed to maximize subscriptions may give a user the treatment that increases her probability of subscription, even if that subscription is to a lower-end product, whereas a policy that focuses on revenues may assign a different treatment to the same user. Therefore, we evaluate the expected length of subscription and revenues for our best-personalized policy and establish that it performs well on these metrics too. However, the magnitudes of gains on these outcomes are different since the treatment can affect long-run outcomes through multiple channels (in addition to conversion).

Our research makes several contributions to the literature. First, from a methodological perspective, we present a framework to design and evaluate personalized targeting policies based on both outcome and CATE estimates. We identify lasso and boosted regression trees as the two ideal outcome estimators for policy design. We also show that CATE estimators such as casual tree and causal forest are unable to effectively personalize policy. Second, from a substantive perspective, we show that SaaS firms can benefit from personalizing the trial length based on each customer's demographics and skill level. Finally, from a managerial perspective, our findings are of relevance to SaaS firms and managers who want to optimize their free trial promotions.

The rest of the paper is organized as follows. In §2 we discuss the related literature. Next, in §3, §4, and §5 we describe the three components of our frameworks – personalized policy design, outcome and CATE estimation, and policy evaluation. In §6, we describe the application setting and data. We present our results in §7 and conclude in §8.

2 Related Literature

First, our paper relates to the theoretical and empirical research on policy design and evaluation that spans marketing, economics, and computer science. In an early theoretical paper, Manski (2004) presents a method that finds the optimal treatment for each observation by maximizing a regret function. More recent papers in this area include Kitagawa and Tetenov (2018) and Athey and Wager (2017). However, the methods in these papers are constrained by the size of the policy space (which in our case is $3^{987,840}$). Therefore, we first estimate outcomes or CATEs, and then use these estimates to develop and evaluate personalized policies.

Recent empirical papers in this area include Lefortier et al. (2016), Swaminathan et al. (2017),

and Simester et al. (2019a). In a particularly relevant paper, Simester et al. (2019b) investigates how data from field experiments can be used to design targeting policies for new customers or new regimes. They find that model-based methods (e.g., lasso) offer the best performance, though this advantage vanishes if the setting and/or consumers change significantly. Like them, we also seek to identify methods that are appropriate for policy design. However, our goal is to compare the ability of different outcome estimation methods with the newly proposed CATE estimation methods (e.g., causal forest) in designing policy within the same regime. Further, we seek to understand and explain why these methods differ in their ability to personalize policy.

Our paper also adds to the small, but growing literature on applications of machine-learning based CATE estimators in marketing. Ascarza (2018) estimates the heterogeneous effect of retention interventions and shows that customers with the highest churn risk are not necessarily those who benefit most from retention interventions. Guo et al. (2017) investigates the heterogeneous effect of information transparency on payments between pharmaceutical firms and physicians using causal forests. Our paper contributes to this literature by showing that tree-based CATE estimators suffer from drawbacks that make them less than ideal for designing personalized targeting policies.

Substantively, our paper adds to three important areas in digital marketing – personalization of marketing actions, field experiments, and free trials. We discuss these in detail below.

A recent stream of marketing papers has focused on the question of "how to personalize marketing actions using machine learning methods": ranking of search engine results (Yoganarasimhan, 2017), mobile ads (Rafieian and Yoganarasimhan, 2018), sequence of ads shown in a mobile in-app setting (Rafieian, 2019a,b), promotions (Hitsch and Misra, 2018), and prices (Dubé and Misra, 2019). We contribute to this research by presenting a comprehensive study of how to design and evaluate personalized targeting policies, and identifying the methods that are ideal for this task.

Our paper also contributes to the body of work on field experiments in digital marketing. A series of papers in this area have looked into the question of how to design and run field experiments such that managers can make correct inferences to overcome selection issues (Johnson et al., 2017), and/or optimize revenues using multi-arm bandits or incorporating the cost of testing a treatment (Schwartz et al., 2017; Feit and Berman, 2019). While these papers are concerned with the design of experiments, our goal is to use the post-experimental data to optimize targeting policies.

Finally, our work relates to research that examines the effectiveness of free trials. Analytical papers in this area have explored mechanisms by which free trials can be effective. They have proposed a multitude of (often conflicting) explanations such as switching costs, software complexity, network effects, quality signaling, and consumer learning (Cheng and Liu, 2012; Dey et al., 2013; Wang and Özkan-Seely, 2018). However, there are only two empirical papers that focus on free

trials. Foubert and Gijsbrechts (2016) build a model of consumer learning and show that while free trials can enhance adoption, ill-timed free trials can also suppress adoption. Using a bayesian learning approach, Sunada (2018) compares the profitability of different free trial configurations. However, neither of these papers explore the question of how to optimize the length of free trials because they have no variation in the length of the free trials offered in their data. Moreover, they both rely on observational data, which limits their ability to draw inferences and form policy. In contrast, we use data from a large-scale field experiment with exogenous variation in the length of free trials to identify the optimal trial length for each user.

3 Personalized Policy Design

We now present the first component of our three-pronged framework – personalized policy design.

3.1 Problem Definition

Consider a setting with $i \in \{1, \ldots, N\}$ independent and identically distributed users, where each user is characterized by a covariate vector $X_i \in X$ of dimension D. Let $W_i \in \mathcal{W}$ denote the treatment or intervention that i receives. $\mathcal{W} = \{0, \ldots, W-1\}$ refers to the set of treatments and the total number of treatments is W. We restrict ourselves to settings where treatments are assigned based on a fully randomized experiment or those where the treatment propensities are a function of observable variables (e.g., an observational study). We use $e(W_i = w, X_i)$ to denote the probability that user i with pre-treatment attributes X_i is assigned treatment w. Finally, let $Y(X_i, W_i)$ denote the outcome for user i when she is allocated treatment W_i.

We begin with a formal definition of a personalized treatment assignment policy.

Definition 1. Personalized treatment assignment policy. π is defined as a mapping between users and treatments such that each user is allocated one treatment, $\pi : X \to \mathcal{W}$.

$\pi(X_i) = W_i^{\pi}$ implies that the policy π assigns treatment W_i^{π} to user i with pre-treatment attributes X_i. Next, we define the objective function that a firm or policy-maker wants to maximize. Let Y denote an outcome of interest. Then:

Definition 2. Reward function. The firm's objective is to choose a policy π such that it maximizes the expectation of outcomes, $\frac{1}{N}\,\mathbb{E}\left[\sum_{i=1}^{N} Y(X_i, W_i^{\pi})\right]$. Thus, for policy π and outcome of interest Y, we can write our reward function as:

$$R(\pi, Y) = \frac{1}{N} \sum_{i=1}^{N} \mathbb{E}\left[Y(X_i, \pi(X_i))\right]. \tag{1}$$

We can now define the optimal personalized policy as follows:

Definition 3. Optimal personalized policy. Given a reward function $R(\pi, Y)$, the optimal personalized policy is given by:

$$\pi^* = \arg\max_{\pi \in \Pi} \left[R(\pi, Y) \right], \tag{2}$$

where Π is the set of all possible policies.

The problem of finding the optimal personalized policy is equivalent to one of finding the policy π^* that maximizes the reward function $R(\pi, Y)$. As discussed in §1, this is a non-trivial problem since the cardinality of the policy space can be quite large. The total number of possible policies is $W^{\prod_{d=1}^{D} c_d}$, when we have D pre-treatment variables and the d-th variable can take c_d different values. Finally, note that the optimal policy can vary based on the outcome of interest.

3.2 Solution Concept

We now discuss the approaches available to find the optimal policy π^*. However, we first need to make a few assumptions on the data generating process for these approaches to be valid.

3.2.1 Assumptions

Assumption 1 (*Unconfoundedness*)**.** The treatment assignment is (conditionally) independent of the potential outcome: $W_i \perp (Y_i(W_i = 1), \ldots, Y_i(W_i = W) \mid X_i)$.

The unconfoundedness assumption holds when we have a fully randomized experiment or when treatment assignment propensities are only function of observables.

Assumption 2 (*Stable Unit Treatment Value Assumption*)**.** SUTVA implies that units do not interfere with each other. It requires the response of a user to only depend on the treatment to which she was assigned, and not the treatments of others around her.

SUTVA is a reasonable assumption in most marketing settings, and it is violated only when there are WOM or network effects.

Assumption 3 (*Positivity*)**.** All users have a positive probability of receiving all values of the treatment variable: $e(W_i = w, X_i) > 0$ for all $w \in \mathcal{W}$ and X_i.

This assumption is satisfied in randomized experiments and in observational studies where there is residual randomness in treatment assignment, conditional on observables. This assumption is necessary for both being able to design policy for the full distribution of Xs over all Ws, as well as for empirically evaluating the performance of a policy π based on the data we have. Intuitively, if there are some combinations of covariates and treatments that have a zero probability of occurring

in our data, then we cannot make inferences in this space nor can we evaluate how a counterfactual policy π would perform in this space.

3.2.2 Optimal Policy Design

There are two distinct approaches to designing an optimal policy. We discuss both below.

- **Policy design using outcome estimates:** In this approach, we first obtain consistent estimates of outcomes, $Y(X_i, W_i)$, for all combinations of w and x. We can write a statistical model to estimate outcomes as follows:

$$f(x, w) = \mathbb{E}[Y|X_i = x, W_i = w] \tag{3}$$

  As long as Assumptions 1 through 3 are satisfied, any flexible model $f(x, w)$ will give us consistent estimates of the outcome y for any combination of covariates and treatments.
  Once we have a consistent estimate of the expected outcome, $\hat{y}(x = X_i, w)$, for all possible treatments for a given covariate vector X_i, we can obtain the optimal personalized policy as:

$$\pi^*(X_i) = w^*, \quad \text{where} \quad w^* = \underset{w \in \mathcal{W}}{\arg\max}\, \hat{f}(x = X_i, w) \tag{4}$$

- **Policy design using CATE estimates:** The second approach to designing a personalized policy is to first obtain consistent Conditional Average Treatment Effects for each pair of treatments, and then use them to assign treatments. This method also follows a two-step procedure.
  - In the first step, we can obtain consistent estimates of CATE, $\tau_{w_j,w_{j'}}(x)$, for each pair of treatments $w_j, w_{j'} \in \{0, \ldots, W-1\}$. Under Assumptions 1 through 3, $\tau_{w_j,w_{j'}}(x)$, can be defined as (Rubin, 1974; Imbens and Rubin, 2015):

$$\tau_{w_j,w_{j'}}(x) = \mathbb{E}\Big[Y(X_i, W_i = w_j) - Y(X_i, W_i = w_{j'}) \mid X_i = x\Big]. \tag{5}$$

    So if we have W treatments, we have to build $\frac{W(W-1)}{2}$ CATE models, where each CATE model gives us an estimate of the pairwise treatment effects for a given $w_j, w_{j'}$, $\tau_{w_j,w_{j'}}(x)$.
  - In the second step, we use the estimated treatment effects to derive the optimal policy as:

$$\pi^*(X_i) = w_j \quad \text{if and only if} \quad \forall\ j' \neq j \quad \tau_{w_j,w_{j'}}(x = X_i) \geq 0 \tag{6}$$

    In practice, we can have situations where three or more pairs of the estimated treatments form a loop, i.e., $\hat{\tau}_{w_j,w_{j'}}(x = X_i) > \hat{\tau}_{w'_j,w_{j''}}(x = X_i) > \hat{\tau}_{w''_j,w_j}(x = X_i)$. This usually happens if

the CATE estimator does not have sufficient data to identify the best treatment. For these observations, we can simply assign the best average treatment (across all observations) as the policy-prescribed treatment.[1]

Note that both the approaches outlined above are consistent and correct. With an infinitely large data and sufficiently flexible models (of outcomes and treatment effects), both methods should give the exact same optimal policy. However, in practice, data are finite and the models used for estimation of outcomes and CATE in high-dimensions make different types of bias-variance trade-offs. Therefore, the optimal policy in an empirical setting can vary significantly based on the exact model used to estimate the outcome or CATEs. Indeed, one of the main objectives of this work is to examine the efficacy of different types of estimators for designing policy.

4 Estimators Used for Policy Design

We now discuss the second component of our framework – estimators used for policy design. In §4.1, we discuss outcome estimation methods and then in §4.2, we discuss CATE estimators.

4.1 Outcome Estimation Methods

Outcome estimators seek to learn a model $f(x, w) = \mathbb{E}[Y|X_i = x, W_i = w]$. We consider the five commonly used outcome estimators: (1) linear regression, (2) lasso, and (3) CART, (4) random forest, and (5) boosted regression trees. We focus on these because of a few key reasons. First, linear regression is the simplest and most commonly used method to model any outcome of interest. Second, lasso was one of the earliest methods proposed to improve the predictions from a linear regression by reducing the out-of-sample variance using variable selection, and hence is worth exploring. Third, CART is a semi-parametric way to model outcomes in complex settings by partitioning the covariate space into areas with the highest within-group similarity in outcomes. Finally, both random forest and boosted regression trees are improvements of CART and have been shown to offer much higher predictive ability than simpler models such as CART, regression, or lasso.

4.1.1 Regression-based models

We start by discussing the two regression-based models: (1) linear regression and (2) lasso.

We can learn a regression model with first-order interaction effects to predict an individual i's subscription as a function of her pre-treatment variables, treatment variable, and the interaction of

[1]There can be multiple optimal policies because two or more top treatments can have the same effect on the outcome. That is, for some combinations of X_i and $j' \neq j$, we can have: $\tau_{w_j,w_{j'}}(x = X_i) = 0$. However, notice that all the solutions give the same reward, i.e., the optimal reward is unique.

both. Specifically, we can train the following model:

$$Y_i = X_i\beta_1 + W_i\beta_2 + X_iW_i\beta_3 + \epsilon_i. \tag{7}$$

β_1 is a vector that captures the effect of the pre-treatment demographic variables (X_i) on the subscription outcome. β_2 is a vector that captures the main effect of the different treatments on Y. Finally, the vector β_3 captures the interaction effect of treatment and pre-treatment demographic variables. This interaction term is important because it helps us in personalizing the policy by capturing how the effectiveness of different treatments varies across individuals (as a function of their pre-treatment attributes).

However, in a high dimensional covariate space, linear regressions with first-order interaction effects will have a large number of explanatory variables. This usually leads to poor out-of-sample fit, that is, such regressions tend to have low bias, but high variance – they tend to overfit on the training sample but perform poorly in new samples, especially if the data generating process is noisy. Since our goal is to design optimal policies for counterfactual situations, the out-of-sample fit is the only metric of importance.[2]

Lasso (least absolute shrinkage and selection operator) addresses this problem by learning a simpler model that uses fewer variables (Tibshirani, 1996). Practically, lasso estimates a linear regression that minimizes the MSE with an additional term to penalize model complexity (Friedman et al., 2010). Formally, we estimate the following model:

$$(\hat{\beta}_1, \hat{\beta}_2, \hat{\beta}_3) = \arg\min \sum_{i=1}^{n} (Y_i - X_i\beta_1 - W_i\beta_2 - X_iW_i\beta_3)^2 + \lambda(||\beta_1||_1 + ||\beta_2||_1 + ||\beta_3||_1),$$

where $||\beta_i||_1$ is the L1 norm of the vector β_i and is equal to the sum of the absolute value of the elements of vector β_i. The nature of this penalty forces some of the coefficients to zero and simplifies the estimated model. This shrinkage property of lasso increases its out-of-sample predictive accuracy since the final model is estimated on a subset of explanatory variables that show the strongest ability to predict the outcome. Intuitively, if there are multiple weak (and correlated) predictors, lasso will pick a subset of them and force the coefficients of others to zero.

As in most machine learning models, model-selection in lasso is data-driven, i.e., λ is a hyper-parameter that is learned from the data (and not assumed). To find the optimal hyper-parameter λ^*, we use a five-fold cross-validation procedure.

[2]We can also add higher-order interaction effects into the regression model. However, we refrain from doing so because it will increase model complexity significantly and exacerbate this problem.

Another regression-based model that is worth considering is elastic net. Elastic net penalizes both L1 norm (like lasso) and L2 norm (like ridge regression). However, in our application setting, the cross-validation procedure chooses the L2 norm coefficient to be zero. So elastic net converges to lasso in our setting. So we do not discuss it in detail here.

4.1.2 Tree-based models

Next, we consider CART, random forests, and boosted regression trees, all of which are tree-based models that share a common semi-parametric approach to building outcome prediction models.

We start with an overview of CART, which was proposed by Breiman et al. (1984). CART recursively partitions the covariate space into sub-regions, and the average of Y in a given region ($E(Y)$) is the predicted outcome for all observations in that region. This type of partitioning can be represented by a tree structure, where each leaf of the tree represents an output region. A general tree model can be expressed as:

$$y = f(x, w) = \sum_{m=1}^{M} \rho_m I(x, w \in R_m), \tag{8}$$

where x, w is the set of explanatory variables, R_m is the m^{th} region of the M regions used to partition the space, ρ_m is the predicted value of y in region m.

Trees are trained by specifying a cost function that is minimized at each step of the tree-growing process using a greedy algorithm. At each stage of this process, the algorithm searches over each value of every explanatory variable in each parent node P to find the next split that minimizes the Sum of Squared Errors (SSE), which is defined as:

$$\text{SSE} = \left(\sum_{i \in L} (Y_i - \rho_L)^2 + \sum_{i \in R} (Y_i - \rho_R)^2 \right), \tag{9}$$

where L and R refer to the left and right splits.[3]

CART, like all machine learning models, can overfit. To control model-complexity and prevent overfitting, we usually add a regularizer to the SSE and specify the cost function as SSE $+ \zeta|T|$. T is the number of terminal nodes in the tree and ζ is the hyper-parameter that captures the weight that we put on tree complexity. Note that this regularizer penalizes complex trees with many splits. We learn the optimal ζ^* using five-fold cross-validation.

CART is popular because it has some good properties. It can accept both continuous and discrete explanatory variables, is not sensitive to the scale of the variables, and allows any number

[3]The notion of left and right does not have any specific meaning and is only used to label the two resulting partitions.

of interactions between features (Murphy, 2012). Thus, a key advantage of CART over regressions is the ability to capture rich nonlinear patters. Further, CART can do automatic variable selection, i.e., it uses only those variables that provide better accuracy in the prediction task for splitting.

Nevertheless, CART often has poor predictive accuracy because it is – (1) discontinuous in nature, (2) is trained using greedy algorithms and thus can converge to a local maximum, and (3) is sensitive to outliers. Therefore, even with regularization, decision trees tend to overfit on the training data and under-perform on out-of-sample data.

We can address these problems using two different techniques – (1) Bagging and (2) Boosting. We discuss both of these and the resulting estimators below.

In general, deep trees have high in-sample fit (low bias), but high out-of-sample variance because of overfitting. However, we can improve the variance problem by bagging or averaging deep trees using boot-strapping. Ho (1995) formalized this idea and proposed random forests. Random forest usually consists of hundreds or thousands of trees, each of which is trained on a random sub-sample of columns and rows. Each tree is thus different from other trees and the average of these random trees is a better predictor than one single tree trained on the full data.

Another approach is to start with shallow trees. Shallow trees have poor in-sample fit (high bias), but low out-of-sample variance. Additively, adding a series of weak trees that minimize the residual error at each step by a process known as boosting can improve the bias problem, while retaining the low variance. This gives us boosted regression trees. Conceptually, boosting can be thought of as performing gradient descent in function space using shallow trees as the underlying weak learners (Breiman, 1998; Friedman, 2001).[4] In this paper, we use XGBoost, a version of boosted trees proposed by Chen and Guestrin (2016) because it is superior to earlier implementations both in terms of accuracy and scalability.

Please see Appendix §A for the set of hyper-parameters that need to be tuned in random forests and XGBoost.

4.2 CATE Estimation Methods

CATE estimators are based on the potential outcomes framework and estimate the treatment effect for any pair of treatments $(w_j, w_{j'})$ at each point x as shown in Equation (5). However, this equation is not very useful in practice when the covariate space is high-dimensional and data are finite because we would not have sufficient observations at each X_i to estimate precise treatment effects. Therefore, the general idea behind modern CATE estimators is to pool observations that are close in the covariate space and estimate CATE for sub-populations instead of estimating CATE at each

[4]It should be noted that bagging is not a modeling technique; it is simply a variance reduction technique. Boosting, however, is a method to infer the underlying model $y = f(x, w)$. Thus, they are conceptually completely different.

point in the covariate space. That is, we can modify Equation (5) as:

$$\tau_{w_j,w_{j'}}(x) = \frac{\sum_{X_i \in l(x), W_i = w_j} Y_i}{\sum 1[X_i \in l(x), W_i = w_j]} - \frac{\sum_{X_i \in l(x), W_i = w_{j'}} Y_i}{\sum 1[X_i \in l(x), W_i = w_{j'}]}, \tag{10}$$

where $l(x)$ is the set of covariates that are fairly similar to x. Intuitively, for each point x, we use the observations in $l(x)$ to estimate treatment effects.

The main question in these methods then becomes how to find the optimal $l(x)$ around each x since manual search is infeasible in a high-dimensional covariate space. On the one hand, if $l(x)$ is too small, then we will not have sufficient observations within $l(x)$, which would result in noisy estimates of treatment effects. On the other hand, if $l(x)$ is too large, then we will not capture all the heterogeneity in the treatment effects, which is essential for personalizing policy. Indeed, if $l(x)$ is the entire data, then we simply have one ATE for all users (which will give us one global policy). Thus, finding the optimal $l(x)$ involves the classic bias-variance trade-off.

Starting with Rzepakowski and Jaroszewicz (2012), a growing stream of literature has focused on developing data-driven approaches to finding the optimal $l(x)$ based on ideas from the machine learning literature. Among these methods, the recently developed causal tree and causal forest methods have been shown to have superior performance. So we focus on them.

4.2.1 Causal Tree

Causal tree is a CATE estimator that builds on CART. The key intuition is that we can use recursive partitioning to estimate CATEs (similar to CART for prediction) if we can come up with an objective function that can identify partitions with similar within-partition treatment effect.[5] Athey and Imbens (2016) show that maximizing the variation in the estimated treatment effects achieves this objective, which can be written as:

$$Var[\hat{\tau}(X)] = \frac{1}{N}\sum_{i=1}^{N} \hat{\tau}^2(X_i) - \left(\frac{1}{N}\sum_{i=1}^{N} \hat{\tau}(X_i)\right)^2 \tag{11}$$

Since $\left(\frac{1}{N}\sum_{i=1}^{N} \hat{\tau}(X_i)\right)^2$ remains constant with respect to any possible next split, the objective function can be also described as maximizing the average of the square of estimated treatment effects. In practice, the algorithm chooses the split that maximizes $Var[\hat{\tau}(X)] - \zeta T$, where the

[5]The first example of such a criterion was proposed by Hansotia and Rukstales (2002): for each potential split, the algorithm calculates the difference between the average outcome of the treatment and control in the right (ΔY_r) and left (ΔY_l) branches of the tree. Then, it selects the split with the highest value of $\Delta\Delta Y = |\Delta Y_r - \Delta Y_l|$. Other splitting criteria include maximizing the difference between the class distributions in the treatment and control groups (Rzepakowski and Jaroszewicz, 2012).

second term is added for complexity control and is analogous to the regularization term in CART.

The algorithm consists of two steps. In the first step, it recursively splits the covariate space into partitions. In the second step, it estimates the treatment effect within each partition ($l(x)$) using Equation (10).[6] Intuitively, this algorithm pools observations with similar treatment effects into the same partition because splitting observations that have similar treatment effects does not increase the objective function (i.e., variation in the post-split treatment effect).

4.2.2 Causal Forest

The causal tree algorithm nevertheless suffers from weaknesses that mirror those of CART (discussed in §4.1.2). To resolve these issues, Wager and Athey (2018) proposes causal forest, which builds on random forests (Breiman, 2001). More broadly, Athey et al. (2019) show that the intuition from random forests can be used to flexibly estimate any heterogeneous quantity, $\theta(x)$, from the data, including CATEs. They suggest a Generalized Random Forest (GRF) algorithm that learns problem-specific kernels for estimating any quantity of interest at a given point in the covariate space. For CATE estimation, the method proceeds in two steps.

In the first step, it builds trees whose objective is to increase the variation in the estimated treatment effects. Each tree is built on a random sub-sample of the data and random sub-sample of covariates. At each step of the partitioning process (when building a tree), the algorithm first estimates the treatment effects in each parent leaf P by minimizing the R-learner objective function. This objective function is motivated by Robinson's method (Robinson, 1988; Nie and Wager, 2017) and can be written as follows:

$$\hat{\tau}_P(\cdot) = \arg\min_{\tau} \left[\frac{1}{n_P} \sum_{i=1}^{n_P} \left(\left(Y_i - \hat{m}^{(-i)}(X_i)\right) - (W_i - \hat{e}^{(-i)}(X_i))\hat{\tau}(X_i) \right)^2 + \Lambda_n(\tau(.)) \right], \quad (12)$$

where n_P refers to the number of observations in the parent partition, $\Lambda_n(\tau(.))$ is a regularizer that determines the complexity of the model used to estimate $\tau_P(.)$s, and $\hat{m}^{(-i)}(X_i)$ and $\hat{e}^{(-i)}(X_i)$ are out-of-bag estimates for the outcome and propensity score, respectively. While any method can be used for estimating $\hat{m}^{(-i)}(X_i)$ and $\hat{e}^{(-i)}(X_i)$, the GRF implementation uses random forests to estimate these values.

Then, it chooses the next split such that it maximizes the following objective function:

$$\frac{n_L \cdot n_R}{(n_P)^2} \left(\hat{\tau}_L - \hat{\tau}_R\right)^2, \quad (13)$$

where n_L and n_R refer to the number of observations in the post-split left and right partitions,

[6] For each x, $l(x)$ is the set of all the other data points that fall within the same partition as x.

respectively. However, instead of calculating the exact values of $\hat{\tau}_L$ and $\hat{\tau}_R$ for each possible split (and for each possible parent), the causal forest algorithm uses a gradient-based approximation of $\hat{\tau}$ for each child node to improve compute speed; see Athey et al. (2019) for details. Thus, at the end of the first step, the method calculates weights, $\alpha_i(x)$, that denote the frequency with which the i-th training sample falls into the same leaf as x in the first step. Formally, $\alpha_i(x)$ is given by $\alpha_i(x) = \frac{1}{B}\sum_{b=1}^{B} \alpha_{bi}(x)$, where $\alpha_{bi}(x) = \frac{\mathbf{1}(X_i \in l_b(x))}{|l_b(x)|}$, B is the total number of trees built in the first step, and $l_b(x)$ is the partition that x belongs to in the b-th tree.

In the second step, the algorithm uses the idea of a weighted kernel regression to calculate the CATE at each point x using weights $\alpha_i(x)$ as follows:

$$\hat{\tau}(x) = \frac{\sum_{i=l}^{N} \alpha_i(x) \left(Y_i - \hat{m}^{(-i)}(X_i)\right) \left(W_i - \hat{e}^{(-i)}(X_i)\right)}{\sum_{i=l}^{N} \alpha_i(x) \left(W_i - \hat{e}^{(-i)}(X_i)\right)^2}, \tag{14}$$

As with all supervised learning models, we need to do hyper-parameter optimization to prevent causal forest from overfitting. We refer readers to Appendix §A for details on this.[7]

5 Policy Evaluation

We now discuss the final component of our three-pronged framework – offline policy evaluation.

To identify the optimal policy among the set of all possible policies (Π), it is important to be able to consistently evaluate the gain from implementing any policy π *without* actually deploying it. As discussed in §1, this is important because deploying a policy in the field is costly in time and money. Moreover, when the policy space is large, it is simply not feasible to test each policy in the field. So we turn to off-policy estimators proposed in the reinforcement learning literature (Sutton and Barto, 2018), where the goal is to estimate the performance of a given policy π using data generated by an experiment or another policy.

In any setting where Assumption 3 is satisfied, there is a non-zero probability that a user i will be assigned to the treatment prescribed by a given policy π. Thus, as shown in Figure 1, we have two types of observations in the data:

[7] Athey and Imbens (2016) propose an additional approach to avoid overfitting in causal tree and causal forest – honest splitting. The main idea behind honesty is to split the data into two parts and use one part for growing each tree (i.e., generating the partitions) and the other part for estimating the treatment effects given a partition. Since the two data-sets are independent of one another, there is a lower likelihood of overfitting. However, honest-splitting comes with its own costs – it reduces the amount of data we have for learning in both stages by half. This can be problematic in settings with low signal-to-noise ratio such as ours and adversely affect the performance of models that use honesty. Indeed, we found that models based on honest splitting lead to worse policies compared to models without honest splitting in our setting. So we do not employ honesty in our analysis.

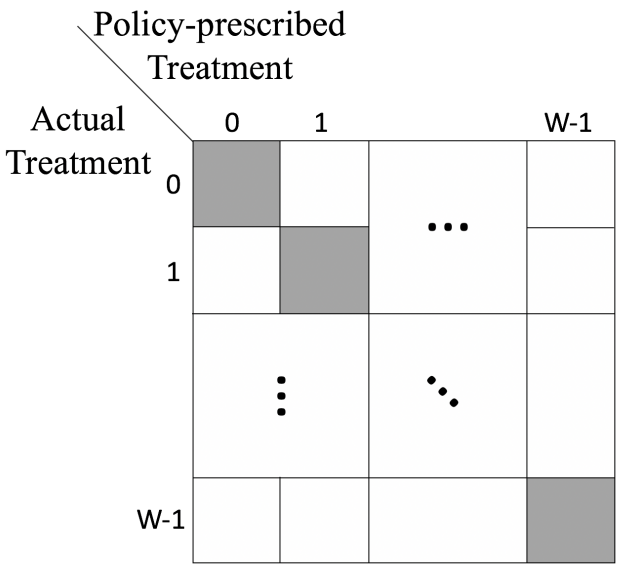


Figure 1: Classification of observations based on the actual treatment they received in the data and the treatment prescribed by the policy. The shaded cells in the diagonal constitute policy congruent observations and the white cells constitute policy in-congruent observations.

- **Policy congruent observations:** Those for whom the observed treatment assignment (W_i) matches the policy-prescribed treatment ($\pi(X_i)$), i.e., $\pi(X_i) = W_i$.
- **Policy in-congruent observations:** Those for whom the observed treatment assignment is different from the policy-prescribed treatment, i.e., $\pi(X_i) \neq W_i$

These two sets of observations form the basis of our evaluation metric: the Inverse Propensity Score (IPS) Estimator for Reward. This estimator has been extensively used in the off-policy evaluation literature (Horvitz and Thompson, 1952; Dudík et al., 2011), and can be defined as:

$$\hat{R}_{IPS}(\pi, Y) = \frac{1}{N} \sum_{i=1}^{N} \frac{1[W_i = \pi(X_i)]Y_i}{e(W_i, X_i)}. \tag{15}$$

The IPS estimator is unbiased as long as Assumption 3 is satisfied. The intuition behind this estimator can be easily understood using Figure 1. Essentially, it takes all the observations in the policy congruent cells and scales them up by their propensity of receiving the treatment assigned to them. This scaling gives us a pseudo-population that received the policy-prescribed treatment. Thus, the average of the outcome for this pseudo-population gives us an estimate of the reward for the true population, if we were to implement the proposed policy in the field.

6 Application: Large Scale Field Experiment on Free Trials Promotions

In this section, we describe our application setting, data, and present some simple descriptive analysis that supports the idea that personalizing free trial lengths can improve firm-level outcomes.

6.1 SaaS Business Model and Free Trial Promotions

One of the major trends in the software industry over the last few years has been the migration of software firms from the perpetual licensing business model to the "Software as a Service" (SaaS) model. In the SaaS model, the software is sold as a service, i.e., consumers can subscribe to the software based on monthly or annual contracts. Global revenues for the SaaS industry now exceed 94 million USD (Gartner, 2019). This shift in the business model has fundamentally changed how software firms market and promote their services. At a high-level, since the software is now a service, sales and trials play a much more important role than traditional advertising and brand building activities. In particular, a new type of customer acquisition strategy that has become popular under the SaaS business model is free trial promotions, where new users are given a limited time to try the software for free.

Free trials can work through three potential mechanisms. First, they give potential consumers a cost-free opportunity to evaluate the quality and usability of the software as well as assess its fit for their individual needs. Second, it can mitigate the learning costs associated with getting familiar with the software. Finally, free trials can function as signals of product quality, i.e., only high-quality firms that are confident that consumers will find their product to be valuable upon examination are more likely to offer free trials (Wang and Özkan-Seely, 2018; Milgrom and Roberts, 1986).

While free trials are quite popular in the SaaS industry, there is no consensus on the optimal length of the trial period. There is substantial variation in the lengths of free trials in the industry, with lengths ranging from anywhere between ranging from one week to six months. There are pros and cons associated with both long and short trials. A short trial period lowers the firm's acquisition costs since the length of time that the company supports a non-paying user is short. It is also less likely to suffer free-riding concerns. For example, if consumers need the software for a specific/defined task which will take only a few days (e.g., set up a website or edit a document/image), a lengthy free trial lets them use the product without paying for it whereas a short trial period will force the customer to purchase it for the period of use. On the other hand, a long trial period can create stickiness/engagement and increases switching-back costs. For example, if a consumer has already created or stored content in the software, it is costly to move it to the local disk or alternative software. Longer trial periods can also enhance consumer learning, i.e., give consumers more time to learn how to use the software and the different product features. Finally, longer trials reduce the

pressure on users to subscribe, which can lead to more positive feelings towards the brand.

While the above arguments make a global case for shorter/longer trial periods, there is also the issue of heterogeneity in consumers. Will the demographics or skill-level of consumers affect whether they will respond better to longer/shorter trial periods? For instance, should we give longer free trials to beginners because they need more time to learn the software? Should students get shorter free trials since they are more likely to free-ride? If there is significant heterogeneity in consumers' response to the length of free trials, then SaaS firms may benefit from implementing a personalized free trial policy. Under such a policy, each consumer can be assigned the trial-length (based on her/his demographic variables) that maximizes her/his responsiveness. In our application, we examine the gains from adopting personalized free trial policies.

6.2 Setting

Our data come from a major SaaS firm that sells a suite of software products. The suite includes a set of related software (e.g., Excel, Word, PowerPoint in Microsoft's MS-Office). The firm is the leading player in its category, with close to monopolistic market power. Users can either subscribe to single-product plans that allow them access to one software product or to bundled plans that allow them to use several products at the same time. Bundles are designed to target specific segments of consumers and consist of a set of complementary products. The prices of the plans vary significantly and depend on the bundle, the type of subscription (regular or educational), and the length of commitment (monthly or annual). Standard subscriptions run from $\$30$ to $\$140$ per month depending on the products in the bundle and come with a monthly renewal option. (To preserve the firm's anonymity, we have multiplied all the dollar values in the paper by a constant number.) If the user is willing to commit to an annual subscription, they receive over 30% discount in price. However, users in annual contracts have to pay a sizable penalty to unsubscribe before the end of their commitment. The firm also offers educational licenses at a discounted rate to students and educational institutions, and these constitute 20.8% of the subscriptions in our data.

6.3 Field Experiment

At the time of this study, the firm used to give users a 30-day free trial for each of its software products, during which they had unlimited access to the product.[8] In order to access the product after the trial period, users need a subscription to a plan or bundle that includes that product.

To evaluate the effectiveness of different trial lengths, the firm conducted a large-scale field experiment that ran from December $1^{st}2015$ to January $6^{th}2016$ and spanned six major geographic

[8]This free trial is at the software product level, i.e., users start a separate trial for each software product, and their trial for a given product expires 30 days from the point at which they started the free trial for it.

	7 Days Trial	14 Days Trial	30 Days Trial	Total
Number of observations (N)	51,040	51,017	235,667	337,724
Percent of total observations	15.11	15.11	69.78	100
Number of subscriptions	7,835	7,635	34,564	50,034
Percent of total subscriptions	15.66	15.26	69.08	100
Subscription rate within group	0.1536	0.1496	0.1467	0.1481

Table 1: Summary statistics of treatment assignment and subscription rates.

markets – Australia and New Zealand, France, Germany, Japan, United Kingdom, and United States of America. During the experiment period, users who started a free trial for any of the firm's four most popular products were randomly assigned to one of 7, 14 or 30 days free trial length buckets. The assignment was at user level, i.e., once a user was assigned to a treatment (trial length), her/his trial length for the other three popular products was also set at the same length. The length of the free trial for other products during this period remained unchanged at 30 days. The summary statistics for the treatment assignment and subscriptions are shown in Table 1.

The experiment was carefully designed and implemented to rule out the possibility of self-selection into treatments, a common problem in field experiments. In our setting, if users can see which treatment (or free trial length) they are assigned to prior to starting their trial, then users who find their treatment undesirable may choose to not start the trial. In that case, the observed sample of users in each treatment condition would no longer be random, and this in turn would bias the estimated treatment effects. Moreover, since the experimenter cannot obtain data on those who choose to not to start their free trials, there is no way to address this problem econometrically.

To avoid these types of self-selection problems, the firm designed the experiment so that users were informed of their trial-length only after starting their trial. In order to try a software product, users had to take the following steps: (1) sign up with the firm by creating an ID, (2) download an app manager that manages the download and installation of all the firm's products, and (3) click on an embedded *start trial* button to start the trial for a given product. Only at this point in time, they are shown the length of their free trial as the time left before their trial expires (e.g., "Your free trial expires in 7 days"). While users can simply quit or choose to not use the product at this point, their identities and actions are nevertheless captured in our data and incorporated in our analysis.

6.4 Data

We have data on 337,724 users, who started a free trial for at least one of the four major products included in the field experiment. For each user i, we observe the following information – (1) Treatment assignment (W_i), (2) Pre-treatment demographic data (X_i), and (3) Post-treatment

Variable	Number of sub-categories	Share of top sub-categories			
		1^{st}	2^{nd}	3^{rd}	4^{th}
Geographic region	6	55.02%	13.66%	9.12%	8.83%
Operating system	8	28.97%	21.40%	14.03%	13.98%
Signup channel	42	81.56%	8.14%	3.47%	0.81%
Job	14	28.20%	21.90%	20.34%	8.46%
Skill	5	69.05%	12.75%	10.77%	7.38%
Business segment	7	35.41%	32.74%	18.40%	7.81%

Table 2: Summary statistics for the pre-treatment categorical variables.

behavioral data (Z_i). The treatment assignment variable denotes the trial length that the user was assigned to – 7, 14, or 30 days. The variables under the latter two are described in detail below.

6.4.1 Pre-treatment demographic data

1. Geographic region: The geographic region/country that the user belongs to (one of the six described in §6.2). It is automatically inferred from the user's IP address.

2. Operating system: The OS installed on the user's computer. It can take eight possible values, e.g., Windows 7, Mac OS Yosemite. It is inferred by the firm based on the compatibility of the products downloaded with the user's OS.

3. Sign-up channel: The channel through which users came to sign-up for the free trial. In total, there are 42 possible sign-up channels, e.g., from the legacy version of the software, from the firm's website, through third-parties, and so on.

4. Skill: A self-reported measure of the user's fluency in using the firm's software suite. This can take four possible values – beginner, intermediate, experienced, and mixed.

5. Job: The user's job-title (self-reported). The firm gives users 13 job-titles to pick from, e.g., student, business professional, hobbyist.

6. Business segment: The self-reported business segment that the user belongs to. Users can choose from six options here, e.g., educational institution, individual, enterprise.

Note that the last three variables are self-reported though not open-ended, i.e., the firm gives users a list and requests them to pick one option from that list. However, users may choose not to report these values, in which case, the missing values are recorded as "unknown". We treat this as an additional category for each of these three variables in our analysis.[9] The list of all the six

[9]Only a small fraction of people chooses to not report these data. For example, the percentage of users with "unknown" Skill and Job is 7.4% and 21.9%, respectively.

pre-treatment variables and their summary statistics are shown in Table 2.

6.4.2 Post-treatment behavioral data

For all the users in our data, we observe all product download and usage data for the duration of their trial period. Further, we observe their subscription and renewal decisions for approximately 24 months (from December 2015 till November 2017). Using these data, we extract the following post-treatment behavioral information:

1. Subscription information: We have data on whether a user subscribes or not, and the date and type of subscription (product or bundle of products) if she does subscribe.

2. Subscription length: Number of months that the user is a subscriber of one or more products/bundles during the 24-month observation period. If a user does not subscribe to any of the firm's products during the observation period, then this number is zero by default. If a user unsubscribes for a period of time and then comes back, her subscription length is the total number of months that she was a paying customer of the firm.[10]

3. Revenue: The total revenue (in scaled dollars) generated by the user over the 2-year observation period. This is a function of the user's subscription date, the products and/or bundles that she subscribes to, and her subscription length.

4. Products downloaded: The date and time-stamp of each product downloaded by the user.

5. Usage information: Each product in the software suite has thousands of functionalities. Functionalities can be thought of as micro-tasks and are defined at the click and key-stroke level; e.g., save a file, click undo, and create a table. The firm captures all the usage data for users during their free trial and stores two key variables associated with usage:

   (a) Total usage: Total count of the functionalities used by the user during her trial period.

   (b) Distinct usage: Number of unique functionalities used by the user during her trial period.

The summary statistics of these post-treatment variables are presented in Table 3. Both subscription length and revenue are shown for: (a) all users and (b) the subset of users who subscribed. There are a couple of points to note regarding the data on subscription length and revenues:

- The minimum subscription length observed in the data for subscribers is zero (for 58 users). These are users who immediately (within one month) unsubscribed after subscribing, in which case the firm returns their money and records their subscription length and revenue as zero.

[10] If a user subscribes to two or more plans, we aggregate the length of subscription all plans and report the total. So the subscription length can be greater than 24 months for such users.

Variable	Mean	Standard Deviation	Min	25%	50%	75%	Max	Number of Observations
Subscription	0.148	0.355	0	0	0	0	1	337,724
Subscription length (all)	2.21	6.35	0	0	0	0	108	337,724
Subscription length (subscribers)	16.02	8.43	0	10	17	22	108	50,034
Revenue (all)	79.13	285	0	0	0	0	20,208	334,223
Revenue (subscribers)	568	552	0	242	420	666	20,208	46,533
No. of products downloaded	1.15	0.40	1	1	1	1	4	337,724
Total usage	2,226	8,071	5	133	454	1,556	543,315	302,646
Distinct usage	102.41	96.61	4	48	75	118	4,033	302,640

Table 3: Summary statistics of post-treatment variables.

Trial Length	Training Dataset	Test Dataset	Total
7 days	35,743	15,274	51,017
14 days	35,901	15,139	51,040
30 days	165,056	70,611	235,667
Total	236,700	101,024	337,724

Table 4: The number of observations for each trial length in each dataset.

- We do not have access to the revenue data for team subscriptions and government subscriptions (which constitute a total of 3501 subscriptions). Hence, the number of observations used to calculate the summary statistics for revenue for subscribers is lower.

The usage data are also missing (at random) for a subset of users and we report the summary statistics for non-missing observations.

6.5 Training and Test Datasets

To implement and test any policy that we design, we first need to partition the data into two independent samples – training data and test data. These are defined as follows:

- **Training Data:** This is the data that is used for both learning the model parameters as well as model selection (or hyper-parameter optimization through cross-validation).
- **Test Data:** This is a hold-out data on which we can evaluate the performance of the policies designed based on the models built on training data.

We use 70% of the data for training (and validation) and 30% for test. See Table 4 for a detailed breakdown of how the data are split across the two data-sets.

It is important to ensure that the joint distribution of the variables is similar in the two data-sets. This is not an issue if the amount of data is very large or the signal-to-noise ratio is low. In our case, neither of these are true; so we take care to ensure that this is indeed the case for the training and test samples. However, there remain some minor differences due to the randomness in splitting.

Data	Treatment	Subscription rate difference	t-statistics	Gain over baseline
Training data	7 days	0.0064	3.07	4.33%
	14 days	0.0021	1.03	1.45%
Test data	7 days	0.0082	2.58	5.59%
	14 days	0.0048	1.51	3.28%

Table 5: Average effect of the 7- and 14-day treatments on subscription; compared to the control condition of 30-day free-trial. Baseline subscription rate (for 30-day case): 14.68 in training data and 14.63 in test data.

6.6 Descriptive Analysis

We now present some model-free evidence on the effect of trial length on subscriptions and the heterogeneity in users' responsiveness (on subscription) to different trial lengths.

6.6.1 Average Treatment Effect of Trial Lengths

In a fully randomized experiment (such as ours), the average effect of a treatment can be estimated by simply comparing the average of the outcome of interest across treatments. We set the 30-day condition as the control and estimate the average effects of the 14 and 7-day treatments on subscriptions for training and test data. The results from this analysis are shown in Table 5.

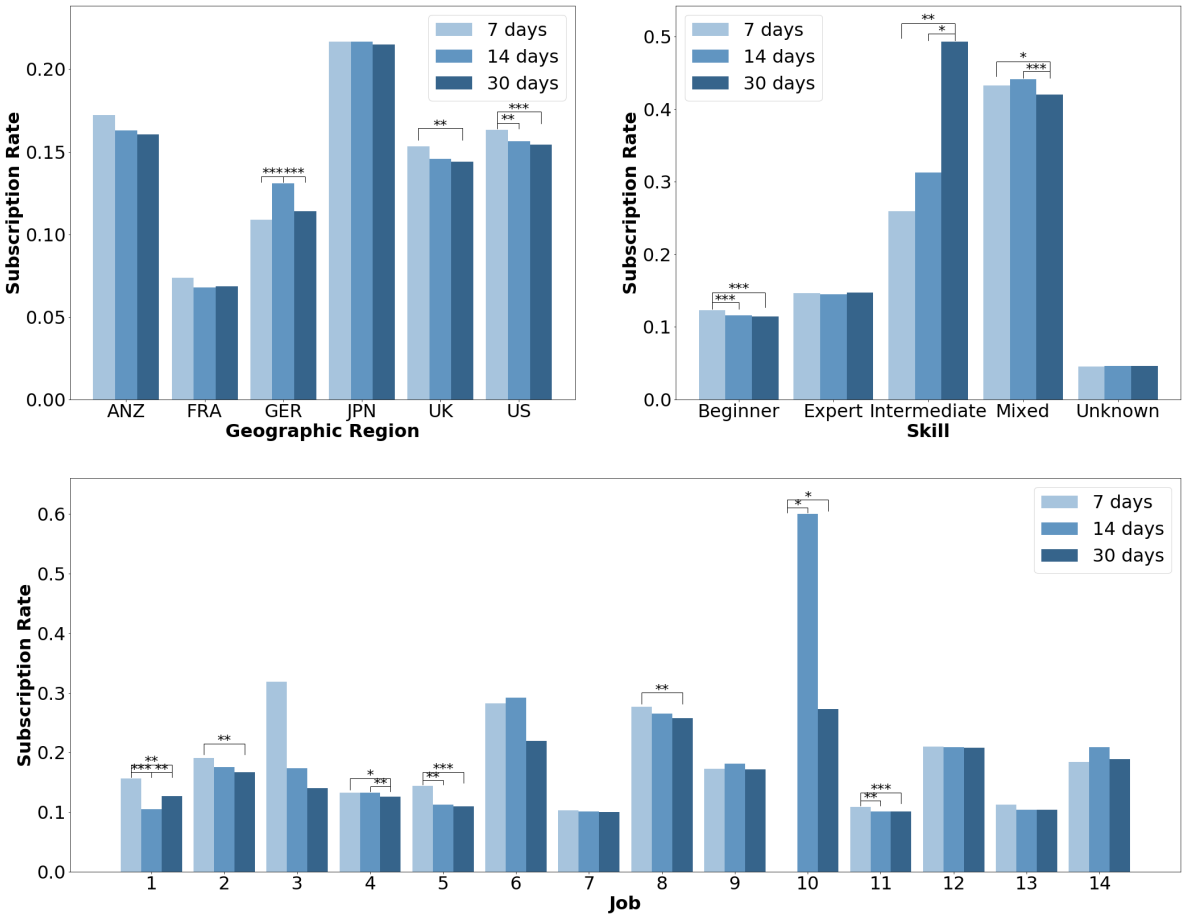
The 7-day trial increases the subscription rate by 4.33% over the baseline of the 30-day condition in the training data and by 5.59% in the test data. However, in both data sets, the effect of the 14-day trial is not significantly different from that of the 30-day trial. These results suggest that a uniform targeting policy that gives the 7-day treatment to all users can significantly increase subscriptions.

We also see that the average treatment effect is fairly small compared to the outcome, which is either zero or one (see Table 5). This effect-size makes intuitive sense since it is natural that the effect of the length of the free trial should be very small compared to other factors that can affect customer acquisition, especially in a near monopolistic market. Nevertheless, the magnitude of this effect implies a low signal-to-noise ratio. As discussed in §1.2, this is one of our estimation challenges, and will affect the empirical performance of the different estimators that we consider.

Finally, note that the gains and subscription rates in the training and test data are slightly different. As discussed earlier, this is due to the randomness in the splitting procedure and the low signal-to-noise ratio. Hence, we should be careful when comparing the performance of models across data sets.

6.6.2 Heterogeneity in User Response

We now examine whether the firm can do better by personalizing its treatment assignment based on a user's pre-treatment characteristics. In the top left panel of Figure 2, we partition the data into six sub-groups based on the user's geographic region and present the average subscription rates for the

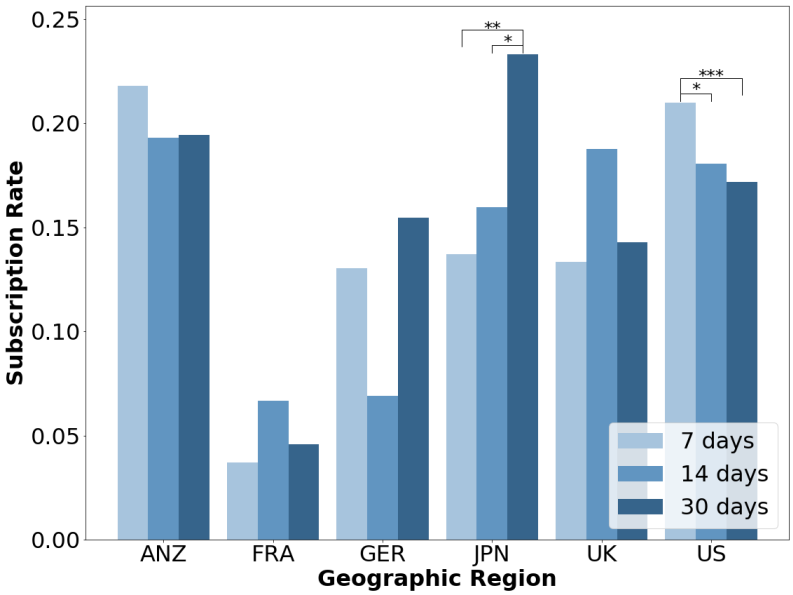


$* \; p < 0.1$, $** \; p < 0.05$, $*** \; p < 0.01$

Figure 2: Heterogeneity in consumers' response to the three trial lengths within three categories – Geographic region, Skill, and Job. (We do not include sub-category names for Job to preserve the firm's anonymity.)

three trial lengths for each region. The results suggest that there is some heterogeneity in response rates by region. For example, the 14-day trial is more effective in Germany while the 7-day trial is more effective in the United States of America. Next, we perform a similar exercise on skill-level and job (see the top right and bottom panels in Figure 2). Again, we find that users' responsiveness to the treatment is a function of their skill level and job. For instance, the 7-day trial is significantly better for Beginners, whereas the 14-day trial is more effective for Mixed-skill users.

We also find that users' responsiveness varies with the interaction of two or more pre-treatment variables. Figure 3 shows the treatment effect for users whose job description is Business Professionals by their geographic region. Based on Figure 2, we know that, on average, the 7-day trial is better than the 30-day trial for Business Professionals (Job $= 2$). However, this effect varies by the user's geographic location. While Business Professionals in the US do better with shorter free trials,



$^{*}\ p < 0.1$, $^{**}\ p < 0.05$, $^{***}\ p < 0.01$

Figure 3: Subscription rates for users whose Job $= 2$ across the six geographic regions.

those in Japan respond better to the 30-day trial.

Together, these results suggest that users' responsiveness to trial lengths is heterogeneous along many pre-treatment variables and their higher-order interactions. So, we develop personalized free trial policies based on the framework outlined in §3, §4, and §5, and present the results in the §7.

7 Results

We start by presenting the results from the first-stage outcome estimation and CATE estimation models in §7.1. Next, in §7.2, we define targeting policies and present the fraction of users who are given each of the three treatments under each policy that we design. Then, we quantify the gains in subscription from adopting different counterfactual policies in §7.3 and we discuss the source of these differences in §7.4. In §7.5, we present some descriptive analysis to explain why/how treatment assignment varies across consumers under the optimal personalized policy. Then, in §7.6, we discuss the revenue improvements under different counterfactual policies.

7.1 First-stage Estimation and Results

We now briefly discuss a few details related to the implementation of the estimators discussed in §4. First, both X_i and W_i are categorical variables in our setting. So we transform each categorical

Method	Mean Squared Error	
	Training Set	Test Set
Linear Regression	0.0932	0.0933
Lasso	0.0933	0.0933
CART	0.0916	0.0920
Random Forest	0.0904	0.0915
XGBoost	0.0905	0.0911

Table 6: Comparison of the predictive performance of the five outcome estimation methods. The MSE for any method are calculated as $\frac{\sum_{i=1}^{N}(\hat{y}_i - y_i)^2}{N}$, where $\hat{y}_i$ is the prediction of y_i and N is the number of data-points in the data-set being considered.

variable into a set of dummy variables for each sub-category (also referred to as one-hot encoding in the machine learning literature). After this transformation, we have 82 dummies for the pre-treatment variables, three treatment dummies, and 246 first-order interaction variables. This gives us a total of 331 explanatory variables for the linear regression and lasso models. For the rest of the tree-based models, we directly feed in the 82 dummy variables for the pre-treatment characteristics and the three treatment dummies. Second, in our setting, propensity scores are known and constant for each observation since treatment assignment is fully randomized. Therefore, we do not estimate them from data for the causal forest algorithm. Finally, we use five-fold cross-validation on the training data to tune hyper-parameters for all the estimators (except regression which does not require hyper-parameter tuning). See Appendix §A for more details on model-selection.

The main result of interest for outcome prediction models is their predictive performance. So in Table §6, we present the MSE for the five outcome estimators on both training and test data. Two main findings emerge. First, we find that linear and lasso are the worst in terms of model-fit, with XGBoost performing the best. This finding is consistent with earlier papers that have found XGBoost to be the best outcome prediction method in tasks involving prediction of human behavior (Rafieian and Yoganarasimhan, 2018; Rafieian, 2019a). However, we also find that all the tree-based models – CART, Random Forest, and XGBoost – suffer from overfitting in spite of hyper-parameter tuning. That is, their performance on the test data is significantly worse than that on training data.

Next, we discuss the results from the CATE estimation methods. In this case, we cannot compare the CATE estimates with any ground truth since we (as researchers and managers) do not know the true treatment effects. So we simply present the distributions of treatment-effects in Appendix B.

7.2 Uniform and Personalized Policies

We start by designing the following three uniform (one length for all) policies:

- π_{30} – This policy prescribes the 30-day treatment for all users. It was used by the firm at the

Policy-prescribed Treatment	π_7	π_{14}	π_{30}	π_{reg}	π_{lasso}	π_{cart}	π_{r_forest}	$\pi_{xgboost}$	π_{c_tree}	π_{c_forest}
7 Days	1	0	0	0.521	0.689	1	0.444	0.697	1	0.911
14 Days	0	1	0	0.322	0.232	0	0.269	0.185	0	0.089
30 Days	0	0	1	0.157	0.079	0	0.287	0.118	0	0
Total	1	1	1	1	1	1	1	1	1	1

Table 7: Fraction of users assigned the 7-, 14-, and 30-day trials under counterfactual policies (in test data).

time of the experiment and we therefore use it as the baseline policy in all our comparisons.

- π_{14} – This policy prescribes the 14-day treatment for all users.
- π_7 – This policy prescribes the 7-day treatment for all users. Since we found that 7 days is the best average treatment (see §6.6.1), this should be the best uniform policy.

Next, to apply the personalized policy design framework from §3, we need to ensure that Assumptions 1 – 3 are satisfied. So we discuss their applicability to our context below:

- Assumption 1 is automatically satisfied because we have a fully randomized experiment.
- Assumption 2 is satisfied because we do not expect any network effects in our setting (since the experiment was run on unconnected users distributed all over the world).
- Assumption 3 is satisfied in all randomized experiments by definition.

We design seven personalized policies based on the estimators discussed in §4: π_{reg}, π_{lasso}, π_{cart}, π_{r_forest}, $\pi_{xgboost}$, π_{c_tree}, and π_{c_forest}. To design these policies, we follow the two optimal policy design methods discussed in §3.2.2. As discussed in §3.2.2, in reality, there is only one optimal policy. In theory, with sufficiently large data and a fully flexible model, we should be able to correctly identify this reward-maximizing optimal policy. However, in practice, each of these methods imposes different semi-parametric restrictions on the functional form of the model and make different bias-variance trade-offs. Thus, with finite data, these policies can vary substantially.

Table 7 presents the fraction of users who are given each of the three treatments (7, 14, and 30 days) under the policies described above. There are a few points worthy of note here. First, policies based on CART and causal tree do not personalize treatment assignment and end up giving the 7-day treatment to all users. Thus, they are equivalent to the best uniform policy: $\pi_7 \equiv \pi_{cart} \equiv \pi_{c\_tree}$. Second, the two policies based on the two outcome estimators – lasso and XGBoost – are somewhat similar in their treatment assignment. Both prescribe the 7-day treatment to $\approx 70\%$ of users, the 14-day treatment to $\approx 20\%$ of users, and the 30-day treatment to $\approx 10\%$ of users. In contrast, π_{reg} and π_{r_forest} prescribe the 7-day treatment to the least number of users whileπ_{c_forest} prescribes the 7-day treatment to 91% of users (and the 30-day treatment to no one).

7.3 Gains from Counterfactual Policies

7.3.1 Empirical Policy Evaluation

We can evaluate the expected reward from each policy described using Equation (15). In theory, in a randomized experiment, the propensity of treatment assignment is orthogonal to the treatment prescribed by any policy π. Thus, $e(W_i = w, X_i) = e(W_i = w)\ \forall\ w\ \in \mathcal{W}$ is known and constant for all observations. However, in practice, within the set of users for whom policy π prescribes w, the empirical treatment propensities ($\hat{e}_{\pi(X_i)}(W_i = w)$) might not be the same as that in the full data. Thus, to correctly estimate the reward under a given policy, we need to use the empirical treatment propensities within each assigned trial length. We therefore modify Equation (15) as follows:

$$\hat{R}_{IPS}(\pi, Y) = \frac{1}{N}\sum_{i=1}^{N}\frac{1[W_i = \pi(X_i)]Y_i}{\hat{e}_{\pi(X_i)}(W_i)}, \tag{16}$$

where $\hat{e}_{\pi(X_i)}(W_i)$ is the probability that a user whom the policy prescribes treatment $\pi(X_i)$ is given W_i. Formally, $\hat{e}_{\pi(X_i)}(W_i) = \frac{\frac{1}{N}\sum_{j=1}^{N} 1[W_j=W_i,\pi(X_j)=\pi(X_i)]}{\frac{1}{N}\sum_{j=1}^{N} 1[\pi(X_j)=\pi(X_i)]}$. Thus, instead of calculating the treatment propensities on the entire data, we now do it on the subset of users who are prescribed $\pi(X_i)$ by the policy π. Intuitively, if we look at Figure 1, instead of calculating treatment propensities on the full square (i.e., all the data), we calculate them separately for each column (i.e., set of users for whom the policy prescribes a treatment).

We present the expected reward from all the policies in Table 8. In all our reward calculations, we use subscription as the outcome of interest. Thus, the expected rewards shown in columns 3 and 4 are the estimated subscription rates for the training and test data, respectively.[11]

We clarify a few points before discussing the results. First, all the results will look more promising on the training data since the policies developed are based on models trained and validated on this data. So, the test data provide a more neutral ground for model comparisons and we focus on it in our discussions. That said, we cannot directly compare the estimated subscription rates (or expected reward) for different datasets (Yi et al., 2013). In general, the performance of a given policy or model on a dataset is a function of: (a) the model used to form the policy, and (b) the joint distribution of the covariates and outcomes in that dataset. Simply put, in Table 8, comparisons within a column are valid but comparisons across columns are less meaningful.

[11] Estimates of the subscription rates (or reward) with theoretical propensities based on Equation (15) are very similar to those shown in Table 8.

Policy category	Policy	Estimated Subscription (%)		Increase in subscription (%)	
		Training Set	Test Set	Training Set	Test Set
One length for all	π_{30} (Baseline)	14.68	14.63	—	—
	π_{14}	14.90	15.11	1.45	3.28
	π_{7}	15.32	15.44	4.34	5.59
Based on outcome estimation	π_{reg}	15.89	15.33	8.21	4.83
	π_{lasso}	15.85	15.62	7.97	6.81
	π_{cart}	15.32	15.44	4.34	5.59
	π_{r_forest}	17.42	14.82	18.67	1.32
	$\pi_{xgboost}$	16.00	15.53	8.98	6.17
Based on CATE estimation	π_{c_forest}	15.57	15.46	6.09	5.71
	π_{c_tree}	15.32	15.44	4.34	5.59

Table 8: Gains in subscription from implementing different counterfactual free-trial policies. The results for policies π_{cart}, π_{c_tree}, and π_7 are the same since they prescribe the 7-days treatment to all users.

7.3.2 Comparison of Counterfactual Policies

The top four policies based on Table 8 are: π_{lasso}, $\pi_{xgboost}$, π_{c_forest}, and π_7, in that order. However, these differences may not be significant. Therefore, in Table 9, we present results from paired t-tests comparing the IPS-reward estimates for the top four methods with each other, as well as with other methods. We refer readers to Appendix §C for details on these tests.

We now discuss the main findings of interest from Tables 8 and 9. First, we find that 7-days for all is the best uniform policy (5.59% increase over the baseline of 30-days for all).

Next, we find that among the personalized policies, the policy based on lasso is the most profitable (6.81% improvement in subscription rate), followed by the policy based on XGBoost (6.17% improvement in subscription rate). Both of these policies are based on outcome estimation. Moreover, the differences between these policies and others are significant, i.e., the reward from π_{lasso} is significantly better than that from other policies (see column 2 in Table 9). Similarly, $\pi_{xgboost}$ is significantly better than all other policies, except π_{lasso} (see column 3 in Table 9).

However, not all outcome estimation methods do well when it comes to policy design. Linear regression and random forest overfit to the training data (high estimated subscription in-sample), but perform poorly on test data.[12] Interestingly, we do not find much correlation between an outcome estimator's predictive ability and its efficacy in policy design (recall Table 6). This is expected since the objective function in outcome estimation methods is predictive ability, which is different from policy design or performance.

An important point of note here is that poorly designed personalized policies (e.g., those based

[12]While this could be due to differences in the data itself, the magnitude of this difference suggests overfitting.

Compared Policy (π)	$\Delta\hat{R}(\pi_{lasso}, \pi)$	$\Delta\hat{R}(\pi_{xgboost}, \pi)$	$\Delta\hat{R}(\pi_{c\_forest}, \pi)$	$\Delta\hat{R}(\pi_7, \pi)$
$\pi_{xgboost}$	0.95 (2.98×10^{-38})	—	—	—
π_{c_forest}	1.66 (1.60×10^{-75})	0.70 (4.35×10^{-22})	—	—
π_7	1.84 (1.47×10^{-83})	0.88 (6.61×10^{-30})	0.18 (2.05×10^{-5})	—
π_{reg}	2.94 ($\leq 10^{-100}$)	1.99 ($\leq 10^{-100}$)	1.29 (5.26×10^{-44})	1.11 (1.38×10^{-30})
π_{r_forest}	8.06 ($\leq 10^{-100}$)	7.10 ($\leq 10^{-100}$)	6.40 ($\leq 10^{-100}$)	6.22 ($\leq 10^{-100}$)

Table 9: Comparison of reward estimates based subscription for top four policies with each other and other counterfactual policies (on test data). $\Delta\hat{R}(\pi_a, \pi_b)$ is defined as $\hat{R}_{IPS}(\pi_a, Y) - \hat{R}_{IPS}(\pi_b, Y)$. For each cell, we show $\Delta\hat{R}(\pi_a, \pi_b)$ multiplied by 10^3 and the p-value for the paired t-test in parenthesis.

on regression and random forest) can actually do worse than the best uniform policy based on average treatment effects. As shown in the last column of Table 9, the reward from π_7 is significantly better than that from the personalized policies, π_{reg} and π_{r_forest}. This problem becomes obvious only when we look at the evaluation metric (IPS reward estimate) on the test data. Simply looking at the relative performance of these policies in the training data can lead to the (incorrect) conclusion that π_{reg} and π_{r_forest} outperform the uniform policy π_7. We, as policy-makers, should therefore be careful in both designing and evaluating personalized policies. It is essential that: (1) we do not conflate a model's predictive ability with its ability to form policy, and (2) we evaluate the performance of each policy on an independent test data with appropriate policy evaluation metrics.

Finally, we find that the recently proposed CATE estimators, causal tree and causal forest, perform poorly when it comes to personalized policy design. As discussed in §7.2, causal tree does not personalize the policy at all since it assigns the 7-day trial to all users. While π_{c_forest} does personalize treatment assignment to some degree, the gain over the simple 7-days for all policy is very marginal. (As shown in column 4 of Table 9, $\Delta\hat{R}(\pi_{c\_forest}, \pi_7)$ is positive and significant, but very small compared to the gain from other methods such as lasso and XGBoost.) Thus we have: $\hat{R}_{IPS}(\pi_{c\_tree}, Y) = \hat{R}_{IPS}(\pi_7, Y) \approx \hat{R}_{IPS}(\pi_{c\_forest}, Y)$.

Our results thus suggest that managers may be better off adopting the best uniform policy instead of investing resources in personalizing policies based on these methods. This is a useful finding since these methods are gaining traction in the marketing literature and researchers are starting to use them for CATE estimation (e.g., Guo et al. (2017)). However, both in marketing research and practice, the end goal is often policy design and not CATE estimation (e.g., decide who should get a price promotion and who should not, or who should be targeted with advertising and who should not be). In such cases, relying on CATE estimators such as causal forest can be sub-optimal. Further, the causal forest model suffers from a lack of interpretability and is more susceptible to hyper-parameter tuning issues. In contrast, lasso is not only easy to specify, estimate, and interpret

but also offers the best performance for policy design (at least in our setting).

7.3.3 Policy Performance and Estimates of Treatment Effects

We now examine if there is any relationship between the individual-level treatment effects (or CATEs) estimated based on a specific method and the performance of the policy based on that method. In Figure 4, we show the CDFs of CATE estimates for the three pairs of treatments ($\tau_{7,30}$, $\tau_{14,30}$, and $\tau_{7,14}$) based on each of the methods discussed in §4. For expositional purposes, we focus on the top panel in the rest of this section. This panel depicts the CATE estimates for the 7 and 30-day treatments, i.e., $\tau_{7,30}$. The discussions for the middle and bottom panels are similar.

The first pattern that stands out is that CATE estimates based on CART, causal tree, and casual forest show very little heterogeneity (see the three vertical lines to the right of zero). This explains why policies based on these methods perform poorly – they are unable to personalize the policy sufficiently to optimize users' response at the individual level. Recall that π_{cart} and π_{c_tree} give the same treatment to all users and π_{c_forest} gives the 7-day treatment to over 91% of users (Table 7).

In contrast, CATE estimates based on linear regression and random forest show the maximum amount of heterogeneity (see the two rightmost curves). This pattern, in combination with the poor performance of π_{reg} and π_{r_forest} on test data (and their extremely good performance on training data) suggests serious overfitting problems. That is, these models seem to infer much more heterogeneity than is true in the data. This, in turn, leads them to personalize policies to a larger extent than is ideal. Recall that π_{reg} and π_{r_forest} show the maximum dispersion (or variation) in treatment assignment compared to other policies (Table 7).

Finally, we see that the CDFs of treatment effects based on lasso and XGBoost lie somewhere in between the above two groups. They show sufficient heterogeneity, but not too much. Hence, policies based on these methods are able to personalize the treatment sufficiently, but also avoid overfitting at the same time. As shown in Table 7, the dispersion in treatment assignment for these two policies is higher than that in π_{cart}, π_{c_tree}, and π_{c_forest}, but lower than that in π_{reg} and π_{r_forest}.

Thus, the ideal estimators for policy design are those that are able to capture sufficient heterogeneity to personalize effectively without overfitting (i.e., capture spurious heterogeneity).

7.4 Understanding the Source of these Differences

We now examine why some policies perform better than others. We present a simple example in §7.4.1 to fix ideas. In §7.4.2, we derive a general mathematical expression to quantify the gains from any policy over the current experiment. In §7.4.3, we use these mathematical expressions to explain the difference in the performance of the various counterfactual policies.

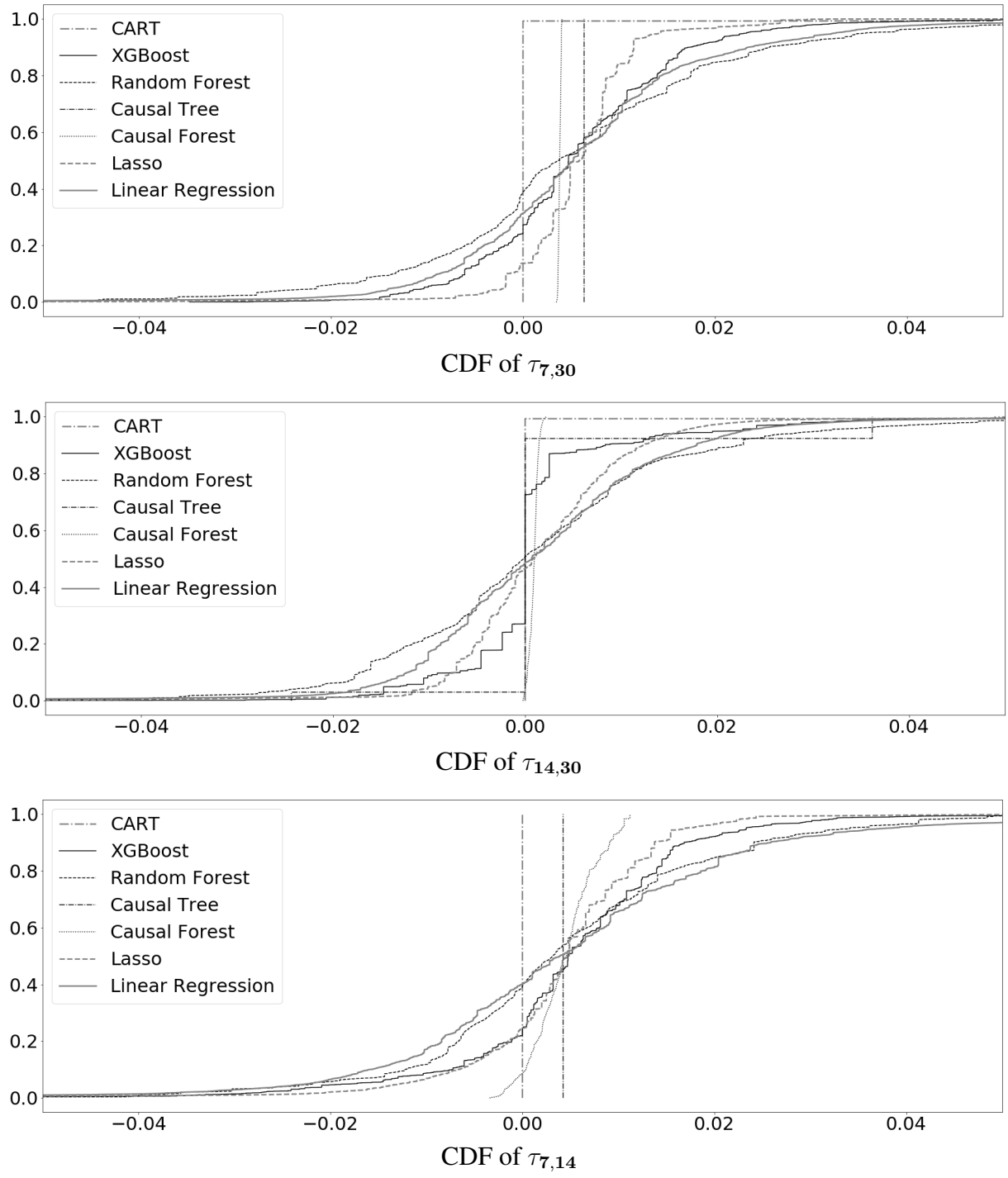


Figure 4: The CDF of estimated CATEs from using different methods (for test data).

7.4.1 Example

Consider a randomized experiment with two groups (A and B) with equal number of individuals, as shown in Table 10. Assume $X_A = X_B = X$, and $W_A = 0$, $W_B = 1$, and consequently $e(X) = 1/2$. Let the average outcome for the two groups be $\bar{Y}_B$ and $\bar{Y}_B$, respectively.

Group	Covariate	Treatment	Outcome
A	$X_A = X$	$W_A = 0$	$\bar{Y}_A$
B	$X_B = X$	$W_B = 1$	$\bar{Y}_B$

Table 10: Overview of an example experiment.

		π_1	π_2
Assigned treatment	$\pi(X)$	$\pi_1(X) = 0$	$\pi_2(X) = 1$
IPS reward	$\hat{R}_{IPS}(\pi, Y)$	$\bar{Y}_A$	$\bar{Y}_B$
Reward improvement	$\Upsilon(\pi, Y)$	$\frac{\bar{Y}_A - \bar{Y}_B}{2}$	$\frac{\bar{Y}_B - \bar{Y}_A}{2}$

Table 11: Overview of reward and improvement under counterfactual policies.

There are two possible policies that we can design in this case: (1) $\pi_1 : \pi_1(X) = 0$ and (2) $\pi_2 : \pi_2(X) = 1$. The IPS reward for both the policies and improvement of each policy over the current experiment $\Upsilon(\pi, Y) = \hat{R}_{IPS}(\pi, Y) - \bar{Y}$ are shown in Table 11.

Note that the improvement from any policy depends on two factors. First, the fraction of people who get a specific treatment (which in the case of both π_1 and π_2 is one). Second, the difference in the expected outcomes for policy congruent and policy in-congruent users. For example, in the case of policy π_1, the average outcome for policy congruent users in our experiment is $\bar{Y}_A$ and that for policy in-congruent users is $\bar{Y}_B$.

7.4.2 Quantifying the Gains from a Policy

We now formalize the intuition behind this simple example for a general case. In a fully randomized experiment, the treatment propensities are the same for all observations i.e., $e(W_i = w, X_i) = e_w$.

So we can therefore write the IPS reward function for any policy π as:

$$\begin{aligned}
\hat{R}_{IPS}(\pi, Y) &= \frac{1}{N}\sum_{i=1}^{N}\frac{1[W_i = \pi(X_i)]Y_i}{e(W_i, X_i)} \\
&= \frac{1}{N}\left[\sum_{w\in\mathcal{W}}\left[\sum_{i:\pi(X_i)=W_i=w}\frac{Y_i}{e_w}\right]\right] \\
&= \frac{1}{N}\sum_{w\in\mathcal{W}}\frac{1}{e_w}\left[\sum_{i:\pi(X_i)=W_i=w} Y_i\right] \\
&= \frac{1}{N}\sum_{w\in\mathcal{W}}\frac{1}{e_w} N e_w \upsilon_w \bar{Y}_{[\pi(X_i)=W_i=w]} \\
&= \sum_{w\in\mathcal{W}} \upsilon_w \bar{Y}_{[\pi(X_i)=W_i=w]}, \qquad (17)
\end{aligned}$$

where $\upsilon_w = \sum_{i=1}^{N}\frac{1[\pi(X_i)=w]}{N}$ is the fraction of observations for whom the the policy π assigns treatment w and $\bar{Y}_{[\pi(X_i)=W_i=w]}$ is the average outcome for policy congruent observations that are assigned treatment w by policy π.

Next, let $\Upsilon(\pi, Y)$ denote the change or improvement in reward from implementing policy π compared to the current experiment. We can formally write $\Upsilon(\pi, Y)$ as:

$$\begin{aligned}
\Upsilon(\pi, Y) &= \hat{R}_{IPS}(\pi, Y) - \frac{1}{N}\sum_{i=1}^{N} Y_i \\
&= \sum_{w\in\mathcal{W}}\left[\upsilon_w \bar{Y}_{[\pi(X_i)=W_i=w]} - \sum_{w'\in\mathcal{W}} \upsilon_w e_{w'} \bar{Y}_{[\pi(X_i)=w, W_i=w']}\right] \\
&= \sum_{w\in\mathcal{W}} \upsilon_w \left[\sum_{w'\in\mathcal{W}} e_{w'}\left(\bar{Y}_{[\pi(X_i)=W_i=w]} - \bar{Y}_{[\pi(X_i)=w, W_i=w']}\right)\right]. \qquad (18)
\end{aligned}$$

Note that $\frac{1}{N}\sum_{i=1}^{N} Y_i$ is not a function of the policy π being evaluated because it is simply the reward of the randomized experiment currently being implemented. It is therefore constant across different policies. Hence, the policy improvement function $\Upsilon(\pi, Y)$ can be used as a metric to compare the performance of different policies and explain why some policies do better than others.

According to Equation (18), gains from a policy π depend on two quantities: (1) For each treatment w, the fraction of observations for which that treatment is assigned (υ_w), and (2) For these observations, the difference in expected outcomes for policy congruent and policy in-congruent users, i.e., $\left(\bar{Y}_{[\pi(X_i)=W_i=w]} - \bar{Y}_{[\pi(X_i)=w, W_i=w']}\right)\ \forall\ w'$. A good policy is one that ensures that – (a) the

difference between policy congruent and in-congruent users is positive and large for each w, and (b) a higher fraction of users (υ_ws) get the more positive and higher differences.

7.4.3 Explaining the Gains from Counterfactual Policies

We now use the expressions derived above to explain the difference in performances of the counterfactual policies discussed earlier. In Table 12, for each policy π, we show: (1) the fraction of users for whom the policy prescribes the three treatments (7, 14, and 30 days), and (2) the subscription rate (or average outcome) of both policy congruent and in-congruent users (e.g., for users for whom the policy prescribes 7 days, the subscription rate of all three sets of users – policy congruent users who received the 7-day treatment, policy in-congruent users who received the 14 and 30 day treatments). Note that the first sub-column within each policy simply shows the fraction of users assigned 7, 14, and 30 days by that policy. It is useful to recognize that these numbers in the bottom panel (for test data) are exactly the same as that in Table 7 from §7.2.

We now discuss the performance of the different counterfactual policies in the training data (top panel of Table 12). First, consider π_{reg}. Here, the difference between policy congruent and policy in-congruent cells for each treatment is always positive, e.g., for the set of users for whom the policy assigns 7 days, the average outcome for policy congruent users is 14.224 (grey cell), which is greater than both 12.36 and 12.569 (the white cells). The same pattern is true for the rest of the policies in the training data – for any treatment w, the average outcome of policy congruent users is always greater than that of policy in-congruent users.

However, things look very different when we look at the performance of these policies in the test data (bottom panel). In the case of π_{reg}, we see that the average outcome for policy congruent users is not always the highest. For example, the average outcome for policy congruent users assigned the 30-day trial is 11.761. However, if these users had instead received the 7 or 14-day trials (which the policy deemed sub-optimal for them), they would have done better (with average subscription rates of 11.809 and 11.993, respectively)! π_{reg} is similarly sub-optimal for users for whom it prescribes the 14-day treatment. Together, these two sets of users constitute 47.9% ($0.157 + 0.322 = 0.479$) of the population. Thus, the policy π_{reg} is sub-optimal for a large fraction of users and the extent of sub-optimality (the differences in the outcomes for policy congruent and in-congruent users) is high. Hence, the overall performance of this policy is poor.

In contrast, if we look at π_{lasso}, we see that for users whom it assigns to the 7 and 14-day treatments, the average outcome for the policy congruent cells is higher than that of policy in-congruent cells. More importantly, these differences are substantial and happen for a large fraction of users ($0.689 + 0.232 = 0.921$). We do find that this policy is somewhat sub-optimal for the users it assigns to the 30-day treatment (16.914 is less than both 17.172 and 17.033). However,

Training Dataset

Trial Length		π_7		π_{reg}		π_{lasso}		π_{r_forest}		$\pi_{xgboost}$		π_{c_forest}	
Optimal	**Actual**	**Frac.**	**Sub. Rate**	**Frac.**	**Sub. Rate**	**Frac.**	**Sub. Rate**	**Frac.**	**Sub. Rate**	**Frac.**	**Sub. Rate**	**Frac.**	**Sub. Rate**
7 Days	7 Days	1	15.320	0.522	14.224	0.688	10.686	0.445	14.170	0.698	13.409	0.913	15.143
	14 Days		14.896		12.360		9.451		10.360		11.969		14.388
	30 Days		14.683		12.569		9.575		10.404		12.153		14.364
14 Days	7 Days	—	—	0.322	18.878	0.233	28.449	0.271	19.897	0.184	22.268	0.087	17.146
	14 Days		—		20.549		30.288		24.139		25.209		20.158
	30 Days		—		19.503		28.547		20.228		22.862		18.084
30 Days	7 Days	—	—	0.157	11.575	0.078	16.824	0.284	12.764	0.118	15.731	—	—
	14 Days		—		11.625		17.042		12.973		15.833		—
	30 Days		—		11.874		18.277		16.128		16.962		—
$\Upsilon(\pi, Y)$		0.508		1.078		1.042		2.612		1.190		0.765	

Test Dataset

Optimal	Actual	Frac.	Sub. Rate	Frac.	Sub. Rate	Frac.	Sub. Rate	Frac.	Sub. Rate	Frac.	Sub. Rate	Frac.	Sub. Rate
7 Days	7 Days	1	15.444	0.521	13.406	0.689	10.538	0.444	11.401	0.697	13.281	0.911	15.145
	14 Days		15.107		12.924		9.741		11.407		12.786		14.764
	30 Days		14.627		12.515		9.621		10.865		12.122		14.252
14 Days	7 Days	—	—	0.322	20.499	0.232	29.720	0.269	22.447	0.185	23.868	0.089	18.524
	14 Days		—		20.193		30.316		20.452		23.570		18.707
	30 Days		—		19.437		28.710		20.630		23.036		18.447
30 Days	7 Days	—	—	0.157	11.809	0.079	17.172	0.287	15.189	0.118	15.037	—	—
	14 Days		—		11.993		17.033		15.710		15.703		—
	30 Days		—		11.761		16.914		14.827		16.232		—
$\Upsilon(\pi, Y)$		0.622		0.511		0.800		-0.002		0.708		0.639	

Table 12: For each counterfactual policy, we show the fraction of users that the policy assigns to the 7, 14, and 30 day treatments, the subscription rate of policy congruent users (grey cells) and in-congruent users (white cells), and $\Upsilon(\pi, Y)$. For example, $\Upsilon(\pi_{lasso}, Y)$ on test data is calculated as follows: $\Upsilon(\pi_{lasso}, Y) = 0.689*[0.152 * 0 + 0.149 * 0.797 + 0.699 * 0.917]+0.232*[0.149 * 0.596 + 0.150 * 0 + 0.700 * 1.606]+0.079*[0.148 * -0.257 + 0.153 * -0.119 + 0.698 * 0] = 0.800$. The top panel is on training data and bottom panel on test data.

these differences are quite small ($16.914 - 17.172 \approx -0.26$ and $16.914 - 17.033 \approx 0.12$) and the fraction of users for whom it makes this mistake is also small (0.079). Overall, this is the reason behind the superior performance of π_{lasso}: it is able to improve the response of a vast majority of users (over 92%) by a substantial amount and when it makes a mistake in treatment allocation, it does so for a small set of users and the magnitude of this mistake is quite small.

Next, we can see that $\pi_{xgboost}$ is the second-most effective policy after π_{lasso}: it is sub-optimal for users whom it prescribes the 14-day treatment. While the difference between the expected outcomes for policy congruent and in-congruent users is not very high, the fraction of users for whom this mistake happens is higher than that in the case of lasso (18.5%).

It is also apparent why the policy based on random forest, π_{r_forest}, performs so poorly – it assigns sub-optimal treatments to all the users. Notice that in no condition (or treatment) is the grey cell the highest for that condition. Finally, we look at the policy based on causal forest, π_{c_forest}. Interestingly, this is the only policy that does not make sub-optimal allocations. For both sets of users (those assigned to the 7-day and 14-day treatments), the policy congruent cells are always highest. However, the problem with this policy is that these differences are minor (e.g., in the case of users whom it assigns to the 7-day treatment, the gain of going from 14-days to 7-days is 0.381). Thus, π_{c_forest} is not able to sufficiently personalize treatments. It is thus similar to the π_7, which also does not make sub-optimal allocations (but also does not personalize treatment assignment).

7.5 Segmentation Based on Policy-prescribed Treatment

So far, we focused on the question of "Who (should get a treatment)". We now examine the question of "Why (should s/he get a specific treatment)". Understanding why some users respond well to longer trials while others respond better to shorter trials can give us insight into consumers' preferences and decision-making process. These insights are valuable for two reasons. First, from the firm's perspective, they can be leveraged to improve other marketing interventions such as pricing. Second, from a research perspective, this gives us a broader theoretical understanding of the heterogeneity in the effectiveness of trial length on subscription, which can be extrapolated to other firms and settings.

We now segment consumers based on their prescribed treatment (according to π_{lasso}, since it is the best policy) and correlate their prescribed treatment with pre-treatment demographic and post-treatment behavioral variables. Since these variables are not randomized across individuals, we cannot make any causal inference on their effect on users' responsiveness. Rather, the goal is to present some suggestive evidence for the underlying mechanisms driving users' behavior.

In Appendix D, we present Tables A4 and Table A5, that show the distributions of pre-treatment demographic variables and post-treatment behaviors for three segments:

- Segment 1: Users who are prescribed the 30-day treatment, i.e., $\pi_{lasso}(X_i) = 30$ days.
- Segment 2: Users who are prescribed the 14-day treatment, i.e., $\pi_{lasso}(X_i) = 14$ days.
- Segment 1: Users who are prescribed the 7-day treatment, i.e., $\pi_{lasso}(X_i) = 7$ days.

We now discuss the attributes and behavior of these three segments.

- **Segment 1:** These individuals are more likely to be *legacy* users. They are more experienced than average, are less likely to be students and hobbyists, and more likely to sign up through the app manager instead of the website. All these factors suggest that these are more likely to be legacy users who are already aware of and using some version of the software. This is also corroborated by their post-treatment behavior – conditional on subscription, e.g., the firm is able to retain them a bit longer than average.
- **Segment 2:** These individuals are more likely to be *learners*. They are more likely to be beginners or mixed skill users, and have the highest usage and subscription rate. Thus, these users benefit the most from the free trial since they seem to actually try the product's features and evaluate the product the most before deciding whether to subscribe or not.[13]
- **Segment 3:** These individuals are more likely to be *free-riders*. A vast majority of them are beginners or students and are the least likely to subscribe. These users are likely to be individuals who need the software for specific tasks, such as course projects. As a result, they are more likely to free-ride if given a lengthy free trial. So, a short free trial leads to a higher subscription rate in this group.

It worth mentioning that our findings disprove some of the previous theories on the relationship between users' experience and the effectiveness of free trials. For example, Dey et al. (2013) argues that beginners should be given longer free trials because longer trials allow them to learn about the product, which increases their likelihood of subscription. However, we find the opposite – shorter trials are more effective for beginners and longer trials are better for experienced users. This is because more experienced users are likely to be familiar with the functionalities in the legacy software and need more time to evaluate these functionalities in the new cloud service and compare it with the legacy version. On the other hand, beginners are just starting to get to know the software, and are unlikely to have enough information to evaluate the software. Given that this firm is the dominant player in this market and sets the standards in this industry, a beginner's decision to subscribe or not is likely driven by the brand name rather than their experience during the free trials.

[13] An interesting question that arises here is this: if these users use the product and learn the software during the free trial period, why is the longer 30-day free trial not optimal for them? There are two possible explanations here. First, with a longer trial, these users might decide to free-ride and use the product for their needs, but not subscribe. Second, it is possible that the features that users learn about later in a free trial are less appealing and have a negative impact on subscription (Heiman and Muller, 1996).

7.6 Long-term Consumer Loyalty and Profitability

So far we have focused on designing and evaluating policies that maximize subscriptions. However, a policy that maximizes subscriptions (or short-run conversions) may not be the best long-run policy if it brings in users who are less "profitable" or less "loyal". For example, a policy that increases subscriptions among students (who get a significant educational discount and hence pay lower prices) and/or users who subscribe to lower-end products/bundles (that are priced much lower than the all-inclusive software suite) at the expense of high-end users can lead to lower revenues. Similarly, some policies may increase subscriptions, but do so at the expense of long-term retention, i.e., they may bring in the less loyal consumers who churn within a short period. Thus, a subscription-optimal policy may in fact be sub-optimal from the perspective of long-run outcomes. So, in this section, we examine the performance of the policies discussed earlier on two important post-subscription outcomes of interest for the firm.

1. Consumer loyalty, as measured by subscription length or the number of months a user subscribes to the service over the two years after the experiment.

2. Consumer profitability, as measured by the revenue generated by the user over the two years after the experiment. (In SaaS settings, revenues and profits can be treated as equivalent since the marginal cost of serving an additional user is close to zero.)

Recall that the summary statistics of subscription length and revenues were shown in Table 3.

We first describe the procedure for deriving the IPS estimates of the average subscription length and revenue under a given policy. For each policy π:

- We first segment users into three groups based on the policy-prescribed treatment: (1) $\pi(X_i) = 7$ days, (2) $\pi(X_i) = 14$ days, and (3) $\pi(X_i) = 30$ days.
- Then, to obtain IPS estimate of average subscription length and revenue under policy π, we use Equation (16) and the observed subscription lengths and revenues as the outcome variable (Y_i). One minor point to note is that we do not have access to revenue data for all subscribers (recall the discussion in §6.4.2). So the revenue calculations are done on the subset of users for whom we have non-missing revenue data.

The IPS estimates of subscription length and revenues for the counterfactual policies are shown in Table 13. (By definition, these IPS estimates are for the full population, and not just subscribers.)

The main takeaway is that π_{lasso} performs the best here also, followed by $\pi_{xgboost}$.[14] Thus, there are no concerns about these policies bringing in more subscriptions at the expense of long-run

[14]Tables A2 and A3 in Appendix §C show the results for paired t-tests comparing the gains in subscription length and revenue across policies.

Category	Policy	Subscription Length				Revenue			
		Estimate (Months)		Increase (%)		Estimate ($)		Increase (%)	
		Training	Test	Training	Test	Training	Test	Training	Test
One Length For All	π_{30} (Baseline)	2.17	2.19	—	—	77.31	78.41	—	—
	π_{14}	2.22	2.32	2.59	5.91	79.86	84.85	3.30	8.21
	π_{7}	2.32	2.33	7.27	6.28	83.51	84.52	8.03	7.80
Based On Outcome Estimation	π_{reg}	2.39	2.31	10.19	5.46	86.73	84.47	12.19	7.73
	π_{lasso}	2.39	2.36	10.42	7.96	86.90	87.51	12.41	11.61
	π_{cart}	2.32	2.33	7.27	6.28	83.51	84.52	8.03	7.80
	π_{r_forest}	2.62	2.25	20.90	3.05	94.15	80.50	21.79	2.67
	$\pi_{xgboost}$	2.41	2.36	11.42	7.85	87.76	86.72	13.52	10.60
Based On CATE Estimation	π_{c_forest}	2.38	2.32	9.91	5.98	85.48	84.08	10.57	7.23
	π_{c_tree}	2.42	2.31	11.80	5.41	86.34	84.27	11.69	7.47

Table 13: IPS estimates of the average subscription length and revenue under different counterfactual policies. Note that all the revenue numbers are scaled by a constant factor to preserve the firm's anonymity. The number of observations used for revenue calculation is $234,237$ in the training data and $99,986$ in the test data. The corresponding numbers for subscription length are similar to Table 4

.

consumer loyalty or revenue.[15]

An interesting empirical pattern here is that the gains in subscription length and revenues are quantitatively different from the gain in subscription (comparing the percentage increases in Tables 8 and 13). We now discuss the source of this difference.

We can expand the expected subscription length (denoted by Y_i^l) conditional on treatment W_i for the population of users as follows:

$$\mathbb{E}(Y_i^l|W_i) = \Pr(Y_i^s|W_i) \cdot \mathbb{E}\left[T_{end} - T_{start}|W_i, Y_i^s = 1\right], \tag{19}$$

where $\Pr(Y_i^s|W_i)$ is the probability that user i will subscribe conditional on receiving treatment W_i and $\mathbb{E}\left[T_{end} - T_{start}|W_i, Y_i^s = 1\right]$ is i's expected length of subscription conditional on receiving

[15]Note that π_{14} outperforms π_7 in the test data (on revenue) even though π_7 is the best policy in the training data. We present a brief explanation for this discrepancy now. In general, estimates from one data set are valid in another data set only when the joint distribution of outcomes and covariates are similar in both data sets. This is usually true when the data set is very large or the signal-to-noise ratio is low . However, as discussed in §6.5, neither of these are true in our case; so there are some minor differences in our training and test data due to the randomness in the splitting procedure. The main difference is this – the distributions of subscription length for the 14-day condition in the training and test data are different. This is however not the case for the 7- or 30-day conditions; see Table A6 in Appendix E. Thus, the estimate of subscription length from training data does not translate well to test data, and this leads to the large difference in the subscription length and revenue estimates across the training and test data sets.

treatment W_i and subscribing ($Y_i^s = 1$). The reason for the discrepancy in the gains on the two outcomes – subscription and subscription length – becomes apparent from Equation (19). If trial length affects not just subscription, but also how long a subscriber will remain loyal to the firm, then the gains in Y_i^l will be naturally different from the gains in subscription. To examine if this is true in our data, we show the summary statistics for $\mathbb{E}\left[T_{end} - T_{start} | W_i, Y_i^s = 1\right]$ for the three trial lengths in Table A6 in Appendix E. We see that there are some small differences in this metric across the three trial lengths, which account for the differences between the gains in subscription and gains in subscription length.

Similarly, we can write the expected revenue (denoted by Y_i^r) conditional on treatment W_i for the population as:

$$\mathbb{E}(Y_i^r | W_i) = \Pr(Y_i^s | W_i) \cdot \mathbb{E}\left[T_{end} - T_{start} | W_i, Y_i^s = 1\right] \cdot \mathbb{E}\left[\text{Price}_i | W_i, X_i, Y_i^s = 1\right]. \tag{20}$$

This is similar to Equation (19), with the additional $\mathbb{E}\left[\text{Price}_i | W_i, X_i, Y_i^s = 1\right]$ term. It suggests that trial length can influence revenues through three channels – (1) subscriptions, (2) length of subscription, and (3) price of the product subscribed. The first two were already discussed in the paragraph above. We now examine whether the products that consumers subscribe to and the prices that they pay are also a function of trial length. That is, we examine whether $\mathbb{E}\left[\text{Price}_i | W_i, X_i, Y_i^s = 1\right]$ is indeed a function of W_i in our data. Note that the price that a subscriber pays is a function of both the product that s/he subscribes to (e.g., single product, all-inclusive bundle) as well her/his demographics (e.g., students pay lower prices for the same product). In Table A7 in Appendix E, we present the distribution of products and subscription type (educational, commercial, or government) by trial length for all the subscribers in our data. Again, we see that there are some minor differences in product and subscription types across trial lengths. However, the main reason why revenue gains are significantly different from subscription gains is likely due to the difference in the samples used to calculate the two gains. Recall that revenue gains are calculated on a sample of $334,223$ users for whom revenue data are available, whereas subscription gains are calculated on the full sample of $337,724$ users.

In sum, there are two main takeaways from this section. First, personalized policies designed to optimize short-term outcomes (such as conversion/subscription) also perform well on long-term outcomes (at least in our setting). Second, the magnitude of gains can be different across outcomes since the treatment can affect long-run outcomes through multiple channels (in addition to conversion). In our setting, we find that trial length only plays a marginal role on other factors that affect revenue or loyalty; so the magnitude of the gains across outcomes are similar (controlling for the sample). However, these effects can vary across settings and policy-makers should keep

them in mind when designing and evaluating personalized policies.

8 Conclusions

Technology and digital marketing firms now have unprecedented access to individual-level data on consumers' attributes and the ability to experiment at scale. This has led to a significant interest in methods to effectively personalize marketing interventions based on these data.

We present a framework for personalized targeting policy design and evaluation that is compatible with a high-dimensional covariate space. Our framework consists of three components. In the first component, we define the optimal policy design problem for a general case and show that there are two distinct approaches available to empirically learn the optimal personalized policy – (1) policy design using outcome estimates, and (2) policy design using CATE estimates. In the second component, we design personalized policies based on five outcome estimators (linear regression, lasso, CART, random forest, and boosted regression trees) and two CATE estimators (causal tree and causal forest). In the third component, we use the Inverse Propensity Score (IPS) reward estimator to evaluate the reward or gain from any targeting policy.

Our proposed approach to policy design and evaluation has a few key advantages. First, it does not impose any parametric restrictions on users' response behavior. Rather, it uses data-driven approaches (i.e., regularization-based methods) to learn users' response function from the data. This is important in settings where the covariate space is high-dimensional and we do not have well-founded theoretical models of users' behavior. Second, because it first learns functions for outcomes/CATEs and then uses them for policy design, it circumvents an unstructured search for the optimal policy over the high-dimensional policy space. Finally, the IPS estimator is valid as long as there is sufficient randomization in the data generating process, and is able to give us a consistent estimate of the reward from any policy without actually deploying it in the field.

We apply our framework to data from a large-scale field experiment on free trials conducted by a leading SaaS firm. The firm manipulated the length of free trials such that new users were randomly assigned to 7, 14, or 30 days of free trial. In this setting, the 7-day trial improves subscription by 5.59% over the baseline of 30 days for all policy. We also find that the two outcome estimators – lasso and XGBoost – offer the best performance when it comes to personalized policy design. In contrast, the two CATE estimators – causal tree and causal forest – perform poorly because they are unable to personalize the policy sufficiently. We also see that personalized targeting policies designed to maximize short-run conversions also perform well on long-run outcomes such as consumer loyalty and profitability in SaaS settings.

In sum, we present a three-pronged framework for designing and evaluating personalized targeting policies and present an application from the SaaS industry. A key takeaway of our research

is that while personalization can lead to substantial benefits, many commonly used methods to personalize are often ineffective. Hence, researchers and managers need to use appropriate policy design and evaluation methods for effective personalization.

One caveat is that we only look at the performance of different estimators in designing policy for the same regime. Simester et al. (2019b) find that model-based estimators like lasso can lose their advantage when either the covariates or regimes change significantly. Hence, it would be useful for future researchers to compare the effectiveness of CATE estimators with that of lasso and XGBoost when regimes shift.

References

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

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

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

S. Athey and S. Wager. Efficient policy learning. *arXiv preprint arXiv:1702.02896*, 2017.

S. Athey, J. Tibshirani, S. Wager, et al. Generalized random forests. *The Annals of Statistics*, 47(2):1148–1178, 2019.

J. Bergstra, D. Yamins, and D. D. Cox. Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures. 2013.

J. S. Bergstra, R. Bardenet, Y. Bengio, and B. Kégl. Algorithms for hyper-parameter optimization. In *Advances in neural information processing systems*, pages 2546–2554, 2011.

L. Breiman. Arcing Classifier. *The Annals of Statistics*, 26(3):801–849, 1998.

L. Breiman. Random forests. *Machine learning*, 45(1):5–32, 2001.

L. Breiman, J. Friedman, C. J. Stone, and R. A. Olshen. *Classification and Regression Trees*. CRC press, 1984.

T. Chen and C. Guestrin. Xgboost: A Scalable Tree Boosting System. In *Proceedings of the 22Nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 785–794. ACM, 2016.

H. K. Cheng and Y. Liu. Optimal software free trial strategy: The impact of network externalities and consumer uncertainty. *Information Systems Research*, 23(2):488–504, 2012.

D. Dey, A. Lahiri, and D. Liu. Consumer learning and time-locked trials of software products. *Journal of Management Information Systems*, 30(2):239–268, 2013.

J.-P. Dubé and S. Misra. Personalized pricing and customer welfare. 2019.

M. Dudík, J. Langford, and L. Li. Doubly robust policy evaluation and learning. In *Proceedings of the 28th International Conference on International Conference on Machine Learning*, pages 1097–1104. Omnipress, 2011.

E. M. Feit and R. Berman. Profit-maximizing a/b tests. *Available at SSRN 3274875*, 2019.

B. Foubert and E. Gijsbrechts. Try it, you'll like it—or will you? the perils of early free-trial promotions for high-tech service adoption. *Marketing Science*, 35(5):810–826, 2016.

J. Friedman, T. Hastie, and R. Tibshirani. Regularization paths for generalized linear models via coordinate descent. *Journal of statistical software*, 33(1):1, 2010.

J. H. Friedman. Greedy function approximation: a gradient boosting machine. *Annals of statistics*, pages 1189–1232, 2001.

Gartner. Forecast: Public cloud services, worldwide, 2016-2022, 4q18 update., 2019. URL `https://www.gartner.com/en/newsroom/press-releases/2019-04-02-gartner-forecasts-worldwide-public-cloud-revenue-to-g`.

T. Guo, S. Sriram, and P. Manchanda. The effect of information disclosure on industry payments to physicians. *Available at SSRN 3064769*, 2017.

B. Hansotia and B. Rukstales. Incremental value modeling. *Journal of Interactive Marketing*, 16(3):35–46, 2002.

A. Heiman and E. Muller. Using demonstration to increase new product acceptance: Controlling demonstration time. *Journal of Marketing Research*, pages 422–430, 1996.

G. J. Hitsch and S. Misra. Heterogeneous treatment effects and optimal targeting policy evaluation. 2018.

T. K. Ho. Random decision forests. In *Proceedings of 3rd international conference on document analysis and recognition*, volume 1, pages 278–282. IEEE, 1995.

D. G. Horvitz and D. J. Thompson. A generalization of sampling without replacement from a finite universe. *Journal of the American statistical Association*, 47(260):663–685, 1952.

G. W. Imbens and D. B. Rubin. *Causal inference in statistics, social, and biomedical sciences*. Cambridge University Press, 2015.

G. A. Johnson, R. A. Lewis, and E. I. Nubbemeyer. Ghost ads: Improving the economics of measuring online ad effectiveness. *Journal of Marketing Research*, 54(6):867–884, 2017.

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

D. Lefortier, A. Swaminathan, X. Gu, T. Joachims, and M. de Rijke. Large-scale validation of counterfactual learning methods: A test-bed. *arXiv preprint arXiv:1612.00367*, 2016.

R. Lewis, J. M. Rao, and D. H. Reiley. Measuring the effects of advertising: The digital frontier. In *Economic Analysis of the Digital Economy*, pages 191–218. University of Chicago Press, 2015.

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

P. Milgrom and J. Roberts. Price and Advertising Signals of Product Quality. *Journal of Political Economy*, 94(4):796–821, 1986.

K. Murphy. Machine learning, a probabilistic perspective, 2012.

X. Nie and S. Wager. Quasi-oracle estimation of heterogeneous treatment effects, 2017.

O. Rafieian. Optimizing user engagement through adaptive ad sequencing. Technical report, Technical report, Working paper, 2019a.

O. Rafieian. Revenue-optimal dynamic auctions for adaptive ad sequencing. Technical report, Technical report, Working paper, 2019b.

O. Rafieian and H. Yoganarasimhan. Targeting and privacy in mobile advertising. 2018.

P. M. Robinson. Root-n-consistent semiparametric regression. *Econometrica: Journal of the Econometric Society*, pages 931–954, 1988.

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

D. B. Rubin. Estimating causal effects of treatments in randomized and nonrandomized studies. *Journal of educational Psychology*, 66(5):688, 1974.

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

E. M. Schwartz, E. T. Bradlow, and P. S. Fader. Customer acquisition via display advertising using multi-armed bandit experiments. *Marketing Science*, 36(4):500–522, 2017.

S. S. Shapiro and M. B. Wilk. An analysis of variance test for normality (complete samples). *Biometrika*, 52 (3/4):591–611, 1965.

D. Simester, A. Timoshenko, and S. Zoumpoulis. Efficiently evaluating targeting policies: Improving upon champion vs. challenger experiments. 2019a.

D. Simester, A. Timoshenko, and S. I. Zoumpoulis. Targeting prospective customers: Robustness of machine learning methods to typical data challenges. *Management Science*, 2019b.

M. A. Stephens. Edf statistics for goodness of fit and some comparisons. *Journal of the American statistical Association*, 69(347):730–737, 1974.

T. Sunada. Customer learning and revenue-maximizing trial design. Technical report, Working Paper, 2018.

R. S. Sutton and A. G. Barto. *Reinforcement learning: An introduction*. MIT press, 2018.

A. Swaminathan, A. Krishnamurthy, A. Agarwal, M. Dudik, J. Langford, D. Jose, and I. Zitouni. Off-policy evaluation for slate recommendation. In *Advances in Neural Information Processing Systems*, pages 3632–3642, 2017.

R. Tibshirani. Regression shrinkage and selection via the lasso. *Journal of the Royal Statistical Society: Series B (Methodological)*, 58(1):267–288, 1996.

S. Wager and S. Athey. Estimation and inference of heterogeneous treatment effects using random forests. *Journal of the American Statistical Association*, 0(0):1–15, 2018. doi: 10.1080/01621459.2017.1319839.

S. Wang and G. F. Özkan-Seely. Signaling product quality through a trial period. *Operations Research*, 66 (2):301–312, 2018.

J. Yi, Y. Chen, J. Li, S. Sett, and T. W. Yan. Predictive Model Performance: Offline and Online Evaluations. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, pages 1294–1302. ACM, 2013.

H. Yoganarasimhan. Search personalization using machine learning. *Available at SSRN 2590020*, 2017.

Appendices

A Hyper-parameter Optimization for the Models Estimated

For each model that we use, we describe the hyper-parameters associated with it and the optimal hyper-parameters that we derive after tuning. In all cases, we use five-fold cross-validation to optimize the hyper-parameters. We then train a model on the entire training data using the optimal hyper-parameters, and report the performance of this model on both the training and test data.

- Least squares does not use any hyper-parameters and hence does not require validation. In this case, we simply train the model on the full training data to infer the model parameters and report the model's performance on both the training and test data.
- For lasso, the validation procedure is straightforward. The only hyper-parameter to tune is the L1 regularization parameter, λ. We use the standard cross-validation procedure implemented in the *glmnet* package in R. It searches over 98 different values of λ ranging from 1.8×10^{-5} to 1.5×10^{-1}, and picks the one that gives us the best out-of-sample performance (based on cross-validation). In our case, it $\lambda = 3.1 \times 10^{-4}$.
- For CART, we use the package *rpart* in R, which implements a single tree proposed by (Breiman et al., 1984). We only need to pick the complexity parameter (ζ) using cross validation in this case. We search over 3886 different values for ζ ranging from 8.6×10^{-11} to 1.7×10^{-1}, and derive the optimal complexity parameter as 5.4^{-5}.
- For Random Forest, we use the package *sklearn* in Python. There are three hyper-parameters in this case – (1) n_{tree}, the number of trees over which we build our ensemble forest, (2) $\max_f$, the maximum number of features the algorithm try for any split (it can be either all the features or the squared root of the number of features), and (3) $n_{\min}$, the minimum number of samples required to split an internal node.

  The standard method for finding hyper-parameters is grid-search. However, grid-search is very costly and time-consuming when we have to tune many hyper-parameters. So we use the hyperopt package for tuning the hyper-parameters in this model. Hyperopt provides an automated and fast hyper-parameter optimization procedure that is less sensitive to researcher's choice of searched hyper-parameter values; see Bergstra et al. (2011, 2013) for details.

  For each of these hyper-parameters, we now define the range over which we search as well as the optimal value of the hyper-parameter are shown below:
  - $n_{tree} \in [100, 1200]$ and $n^*_{tree} = 1000$
  - $\max_f \in \{n, sqrt(n)\}$ and $\max^*_f = n$
  - $n_{\min} \in [10, 300]$ and $n^*_{\min} = 70$
- XGBoost also has many hyper-parameters that need tuning. However, we found that our results is sensitive to only three of the parameters: α, η, and $d_{\max}$. The first parameter, α, is an L1 regularization parameter, η is the shrinkage parameter or learning rate, $d_{\max}$ is maximum depth of trees. Again, we use the hyperopt package to search over a wide range of parameter values. The optimal values are shown below:
  - $\alpha \in \{0.1, 0.2, 0.5, 1, 2, 5, 10, 15, 20, 25\}$ and $\alpha^* = 20$
  - $\eta \in [0, 1]$ and $\eta^* = 0.59$
  - $d_{\max} \in \{6, 8, 10, 12\}$ and $d^*_{\max} = 12$
- Causal tree has two hyper-parameters that needs tuning – (1) the complexity parameter (ζ) and the minimum number of treatment and control observations in each leaf (q). We use the cross-validation

procedure in the "causalTree" package in R for tuning ζ. We manually tune q using grid-search over the range $[100, 1000]$ in increments of 100. We search over all possible values of ζ for each q.

The optimal hyper-parameters for the three trees (one for each pair of treatments) are:

- The tree for 7 and 14 days pair: $\zeta = 1.8e-05$ and $q = 100$.
- The tree for 7 and 30 days pair: $\zeta = 8.0e-06$ and $q = 100$.
- The tree for 14 and 30 days pair: $\zeta = 3.0e-06$ and $q = 900$.

- Causal forest has five hyper-parameters that need to be tuned: (i) $frac$, the fraction of data that is used for training each tree, (ii) $mtry$, the number of variables tried for each split, (iii) max_imb the maximum allowed imbalance of a split, (iv) imb_pen, a penalty term for imbalanced splits, (v) q, the minimum number of observations per condition (control, treatment) in each partition.[16]

  We used the hyper-parameter optimization procedure available in the grf package for tuning these hyper-parameters. The optimal hyper-parameters for each model are shown below:

  - 7-14 days pair: $frac = 0.5$, $max\_imb = 0.11$, $imb\_pen = 2.03$, $mtry = 13$, and $q = 1651$.
  - 7-30 days pair: $frac = 0.5$, $max\_imb = 0.13$, $imb\_pen = 2.49$, $mtry = 1$, and $q = 4$.
  - 14-30 days pair: $frac = 0.5$, $max\_imb = 0.20$, $imb\_pen = 5.11$, $mtry = 7$, and $q = 121$.

[16]For more information please visit https://github.com/grf-labs/grf/blob/master/REFERENCE.md

B CATE Estimates from Casual Tree and Causal Forest

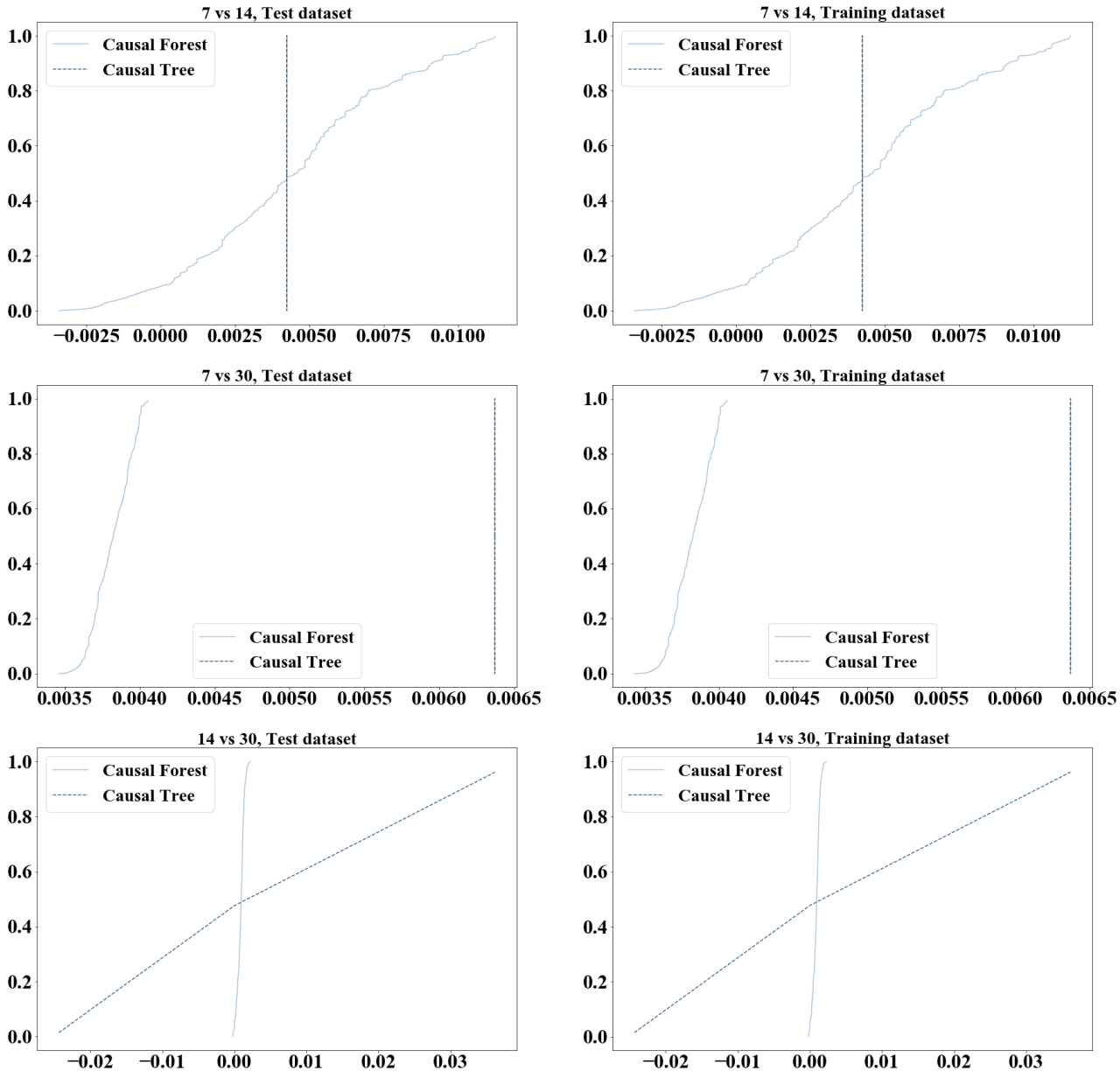


Figure A1: The CDFs of treatment effects for each pair of treatments from the causal tree and causal forest models (on training and test data).

C Paired t-tests

Policy Category	Policy	Subscription (%)		Subscription Length		Revenue	
		Mean	SD	Mean	SD	Mean	SD
One length for all	π_{30} (Baseline)	14.63	0.13	2.189	0.023	78.33	1.10
	π_{14}	15.10	0.28	2.317	0.053	84.83	2.42
	π_{7}	15.44	0.30	2.325	0.051	84.51	2.44
Based on outcome estimation	π_{reg}	15.33	0.28	2.309	0.049	84.50	2.34
	π_{lasso}	15.63	0.27	2.364	0.050	87.56	2.37
	π_{r_forest}	14.82	0.26	2.256	0.046	80.56	2.02
	$\pi_{xgboost}$	15.53	0.27	2.362	0.052	86.77	2.38
Based on CATE estimation	π_{c_forest}	15.46	0.30	2.319	0.051	84.11	2.39
	π_{c_tree}	15.44	0.30	2.306	0.053	84.32	2.42

Table A1: Mean and standard deviation of IPS reward estimations for subscription rate, subscription length (in months), and revenue (scaled) based on 1000 rounds of bootstrap samples drawn from the test data.

We now examine whether the rewards from the policies are statistically different from each other. A straightforward approach to do this is to run paired Student's t-tests. To do that, we generate 1000 bootstrap samples of our test data and estimate the IPS reward of each counterfactual policy on each sample for the three outcomes of interest – subscription rate, subscription length, and revenue (scaled). The summary statistics of these IPS rewards are shown in Table A1.

A key assumption necessary for the validity of the Student's t-test is for the data to be generated from a normal distribution. So we use the Shapiro-Wilk test (Shapiro and Wilk, 1965) and Anderson test (Stephens, 1974) to test the hypothesis that the distributions of bootstrap IPS reward estimates are normal. Neither of these tests could reject the null hypothesis that the IPS rewards for each of the counterfactual policies come from a normal distribution (even at a 10% confidence interval).

The results from the paired t-tests for subscription rate are shown in Table 9 in §7.3.2. We present the results for the paired t-tests for subscription length and revenue in Tables A2 and A3 below.

Compared Policy (π)	$\Delta\hat{R}(\pi_{lasso}, \pi)$	$\Delta\hat{R}(\pi_{xgboost}, \pi)$	$\Delta\hat{R}(\pi_7, \pi)$	$\Delta\hat{R}(\pi_{cf}, \pi)$
$\pi_{xgboost}$	0.19 (7.43 $\times 10^{-2}$)	—	—	—
π_7	3.90 ($\leq 10^{-100}$)	3.71 ($\leq 10^{-100}$)	—	—
π_{c_forest}	4.45 ($\leq 10^{-100}$)	4.26 ($\leq 10^{-100}$)	0.55 (6.88 $\times 10^{-13}$)	—
π_{14}	4.70 ($\leq 10^{-100}$)	4.51 (1.32$\times 10^{-93}$)	0.80 (3.31 $\times 10^{-4}$)	0.25 (1.29$\times 10^{-1}$)
π_{reg}	5.46 ($\leq 10^{-100}$)	5.27 ($\leq 10^{-100}$)	1.56 (6.64 $\times 10^{-22}$)	1.00 (7.58 $\times 10^{-12}$)
π_{r_forest}	12.03 ($\leq 10^{-100}$)	10.59 ($\leq 10^{-100}$)	6.88 ($\leq 10^{-100}$)	6.33 ($\leq 10^{-100}$)

Table A2: Comparison of reward estimates based on subscription length for the top six policies with other counterfactual policies (on test data). The difference numbers are multiplied by 10^2. For each cell, we show the mean difference and the p-values for the paired t-test in parenthesis. Note that we use $\Delta\hat{R}(\pi_a, \pi_b)$ to show $\hat{R}_{IPS}(\pi_a, Y) - \hat{R}_{IPS}(\pi_b, Y)$ in the header.

Compared Policy (π)	$\Delta\hat{R}(\pi_{lasso}, \pi)$	$\Delta\hat{R}(\pi_{xgboost}, \pi)$	$\Delta\hat{R}(\pi_{14}, \pi)$	$\Delta\hat{R}(\pi_7, \pi)$
$\pi_{xgboost}$	0.79 (1.99 $\times 10^{-38}$)	—	—	—
π_{14}	2.72 ($\leq 10^{-100}$)	1.94 (4.14$\times 10^{-87}$)	—	—
π_7	3.04 ($\leq 10^{-100}$)	2.26 ($\leq 10^{-100}$)	0.32 (1.67 $\times 10^{-3}$)	—
π_{reg}	3.06 ($\leq 10^{-100}$)	2.27 ($\leq 10^{-100}$)	0.33 (2.53 $\times 10^{-5}$)	0.01 (4.54 $\times 10^{-1}$)
π_{c_forest}	3.45 ($\leq 10^{-100}$)	2.66 ($\leq 10^{-100}$)	0.72 (1.22 $\times 10^{-11}$)	0.40 (1.00 $\times 10^{-24}$)
π_{r_forest}	7.00 ($\leq 10^{-100}$)	6.22 ($\leq 10^{-100}$)	4.28 ($\leq 10^{-100}$)	3.96 ($\leq 10^{-100}$)

Table A3: Comparison of reward estimates based on revenue (scaled by a fixed number) for the top six policies with other counterfactual policies (on test data). For each cell, we show the mean difference and the p-values for the paired t-test in parenthesis. Note that we use $\Delta\hat{R}(\pi_a, \pi_b)$ to show $\hat{R}_{IPS}(\pi_a, Y) - \hat{R}_{IPS}(\pi_b, Y)$ in the header.

D Appendix for §7.5

Variable	Value	Population	Policy Assigned Treatment		
			30 Days	14 Days	7 Days
Country	*United States*	54.9%	55.9%	45.8%	57.9%
	Germany	8.9%	8.2%	15.9%	6.6%
	Japan	7.9%	9.7%	11.5%	6.4%
	Other	28.3%	26.2%	26.8%	29.1%
	Total	100%	100%	100%	100%
Operating System	*Windows 10*	29.0%	9.0%	11.9%	37.0%
	Windows 7	21.5%	46.8%	21.7%	18.5%
	Windows 8.1	14.0%	1.0%	22.6%	12.7%
	El Capitan	13.9%	30.7%	17.9%	10.5%
	Yosemite	13.4%	3.3%	21.9%	11.7%
	Other	8.2%	9.2%	4.0%	9.6%
	Total	100%	100%	100%	100%
Skill	*Beginner*	68.9%	33.8%	41.1%	82.3%
	Experienced	12.8%	41.6%	22.0%	6.4%
	Mixed	10.7%	2.0%	35.0%	3.6%
	Unknown	7.5%	21.8%	2.0%	7.7%
	Intermediate	0.0%	0.1%	0%	0%
	Total	100%	100%	100%	100%
Job	*Student*	28.1%	12.7%	13.1%	34.9%
	Unknown	22.0%	69.8%	16.2%	18.5%
	Hobbyist	20.0%	7.3%	20.0%	21.5%
	Other	29.9%	10.2%	50.7%	25.1%
	Total	100%	100%	100%	100%
Signup Channel	*Website*	81.6%	65.2%	76.1%	85.3%
	App Manager	8.2%	17.7%	5.2%	8.1%
	Other	10.2%	17.1%	18.7%	6.6%
	Total	100%	100%	100%	100%

Table A4: Distribution of users' pre-treatment attributes for the three segments: those assigned to the 7-day condition, those assigned to the 14-day condition, and those assigned to the 30-day condition. (For each categorical variable, we show the fractions only for the top few categories in the interest of space.)

Variable	Population	Policy Assigned Treatment		
		30 Days	14 Days	7 Days
	Mean			
Subscription rate	14.8%	17.0%	29.1%	9.8%
Revenue	$536	$545	$546	$524
Retention (months)	16.1	17.3	15.9	16.1
Downloaded Products	1.15	1.14	1.20	1.13
Usage				
Total used functions	2,226	2,491	3,256	1,844
Distinct used functions	102	101	120	97
	Fraction			
Purchased Bundle				
Bundle I	55.8%	57.5%	55.4%	55.9%
All inclusive	21.5%	16.7%	21.6%	21.7%
Single Product	19.2%	20.2%	19.4%	19.8%
Other	3.5%	5.6%	3.6%	2.6%
Total	100%	100%	100%	100%
Subscription Type				
Commercial	79.1%	79.0%	81.0%	77.3%
Education	20.7%	20.6%	18.9%	22.7%
Other	0.2%	0.4%	0.1%	0%
Total	100%	100%	100%	100%

Table A5: Means and distributions of users' post-treatment attributes for the three segments: users assigned to the 7-day condition, users assigned to the 14-day condition, and users assigned to the 30-day condition.

E Appendix for §7.6

Trial Length	Dataset	Subscription Length						
		Mean	Std	Min	25%	50%	75%	Max
7 Days	Training	16.25	8.72	0	10	18	23	73
	Test	16.04	8.65	0	9	18	23	53
	All	16.19	8.70	0	10	18	23	73
14 Days	Training	15.95	8.47	0	9	17	22	67
	Test	16.39	8.48	0	10	18	22	56
	All	16.09	8.48	0	10	18	22	67
30 Days	Training	15.89	8.31	0	10	17	22	96
	Test	16.12	8.47	0	10	18	22	108
	All	15.96	8.36	0	10	17	22	108

Table A6: The summary statistics of the subscribed users' total months of subscription.

Trial	Subscribed Bundle					Subscription Type		
Length	1	All Inclusive	Single Product	4	5	Commercial	Education	Government
7 Days	55.24	22.02	19.44	1.68	1.62	78.57	21.30	0.13
14 Days	55.99	21.75	19.12	1.79	1.34	79.62	20.25	0.13
30 Days	55.98	21.63	18.95	1.82	1.62	79.02	20.84	0.13

Table A7: The fraction of each subscription bundle and type. We do not reveal the names of some of the bundles to preserve's the firm's anonymity.