Erasmus University Rotterdam
Erasmus School of Economics
Bachelor Thesis Econometrics and Operations Research

Optimizing Personalized Online Advertisement

Ward Broos (582309)

Supervisor:	H. Deng
Second assessor:	Prof. dr. ir. R. Dekker
Date final version:	July 2024

Abstract

In the fast-evolving field of digital marketing, personalized online advertisements remain crucial for direct communication with consumers. However, the effectiveness of online advertising campaigns hinges significantly on the ability to personalize content to meet diverse preferences. This study aims to explore the best personalized policy, using several methods, for online advertisements to enhance consumer engagement and response rates. It will not only investigate whether customers visit the advertisements but also whether they make a purchase (conversion). Additionally, the policies will be evaluated using the inverse propensity score estimator and the doubly robust estimator to show that for our best model, an increase in conversion of 0.0294% and 0.389% for visit can be found, compared with addressing every customer a treatment. Finally, we show that personalized policies do not always outperform uniform policies and that heterogeneity takes a significant role in creating personalized policies.

1 Introduction

Over the last few years online advertising has changed significantly. The desire of companies to capture the attention of their audience has led to an increasingly important role in personalized email campaigns. In these campaigns, the challenge lies not only in reaching potential customers but also in engaging them effectively through personalized content that meets individual preferences and behaviors. Therefore, these personalized email campaigns are far from everyone's cup of tea and can sometimes cause frustration for customers. Something that often results in a counterproductive effect than what the original intent of the advertisement was. For this reason, companies must determine which types of advertisement to send to each customer, and when it is better to pass by certain customers. As a result, it is of interest to develop personalized policies that provide a customized treatment, to maximize the outcome variable.

To determine which personalized policy performs best, this research seeks to optimize targeted online marketing strategies, specifically focusing on personalized email advertisements. The goal is to improve the campaigns' conversion and visit rates. This intuitive raises a crucial question that we aim to answer in our research. Namely, how can we optimize personalized email advertising to maximize conversion and visit rates?

The optimization starts with capturing customer's heterogeneity responses to email advertisements. Therefore, a two-step approach is applied to design a personalized policy, since an unstructured search for the optimal policy is not feasible in our high-dimensional setting. First, using a model as a function of the customer's characteristic variables. Then, making a prediction with the model and the different treatments for each customer. This allows us to finally assign the optimal treatment for each customer, that is the treatment for which the probability of the outcome is the highest. The allocation of optimal treatment forms then a personalized policy. Since we seek to optimize personalized online advertisement, several policies based on different outcome estimators will be used. Despite the simple models, logistic lasso and logistic regression, tree-based methods are used, since they are able to model complex relations within large data. Heterogeneous treatment estimator policies should also be designed, to seek for a different approach to partition within the tree models. Second, the designed policies are evaluated using the inverse propensity score reward estimator (IPS) and the doubly robust estimator (DR) and are compared to the uniform policies, which are found by giving all customers the same treatment and calculating their effect. This way, a conclusion can be formed for optimizing personalized online advertising based on policy evaluation.

This research brings the following aspects to the current literature. First, the effect of advertisements on the visit and conversion rate will be examined. Then this effect will be improved by designing different personalized policies. Eventually, an approach to design and evaluate personalized policies to marketing companies will be presented. The poor performance of some well-known estimators while designing these policies is highlighted, emphasizing the necessity of offline policy evaluation.

This paper is organized as follows: First, the existing literature is discussed. In Section 3 we present the data that is used. In Section 4, the methods used in this paper are presented. The numerical results are discussed in Section 5. At last, we present our conclusion and suggest directions for future research.

2 Literature

Targeting customers with personalized policies is a hot topic in modern literature. Ranging from medical treatment in healthcare to creating personalized exercises in fitness. All seeking to find heterogeneity in customer response to treat them differently.

Firstly, understanding how much personalized advertisement plays a role in online advertisement, we delve into the research of Yan et al. (2009). They provide an empirical study on the click-through log of advertisements collected from a commercial search engine, to explore in what quantity behavioral targeting contributes to online advertising. Impressive results were found on the impact of behavioral targeting in online advertising. As personalized advertising has shown to be successful, we aim to find efficient approaches. One approach was done in the research by Yoganarasimhan et al. (2023). Namely, they examined significant heterogeneity in customer response to different free trial periods. This shows that companies could benefit from assigning customers different trial periods, according to his or her skills and origins. In their research different outcome estimators are compared, of which the lasso-based personalized policy appears to be the best.

Despite the focus on the lasso method, machine learning appears to be a widely used method in the personalization of digital products and promotions.

Ban and Keskin (2021) did comparable research about personalized dynamic pricing with the use of machine learning. Where there was also a search for customer heterogeneity in a large-scale field experiment. Since machine learning models learn from performing problem-solving operations and algorithms, their use is only becoming more important for personalized marketing. Perlich et al. (2014) delves deeper into the use of machine learning in personalized advertising. As they describe a multistage transfer learning system for targeted display advertising to improve the effectiveness of advertising campaigns, using machine learning techniques.

Policy evaluation plays a critical role in the development and refinement of targeted policies, offering a structured approach in understanding the effectiveness and efficiency of public programs. Besides this, it is also very important before implementing personalized targeting, when dealing with different performances of the policies. Such off-policy evaluation has become very popular within this field. Hanberger (2001) stated the importance of the policy evaluation, hence addressing wrong medicines could have catastrophic consequences. Simester et al. (2020) examines effective ways for managers to evaluate targeting policies. They offer two insightful observations. Firstly, they point out that randomization by action is better than randomization by policy, since it enables us to assess any policy using an off-policy evaluation method. Secondly, they point out that we should understand that, when comparing two policies, if they both advise a customer to do the identical action, there is absolutely no difference in how well the policy performs for those customers.

As for our policy evaluation, we focus on the inverse propensity score reward estimator as mentioned by Yoganarasimhan et al. (2023). Stating that the IPS is accurate if the propensity scores are estimated accurately. Dudík et al. (2011) stated that the doubly robust estimator could be seen as an improvement upon the IPS estimator as one of the major advantages of the DR estimator over the IPS lies in variance. The DR estimator obtains potentially lower variance allowing more stable estimates and faster convergence rates, both very crucial for effective policy learning and development. Kang and Schafer (2007) also examined the DR estimator, and marked that the DR estimator obtains lower variance when optimizing the propensity score model with the outcome regression model. A small variation immediately causes faster convergence, which ensures that the model is more effective and efficient for the data used. This is a very important factor in determining personalized policies. Because of this, both estimators are considered while evaluating the policies.

3 Data

The data was captured on the Criteo AI lab and is related to a direct marketing campaign with over twenty-five million customers. The CRITEO-UPLIFT-1 dataset was collected using various incrementality tests, that is a specific type of randomized trial where an arbitrary portion of the population is not targeted by advertising. This dataset includes thirteen million customers, with each customer having multiple behaviors while responding to marketing efforts that were captured on the advertiser's website during the two-week test period.

For each consumer, the following information was captured: twelve pre-treatment variables indicating various customer and behavioral characteristics. A binary treatment variable that specifies whether the customer received the marketing treatment. Both visit and conversion are represented as binary variables, indicating whether the customer converted or visited after the market treatment. For privacy reasons, the feature 'names' are anonymized, and their values are randomly projected to maintain predictive power, making it impossible to identify the original context. Because of the vast size of the Criteo data, a random sample was generated. This sample includes 100,000 customers with an overall treatment ratio of 84.9%, similar to the original data. According to Table 4, the sample data statistics are significant and represent the actual data accurately. The sample data is split up into two independent samples. Namely, training data and test data. The training data is utilized to learn model parameters as well as to select models. The test data serves as a hold-out for evaluating the performance of personalized policies, designed on models built with training data. 70% of the data is used for training, with the remaining 30% for testing. Since the data is randomly divided across the two samples, the two data sets might differ slightly. This will be used when comparing the outcomes.

The average effect of sending an advertising mail can be estimated by simply comparing the percentages of the outcome of interest (conversion or visit). Since the mailing was randomly selected in the experiment, it rules out the probability of self-selection into treatments, which is a common problem in field experiments. Table 1 shows a summary statistics of the sample data for the treatments. Assigning advertising emails increased website visitors by 1.31 percentage points and conversion by 0.102 percentage points. When looking at the total data, as can be found in Table 5, conversion improved by 0.115 percentage points, while visits increased by 1.034 percentage points. As a result, issuing an advertising mail will significantly increase visits and purchases.

Table 1: Summary Statistics of Conversion and Visit rates, and Treatment Assignment.

Sample data	No treatment	Treatment	Total
Number of observations (N)	15,016	84,984	100,000
Percent of total observations	15.016	84.984	100
Number of visits	540	4,169	4709
Percent of total visits	11.467	88.533	100
Number of conversions	31	262	293
Percent of total conversions	10.580	89.419	100
Visit rate within group (in %)	3.596	4.906	4.709
Conversion rate within group (in %)	0.206	0.308	0.293

4 Methodology

In our sample, the following was observed for each customer i: (1) twelve pre-treatment explanatory variables (X_i), (2) the treatment assignment (W_i), and (3) the conversion or visit indicator as the outcome variable (Y_i). With these variables a certain model $f(x, w) = \mathbb{E}[Y \mid X_i = x, W_i = w]$ can be learned. This model is the expected value of the outcome variable, given the treatment and pre-treatment variables. First, every customer is assigned to treatment $W_i = 1$ and $W_i = 0$ separately. Both models are then predicted, in a way that $\hat{f}_{(x,0)}$ and $\hat{f}_{(x,1)}$ are obtained. As the aim is to optimize online advertisement and therefore maximize the outcome variable, the model with the highest estimated probability or value of the outcome variable is labeled as the optimal policy for that customer, $w^* = \arg\max_{w \in W} \hat{f}(X_i = x, w)$. By doing this for all consumers, a personalized policy is designed, and can be formulated as $\pi_f(X_i) = w^*$. The customer is indifferent if the treatment variable's probabilities are equal. After that, the client gets no treatment. This is due to the expenses associated with mail assignments. The description of the several policies that are used are shown below.

4.1 Logistic regression

As the outcome variable is predicted in probabilities, the logistic regression model is used. That is based on the logistic function. It estimates the probability of $Y_i = 1$ given a set of predictors, Equation (1). Our model contains interaction terms to investigate the potential combined effects of variables X_i and W_i on the outcome variable Y_i. These terms were included in the model to establish their predictive value and to find the effectiveness of different treatments across customers. The logistic regression can be estimated instantaneously, as it does not apply hyper-parameters tuning. Further, the logistic regression is known for its poor out-of-sample performance, which is of great value when creating personalized policies. Therefore, we suspect that this method will perform poorly.

$$P(Y_i = 1) = \frac{1}{1 + \exp(-(\beta_0 + X_i\beta_1 + W_i\beta_2 + X_iW_i\beta_3))} \tag{1}$$

4.2 Logistic Lasso

In the research by Yoganarasimhan et al. (2023) different outcome estimators are compared, of which the lasso-based personalized policy appears to be the best. The logistic lasso is applied to capture the probability per treatment. Differing from the lasso model it estimates a logistic regression to minimize the MSE, with an additional term to penalize model complexity. Similar to the logistic regression, the logistic lasso contains interaction terms to find the effectiveness of different treatments across customers. The estimates of the logistic lasso can be shown as follows:

$$
\begin{aligned}
(\hat{\beta_0}, \hat{\beta_1}, \hat{\beta_2}, \hat{\beta_3}) = \arg\min \Bigg\{ & - \sum_{i=1}^{N} \left[Y_i \log(P(Y_i = 1)) + (1 - Y_i) \log(1 - P(Y_i = 1)) \right] \\
& + \lambda \left(||\beta_1||_1 + ||\beta_2||_1 + ||\beta_3||_1 \right) \Bigg\}
\end{aligned}
\tag{2}
$$

4.3 Tree-based methods

In addition to regression models, machine learning models are applied. Not only do they have different outcome estimators, but these policies have been proven in the literature to function effectively for personalized policies. According to Chen and Guestrin (2016) and Breiman (2001), tree-based approaches have high predictive power and are capable of handling complexity. These methods offer diverse modeling capabilities, allowing them to capture complexity that simpler models might overlook.

4.3.1 Classification and Regression Tree

The Classification and Regression Tree (CART) recursively partitions the data into subsets based on the values of input features. Within each of these partitions, the mean value of Y is then noted as the predicted outcome, E(Y), for all observations. This results in a tree-like structure, where each internal node represents a decision based on a feature, each branch represents an outcome of that decision, and each leaf node represents a predicted outcome.

Let M be the regions that are used to partition the data. ρ_m is the predicted value of y in region R_m, and R_m is denoted as the m^{th} region. This gives the following expression for the CART model:

$$
y = f(x, w) = \sum_{m=1}^{M} \rho_m I((x, w) \in R_m) \tag{3}
$$

In general, trees are trained by specifying the mean squared error that will be minimized at each step of the tree-growing process using an algorithm. Often a penalty term is added to the cost function, since overfitting is a well-known problem when working with trees. The weights for this penalty term are learned from cross-validation on the data. Since CART often has poor predictive accuracy due to its discontinuous nature and sensitivity to outliers, random forest and XGBoost are introduced.

4.3.2 Random Forest

Typically the random forest consists of many trees. Where each tree is individually created during the training phase on a bootstrap sample. A random subset of characteristics is selected for splitting the tree. As done so, it adds randomness to the model and ensures that every tree is uniquely created. This helps to reduce variation and improve its ability to perform well on new data. Consequently, the average prediction of the random forest trees tends to be a better predictor than a single-trained tree. Equation (4) shows the formulation of such prediction. The random forest is built upon the structure of the CART model in Equation (3), but interprets multiple trees and averages their predictions. Let T be the number of trees in the model. Each tree t partitions the data into M regions. $R_m^{(t)}$ is the $m-th$ region of tree t. Such that, $\rho_m^{(t)}$ is the predicted value of y in region $R_m^{(t)}$. $I(x,w)$ denotes a indicator function that equals one if the observation(x,w) falls within the region R_m and zero else.

$$y = f(x,w) = \frac{1}{T}\sum_{t=1}^{T}\sum_{m=1}^{M_t}\rho_m^{(t)} I((x,w) \in R_m^{(t)}) \tag{4}$$

4.3.3 Extreme Gradient Boosting

XGBoost improves predictive performance through gradient boosting, where the errors of the tree are corrected by a subsequent newly created tree. Eventually leading to more accurate models. A gradient descent algorithm is used to minimize the specified loss function. By doing so the XGBoost is able to capture complex patterns within the data, that might be missed by random forest or CART, making it robust and very suitable for high-dimensional data.

$$y = f(x,w) = F_T(x,w) = F_0 + \sum_{t=1}^{T}\eta\sum_{m=1}^{M_t}\rho_m^{(t)} I((x,w) \in R_m^{(t)}) \tag{5}$$

The model of XGBoost is shown above. The difference from Equation (4) is that each prediction is updated with a learning rate, rather than averaging the sum of the predictions. F_0 is the initial prediction, that is updated for all T trees with the learning rate η. Eventually after T trees the final prediction $F_T(x,w)$ is constructed.

4.4 Heterogeneous Treatment Effect Estimators

Moreover, heterogeneous treatment estimators, such as causal tree and causal forest, are also implemented. Athey and Imbens (2016) describe that these heterogeneous treatment estimators were developed to better explain the variation in different treatments within distinct sub-populations, and so perform better than tree-based policies. These approaches present a varied image of how various components of a population react to different policies. As it first obtains consistent estimates of heterogeneous treatment effects for each pair of treatments for all customers and then uses them to assign treatments.

Consistent estimates of individual-level treatment effects are obtained, $\tau_{w_1,w_0}(x)$, for the binary pair of treatments w. Hence the aim is to optimize online advertisement, the design of the optimal policy is done as follows. First, $\hat{\tau}_{w_1,w_0}$ is estimated for all customers. Then, treatment is assigned as the optimal policy if $\hat{\tau}_{w_1,w_0} > 0$. Vice versa, if $\hat{\tau}_{w_1,w_0} < 0$ the outcome variable is more likely to be obtained when assigning no treatment, so w_0 is labeled as optimal policy. Note that if $\hat{\tau}_{w_1,w_0}$ equals zero, the probability of the outcome variable for w_1 and w_0 is indifferent. Treatment w_0 is then considered best due to the expense of sending emails and shall be allocated as optimal policy for the customer.

4.4.1 Causal Tree

The causal tree is based on the CART model. However, the difference lies in its partition. The causal tree partitions the covariate space into regions with similar within-partition treatment effects, whereas the CART model divides the space to maximize prediction ability. The causal tree estimates the treatment effect for each region, $l(x)$, using Equation (6). As separating observations with the same treatment effect has no effect on improving the target function, the algorithm pools decision tree observations with the same treatment effects. Due to its pooling, the main target for heterogeneous treatment effect estimators lies in finding the optimal $l(x)$.

Equation (6) forms a modified method that is applied to estimate individual-level treatment effects. To deal with the high-dimensional covariate space, the modern heterogeneous treatment effects estimators aim to pool observations that are near the covariate space and place those observations in certain regions. To then estimate the conditional mean treatment effect for sub-populations, under standard assumptions.

$$\hat{\tau}_{w_1,w_0}(x) = \frac{\sum_{X_i \in l(x), W_i = w_1} Y_i}{\sum 1[X_i \in l(x), W_i = w_1]} - \frac{\sum_{X_i \in l(x), W_i = w_0} Y_i}{\sum 1[X_i \in l(x), W_i = w_0]} \tag{6}$$

4.4.2 Causal Forest

Since the causal tree algorithm tends to suffer from the same weaknesses as those of the CART, causal forest is introduced. Based on the design of random forest, causal forest can be used to flexibly estimate any heterogeneous quantity from the data, including heterogeneous treatment effects. And improve robustness and accuracy in estimating treatment effects. Causal forests consist of multiple trees generated during the training phase using a bootstrap sample. The objective is to create partitions where the treatment effects are as homogeneous as possible within each region. Consequently, the predicted treatment effect should be consistent within each region. The estimated treatment effect $\hat{\tau}_{w_1,w_0}$ for a total of T trees, is the average over all the $t-th$ tree treatment effect estimates, $\hat{\tau}_{w_1,w_0,t}(x)$, as is shown below:

$$\hat{\tau}_{w_1,w_0}(x) = \frac{1}{T}\sum_{t=1}^{T}\left(\frac{\sum_{X_i \in l(x), W_i = w_1} Y_i}{\sum 1[X_i \in l(x), W_i = w_1]} - \frac{\sum_{X_i \in l(x), W_i = w_0} Y_i}{\sum 1[X_i \in l(x), W_i = w_0]}\right) \tag{7}$$

4.5 Personalized Policy Evaluation

4.5.1 Inverse Propensity Score

The Inverse Propensity Score estimator is used to evaluate our designed personalized policies. The IPS estimates the causal effect of a treatment. Because the treatment assignment happened non-randomly, the IPS adjust for biases. Providing a fair assessment of each policy's effectiveness. Through this comprehensive evaluation, we can identify the most promising personalized policy that optimizes the desired outcomes. The IPS approach relies on the propensity score, which is the probability of receiving a treatment given the observed covariates.

The propensity score is estimated using a counting-based approach. For every personalized policy, this approach counts the occurrences of treatment and policy combinations to estimate the empirical propensity score. The empirical propensity score in the counting-based approach is the fraction of the number of times a certain treatment W_i is observed given its prescribed treatment by the policy π_i, divided by the total number of observations where the policy applies the same treatment. The empirical propensity score $\hat{e}_{\pi(X_i)}(W_i)$ is given in Equation (8), representing the probability of receiving the treatment under a personalized policy $\pi(X_i)$.

$$\hat{e}_{\pi(X_i)}(W_i) = \frac{\frac{1}{N}\sum_{j=1}^{N} 1[W_j = W_i, \pi(X_j) = \pi(X_i)]}{\frac{1}{N}\sum_{j=1}^{N} 1[\pi(X_j) = \pi(X_i)]} \tag{8}$$

With the propensity score the estimated IPS reward of a certain policy $\hat{R}_{IPS}(\pi, Y)$ can be formulated, as in Equation (9). The indicator function $\mathbb{I}(W_i = \pi(X_i))$ denotes the treatment that ought to coincide with the personalized policy's treatment and is equivalent to one if so.

$$\hat{R}_{IPS}(\pi, Y) = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{\mathbb{I}(W_i = \pi(X_i)) \cdot Y_i}{\hat{e}_{\pi(X_i)}(W_i)}\right) \tag{9}$$

4.5.2 Doubly Robust Estimator

The doubly robust estimator is employed as an additional method of evaluation to reach more complete results. Dudík et al. (2011) shows that the DR estimator combines the estimation power of the IPS estimator and that from the direct outcome regression model (DM). The DM forms an estimate of the expected outcome conditioned on the pre-treatment variables and policy assigned treatment. Note that the DM method estimates without any knowledge of the policies. Therefore, it might occur that the approximating of the estimate is mainly in areas that are not relevant to the reward. Since the DR estimate is a broadening of the IPS estimate we try to evaluate the policies better. This hybrid form of the models ensures that the estimator remains consistent as long as the models are well specified. This enhances the reliability and robustness of the policy evaluation. The DR estimator is given in Equation (10). Where $\hat{Y}_z$ is the expected reward Y_i given treatment z. Compared to the IPS estimator, the DR estimator will be accurate if at least one of the estimators, $\hat{e}_{\pi(X_i)}(W_i)$ or $\hat{Y}_z$ are accurate.

$$\hat{R}_{DR}(\pi, Y) = \frac{1}{N}\sum_{i=1}^{N}\left(\frac{(Y_i - \hat{Y}_{W_i}(x))\mathbb{I}(W_i = \pi(X_i))}{\hat{e}_{\pi(X_i)}(W_i)} + \hat{Y}_{\pi(X_i)}\right) \tag{10}$$

5 Results

From the statistics in Section 3 it became clear that corporations gain conversion and visit rates by assigning advertising mailings. Consequently, the personalized policies will now be explore in further detail, as previously mentioned in Section 4. The outcomes of the personalized policy evaluation and the advantages over the non-personalized policies will be covered in this section.

Two uniform policies are defined, for the non-personalized policies. π_0 is the uniform policy that assigns no treatment to all its customers. On the other hand, π_1 assigns a treatment to every customer. With the use of the IPS and DR estimators, the rewards are estimated for each policy and can be found in Table 2. High rewards indicate a higher outcome rate, demonstrating how well the model adapts the policy to the specific characteristics and preferences of each customer. What is also shown in the table is the number of treated customers. That is the number of customers that were assigned a treatment, within the sample data. Since the π_1 treats all customers, the treatment column is set to the size of the data, which for our case is 30.000 for the test data, and vice versa 0 for π_0.

Focusing on the IPS evaluation first, it can be seen from Table 2, that the uniform treatment approach of sending treatment to all customers, resulted in mostly higher or comparable estimated reward rates for the outcome variable. Such that most personalized policies, despite their complexity, failed to deliver significantly better outcomes. Only π_{Lasso} for conversion, and $\pi_{XGBoost}$ for visit were able to estimate a higher IPS reward. Compared to the IPS, the DR evaluation looks somewhat brighter. For visit the personalized policies of π_{LR}, π_{Lasso}, $\pi_{XGBoost}$ and, $\pi_{RForest}$ do perform better compared to the uniform ones. For conversion this applies for π_{Lasso} and $\pi_{XGBoost}$. Policies based on heterogeneous treatment effects estimators, causal tree and causal forest, do not personalize treatment assignment and end up giving each customer a treatment equal to one, that is $\pi_1 \equiv \pi_{CTree} \equiv \pi_{CForest}$. π_{CART} performs the poorest, as it ends up treating no single customer. Further, something that could be seen as remarkable is that, for optimizing the hyper-parameters of causal forest the IPS and DR estimator reward degraded. For example, without tuning an IPS reward for conversion of 0.0032 with 24910 customers assigned to the treatment was found. This means that optimizing the hyper-parameters for causal forest deteriorates the estimated rewards.

Looking at the rewards on the train data in Table 3, it can be found evident that policies based on non-heterogeneous treatment effects function better than the uniform ones. Again, π_{CTree}, $\pi_{CForest}$, and π_{CART} fail to assign personalized treatment. That the rewards perform better on the train data than on the test data is not surprising. Outperforming train data is a common phenomenon within machine learning, where the primary cause is overfitting. Due to overfitting, the model does not generalize well on new data(in this research test data). The fact that the random forest performs well on train data but not on test data may be due to model bias. Decision tree methods, like random forest, are very sensitive to overfitting. Having a deep decision tree can perform poorly on the test data but well on the train data. The slight variations and the sample size in the test and train data could also be a factor in the observed discrepancies in the results. Therefore, it can be concluded that a prediction made using a model that has been trained on train data will perform better. As a result, more personalized policies perform better than the uniform policies.

Table 2: IPS and DR rewards on test data.

	Conversion			Visit		
Policy	IPS	DR	Treated	IPS	DR	Treated
π_{LR}	0.002514462	0.002481429	27531	0.04788889	0.04740907	29836
π_{Lasso}	0.003142166	0.003213283	5624	0.04803754	0.04774164	13798
π_{CART}	0.002205558	0.002205558	0	0.03573004	0.03573004	0
$\pi_{RForest}$	0.002527458	0.002309057	5373	0.046085658	0.04608566	23437
$\pi_{XGBoost}$	0.003098368	0.003193879	24547	0.04788889	0.04849897	29823
π_{CTree}	0.003102175	0.002900225	30000	0.04861384	0.04459967	30000
$\pi_{CForest}$	0.003102175	0.002900225	30000	0.04861384	0.04459967	30000
π_1	0.003102175	0.002900225	30000	0.04861384	0.04459967	30000
π_0	0.002205558	0.002205558	0	0.03573004	0.03573004	0

The relationship between the estimates of treatment effects and the performance of the policy is examined. Figure 1 shows the cumulative density function (CDF) for conversion and visit for all the designed personalized policies. Low heterogeneity, that occurs within

Table 3: IPS and DR rewards on train data.

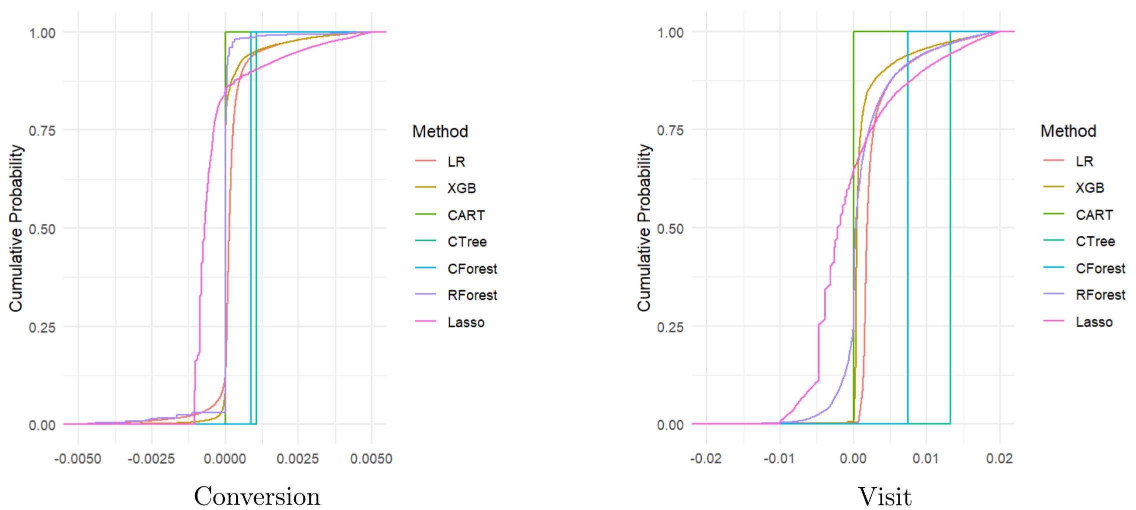
	Conversion			Visit		
Policy	IPS	DR	Treated	IPS	DR	Treated
π_{LR}	0.003303817	0.003568392	64202	0.04954432	0.04887055	69545
π_{Lasso}	0.003074454	0.003103019	12745	0.04881205	0.04855096	32209
π_{CART}	0.002003434	0.002368053	0	0.036061820	0.03964438	0
$\pi_{RForest}$	0.003762671	0.003812206	12415	0.065389023	0.06333313	54120
$\pi_{XGBoost}$	0.003663866	0.003488562	69200	0.04988612	0.04937113	65274
π_{CTree}	0.003074700	0.003041969	70000	0.049245606	0.04847139	70000
$\pi_{CForest}$	0.003074700	0.003041969	70000	0.049245606	0.04847139	70000
π_1	0.003074700	0.003041969	70000	0.049245606	0.04847139	70000
π_0	0.002003434	0.002368053	0	0.036061820	0.03964438	0

the CART, causal forest, and causal tree methods, is clearly visible within the CDF graph. This refers to the vertical lines positioned at or to the right of zero. This lack of variability indicates that these policies do not differentiate sufficiently between individual users, leading to poor performance, as these policies cannot properly improve customer's responses. It does not come as a surprise that the estimated rewards of π_{CART}, which moves vertically on zero, is equal to π_0. As π_{CART} does not differentiate between the customers, it assigns all customers to no treatment. Similar applies for $\pi_{CForest}$ and π_{CTree}. Due to the low heterogeneity and a constant difference in predicted probability, that is greater than zero, it assigns all customers to a treatment. Resulting in an equal reward as π_1. In contrast, the treatment effect estimates based on Lasso exhibits the highest degree of heterogeneity, as can be seen in the CDF graph. This indicates that this model captures a wide range of treatment effects across individuals, leading to an improvement in estimated reward (Table 2). However, this is not the case for the outcome of the variable visit in the IPS estimate. This may suggest that it suffers from overfitting.

In the mid-range of Figure 1, the CDFs of treatment effect estimates from XGBoost, logistic regression, and random forest can be found. This indicates that these policies show a moderate degree of heterogeneity. Thus, these policies can effectively design personalized policies, while not causing overfitting. Nevertheless, this only applies to XGBoost, given the poor IPS and DR rewards of logistic regression and random forest. The ability of XGBoost to show an appropriate level of heterogeneity in treatment effect estimates, while still performing good in the evaluation, demonstrates its good performance. Thus, taking the DR and the CDFs of treatment effect into account, XGBoost optimizes personalized advertisement most effectively, while not suffering from the negative effects of overfitting.

In a deeper analysis of the CART model, its poor performance can be explained in two different scenarios. As the tree of the CART model is pruned with a very small complexity parameter, it mainly focuses on the $f1$ variable, leaving the treatment variable entirely out

Figure 1: CDF of Estimated conditional average treatment effects for conversion and visit on the test data



of the tree. As a consequence, no distinction can be made between a treatment or not. When there does not exist a distinction between treatments, every customer is assigned to no treatment. If the model is used without any pruning, the treatment variable is not significant. Even though the treatment variable is conditional, it influences only a very small part of the tree, with the majority of the other variables being more prominent. Further, when the tree is not pruned, it becomes excessively large, leading to substantial bias and overfitting. This causes that the treatment variable is not incorporated into the tree.

As mentioned by Yoganarasimhan et al. (2023), one would expect that personalized policies should work better than non-personalized policies. Nevertheless, π_1 performs equivalent to almost all policies or even better, especially on the IPS rewards. This raised the question of whether the poor performance is due to the policies or some other factor. Looking at the statistics of the data, it can be concluded that the poor performance in certain policies does not depend on the policy itself, but is due to the data. Since multiple random samples were tested and gained similar outcomes. A closer inspection of the sample reveals significant variation within the different variables. The statistics of the variables can be seen in the appendix in Table 6. Here it can be seen that for some variables the kurtosis and skewness differ a lot from the normal distribution. For eight of the twelve pre-treatment variables, the kurtosis exceeds the normal value between $[-3, 3]$. For example, the $f1$ variable has a kurtosis of 321.6. This normally suggests the presence of outliers. However, when zooming in on the values of these particular poor performance variables, we find something noticeable. Namely, almost all the values within a variable are the same, which can be seen in the frequency histogram in Figure 2. The specific distribution of values within $f1$ are shown in Table 7. The unequal distribution found in a variable ensure that, whenever a certain value deviates from the 'most common' value, it becomes immediately an outlier. For all variables that exceed the interval of the normal distributed kurtosis, the same result is found. Thus, there exists a coherence between the extreme value of the kurtosis and the number of values that are equal within a variable. Namely, if all values are virtually the same, a single deviation is immediately regarded as an extreme outlier. Because of all this, data trimming has become a very impossible task, only data clustering could be a sufficient option. To ensure that this high kurtosis is not due to the sample data, the statistics of the entire data were analyzed and found similar high kurtosis for the same variables.

In short, this high frequency of the same values within a variable that occurs in eight of the twelve variables results in homogeneity. Earlier we saw in Figure 1 that under some policies this was already occurring. This homogeneity among variables causes personalized policies to perform poorly. This is because personalized policies per definition target the heterogeneity of customers, in order to make a prediction for each person, with unique characteristic properties, and finally assign this specific prediction to this person. Therefore, because this heterogeneity is missing, some personalized policies are unable to create a better policy than the non-personalized policy. This suggests that advertising firms may sometimes be better off adopting uniform policies instead of investing in complex personalizing policies based on those methods. Further, in the comparison of evaluation policies, it was found that the DR generally performs better. This is because the DR uses the IPS with a corrective factor. Especially for visit their is an improvement present in the policies compared to the uniform policies.

6 Conclusion

This research compares personalized email advertising policies. Where several personalized policies are evaluated using the DR and IPS estimators. After evaluation, it becomes clear that the personalized policies are better evaluated using the DR estimator. As the DR estimator implies the IPS with a corrective factor. Within this specific DR estimator the $\pi_{XGBoost}$ and π_{Lasso} performed best, resulting in an increase in conversion and/or visit rate, to the extent that there are policies for which personalization is profitable. As $\pi_{XGBoost}$ not only performs better than the uniform policies but also addresses a higher amount of treatments to the customers, it forms the optimal personalized policy for online advertisement.

The CDF analysis shows that methods such as CART, causal forest, and causal tree perform poorly due to high homogeneity. In contrast, XGBoost along with Lasso, demonstrated a moderate level of heterogeneity, balancing personalization without overfitting, and confirming their better performance. Finding heterogeneity among the customers was very difficult due to data characteristics. Extremely high kurtosis is formed because the features within the variables are almost all the same, hindering the effectiveness of personalized policies. This results in a disappointing performance of some personalized policies. Finally, the extremely poor performance of the CART model can be explained either through excessive pruning or lack of significant inclusion.

Further research should focus on improving data quality, by employing advanced preprocessing techniques, and developing adaptive learning models to be able to adjust to changing data characteristics and customer behaviors. Through this implementation, personalized marketing strategies can be expected to be further optimized, guaranteeing better outcomes for advertising companies.

References

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

Ban, G.-Y. and Keskin, N. B. (2021). Personalized dynamic pricing with machine learning: High-dimensional features and heterogeneous elasticity. *Management Science*, 67(9):5549–5568.

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

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

Dudík, M., Langford, J., and Li, L. (2011). Doubly robust policy evaluation and learning. *Machine Learning*, 1.

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

Kang, J. D. and Schafer, J. L. (2007). Demystifying double robustness: A comparison of alternative strategies for estimating a population mean from incomplete data.

Perlich, C., Dalessandro, B., Raeder, T., Stitelman, O., and Provost, F. (2014). Machine learning for targeted display advertising: Transfer learning in action. *Machine learning*, 95(1):103–127.

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

Yan, J., Liu, N., Wang, G., Zhang, W., Jiang, Y., and Chen, Z. (2009). How much can behavioral targeting help online advertising? In *Proceedings of the 18th international conference on World wide web*, pages 261–270. Association for Computing Machinery.

Yoganarasimhan, H., Barzegary, E., and Pani, A. (2023). Design and evaluation of optimal free trials. *Management Science*, 69(6):3220–3240.

7 Appendix

7.1 Hyper-parameter Optimization for the Models Estimated

In addition to the optimized hyper-parameters for the Lasso model, we do the same for the other models. Where for each model a five-fold cross-validation is used to optimize the hyper-parameters. The optimization of the models chance be described as follows:

- In the lasso model, The standard cross-validation procedure is implemented in the *glmnet* package in R. In our case the best λ is equal to 2.12×10^{-4} for conversion and 6.92×10^{-4} for visit.

- For the CART model, the GridSearchCV was used from the sklearn Python package. For which we found an optimal complexity parameters (ζ of 0.307 for conversion and 2.2 $\times 10^-4$.

- For the Random Tree model, the GridSearchCV from the sklearn Python package was also used. For the Random Forest we optimized three hyper-parameters. (1) η_{tree}, the number of trees over which the ensemble forest is build. (2) max_f, the maximum number of features the algorithm try for any partition. (3) η_{min}, the minimum number of samples required to partition an internal node.

  All three parameters are optimized within an interval, and is shown below.

  - $\eta_{tree} \in$ [10,200] and $\eta_{tree}^{*Conversion} = 28$, $\eta_{tree}^{*Visit} = 154$.
  - $max_f \in$ n, sqrt(n), log2(n) and $max_f^{*Conversion} = log2(n)$, $max_f^{*Visit} = sqrt(n)$.
  - $\eta_{min} \in$ [1,10] and $\eta_{min}^{*Conversion} = 3$, $\eta_{min}^{*Visit} = 9$.

- For the XGBoost method follow the hyper-parameter tuning from Yoganarasimhan et al. (2023), where they optimize three parameters. (1) α is an L1 regularization parameter. (2) η is the learning rate. (3) d_{max} is the maximum depth of the trees. For the optimization the grid expansion function from the "caret" package in R is used. The optimal values with its intervals are as follows:

  - $\alpha \in$ 5,10,25,50,100,150,200 and $\alpha^* = 50$
  - $\eta \in$ 0.01,0.1,0.3,0.5 and $\eta^* = 0.1$
  - $d_{max} \in$ 4,6,8,12 and $d_{max}^* = 4$

- For the Causal Tree we make use of the "causalTree" and "caret" packages in R. Where the complexity parameter ζ is tuned using "causalTree" and the minimum number of treatment and control observations in each leaf (q) is tuned using grid-search.

  - $q \in$ [1,100] and $q^* = 1$

- $\zeta \in$ [0.0001,0.01] and $\zeta^* = 5 \times 10^{-4}$

- Last, three hyper-parameters were tuned for the Causal Forest. (1) $mtry$, the number of variables tried for each split. (2) max_imb, the maximum allowed imbalance of a split. (3) q, the minimum number of observations per condition(control, treatment) in each partition. These hyper-parameters were tuned using the sklearn Python package, for which we found the following:
  - $q \in$ [1,15] and $q^{*Conversion} = 2$, $q^{*Visit} = 5$
  - $max_imb \in$ [0,0.05] and $max_imb^{*Conversion} = 0.08679$, $max_imb^{*Visit} = 0.3516$
  - $mtry^{*Conversion} = 5$, and $mtry^{*Visit} = 6$

Table 4: Summary statistics of conversion and visit.

	Full data		Sample data	
Variable	Mean	Standard deviation	Mean	Standard deviation
Visit	0.0470	0.212	0.0471	0.212
Conversion	0.00292	0.0539	0.00293	0.0541

Table 5: Summary Statistics of Conversion and Visit rates, and Treatment Assignment for the CRITEO-UPLIFT1 dataset

CRITEO-UPLIFT1 dataset	No treatment	treatment	Total
Number of observations (N)	2,096,937	11,882,655	13,979,592
Percent of total observations	14.999	85	100
Number of visits	80,105	576,824	656,929
Percent of total visits	12.194	87.806	100
Number of conversions	4,063	36,711	40,774
Percent of total conversions	9.965	90.035	100
Visit rate within group (in %)	3.820	4.854	4.699
Convertion rate within group (in %)	0.194	0.309	0.292

Table 6: Statistics of pre-treatment and treatment variables in test data

	Mean	Sd	Median	Min	Max	Range	Skew	Kurtosis	Se
f0	19.609	5.369	21.930	12.616	26.745	14.128	-0.245	-1.626	0.031
f1	10.070	0.105	10.060	10.060	15.070	5.010	14.219	321.622	0.001
f2	8.444	0.298	8.214	8.214	9.052	0.838	0.824	-0.975	0.002
f3	4.179	1.336	4.680	-5.318	4.680	9.998	-3.179	10.395	0.008
f4	10.339	0.352	10.281	10.281	18.353	8.072	9.078	108.263	0.002
f5	4.030	0.418	4.115	-4.944	4.115	9.060	-6.830	63.383	0.002
f6	-4.176	4.596	-2.411	-25.334	0.294	25.629	-1.137	0.777	0.027
f7	5.099	1.197	4.834	4.834	11.993	7.159	4.559	19.555	0.007
f8	3.933	0.057	3.972	3.665	3.972	0.307	-1.600	1.824	0.000
f9	16.049	7.029	13.190	13.190	65.006	51.816	2.818	7.747	0.041
f10	5.333	0.167	5.300	5.300	6.474	1.173	5.349	27.942	0.001
f11	-0.171	0.025	-0.169	-1.023	-0.169	0.855	-15.269	304.275	0.000
Treat	0.849	0.358	1.000	0.000	1.000	1.000	-1.948	1.794	0.002

Table 7: Distribution of values within variable $f1$ in test data

Value	10.0596	10.6795	11.1193	11.4604	11.7392
Frequency	29611	281	59	27	11
Value	11.9748	12.1789	12.5201	12.7988	15.0696
Frequency	2	5	2	1	1

8 Programming code

The code to obtain all results consists of several parts. $TestData$ and $TrainData$ represents the used sample data. The designed personalized policies are constructed in Cart.py, causalForest.py, causalTree.R, Lasso.R, Linear.R, RandomForest.py, and XGBoost.R. Where the hyper-parameters optimization is incorporated into the files. To evaluate the personalized policies the estimators are given in InversePropensityScoreestimator.R and DoublyRobustestimator.R. A more detailed explanation of the codefiles can be found in the ReadMe.txt.

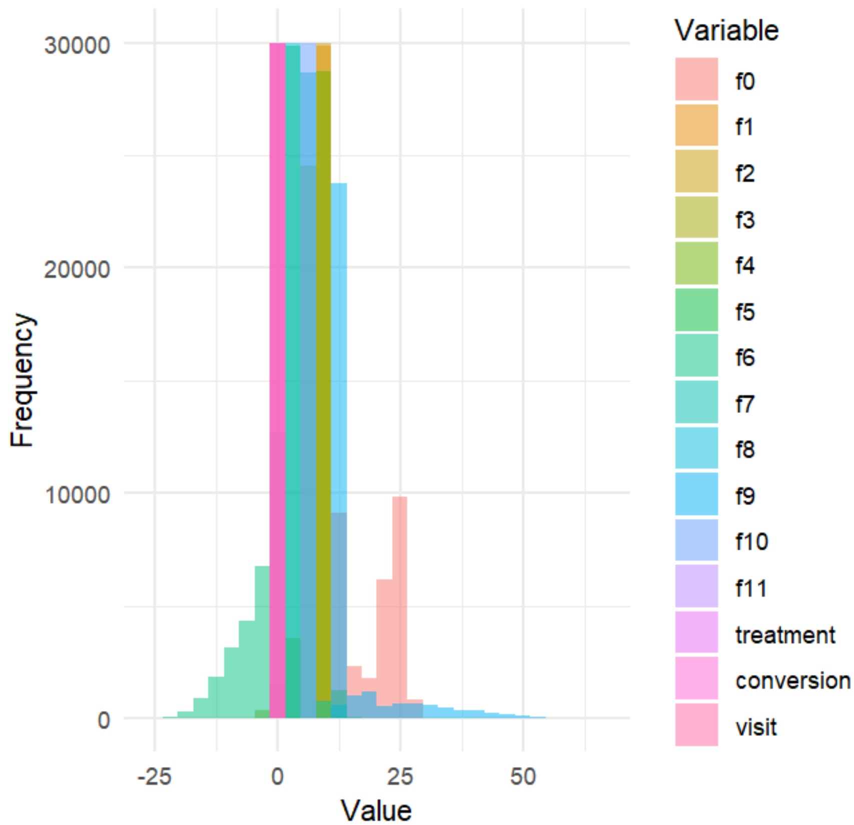


Figure 2: Histogram of distribution of values within the variables in test data