Erasmus University Rotterdam

Erasmus School of Economics

Bachelor Thesis

BSc2 Econometrics and Economics

Design and Evaluation of Optimal Online Advertising Policies

Author:
Caspar de Bruyn

Student Number:
529969

Supervisor:
Hong Deng

Second Assessor
dr. Phyllis Wan

July 2, 2023

Abstract

Online advertising plays a huge role in present-day marketing, and personalization based on online user data has never been this easy. We use data from several incrementality experiments to examine the effects of personalized targeting policies on the website visit rates and conversion rates resulting from online advertising. We find that a personalized targeting policy based on a lasso model increases the website visit rates by 26.90% over showing no advertisements, although it does not significantly outperform a uniform policy of showing all users the advertisement. We also find that a policy which maximizes website visits performs fairly well on conversion rates, although optimizing over conversion rates directly does lead to better performance on this metric. Interestingly, a uniform policy showing advertisement to all users performs slightly better on maximizing conversion rates. Furthermore, we show that personalized policies are not always better than a uniform policy by comparing several machine learning methods, showing that designing and evaluating personalization policies should be done cautiously. Lastly, we compare two advertising cost models and give conditions under which model which policy performs better.

1 Introduction

When browsing the internet, there is no escaping from it: advertisements. In 2021 alone, worldwide expenditures on digital advertising reached 522.5 billion U.S. dollar, which includes advertisements on mobile, laptop, and desktop. This number is expected to rise to over 800 billion U.S. dollar in 2026 (Statista, 2023). These numbers are not that shocking, given the fact that in 2021 the average person spent 6 hours and 58 minutes a day on a device connected to the internet (Howarth, 2023). In addition to this, it is often way cheaper to advertise online, especially because it is easier to reach a large audience compared to using offline advertising. An added benefit of online advertising is the fact that through user accounts and the use of cookies, there is access to many user features and preferences, making it easier to target and personalize advertising. This paper focuses on the personalization of online advertisement targeting.

In online advertising, there are two common payment models. The first one is pay per click (PPC), where the company pays the publisher each time a user clicks on the advertisement. This payment model is most common in search engine advertising. This is an effective form of advertising as the advertisement is related to the search query of the user and therefore targeted at the user's interests. The other most common model is pay per impression (PPM). In this method, the advertiser pays the publishing service each time an advertisement is shown to a user (Goldfarb, 2014). In case we wish to personalize advertisement, the optimal personalization policy also depends on the payment model of advertisement. In a pay per click model, there is no direct marginal cost of showing more people the advertisement, but an indirect one by an increase in users who (potentially) click the ad. However, if the policy maximizes the click-through rate, it is desirable that this policy also increases the conversion rate. Otherwise, the marginal costs of advertisement will be higher than the marginal benefit. In a pay per impression model, there are direct marginal costs of showing the advertisement. It may therefore be of interest to not show the ad to users who are unlikely to click the ad, let alone purchase the product.

This raises the question for firms if it is effective to show all users of a certain online platform their advertisement, or if they should target only specific customers. Therefore, the main question this paper tries to answer is: *Is the effect of advertising heterogeneous across consumers, and if so, how can the assignment of advertisements be personalized based on consumer characteristics, and what are the gains from personalization?* Additionally, it is investigated if policies which optimize website visits also translate to higher conversion rates. Lastly, we assess what policies are more effective under different advertising cost models. This paper is a continuation of the paper by Yoganarasimhan et al. (2022) on the design of personalized policies on trial lengths for online software. This paper uses a similar approach, but applies it to a different context, namely online advertising. In addition to their methods, this paper deploys another method of designing personalized policies, as well as an additional evaluation score.

In order to answer our research questions, we adopt a three-step approach to design and evaluate personalized advertising policies. To this end, we use data of several incrementality tests where an arbitrary part of the population does not see any advertisements. We consider two outcome variables of interest, namely website visits, and actual conversion. Website visits are of interest as visits create brand awareness, and users might find products they wish to buy which were not initially shown in the advertisement. Brand awareness is an important asset for a company, as in the long run this has the potential to increase performance (Bernarto et al., 2020). Conversion rates are of importance as this is the main source of income for a firm.

First, we find that a uniform policy where all users are shown advertisements leads to an increase of 22.98% over not showing any advertisements at all. Next, we develop a 2-step approach of designing personalized policies. In the first step, we train a lasso (Least Absolute Shrinkage and Selection Operator) regression which models visit rates as a function of user features, treatment, and the interaction of these two. In the second stage, we use outcome predictions of this model in order to assign each user the optimal treatment. Afterwards, we use the Inverse Propensity Score (IPS) estimator to evaluate this policy, and compare this with a different off-policy evaluation estimator, namely the Doubly Robust (DR) estimator, which are popular in the counterfactual policy evaluation literature (Dudík et al., 2011).

We find that this personalized lasso policy increases expected visit rates by 26.90% over not showing any advertisements at all. It does however not significantly outperform the uniform policy of assigning all users treatment. However, it does assign only 32.93% of users to the treatment group, which could lead to a sharp decrease in advertising costs under a PPM model. This finding is in line with previous literature, where the effect of personalization in advertisement is shown to be only slightly positive (Rafieian & Yoganarasimhan, 2021). We also find that, in our context, the IPS estimates and DR estimates do not differ significantly, and both give a good estimate of the expected gain.

Next, we find that the policy which optimizes visit rates increases the conversion rates by 72.73% over not showing any advertisements, but it performs worse than the best uniform advertising policy. Also, the personalized policy which optimizes conversion rates does not outperform the best uniform policy, although the difference is moderate. Again, the personalized policy shows way less users the advertisement, namely only 24.99% of users, which could lead to lower costs.

Furthermore, we compare the performance of the lasso policy with several other popular machine learning techniques. We consider five other outcome estimators, namely 1) linear regression, 2) classification and regression trees (CART), 3) random forests, 4) XGBoost, and 5) Artificial Neural Networks (ANNs). In addition to this, we use two heterogeneous treatment effect estimators, namely 1) causal tree and 2) causal forests. We find that the policy based on lasso continues to perform the best, no matter the policy evaluation estimator we use, followed by the causal tree and linear regression (both around 4.45%). Interestingly, CART and XGBoost are unable to personalize policies at all, and therefore assign all users to the treatment group. Random forest and artificial neural network perform even worse than the best uniform policy, meaning we would be better off by not personalizing at all. The two recently popularized conditional average treatment effect (CATE) estimators causal tree and causal forest do perform better than the best uniform policy. This is however only by a slight amount, even though these methods were performed to specifically model treatment effects. Note that the findings are specific to the context of this paper, but still provide the managerial insight that firms should not naively pick a popular modelling technique and expect it to lead to better performance.

Lastly, we compare the lasso policy based on both outcomes with the uniform policy, and under which conditions and cost model each policy should be preferred. We find that the lasso policy optimized over conversion rates requires lower profit margins in order to be profitable than the policy optimized over visit rates under both a PPM and a PPC model. Moreover, we find that a personalized policy is often more profitable than a uniform one under the PPM model, but it is the other way around under the PPC model.

This paper contributes to existing literature by comparing many machine learning approaches in the field of policy design, with the addition of quickly popularized neural networks and heterogeneous outcome estimators. Furthermore, it compares two different policy evaluation estimators in a field experiment, whereas until now this has mostly been done in synthetic or simulated settings. From a managerial perspective, we provide a framework of designing and evaluating personalized policy design, which managers can use to design, evaluate, and implement personalization policies in advertisement targeting. Most importantly, we show that not all popular estimators perform well on policy design, and therefore the evaluation is the most critical step. We also show that policies which optimize over visit rates do not perform particularly poorly on conversion rates, and may therefore, based on the goal of the advertising campaign, may be preferred. Additionally, an analysis of which policy to deploy in which advertisement pricing model is given, which is useful for managers as we show that the optimal policy may differ between cost models.

In the remainder of this paper, we first discuss the theoretical background of our problem at hand. Next, we describe our data and the plan of approach. After this, the used estimators are discussed in detail. Then, we present and discuss the findings of the research. We end with a discussion on the limitations of the research and pathways for future research.

2 Theoretical Background

Advertising is one of the most important marketing strategies for any brand. Without any form of advertising, consumers might not know of the brands existence, let alone have any knowledge on the quality of the brand. Bagwell (2007) considers three possible effects of advertisement, namely the persuasive effect, the informative effect, and the complementary effect. The persuasive effect implies that advertisement leads to a change in taste of the consumer and leads to brand loyalty. The persuasive effects therefore means that advertisement has an anti-competitive effect, as it has no actual value to consumers, but creates artificial product differentiation. The informative effect means that advertising reduces the amount of imperfect consumer information, and therefore helps consumers choose the correct product or service. The complementary effect says, contrary to the persuasive effect, that advertisement does not directly alter the choice of the consumer. However, it assumes that consumers have a stable set of preferences, and that advertisements should directly appeal to those preferences. Especially this last effect indicates a need for personalized advertisement. If advertisements can directly enter a consumer's preferences, the likelihood of the advertisement being effective increases. This research focuses on the heterogeneous effect of online advertising. In the remainder of this section we discuss the goal, the design, and evaluation of personalization policies.

2.1 Personalization

The main reason we are interested in personalization policies in marketing, is the fact that it could lead to an increase in a company's performance. For instance, personalization leads to improvement of a customer's browsing or shopping experience (Adomavicius and Tuzhilin, 2005; Arora et al., 2008). Furthermore, it could lead to an increase in profits, customer lifetime value, and brand loyalty (Adomavicius and Tuzhilin, 2005; Arora et al., 2008). Anshari et al. (2019) show that personalization can lead to an increase in customer acquisition. Rossi et al. (1996) find that personalized couponing lead to an increased gain of 155% over a uniform couponing policy, and Ansari and Mela (2003) find an increase of 62% in click-through probabilities when sending customized e-mails to potential customers. De Keyzer et al. (2015) show that personalizing advertising on only gender already increased the click through intention of users when shown advertisements on Facebook. Becker et al. (2017) find that personalizing the channels through which users are shown advertisements increases the conversion probability of customers. However, Kim et al. (2022) show that when advertisements became too personalized, there was a negative effect on the attitude of users towards a brand and the advertisements, meaning that making advertisements too personal might have negative consequences for the brand. In addition to this, users often value their privacy while also wanting personalized services, the so-called privacy-paradox (Barth & De Jong, 2017). Malheiros et al. (2012) find that greater personalization of ad content almost doubles the time that users spent looking at the advertisement, but at the same time also increased discomfort experienced by the user by 80%. This implies that when designing a personalization policy, the selection of the correct user features to personalize on should be carefully selected. Goldfarb and Tucker (2011) find that the targeting in advertising is effective, but this effect is limited. This is supported by the findings of Yoganarasimhan (2020) and Rafieian and Yoganarasimhan (2021), who find positive but only slight gains from personalized digital interventions. This paper contributes to the existing literature on personalization by discovering methods which select the futures to personalize on, and therefore implicitly try to minimize discomfort of users while still forming an effective advertising model. Kietzmann et al. (2018) note that the use of artificial intelligence could improve the targeting policy of advertisers, making sure the right user is shown the right advertisement, while using information as efficient as possible.

2.2 Design of Personalized Policies

The design of a personalized policy is a complex task on its own. The amount of customer data available is increasing every day, and deciding which variables are relevant for the design of the policy is not straightforward, yet important. Therefore, methods which select these variables are popular in the field of policy design. Here, regularization methods like lasso or Elastic-net, and tree-based methods like classification and regression trees (CART) and random forests are often used. In the two following sections, the design and evaluation of personalized policies are described.

A personalized policy prescribes each user a treatment which maximizes an outcome of interest. However, a major issue in designing personalized policies is the fact that in an experiment there is only an observed outcome per individual under one of the treatments, meaning that it is impossible to directly compare the effect of treatment(s) for the same individual. Therefore, often a two-step approach is used in order to design the optimal personalized policy. In the first step, a model is estimated and used to predict the outcome variable under each treatment as accurate as possible given and individual's characteristics. In the second step, this model is used to assign each individual the treatment which would lead to the highest possible value of the outcome variable (Guelman et al., 2015). This two-step model is often used in combination with machine learning based models like regression trees, random forests, and variations on these methods (like boosted trees) (Athey and Imbens, 2016; Wager and Athey, 2018; Yoganarasimhan, 2020). In addition to this, there are also methods based on k-nearest neighbours, causal trees, and causal forests (Wager & Athey, 2018). These methods aim to first split all individuals in sub-groups where individuals within a sub-group are roughly the same (based on characteristics in the case of k-nearest neighbours, or based on the CATE for causal trees and forests). Then, within each sub-group the average treatment effect is estimated, which can be interpreted as the CATE for these individuals. The policy based on these methods assigns each individual to the treatment which gives them the highest CATE.

A method not yet used often in marketing, and especially not in the design of personalized policies, are ANNs. In this research, an ANN will also be used to this end, and therefore contributes to existing literature. ANNs come with both pros and cons, which will now be discussed.

The first advantage of ANNs is that they do not require any pre-existing knowledge on the model specification (Heinrichs & Lim, 2008). Additionally, the standard regression assumptions need not be satisfied, meaning it can also handle nonlinear models, nor do they need any assumptions on the distribution of errors and covariates (Gorr et al., 1994). Furthermore, there is no issue of multicollinearity (DeTienne & DeTienne, 2017). Next, ANNs can easily adjust to new information added without having to be re-estimated as a whole, making them effective to use in dynamic contexts (Rumelhart et al., 1994).

However, there are some downsides to using ANNs. The first one being the fact that they are somewhat of a black box, meaning that model parameters are not easily interpretable. They are therefore effective for predicting outcomes, but not so much for understanding mechanisms (Stern, 1996). Also, when the standard regression assumptions are met, and the model is correctly specified, standard linear analysis tends to outperform ANNs. In this case, regression models are preferred, as the relationship between input and output variables can be interpreted, which is where ANNs are lacking (Warner & Misra, 1996). However, as consumer behaviour is often unpredictable and of non-linear nature, it is quite unlikely that in practice all these conditions are met (DeTienne & DeTienne, 2017).

2.3 Off-Policy Evaluation

Nowadays, policy evaluation is a quintessential part of impact evaluation and governmental policy making. It arose from the idea that more rational policy design could be achieved by critically and analytically evaluating a policy, especially before the actual policy is implemented. Whereas systematic policy evaluation was first mostly used for regulatory policies, it later became popular in the field of environmental impact evaluation (Adelle & Weiland, 2012). This type of ex ante policy evaluation is important, as the implementation of these policies is expensive, especially when the desired result is not achieved. For this reason, off-policy evaluation has recently became more and more popular in the field of medicine, as giving the wrong medicine to a person can have detrimental consequences (Hanberger, 2001). Since the amount of available data is increasing every day, data-driven policy evaluation, often based on machine learning methods, has gained traction. Bertot and Choi (2013) claim that the use of big data in policy making and evaluation can have large positive impact on these policies. Later, the concept of policy-evaluation was also found in the evaluation of advertising policies, for instance in Lambert and Pregibon (2007) and Chan et al. (2010).

As mentioned in Section 2.2, we only observe outcomes for each individual under one treatment, and therefore use a 2-step approach to find the optimal policy. In order to estimate the gains from this policy, without observing an individual under all treatments, off-policy evaluation estimators are used. Commonly used estimators are the Direct Method (DM) and the IPS estimator. The first method averages the estimated reward per individual when assigned treatment by a given policy. The IPS estimator makes an approximation of the fraction of the population which receives a certain treatment, the propensity score, and uses this estimate to re-scale rewards from the policy. See section 4.2 for further mathematical details. Both methods require some assumptions in order to be unbiased. DM requires that the estimated outcome under the policy needs to be unbiased. However, in practice it turns out that this is often not the case. The estimates of the rewards under the policy are made without prior knowledge of the policy, and might therefore not be accurate (Beygelzimer & Langford, 2009). The IPS does not require this assumption to be unbiased. However, in order for the IPS to be accurate we require that the propensity scores are accurate. In practice, this assumption is more likely to hold than the unbiasedness assumption for DM. However, the IPS estimator tends to have a higher variance than the DM estimator (Dudík et al., 2011; Dudík et al., 2014).

Dudík et al. (2011) therefore suggest using the DR estimator. This estimator combines the previous methods into one estimator. Again, the exact mathematical details are in section 4.2. In order to be unbiased, this estimator requires only one of the two estimates, the expected reward or the treatment probability, to be unbiased. In general, the variance of the DR estimator is larger than the one of the DM estimator, but smaller than the variance of the IPS estimator (Dudík et al., 2011; Dudík et al., 2014). These papers also find that the DR estimator tends to estimate expected rewards more accurately than the IPS estimator.

3 Data Collection and Description

The data used is the CRITEO-UPLIFT1 data set as used by Diemert et al. (2018). The data was collected from several incrementality tests, where an arbitrary sub-sample of the population is prevented from being targeted from advertisement. The full data set has about 25 million observations, with a treatment ratio of 84.6% However, the set has been randomly sub-sampled in order to remain privacy, and to make the analysis computationally feasible. This sub-sample consists of 50,000 observations with a treatment ratio of 85.1%. Such high treatment ratios are quite regular in related research, as it can be quite costly to a company to not show any advertisements. For each user we observe the following: twelve user features, a binary treatment variable, and two binary outcome variables, one indicating whether the user visited the advertiser's website during the test period, the other indicating if the user converted during the test period. The test period was two weeks long. In order to remain privacy, the user feature names are anonymized, and values are projected randomly. This to ensure that it is practically impossible to recover the initial features, while retaining predictability. The summary statistics of the covariates are presented in Table 1.

Table 1 Mean values of user features and outcome variables.

	Full Sample	**Control Group**	**Treatment Group**
f0	19.642	19.598	19.650
f1	10.069	10.067	10.070
f2	8.446	8.448	8.446
f3	4.183	4.223	4.176
f4	10.336	10.332	10.337
f5	4.030	4.044	4.028
f6	-4.147	-4.009	-4.171
f7	5.102	5.064	5.108
f8	3.934	3.935	3.934
f9	15.989	15.833	16.016
f10	5.334	5.332	5.334
f11	-0.171	-0.170	-0.171
N	50,000	7447	42553

In terms of outcome variables, *visit* and *conversion* are binary variables, where a value of 1 means the user visited the website/was converted. The summary statistics of the outcome variables can be found in Table 2.

Table 2 Summary Statistics of Visit and Conversion Outcomes.

Variable	Mean	Std. Dev.	Min.	25%	50%	75%	Max.	N
Visit	0.046	0.209	0	0	0	0	1	50,000
Conversion (all)	0.003	0.051	0	0	0	0	1	50,000
Conversion (visitors)	0.058	0.233	0	0	0	0	1	2294

The data set will be split into a training set and a test set, using a 70-30% split. The training set will be used for model estimation, as well as hyper-parameter optimization. The test set will be used for model evaluation of policy performance. Note that the joint distribution of variables between the two sets should be the same in theory due to the randomization, but in practice might differ slightly. For our approach to work, we need the two sets to follow the same distribution (Simester et al., 2020), which we will assume throughout this paper.

The data and code used for the analysis can be fount in the code package. A short description of the code can be found in Appendix C. A more extensive description is found in the *readme.txt* file in the code package.

4 Methodology

4.1 Optimal Policy Design

In order to design the optimal policy, we make use of the lasso estimator. Let $i \in \{1, \ldots, N\}$ be the set of users, where each user has characteristics described by covariate vector $X_i \in X$. Next, $W_i \in \mathcal{W}$ denotes the treatment individual i receives. The set of W possible treatments is given by $\mathcal{W} = \{0, \ldots, W-1\}$. Lastly, let $Y(X_i, W_i)$ be the outcome variable, dependent on covariates X_i and treatment W_i. In this paper, the covariate vector X_i consists of the 12 user features. Treatment $W_i \in \{0, 1\}$, where 0 means no advertisement was shown. Lastly the outcome variable Y is either the visit or conversion variable.

Define a personalized treatment assignment policy, π, as a mapping from a users characteristics to one treatment, $\pi : X \to \mathcal{W}$. The set of all possible policies is given by Π. The objective is to find such a policy π which maximizes a reward function defines as the expected outcome, $R(\pi, Y) = \frac{1}{N}\sum_{i=1}^{N} \mathbb{E}\left[Y\left(X_i, \pi(X_i)\right)\right]$. For a given reward function, the optimal policy π^* is then given by

$$\pi^* = \arg\max_{\pi \in \Pi} \left[R(\pi, Y)\right]. \tag{1}$$

In order to find the optimal personalized policy, a 2-step approach is used. For this approach, we need three standard assumptions, namely 1) unconfoundedness, 2) Stable Unit Treatment Value Assumption, and 3) positivity. Given proper randomization, assumptions 1 and 2 are likely satisfied, although unconfoundedness is not guaranteed by randomization (Sävje, 2021). In addition to this, assuming no communication between users, assumption 2 is also satisfied. Assuming all users had a probability of being put in either the treatment or control group, the assumption 3 is also satisfied. Further discussion on the unconfoundedness assumption is found in section 4.2.

Now define $f(x, w) = \mathbb{E}\left[Y | X_i = x, W_i = w\right]$, where $f(\cdot)$ is a lasso model. The set of parameter estimates in the model is then given by:

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

where λ is a hyper-parameter to be selected using 5-fold cross-validation with the *glmnet* package in R. Furthermore, $\|\beta_j\|_1$ is the L1 norm of the vector β_j. A desirable property of lasso is that when several predictors are (weakly) correlated, it picks a subset of predictors, and sets the others to zero. The model in Equation (2) contains a total of 25 explanatory variables, namely the treatment variable W, the twelve covariates X and the interaction between these two, XW. With this many variables, the selection property is desirable, as otherwise it is likely that the model would overfit on the training data.

The second step is to estimate the expected outcome $\hat{y}\left(x = X, w\right)$ using the lasso model. We then obtain the optimal personalized policy for individual i as follows:

$$\pi_{lasso}\left(X_i\right) = w^*, \quad \text{where } w^* = \arg\max_{w \in \mathcal{W}} \hat{y}\left(x = X_i, w\right). \tag{3}$$

4.2 Policy Evaluation

In order to evaluate and quantify the gains of the personalization policy, the policy π_{lasso} is compared to two uniform policies, namely π_0, and π_1. Here, π_0 is the policy with no advertising, and π_1 denotes a policy where all users are shown the advertisement. We take π_0

as the baseline policy. We evaluate the estimated reward from the policies using two different estimators, namely the IPS and the DR estimator.

The first one is given by:

$$\hat{R}_{IPS}(\pi, Y) = \frac{1}{N}\sum_{i=1}^{N}\frac{\mathbb{1}\left[W_i = \pi\left(X_i\right)\right]Y_i}{\hat{e}_{X_i}\left(W_i\right)}, \tag{4}$$

where $\hat{e}_{X_i}(W_i)$ is the propensity score, namely $\Pr(W_i = w_i|X_i)$, where w_i is the treatment received by user i. Given that a study is unconditionally unconfounded, the propensity score is the same for the whole population, and we can estimate it using the treated fraction, or the treatment assignment probabilities if the assignment model is known (Sävje, 2021). However, this assumption needs to be tested. We regress treatment on covariates of all users in order to test if the treatment is independent of user features. We reject the null hypothesis of features being jointly insignificant at a 5% level ($p = 0.033$), and therefore conclude treatment is likely not independent of user features. Hence, the propensity scores are not equal for the entire population. Therefore, we model the propensity scores as a function of the user features, making the study conditionally unconfounded. The propensity scores are calculated by regressing treatment on the covariates X_i. As a sanity check we verify if these estimated propensity scored lie between 0 and 1, which is the case.

Next, the DR estimator is given by the following:

$$\hat{R}_{DR}(\pi, Y) = \frac{1}{N}\sum_{i=1}^{N}\frac{\mathbb{1}\left[W_i = \pi\left(X_i\right)\right]}{\hat{e}_{X_i}\left(W_i\right)}\left(Y_i - \hat{Y}_{W_i}\right) + \hat{Y}_{\pi(X_i)}, \tag{5}$$

where $\hat{Y}_p$ is the expected reward Y_i given treatment p.

As discussed in section 2.3, the main difference in performance between the two estimators is in the assumptions needed for unbiasedness. The IPS estimator requires the propensity scores to be accurate, whereas the DR estimator requires either the propensity scores or the estimates $\hat{Y}_p$ to be accurate in order to be unbiased.

4.3 Model Comparison

In addition to the lasso model as described in Section 4.1, Several other models will also be considered as a comparison of performance. This will be done using five outcome estimators, namely 1) linear regression, 2) CART, 3) random forests, 4) XGBoost, and 5) ANNs. Furthermore, two additional CATE estimators are considered, namely 1) causal trees and 2) causal forests. All these methods will be used to design policies, and are evaluated based on their estimated visit rate. Note that the DR estimator uses the predicted outcome in its calculation, and therefore this estimator can not be used in combination with the CATE estimators, as these do not make a prediction of the outcome variable. The mathematical details of these methods will now be discussed in more detail.

Linear model

The linear regression model is estimated using Ordinary Least Squares (OLS) in the following model:

$$Y_i = X_i\beta_1 + W_i\beta_2 + X_iW_i\beta_3 + \varepsilon_i. \tag{6}$$

A linear regression needs no parameter tuning, and can therefore be estimated directly. The interpretation of the coefficients is the same as for the lasso model. Linear regression models are, given the standard OLS assumption, unbiased. However, they often have poor out-of-sample performance due to the large number of variables, leading to higher variance and possible overfitting. As out-of-sample performance is of high importance when creating personalized policies, this method will likely perform poorly.

Tree-based Methods

Tree-based methods partition the covariate space into subregions recursively. This is done based on a certain criterion, mean squared error in our case. After the partitioning is done, the average value of the outcome variable in a region is the predicted value for that region. A CART model can be represented as follows:

$$y = f\left(x, w\right) = \sum_{k=1}^{K} \rho_k \mathbb{1}\left(x, w \in R_k\right), \tag{7}$$

where K is the total number of subregions, R_k is the k^{th} subregion, and ρ_k is the predicted value in that region. The CART model has one hyper-parameter that requires tuning, namely the complexity parameter ζ. This is a penalty term on the number of terminal nodes of the tree, used to prune a large tree (James et al., 2013). This parameter will be tuned using GridSearchCV from the sklearn API (Pedregosa et al., 2011) in python.

CART models are simple and easy to interpret, but often have poor performance compared to other supervised learning approaches due to high prediction variance, as well as them being quite sensitive to outliers (James et al., 2013). Therefore, there are methods to improve the performance of these regression trees, namely bagging, random forests, and boosting. We will discuss these methods next.

The idea of bagging is the process of growing multiple regression trees based on several distinct training sets, and then averaging the predicted outcomes of all these trees to form a final prediction. These trees are grown deep, yielding low bias but high variance. Averaging out predictions reduces this variance. As there is often no access to more than one training set, these training sets are generated using bootstrapping.

The next variance reduction method for regression trees is the random forest regression. Random forests work in a similar way to bagging, but instead of bootstrapping samples, each tree in a random forest is grown using a subset of observations and predictors of the original data set. Therefore, each tree is different. The final prediction is again made by averaging out the prediction of each tree in the forest. A random forest has three hyper-parameters, namely the number of trees n_tree, the maximum number of features the alogrithm tries at each split max_f, and the the minimum number of samples required to split a node n_min. These parameters are tuned using the hyperopt package in Python.

The last tree-based method considered is boosting. Boosting is performed by first fitting a shallow tree. The residuals of this shallow tree are calculated, and then a new tree is grown based on these residuals instead of on the actual outcomes. In the end, all these trees are added to form a final model. Boosting is different from the last two methods, as each time a new tree is grown, this tree depends on the last tree. A general boosting algorithm is given by Algorithm 1.

Algorithm 1 Regression Tree Boosting

1. Let $\hat{f}(x) = 0$ and $r_i = y_i$ for all i.
2. for $b = 1, 2, \ldots, B$, do:
   (a) grow tree $\hat{f}^b$ with d_max number of splits to the data (X, r)
   (b) $\hat{f}(x) = \hat{f}(x) + \eta \hat{f}^b(x)$
   (c) $r_i = r_i - \eta \hat{f}^b(x_i)$
3. $\hat{f}(x) = \sum_{b=1}^{B} \eta \hat{f}^b(x)$

In this paper we use XGBoost, which was proposed by Chen and Guestrin (2016). XGBoost requires tuning of six parameters, but often shows to only be sensitive to few of them (Yoganarasimhan et al., 2022). We therefore optimize over the L1 regularization parameter on the loss function α, the shrinkage paramater or learning rate η, the L2 regularization parameter on the weights λ, and the maximum depth of trees d_max. These are also tuned using hyperopt in Python.

CATE Estimators

The next class of models we consider are the CATE estimators. Instead of predicting an outcome for each observation, these methods predict the heterogeneous treatment effect for each individual, and use these estimates to assign treatment. In our setting this heterogeneous treatment effect can be defined as

$$\tau(x) = \mathbb{E}\left[Y(X_i, W_i = 1) - Y(X_i, W_i = 0) | X_i = x\right]. \quad (8)$$

A policy assigns treatment $W_i = 1$ to a user i if and only if $\tau(x_i) > 0$. However, when the sample space is finite and the covariate space is large, we often do not have enough observations with the same X_i to estimate the precise treatment effect. Therefore, we use methods which pool observations whose covariates are close to each other in the covariate space, and estimate the heterogeneous treatment effects for these pooled observations. The main challenge in these methods is the way of chosing the sub-samples. If the sub-samples are chosen on a covariate sub-space that is too small, there will not be enough observations to accurately estimate the treatment effect for each sub-sample. Contrarily, if the covariate sub-space in each pool is to large, the model will not capture enough heterogeneity. Therefore, data-driven approaches for finding the best covariate sub-spaces based on machine learning methods have been developed, of which we will discuss causal trees and causal forests.

The causal tree method is similar to CART, except that it uses a different splitting criterion. Whereas CART uses the MSE as a splitting criterion, causal trees split based on similarity of treatment effects within each partition. Athey and Imbens (2016) find that maximizing the variance of the treatment effect within each partition, with an added complexity penalty term, achieves this goal. The objective function to be maximized than becomes $Var\left[\hat{\tau}\left(X\right)\right] - \zeta T$, where T is again the number of terminal nodes. In addition to ζ, causal trees have the additional hyper-parameter q, indicating the minimum of treatment and control observations in each leaf. ζ is tuned using the CausalTree package in R and q is found by manual grid search.

Next, we consider causal forests or the generalized random forest. This algorithm combines the ideas of causal trees and predictive random forests. Again, each tree is built on a random sub-sample of observations and covariates. Every time a split is made, the treatment effect in the parent leaf P is estimated by minimizing the R-learner objective function, given by:

$$\hat{\tau}_P(\cdot) = \arg\min_{\tau} \left[\frac{1}{n_P} \sum_{i=1}^{n_P} \left(\left(Y_i - \hat{m}^{(-i)}\left(X_i\right) \right) - \left(W_i - \hat{e}^{(-i)}\left(X_i\right) \right) \hat{\tau}\left(X_i\right) \right)^2 + \Lambda_n(\tau(.)) \right]. \quad (9)$$

Here, n_P is the number of observations in the parent node, $\Lambda_n(\tau(.))$ is a regularization term which decides the complexity of the model. $\hat{m}^{(-i)}$ is an estimate of the outcome, and $\hat{e}^{(-i)}$ is the estimate of the propensity score. Here $-i$ denotes the set of out of bag observations. The next split is chosen by maximizing the following function:

$$\frac{n_L \cdot n_R}{(n_P)^2} \left(\hat{\tau}_L - \hat{\tau}_R\right)^2, \quad (10)$$

where n_L and n_R are the number of observations in the nodes left and right of the split respectively. In the second step of the algorithm, a weighted kernel regression is performed to estimate the treatment effect at each point x using weights $\alpha_i(x)$:

$$\hat{\tau}(x) = \frac{\sum_{i=l}^{N} \alpha_i(x) \left(Y_i - \hat{m}^{(-i)}\left(X_i\right)\right) \left(W_i - \hat{e}^{(-i)}\left(X_i\right)\right)}{\sum_{i=l}^{N} \alpha_i(x) \left(W_i - \hat{e}^{(-i)}\left(X_i\right)\right)^2}, \quad (11)$$

where $\alpha_i(x)$ is the frequency with which the i-th training sample ends up in the same leaf as x in the first step.

Causal forest has five tunable hyper-parameters, namely 1) frac, the fraction of the data used when training each tree, 2) max_imb, indicating the maximum imbalance allowed in every split, 3)n_min, the number of samples required to split a node, 4) mtry, the number of considered covariates in each split, and 5) q, the minimal number of observations in each partition from both the treatment and control group. These parameters are tuned using CausalForestDML in python.

Artificial Neural Networks

Artificial neural networks are based on biological neural networks in the brains of animals. A neural network has several layers which consists of nodes called artificial neurons. The neurons are connected between layers through edges, through which signals with information are transmitted, similar to the synapses in the human brain. In a neural network, these signals are real numbers. See Figure 1 for a simple example of an ANN.

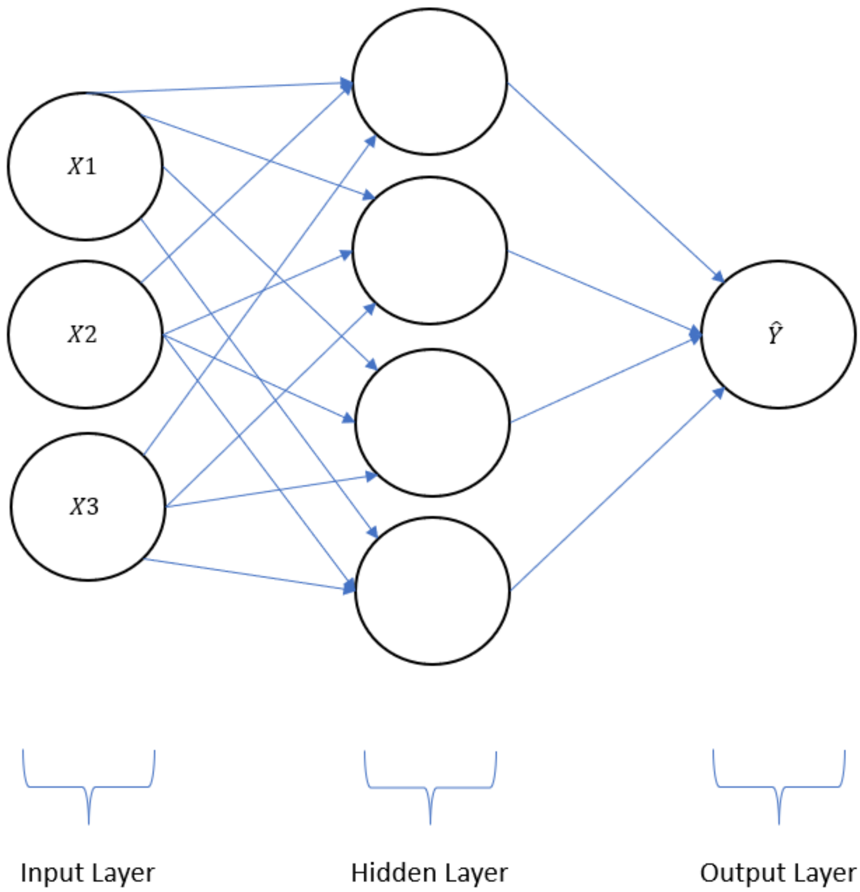

Figure 1 Simple representation of a two-layer neural network

In the first layer, or input layer, all explanatory variables are inserted into the model. Then, in each node of the hidden layer a weighted sum of the explanatory variables is calculated, to which then a transformation is applied. In other words, in each node j in the hidden layer the following operation is applied:

$$\phi\left(\sum_{i=1}^{m} w_{ij}x_i\right) \tag{12}$$

Where x_i is the i-th explanatory variable, and w_{ij} is the weight of the signal of from the i-th explanatory variable to the j-th node in the hidden layer. Next, the output of the nodes in the hidden layer is passed to the output layer, where again a transformation of the weighted sum of these signals is calculated, which then gives the final prediction.

The transformation function $\phi(\cdot)$ is called the activation function. The discussion on activation functions will be based on the paper by Sharma et al. (2017). The most common activation functions are:

1. Binary step function ($\phi(x) = 1$ if $x \geq 0,\ 0$ otherwise)
2. Linear function ($\phi(x) = ax$ with a some constant)
3. Sigmoid function ($\phi(x) = \frac{1}{1+e^{-x}}$)
4. Tanh function ($\phi(x) = \frac{e^{2x}-1}{e^{2x}+1}$)
5. Rectified linear unit (ReLU) function ($\phi(x) = \max(0, x)$)

Of these functions, the binary step function is most suited for binary outcome variables. However, since we do not wish to estimate a 0/1 outcome when designing the policies, but wish to find a continuous prediction, we can not use this function. The linear function is useful when an intepretable model is desired, however that is not the aim of our study. Additionally, it is not good in modelling complex patterns. The sigmoid and tanh function are smooth functions that have outputs from 0 to 1 and -1 to 1 respectively. The ReLU is the most used function, as it makes sure not all neurons are activated at the same time, increasing efficiency. We make use of the ReLU function in the hidden layer, and the sigmoid function in the output layer such that we get output between 0 and 1.

The weights of the model are learned by optimizing a cost function. One of the most common cost functions is the MSE, but other cost functions can be used as well. In this paper we use the MSE as cost function. There is no closed form solution for the weights, and they are found using gradient descent. Gradient descent is an optimization method where in each iteration the parameters in the model are updated using the gradient of the cost function. In other words, in each iteration the parameters are updated as follows:

$$\theta_{n+1} = \theta_n - \eta \nabla C(\theta_n), \tag{13}$$

where θ is the vector of parameters, η is the learning rate, and $C(\cdot)$ is the cost function. When the data set used to train the model is large, this procedure can take a long time. Therefore, often variations of gradient descent are used to reduce computation time, as well as decrease the variance of parameter updates (Ruder, 2016). These versions use either only one observation per iteration (Stochastic Gradient Descent) or a subset of the observations (Mini-Batch Gradient Descent). In this paper we make use of Mini-Batch Gradient Descent. Once the algorithm has run through the whole data set once, this is called one epoch. We use 100 epochs.

There are several hyper-parameters in an ANN model. First, there is the number of hidden layers n_layers. As we do not have that complex of a model, one hidden layer should suffice. Next, the number of neurons in the hidden layer n_nodes needs to be chosen. We make use of a rule-of-thumb method suggested by Karsoliya (2012) of selecting the number of neurons, namely that the number of neuron should be around two thirds of the number of input nodes, and between the number of input and output nodes. We therefore try values around eight neurons in the hidden layer, and find that seven nodes in the hidden layer leads to the lowest in-sample MSE, but eight nodes gives the lowest out-of-sample MSE. As mentioned earlier, out-of-sample performance is crucial in policy design, and we therefore choose for 8 neurons in the hidden layer. Lastly, we have the learning rate η and the mini-batch size b_size. For this we use the standard options in TensorFlow in python, which are 0.01 for η, and 32 for b_size.

Table 7 in Appendix A gives an overview of all the models, the hyper-parameters that need tuning, and the tuning method. Appendix A also gives the parameter values searched over, and the final parameter values.

5 Results

The expected visit rates of the two uniform policies and the personalized policy based on lasso are presented in the top two panels of Table 3.

Table 3 Gains in website visits using different counterfactual advertisement policies, based on the IPS estimator.

Policy Category	**Policy**	**Estimated Visit (%)**		**Increase in Visit (%)**		**Prescribed Treatment (%)**	
		Training Set	Test Set	Training Set	Test Set	Training Set	Test Set
Personalized based on lasso	π_{lasso}	4.87	4.52	22.98	26.90	33.48	32.93
Uniform	π_0	3.96	3.56	-	-	0	0
	π_1	4.85	4.38	22.24	22.98	100	100
Personalized	π_{reg}	4.90	4.44	23.63	24.73	39.25	39.30
	π_{CART}	4.85	4.38	22.24	22.98	100	100
	π_{rf}	4.72	4.28	19.01	20.10	31.48	31.54
	$\pi_{XGBoost}$	4.85	4.38	22.24	22.98	100	100
	π_{ANN}	4.34	4.32	9.43	21.44	25.61	25.67
	π_{c_tree}	5.11	4.47	29.09	25.59	28.11	27.80
	π_{c_forest}	4.82	4.40	21.63	23.63	99.62	99.58

First, we see that assigning all users to the treatment group (π_1) leads to an increase in estimated website visits of about 23% in the test set. Furthermore, the personalized policy based on lasso π_{lasso} leads to an even bigger increase in website visits, namely of 26.90%. In order to assess if these differences are significant, the full data set is repeatedly split into a 70%-30% train-test split, for a total of 30 times. Then, for each split the lasso model is estimated, and the IPS rewards for the three policies is calculated. Finally, a two-sample paired t-test is performed to assess the significance of these differences. The paired t-test between the two uniform policies has a t-statistic and p-value of -14.7 and 0.000 respectively, rejecting the null-hypothesis of equal means. Therefore, we conclude that π_1 significantly outperforms π_0. Next, the paired t-test between π_{lasso} and π_1 has a t-statistic of 0.008 with a p-value of 0.999. This means that we do not reject the null-hypothesis, and cannot conclude that one policy significantly outperforms the other. This is in line with the findings of the recent literature on personalized advertising (Goldfarb and Tucker, 2011; Yoganarasimhan, 2020; Rafieian and Yoganarasimhan, 2021), which also shows positive but relatively small gains form personalization.

5.1 Robustness Checks and Comparisons

As mentioned, the IPS estimator needs the assumption of accurate propensity scores in order to be unbiased. We therefore also compute the DR estimates and compare these to the IPS rewards. Note that we can not make DR estimates for the uniform policies, as these do not make any predictions on the outcome variable. The DR estimate of the lasso policy is shown in the top panel of Table 4.

Table 4 Gains in website visits using different counterfactual advertisement policies, based on the DR estimator.

Policy	**Estimated Visit (%)**		**Increase in Visit (%)**	
	Training Set	Test Set	Training Set	Test Set
π_{lasso}	5.17	4.78	30.42	34.18
π_{reg}	4.84	4.38	22.21	23.15
π_{CART}	4.83	4.36	21.99	22.55
π_{rf}	4.90	4.09	23.70	14.83
$\pi_{XGBoost}$	4.85	4.38	22.25	22.94
π_{ANN}	4.59	4.30	16.04	20.84

Note. The increase percentages are compared to the IPS estimate of the policy π_0. For the uniform policies, and the policies based on CATE estimators we cannot make a DR estimate as these do not have outcome estimates.

We see that the DR estimator has slightly higher values than the IPS estimates, namely 5.17% and 4.78% compared to 4.87% and 4.52%. We again use the same testing procedure to assess the significance of the difference between the two estimators, and find a t-statistic of -1.320 with a p-value of 0.197, concluding that the two estimators do not differ significantly. We also test if the DR estimate for the policy π_{lasso} differs significantly from the IPS estimate for the uniform policy π_1. The t-statistic equals 0.723 with a p-value of 0.478. We again conclude that the lasso policy does not significantly outperform the uniform policy of assigning all users treatment.

Next we look at the other machine learning methods and see how these perform. The IPS results are in the bottom panel of Table 3, and the DR estimates in the bottom panel of Table 4. First, based on the IPS estimates, we notice that the polcies based on CART and XGBoost have the exact same estimates as the uniform policy π_1. This is because these methods were unable to create splits on the treatment variable, and only created subregions based on the user features. Therefore, the best policy is the one who assigns all users to the treatment group. We also see that the causal forest policy assigns almost everyone to the treatment group. All other methods assign approximately 30% of the users to the treatment group, and the linear regression model about 40%. Next, we find that the policies based on linear regression and causal trees perform better than the lasso policy in the training set, but perform worse on the test set. This is likely due to these methods having low bias, and therefore good in-sample fit, but have high variance and thus worse out-of-sample fit. Another interesting thing to notice is that the random forest policy has quite a bit worse performance than the lasso policy, but the MSE values are lower for the random forest model (see Appendix B), and several other estimators show similar relationships. We therefore conclude that there is little correlation between predictive performance and policy performance. Lastly, we see that the addition of a neural network did not give a better policy performance than the other methods, only outperforming the uniform policy of no advertising, and the random forest policy on the test set. Similar conclusions hold when using the DR estimator of policy evaluation, although now the policy π_{lasso} outperforms all the other methods on the training set as well.

5.2 Conversion

Next, we are interested in the expected conversion rate after personalization. To this end, we perform the same two-step procedure as described in section 4.1 using a lasso model, but this time the dependent variable Y_i is the conversion variable. The results are in Table 5.

Table 5 IPS estimates of conversion rates under different counterfactual policies.

Policy Category	**Policy**	**Estimated Conversion (%)**		**Increase in Conversion (%)**		**Prescribed Treatment %**	
		Training Set	Test Set	Training Set	Test Set	Training Set	Test Set
Uniform	π_0	0.15	0.11	-	-	0	0
	π_1	0.30	0.22	96.43	102.52	100	100
Personalized	π_{lasso}	0.29	0.20	87.71	81.74	25.14	24.99

We see that personalizing treatment using the lasso model no longer outperforms the uniform policy of assigning every user treatment, although they differ only slightly. It still outperforms π_0 by a lot, namely 87.71% and 81.74% on the training and test set respectively. Notice that the percentage increase in conversion rates for π_1 and π_{lasso} compared to π_0 is substantially higher than the increase in visit rates as shown in Table 3. This can be explained by looking at the definition of the conversion probability:

$$Pr\left(Y_i^c = 1|W_i\right) = Pr\left(Y_i^v = 1|W_i\right) \cdot Pr\left(Y_i^c = 1|W_i, Y_i^v = 1\right) \tag{14}$$

Where Y_i^c is the conversion variable of individual i, and Y_i^v is the visit variable of individual i. This equation shows why the increase in conversion is different than the increase in website visits. The advertisement does not only influence the visit probability, but also the probability of conversion. Therefore, the gains in these outcomes are naturally different. This formula also shows that increasing the conversion probability should lead to an increase in the conversion probability.

Next, it is assessed if the policy which maximizes visits also translates to the highest conversion rates. This is done by calculating the IPS estimator using conversion rates as outcome variable Y_i, based on π_{lasso} which optimized visits. Note that we can not use the DR estimator here, as we do not have en estimate for this outcome under the policy which optimizes the visits.

Table 6 IPS estimates of the visit and conversion rates under lasso policies optimized under each outcome.

Data set	Outcome optimized on	Visit	Conversion
Training set	Visit	4.87	0.27
	Conversion	4.71	0.29
Test Set	Visit	4.52	0.19
	Conversion	4.28	0.20

We see that optimizing over the visit rate does indeed lead to higher expected visit rates than optimizing over conversion rates, namely by 3.40% on the training set, and by 5.61% on the test set. We see however that these increases are moderate.

Similarly, optimizing over conversion rates instead of visit rates increases the expected conversion rates only slightly, namely by 7.41% and 5.26% on the training set and test set respectively. The best variable to optimize over, either visit rates or conversion rates, depends on the goal of the advertisement campaign, as well as the cost of advertisement. The latter will be discussed in the following section.

6 Cost Analysis

As mentioned in Section 1, the cost model of advertising decides which policy is optimal. In this section, we analyse which method is most effective under two different cost models, namely PPM and PPC. We first introduce some notation. Define the cost of advertising under the PPM model as c_i and under the PPC model as c_c, and the price of the product advertised p. Next, let the number of advertisements shown be A, the number of products sold Q, and the number of people who clicked the advertisement T. Assuming the seller only has advertising costs and no other costs, under the PPM model, the total profit over the advertisement period can be defined as:

$$Profit_{PPM} = Qp - Ac_i, \tag{15}$$

and under the PPC model:

$$Profit_{PPC} = Qp - Tc_c. \tag{16}$$

Following these equations, we find that the break-even profit-margin $\frac{p-c}{c} * 100\%$ under the PPM model is equal to $\frac{A-Q}{Q} \times 100\%$, and under the PPC model it is $\frac{T-Q}{Q} \times 100\%$. For instance, when using the policy π_{lasso} optimized over visit rates, we get a break-even profit-margin under the PPM model on the test set of $\frac{32.93-0.19}{0.19} = 17231.58\%$. In order to calculate this margin under the PPC model, we first need to know how many people who received treatment also clicked the advertisement, which is equal to the product of the policy assigned treatment and the estimated outcome under this policy. For the policy π_{lasso} optimized over visits, this equals 4.26% in the test set. The break-even profit-margin under the PPC model on the test set is therefore equal to $\frac{4.26-0.19}{0.19} = 2110.53\%$. This shows that under the PPC model, the required profit-margin is way lower than under the PPM model when using π_{lasso}. However, often the cost of advertising is higher in PPC models, exactly for this reason (Asdemir et al., 2012). In a similar fashion, we calculate these margins for the policy π_1. This leads to, under the PPM model, a margin of $\frac{100-0.22}{0.22} \times 100\% = 45354\%$, and under the PPC model it is $\frac{4.38-0.22}{0.22} \times 100\% = 1890.91\%$. We see that, as expected, under a non-personalized policy where everyone is shown the advertisement and the advertisement costs follow a PPM model, a substantially higher break-even profit-margin is required than under the personalized policy π_{lasso}. However, under a PPC model, the uniform policy requires a lower break-even profit-margin than the personalized policy. This is logical, as under a PPM model the company pays per advertisement shown, and the lasso model prescribes only about 33% of users treatment, which is only a third of the advertisements shown under the uniform policy. The uniform policy does lead to a slightly higher expected conversion rate, but this increase is moderate.

When, using the lasso policy optimized on conversion rates, the break-even profit-margin under the PPM model on the test set is equal to $\frac{24.99-0.20}{0.20} \times 100\% = 12395\%$. The expected click through rate in the test set under the lasso policy optimized on conversion rates is 3.99%. This leads to a required profit-margin under the PPC model of $\frac{3.99-0.20}{0.20} \times 100\% = 1895\%$. We see that both under the PPM and the PPC model, the policy optimized over conversion rates requires a lower profit margin than the policy optimized over visit rates to be profitable. This is due to the fact that both the expected conversion is higher, and the percentage of people assigned treatment is lower.

Lastly, we assess under which conditions π_{lasso} is more desirable than π_1. To this end, define $\{Q_l, A_l, T_l\}$ and $\{Q_u, A_u, T_u\}$ as the aforementioned variables Q, A, and T under the lasso and uniform policy respectively. Under the PPM model, the policy π_{lasso} is preferred if $Q_l p - A_l c_i > Q_u p - A_u c_i$, and similarly under the PPC model when $Q_l p - T_l c_c > Q_u p - T_u c_c$. We again use the policy π_{lasso} which was optimized over conversion rates, as this policy requires a lower profit margin than the one optimized over visits. In the test set, we obtain the following condition under the PPM model: $0.20p - 24.99c > 0.22p - 100c$, which comes down to a profit margin below 374950%. Under the PPC model we get $0.20p - 3.99c > 0.22p - 4.38c$, implying a profit margin below 1850%. Under these conditions, the lasso policy optimized over conversion rates is preferred over the uniform policy π_1. This gives a simplified example of how firms can evaluate different policies under two common cost models for online advertising, and make a decision based on this evaluation.

Note that the profit margins on advertising shown in this section are extremely high and quite unrealistic, as a typical margin on advertising is somewhere between 200% and 600% (Brown, 2023). However, the comparison of margins between the different policies is still useful and the general procedure of the cost model analysis is still applicable in practice. Furthermore, the relative difference of the margins between the different policies is intuitive, as we do expect a policy assigning everyone treatment to require a higher profit margin under a CPM model compared to a policy assigning only few people treatment. The other comparisons made in this section remain intuitive in a similar manner.

7 Discussion and Conclusion

Online advertising is a fundamental part of promotional activities of all companies nowadays. In this paper, we tried to answer the question *Is the effect of advertising heterogeneous across consumers, and if so, how can the assignment of advertisements be personalized based on consumer characteristics, and what are the gains from personalization?* This effect of advertising strategies on website visits and conversion rates is examined using data from several incrementality experiments. Using the IPS estimator for off-policy evaluation, We find that in terms of uniform policies, the policy showing advertisements to all users performs better than showing no advertisements at all, with an increase in visit rates of about 23%, and more than doubling the expected conversion rate.

Next, a personalization strategy based on lasso regression is developed to examine heterogeneous effects of advertising on users decisions. We find that this policy increases expected visit rates by about 27% over showing no advertisements at all, and about 3% over showing all users advertisements, although this last increase is not statistically significant.

As the IPS estimator makes quite a strict assumption in order to be unbiased, we also estimate expected pay-offs using the DR off-policy estimator. We find that in our context, the estimates of both estimators do not differ significantly, *Is the effect of advertising heterogeneous across consumers, and if so, how can the assignment of advertisements be personalized based on consumer characteristics, and what are the gains from personalization?* and lasso still outperforms the uniform policies, although not necessarily by a statistically significant amount.

The performance of the lasso policy is also compared to other methods in machine learning, using both popular outcome estimators like random forests and the recently more popular artificial neural network, as well as recently proposed CATE estimators like causal trees and causal forests. We find that these policies do not outperform the lasso policy, which is in line with the findings of Yoganarasimhan et al. (2022). The random forest policy even performs worse than the best uniform policy, whereas CART and XGBoost are unable to differentiate between users at all. Additionally, the recently popularized causal tree and causal forest also fail to outperform the lasso policy, as well as the policy based on neural networks, despite its increasing popularity. An important takeaway from a managerial standpoint is that companies should not blindly choose a popular estimator and assume that it leads to a good personalization policy, but should carefully evaluate several policies, for instance using IPS or DR, to decide what strategy to deploy.

Next, we find that a personalized lasso policy which optimizes conversion rates does not lead to an increase in conversion over the best uniform policy, although the differences are small. We also find that the policy optimized over website visits performs slightly worse on expected conversion rates than a policy explicitly maximizing conversion rates, though the difference is moderate. The same holds the other way around. This implies that based on the goal of the advertising campaign, a firm should optimize over different outcome variables. If the only goal is to sell more products in the short run, optimizing over conversion is most likely the best strategy, especially since this is cheaper under certain cost models since the prescribed treatment percentage is lower. However, optimizing over visit rates still leads to only slightly lower sales. Therefore, if the firm wishes to expand their brand awareness, it should likely optimize over visits as more people are then exposed to the advertisement, and more users visit the website. In the short run this might lead to lower sales, but brand awareness could lead to an increased long term performance (Bernarto et al., 2020).

Lastly, we look into different cost systems, and which policy is preferred under these cost systems. We find that the personalized lasso policy which optimizes over conversion rates requires lower profit margins than the policy optimized over visit rates in order to be profitable, no matter what cost system. Furthermore, we find that under a PPM model, a personalized strategy is often more profitable than the best uniform strategy, whereas under a PPC model it is the other way around. This shows that when choosing what advertising policy to use, firms should take into consideration what cost model they are dealing with, and choose their strategy accordingly.

8 Limitations and Further Research

This paper comes with several limitations. First, not all available data was used, but only a subset of observations due to computational limitations. Even though the subsetting was done randomly, and the distribution of the subset should be similar to the full data set, this might lead to small biases. In addition to this, there is little managerial motivation to not use all available data.

Secondly, as the user features were anonymized completely, both in the naming and the levels of the variables, interpretation of the personalization policies is impossible and we cannot identify underlying mechanisms. Though finding these mechanisms were not the goal of this research, it does give managers insight as to what and who to target on, making the advertising process more efficient. It can therefore be of importance to perform a similar study with known user features.

Another limitation is related to the test period of the experiment the data was retrieved from. The data was collected over a time span of only two weeks, meaning long term effects of advertisement are unobservable. Depending on the type of product advertised, it is likely that a user who is shown an advertisement might take more than two weeks to decide on buying a product. Furthermore, users shown the advertisement at the end of the test period would have to buy the product within the last few days in order to be considered a converted user. Therefore, the information on the actual effect of the advertisement is incomplete.

Furthermore, in computing the IPS and DR estimates, we use a propensity score following an estimated linear model. However, we can not know for sure if these scores are correct, especially if treatment assignment also depends on unobserved covariates. In this case, the study would not be conditionally unconfounded either, which we assumed to be true throughout the paper. It may therefore be useful to perform a similar research with an unconditionally unconfounded study, or a study where the actual propensity scores are known. In addition to this, we use the DR estimator as first proposed by Dudík et al. (2011). However, in more recent literature different forms of the DR estimator are proposed, which might be more optimal. Li and Shen

(2020) use a DR estimator which does not calculate the average absolute outcome (visit rates for instance), but it estimates the average actual uplift. Kennedy (2020) extend this idea, where the uplift is calculated per individual, and afterwards regressed on the observed confounders to obtain the estimated uplift per individual, which is again averaged in the end. This method leads to more efficient estimates than the DR estimator used throughout this research. It may therefore be of use to explore if other definitions of this estimator would lead to different outcomes.

Next, for the models that require parameter tuning, some parameters were handpicked, whereas tuning all parameters should lead to better model performance. This is mainly the case for the ANN and XGBoost models. However, additional tuning would lead to better predictive performance, but we have shown that this does not automatically lead to higher gains in visit and conversion rates, and may therefore not lead to better policy design. This is something worth wile to be further investigated.

An interesting finding is that some personalization policies perform worse than the best uniform policy, with some policies being unable to personalize at all. Furthermore, we find that methods explicitly designed to model treatments effects do not necessarily perform well. A useful followup to this study is to investigate if these findings are generalizable, or arise due to the structure and nature of the data.

References

Adelle, C., & Weiland, S. (2012). Policy assessment: The state of the art. *Impact assessment and project appraisal*, *30*(1), 25–33.

Adomavicius, G., & Tuzhilin, A. (2005). Personalization technologies: A process-oriented perspective. *Communications of the ACM*, *48*(10), 83–90.

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

Anshari, M., Almunawar, M. N., Lim, S. A., & Al-Mudimigh, A. (2019). Customer relationship management and big data enabled: Personalization & customization of services. *Applied Computing and Informatics*, *15*(2), 94–101.

Arora, N., Dreze, X., Ghose, A., Hess, J. D., Iyengar, R., Jing, B., Joshi, Y., Kumar, V., Lurie, N., Neslin, S., et al. (2008). Putting one-to-one marketing to work: Personalization, customization, and choice. *Marketing Letters*, *19*, 305–321.

Asdemir, K., Kumar, N., & Jacob, V. S. (2012). Pricing models for online advertising: Cpm vs. cpc. *Information Systems Research*, *23*(3-part-1), 804–822.

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

Bagwell, K. (2007). The economic analysis of advertising. *Handbook of industrial organization*, *3*, 1701–1844.

Barth, S., & De Jong, M. D. (2017). The privacy paradox–investigating discrepancies between expressed privacy concerns and actual online behavior–a systematic literature review. *Telematics and informatics*, *34*(7), 1038–1058.

Becker, I. F., Linzmajer, M., & von Wangenheim, F. (2017). Cross-industrial user channel preferences on the path to online purchase: Homogeneous, heterogeneous, or mixed? *Journal of Advertising*, *46*(2), 248–268.

Bernarto, I., Berlianto, M. P., Meilani, Y. F. C. P., Masman, R. R., Suryawan, I. N., et al. (2020). The influence of brand awareness, brand image, and brand trust on brand loyalty. *Jurnal Manajemen*, *24*(3), 412–426.

Bertot, J. C., & Choi, H. (2013). Big data and e-government: Issues, policies, and recommendations. *Proceedings of the 14th annual international conference on digital government research*, 1–10.

Beygelzimer, A., & Langford, J. (2009). The offset tree for learning with partial labels. *Proceedings of the 15th ACM SIGKDD international conference on Knowledge discovery and data mining*, 129–138.

Brown, R. (2023). 3 Tips For Ideal Advertising Cost of Sales (ACoS) on Amazon — Pattern. https://pattern.com/blog/amazon-acos-advertising-cost-of-sales/#:~:text=You'll%5C%20need%5C%20to%5C%20consider,can%5C%20affect%5C%20the%5C%20ideal%5C%20ACoS.

Chan, D., Ge, R., Gershony, O., Hesterberg, T., & Lambert, D. (2010). Evaluating online ad campaigns in a pipeline: Causal models at scale. *Proceedings of the 16th ACM SIGKDD international conference on Knowledge discovery and data mining*, 7–16.

Chen, T., & Guestrin, C. (2016). Xgboost: A scalable tree boosting system. *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, 785–794.

De Keyzer, F., Dens, N., & De Pelsmacker, P. (2015). Is this for me? how consumers respond to personalized advertising on social network sites. *Journal of Interactive Advertising*, *15*(2), 124–134.

DeTienne, K. B., & DeTienne, D. H. (2017). Neural networks in strategic marketing: Exploring the possibilities. *Journal of Strategic Marketing*, *25*(4), 289–300.

Diemert, E., Betlei, A., Renaudin, C., & Amini, M.-R. (2018). A large scale benchmark for uplift modeling. *KDD*.

Dudík, M., Erhan, D., Langford, J., & Li, L. (2014). Doubly robust policy evaluation and optimization.

Dudík, M., Langford, J., & Li, L. (2011). Doubly robust policy evaluation and learning. *arXiv preprint arXiv:1103.4601*.

Goldfarb, A. (2014). What is different about online advertising? *Review of Industrial Organization*, *44*, 115–129.

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

Gorr, W. L., Nagin, D., & Szczypula, J. (1994). Comparative study of artificial neural network and statistical models for predicting student grade point averages. *International Journal of Forecasting*, *10*(1), 17–34.

Guelman, L., Guillén, M., & Pérez-Marín, A. M. (2015). A decision support framework to implement optimal personalized marketing interventions. *Decision Support Systems*, *72*, 24–32.

Hanberger, A. (2001). What is the policy problem? methodological challenges in policy evaluation. *Evaluation*, *7*(1), 45–62.

Heinrichs, J. H., & Lim, J.-S. (2008). Impact of marketing model application and competitive intelligence utilization on strategic response capability. *Journal of Strategic Marketing*, *16*(2), 91–110.

Howarth, J. (2023). Alarming average screen time statistics (2023). *Exploding Topics*. https://explodingtopics.com/blog/screen-time-stats

James, G., Witten, D., Hastie, T., & Tibshirani, R. (2013). *An introduction to statistical learning* (Vol. 112). Springer.

Karsoliya, S. (2012). Approximating number of hidden layer neurons in multiple hidden layer bpnn architecture. *International Journal of Engineering Trends and Technology*, *3*(6), 714–717.

Kennedy, E. H. (2020). Towards optimal doubly robust estimation of heterogeneous causal effects. *arXiv preprint arXiv:2004.14497*.

Kietzmann, J., Paschen, J., & Treen, E. (2018). Artificial intelligence in advertising: How marketers can leverage artificial intelligence along the consumer journey. *Journal of Advertising Research*, *58*(3), 263–267.

Kim, J. J., Kim, T., Wojdynski, B. W., & Jun, H. (2022). Getting a little too personal? positive and negative effects of personalized advertising on online multitaskers. *Telematics and Informatics*, 101831.

Lambert, D., & Pregibon, D. (2007). More bang for their bucks: Assessing new features for online advertisers. *Proceedings of the 1st international workshop on Data mining and audience intelligence for advertising*, 7–15.

Li, X., & Shen, C. (2020). Doubly robust estimation of causal effect: Upping the odds of getting the right answers. *Circulation: Cardiovascular Quality and Outcomes*, *13*(1), e006065.

Malheiros, M., Jennett, C., Patel, S., Brostoff, S., & Sasse, M. A. (2012). Too close for comfort: A study of the effectiveness and acceptability of rich-media personalized advertising. *Proceedings of the SIGCHI conference on human factors in computing systems*, 579–588.

Pedregosa, F., Varoquaux, G., Gramfort, A., Michel, V., Thirion, B., Grisel, O., Blondel, M., Prettenhofer, P., Weiss, R., Dubourg, V., Vanderplas, J., Passos, A., Cournapeau, D., Brucher, M., Perrot, M., & Duchesnay, E. (2011). Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, *12*, 2825–2830.

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

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

Ruder, S. (2016). An overview of gradient descent optimization algorithms. *arXiv preprint arXiv:1609.04747*.

Rumelhart, D. E., Widrow, B., & Lehr, M. A. (1994). The basic ideas in neural networks. *Communications of the ACM*, *37*(3), 87–93.

Sävje, F. (2021). Randomization does not imply unconfoundedness. *arXiv preprint arXiv:2107.14197*.

Sharma, S., Sharma, S., & Athaiya, A. (2017). Activation functions in neural networks. *Towards Data Sci*, *6*(12), 310–316.

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

Statista. (2023). Digital advertising spending worldwide from 2021 to 2026. https://www.statista.com/statistics/237974/online-advertising-spending-worldwide/#:

~:text=It%5C%20was%5C%20calculated%5C%20that%5C%20the,would%5C%20reach%5C%20836%5C%20billion%5C%20dollars.

Stern, H. S. (1996). Neural networks in applied statistics. *Technometrics*, *38*(3), 205–214.

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

Warner, B., & Misra, M. (1996). Understanding neural networks as statistical tools. *The american statistician*, *50*(4), 284–293.

Yoganarasimhan, H. (2020). Search personalization using machine learning. *Management Science*, *66*(3), 1045–1070.

Yoganarasimhan, H., Barzegary, E., & Pani, A. (2022). Design and evaluation of optimal free trials. *Management Science*.

Appendix

A Parameter Estimation

Table 7 Overview of used methods, their hyper-parameters, and the method of parameter tuning

Model	Hyper-Parameters	Tuning Method and Language
Linear regression	-	None - R
Lasso	λ	glmnet - R
CART	ζ	GridSearchCV - Python
RF	n_tree, max_f, n_min	hyperopt - Python
XGBoost	α, η, λ, d_max	hyperopt - Python
Causal Tree	ζ, q	CausalTree & grid search - R
Causal Forest	frac, max_imb, n_min, mtry, n_trees	causalForestDML - Python
ANN	n_nodes, n_layers, η, b_size	TensorFlow & grid search - Python

Below, the search space and optimal value for each parameter are given

Lasso:

- $\lambda \in [9.7 \times 10^{-6},\ 9.7 \times 10^{-2}]$ and $\lambda^* = 2.0 \times 10^{-5}$

CART:

- $\zeta \in [1 \times 10^{-10},\ 2 \times 10^{-1}]$ and $\zeta^* = 1.0 \times 10^{-4}$

Random Forest:

- $\text{n\_tree} \in [100,\ 1200]$ and $\text{n\_tree}^* = 600$
- $\text{max\_f} \in \{k, \sqrt{k}\}$ and $\text{max\_f}^* = k$, where k is the number of features
- $\text{n\_min} \in [10,\ 300]$ and $\text{n\_min}^* = 70$

XGBoost:

- $\alpha \in \{1, 5, 10, 15, 20, 25\}$ and $\alpha^* = 20$
- $\eta \in [0,\ 1]$ and $\eta^* = 3.8 \times 10^{-2}$
- $\lambda \in \{0, 1, 5, 10, 15\}$ and $\lambda^* = 10$
- $\text{d\_max} \in \{6, 8, 10, 12\}$ and $\text{d\_max}^* = 6$

Causal Tree:

- $\zeta^* = 5.4 \times 10^{-6}$
- $q \in [100, 1000]$ and $\text{q}^* = 300$

Causal Forest: *Note that CausalForestDML does not offer the option to retrieve optimal parameters.*

- frac $\in (0,\ 1]$
- max_imb $\in [0,\ 0.5]$
- n_min $\in [2, N]$, where N is the total number of observations
- mtry $\in \{k, \sqrt{k}, \log_2 k\}$, where k is the number of features
- n_trees > 1

ANN: *Note that only the number of nodes was tuned using grid search, the other parameters were set manually*

- n_nodes $\in \{6, 7, 8, 9\}$ and n_nodes$^* = 8$
- n_layers$^* = 1$
- $\eta^* = 0.01$
- b_size$^* = 32$

B Mean Squared Errors of Personalization Policies

Table 8 Comparison of the predictive performance of policy design methods.

Method	**Mean Squared Error**	
	Training Set	Test Set
Linear Regression	0.0330	0.0308
Lasso	0.0329	0.0308
CART	0.0313	0.0301
Random Forest	0.0294	0.0293
XGBoost	0.1326	0.1322
ANN	0.0314	0.0300

Note. Notice that there is no MSE for the CATE estimators, as we do not know the actual values of the CATE.

C Code Description

The code provided in the code package consists of several parts. *Data_Preparation.R* samples the data and creates the train/test split. *Summary_Statistics.R* computes the values shown in Table 1 and Table 2. Next, *Lasso_And_Lr.R, CART.py, Random_Forest.py, XGBoost.py, ANN.py, Causal_Tree.R, and Causal_Forest.py* are used to design and evaluate personalized policies. Lastly, *Lasso_Significance_Analysis.R* is used to compute the significance of differences in expected gain between several policies. A more detailed explanation of the code is found in the *readme.txt* in the code package.