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

Optimizing Personalized Social Media Advertisements

Andrea Jaspers (558662)

Abstract

This research aims to analyze several machine learning methods for optimizing the effectiveness of customized social media advertisements. The data is retrieved from a randomized field experiment that investigates the extent to which the profits of social advertisements are influenced by the heterogeneous characteristics of the advertised products and the social media users. In the context of this paper, social advertising is illustrated by the degree to which connections of the viewer of the advertisement are engaged with the ad, which is reflected by giving the advertisement a "like" on social media. By combining a wide range of heterogeneous features of both the advertised products and the social media users, the optimal treatment is determined that indicates the number of friends' likes that should be presented below the ad. Several machine learning techniques are used to formulate the optimal allocation of treatments for the user-advertisement pairs, including a logistic regression, lasso, classification and regression tree (CART), random forest, XGBoost, causal tree and causal forest model. In addition to this, an ensemble learning model is constructed that combines the three best performing models with the aim of improving the overall predictive power. The personalization techniques are evaluated using the inverse propensity score (IPS) and doubly robust (DR) off-policy evaluation estimators, which result in corresponding conclusions about the performance of the models. The ensemble learning model outperforms all of the other models and increases the predicted click-rates of the personalized advertisements by a range of 13.6% - 19.8% for the IPS and DR estimates.

Supervisor:	H. Deng
Second assessor:	Prof. Dr. ir. R. Dekker
Date final version:	1st July 2024

1 Introduction

The prominence of social media in the marketing field has emerged substantially over the past few years, due to the fact that it provides an extremely attractive platform for marketing campaigns to reach their target audience in an easy and effective way (Alalwan, 2018). In addition to this, the simplicity and efficiency of social media advertising campaigns allows companies to grow relationships with their potential customers by engaging with individual preferences, at a relatively low cost compared to other marketing strategies such as banners or commercials (Raudeliūnienė et al., 2018). Therefore, the potential of social media advertising is auspicious, and customizing the social media advertisements provides an interesting field of research for further enhancing the commercial benefits (Van Doorn & Hoekstra, 2013).

Huang et al. (2020) investigate a specific form of social media advertising, in which social cues, i.e. "likes" for the ad given by the connections of the social media users, are presented below the ad to encourage the ad engagement of the viewer. The paper by Huang et al. (2020) analyzes the extent to which the heterogeneity across the products and the social media users affects the payoff of the advertisements in which the social cues are used. For their research, they conduct a randomized field experiment that measures the effectiveness of advertisements that are presented on the social media platform "WeChat", which reached more than 37 millions users during their experiment and aims to investigate the differences in the reaction rates across 71 advertised products. The social media platform WeChat is a multifunctional social media, messaging and mobile payment app that quickly evolved into one of the most popular social media apps in the world after its launch in 2011. In 2023, more than 1.3 billion users used the platform to send text and voice messages, engage in video calls and update their connections on the social feed called "Moments", which is also used to present advertisements (Oberlo, 2024). The platform was developed by the Chinese company Tencent and is an essential part of daily life for millions of users around the world, although the majority of the users, roughly 60%, are located in China (Oberlo, 2024).

The research of this paper focuses on determining the optimal strategies for personalizing the number of social cues that are presented below an advertisement, with the aim of maximizing the total number of clicks on the ad and magnifying the commercial potential of the marketing campaigns. Data from the large-scale randomized field experiment of Huang et al. (2020) is used to construct customized advertisements for users of the social media platform WeChat, in order to increase the ad engagement and positively influence the adoption of the advertised products. Several machine learning methods are used to assign either zero, one or multiple social cues (e.g. likes of connections) to the user-advertisement pairs to inflate the rate of response for the specific ad. The main research question is therefore formulated as: *how to design and evaluate optimal personalized social advertisements based on heterogeneous user and product characteristics?*

The methods that are used to design the personalized advertisements are similar to the ones used by Yoganarasimhan et al. (2023), who use several machine learning and causal inference techniques to customize free trial periods for a software company based on individual-level predictions of the software subscription rates. For this specific research, the prediction of the outcome of interest is related to whether the viewer of an ad decides to click on the ad after seeing it for the first time. Multiple techniques are used to generate the personalization strategies,

including a logistic regression, lasso method, classification and regression tree (CART), random forest and XGBoost algorithm. Furthermore, for the causal inference methods, causal tree and causal forest (generalized random forest) techniques are used to estimate the heterogeneous treatment effects. In addition to this, an ensemble learning model consisting of the best performing models is investigated as an extension to examine whether this combined model possibly further enhances the profits of personalizing the advertisements. The performance of the personalized advertisements is evaluated by considering the inverse propensity score (IPS) off-policy estimator, following a similar approach to the evaluation method used by Yoganarasimhan et al. (2023). Additionally, the doubly robust (DR) estimator is analyzed in order to be able to draw more definite conclusions about which model is optimal, as it is more robust to model misspecifications and noise in the rewards than other off-policy estimators (Neugebauer & van der Laan, 2005). The main finding of this paper indicates that the ensemble learning model generates the highest predictions for the estimated click-rates of the customized social media advertisements, by anticipating improvements between 13.6% - 19.8% for the IPS and DR estimators, compared to the original click-rates of the advertisements that were not customized. Therefore, combining several machine learning techniques into an ensemble model results in favourable outcomes regarding the commercial potential of personalized advertisements for this dataset.

By gaining insights into the mechanisms behind which different types of consumers respond to advertisements of different types of products, marketing managers and sales departments can enhance their response rates and maximize their revenues, allowing them to optimize their business operations. The research of Huang et al. (2020) investigates these effects, but does not examine the possibilities of delivering personalized social advertisements that are customized according to the observed heterogeneous characteristics of the products and the WeChat users. Therefore, this research adds to the existing literature by investigating a different approach to social media marketing, which combines the features of the social media users and the advertised products to produce personalized social media advertisements that generate higher response rates than advertisements that are not tailored at the individual level. Moreover, combining several customization techniques in an ensemble model has not been studied extensively in the literature yet and therefore allows for the evaluation of an additional method that has not been investigated in the field of customized social media marketing. In addition to this ensemble learning technique, the research of Yoganarasimhan et al. (2023) is extended by evaluating the off-policy doubly robust estimator, which allows for a more robust comparison of the effectiveness of the personalization methods. By comparing the results of two off-policy evaluation metrics, the final conclusions regarding the performance of the models are more decisive and trustworthy.

The remaining part of this paper adheres to the following structure. First, the existing literature related to using machine learning techniques for marketing purposes is discussed in Section 2. Section 3 provides a description of the dataset and presents the summary statistics of the variables that are examined. The methodological approach that is used to implement and evaluate the customization techniques is described in Section 4. The results for the IPS and DR estimates of the click-rates are discussed in Section 5. Finally, Section 6 provides conclusions regarding the optimal model for personalizing the social media advertisements.

2 Literature review

Customization strategies are a rapidly growing field of research in the marketing literature, exemplified by the use of personalized emails or website content to directly target the desired audience. In this context, machine learning techniques are extremely useful for analyzing large amounts of data on consumer behaviour and tailoring marketing strategies accordingly. For instance, Sun et al. (2019) investigate the effect of various price promotions for a large retailer over a 5-month period and find that a partially profiled lasso model is optimal for relating promotion results to heterogeneous product characteristics, which follows a similar approach to the optimal lasso method used by Yoganarasimhan et al. (2023). Optimizing personalized treatments is further explored by Guelman et al. (2015), who construct causal conditional inference trees and forests, similar to the methods of this paper, with the aim of customizing treatment choices at the individual level. The results indicate that the proposed random forest method outperforms the alternatives for selecting the best targets for cross-selling an insurance product and retaining the customers in the long-term.

Another study in which machine learning algorithms are used to personalize marketing strategies is conducted by Deligiannis et al. (2020), who implement the XGBoost regression method to determine the timing and content of customized marketing messages. The evaluation results indicate that the algorithm contributes to significant improvements regarding the marketing effectiveness, compared to the standard approach of scheduling product repurchase reminders at fixed time intervals. Van Wezel & Potharst (2007) conclude that the application of various ensemble learning methods improves the prediction quality of flexible models like decision trees, which is in line with the conclusions for the ensemble model of this paper.

Song et al. (2023) investigate the potential of combining multiple models, in this case the whale optimization algorithm (WOA), a lasso technique and the extreme gradient boosting (XGBoost) framework to improve the overall predictive accuracy of estimating the amount of gas emission in a mining face. Their ensemble modelling technique reduces the mean squared error and root mean squared error of the predictions by roughly 8% and 4% respectively, resulting in substantial health benefits for the mine workers. A different context in which a combination of machine learning models is used to enhance the final predictive abilities is set by Yigit et al. (2020), who use an XGBoost-lasso ensemble modelling approach to establish an advanced football player value assessment model that can be used to support the transfer decisions of football clubs. Overall, the ensemble model is discovered to improve the quality of the valuations of the football players, compared to the performance of the individual machine learning models.

The use of off-policy evaluation techniques is examined by Kiyohara et al. (2024), who analyze several common off-policy estimators, including the inverse propensity score (IPS) estimator. The main finding of their research indicates that a latent IPS estimator is optimal for the off-policy evaluation of the dataset in the paper, as it substantially reduces the variance of the estimator without imposing restrictive assumptions on the reward function structure. Selecting the optimal estimator for off-policy evaluation is further explored by Neugebauer & van der Laan (2005), who compare the performance of the G-computation, inverse probability of treatment weighted (IPTW) and doubly robust estimator, and conclude that the doubly robust estimator is the most robust for point treatment studies, indicating that it is preferred when evaluating

the causal inference methods that are analyzed in this paper.

3 Data

The data for this research is extracted from the randomized field experiment that was performed by Huang et al. (2020), which consisted of a 21-day period that started in December 2015 and was conducted on a random sample of more than 37 million users of WeChat. The analysis was targeted at advertisements that are displayed in the news feed "Moments" on WeChat, which is similar to for example the Facebook news feed and allows users to post images, videos or textual updates for their friends and connections. By liking, commenting or clicking on the advertisements that are presented on the news feed, WeChat users can express their attitude towards the ad and articulate their preferences and opinions. For this research, only the likes of connections for the specific ad are used to represent the social cues, because comments can vary widely in their content and therefore do not allow for a clean estimation of the amount of social influence. If a WeChat user clicks "like" below a specific ad, they present their appreciation of the ad to their first-degree social connections, who see the like appear immediately afterwards on their WeChat Moments feed. The social advertising effectiveness is measured by the degree to which social cues result in a reaction of the viewer of the ad, which is measured by the number of click-throughs that lead to for example the advertisers' profile page, a landing page or a product photo. An example of WeChat Moments advertisements with different social cues (e.g. likes) is presented in Figure 1 below.

During the experiment, users were randomly matched with advertisements that were assigned to one of three groups with an equal probability of 8%; one control group without displayed likes (no social cues), one treatment group with a maximum of one displayed like (one social cue) and a second treatment group where multiple likes of various connections were displayed below the ad (multiple social cues). The other users remained out of the experiment with a probability of 76%. Figure 1 (Huang et al., 2020) represents the differences between the control and treatment groups through examples of social advertisements that were presented for each group.

Note. This figure illustrates the control condition (without any social cue), the first treatment group (with a maximum of one like), and the second treatment group (with the organic number of likes), from left to right, respectively.

Figure 1: Example of WeChat Moments ad for the control and two treatment groups (Huang et al., 2020)

Because the presence and number of social cues that were displayed for every ad were assigned randomly, the estimates of the social influence effects are unbiased. Every ad stayed in the news feed for 48 hours and users were only allowed to see one ad at a time, which decreases the exposure to statistical interference of the different ads. After 48 hours, users could be assigned to a new treatment/control group if a new ad was presented to them. Users were uninformed and unaware that they were participating in the experiment, so the possibility of self-selection is ruled out in the experiment.

Eventually, the experiment generated a sample of advertisements for 4.884.070 users and 71 distinct products across 25 product categories. The product categories cover a wide spectrum of classification possibilities, including for instance "Smart phone", "Home furnishing" and "Tourist spot". The frequencies of the unique values of the product categories are relatively equal and range from 1% to 10% of the total dataset, which indicates that each unique value is well-represented. Therefore, the distribution of the unique values of the product categories is representative and the variable is not too granular. Due to a non-disclosure agreement, the complete dataset of the experiment is not available, so a randomized sub-sample consisting of 10.000 observations is used to perform the research of this paper.

Every user-advertisement pair is identified as a unique observation, which is therefore specified as a match between the user and the advertisement that is presented to the user. The simulated dataset contains information about the characteristics of the products ($\mathbf{Z}$) and the WeChat users ($\mathbf{X}$) which is captured by matrix $\mathbf{F}$ for the users and products. Every row of the matrix, denoted by $\mathbf{f_{ij}}$, indicates the user and product characteristics of user i that is presented with an advertisement of product j. The allocated treatment/control group is denoted by W_{ij} for user-ad pair ij, which indicates either the control group, or treatment group 1 or 2. The outcome for the user-ad pairs ij is reflected by the binary variable $Y_{ij}(\mathbf{f_{ij}}, W_{ij})$, which is a dummy variable that represents whether user i has clicked on the specific advertisement j after seeing it for the first-time ($Y_{ij} = 1$). A more detailed description of the covariate matrix $\mathbf{F}$, treatment W_{ij} and outcome $Y_{ij}(\mathbf{f_{ij}}, W_{ij})$ is specified in the theoretical framework in Section 4.1.

Regarding the characteristics of the users $\mathbf{X}$, the gender of the users is given and the ages are separated into 8 different age categories. In addition to this, the cities of the users are identified as either first, second or third-class cities, which is typically used in China to classify cities based on for example population size and the amount of economic development. The gender, age and city classification of the users' connection who liked the social advertisement is also provided. If a user was assigned to treatment group 2 and received multiple social cues (e.g. likes), the gender, age and city classification of the connection whose like was presented first is given. The summary statistics for the pretreatment categorical variables of the users are presented in Table 1, where the percentages for the shares of the top subcategories are displayed and the first subcategory includes the largest number of observations, relative to the other subcategories. For the complete simulated dataset, the percentages of all subcategories were relatively equal, which is also noticed from the percentages in Table 1. The summary statistics for the characteristics of the connections whose likes were shown below the ads are similar to the summary statistics of the users who viewed the ads, which are presented in Table 1.

Variable	# of subcategories	Share of top subcategories		
		First	*Second*	*Third*
User age group	8	12.7% (46-50)	12.7% (26-30)	12.7% (50+)
User city	3	34.1% (city level 2)	33.1% (city level 1)	32.8% (city level 3)
Gender	2	50.3% (males)	49.7% (females)	

Table 1: Summary statistics of the pretreatment categorical variables of the users (X_i)

Regarding the pretreatment variables for the products **Z**, the products are identified as either an experience or a search good and as a status or a non-status good, besides the fact that they are classified into one of the 25 product categories. Additionally, the brands of the goods are classified as either large or small, by using a dummy variable that indicates whether the brand belongs to the 100 Best Global Brands, as rated by Interband. Table 6 in Section A of the Appendix provides descriptions of the product classifications and presents an overview of the user-ad pairs that were assigned to the different product types. The brands of the goods are classified as large brands for 16.2% of the products or as small brands for 83.8% of the products.

The summary statistics for the original reaction rates of the control and treatment groups are presented in Table 2, which provides an overview of how many users decided to click on the ad after it was first presented to them, for every treatment group. The first row of Table 2 indicates that the number of observations that is included in every control / treatment group is relatively equal. Furthermore, Table 3 presents the differences between the social cues for the control and treatment groups. The mean and standard deviations of the number of likes for the different treatment groups are based on the complete dataset of the randomized field experiment by Huang et al. (2020), as the number of displayed likes per user-ad pair are not specified in the simulated dataset that is used for this paper.

	Control group	Treatment group 1	Treatment group 2	Total
# observations N	3.360	3.356	3.284	10.000
% total observations	33.6%	33.6%	32.8%	100%
# first-time responses	1677	1661	1659	4997
% of total first-time responses	33.6%	33.2%	33.2%	100%
First-time reaction rate within group	50.0%	49.5%	50.5%	50.0%

Table 2: Summary statistics of the click-through rates for the control/treatment groups

Group	Mean # likes	SD # likes	Max # likes	Min # likes
Control	0.000	0.000	0	0
Treatment 1	1.000	0.000	1	1
Treatment 2	1.672	1.742	100	1

Table 3: Summary statistics of the displayed likes (social cues) for the complete dataset

4 Methodology

In order to design and evaluate the personalized advertisements, the data is partitioned into two independent samples with identical joint distributions, although some minor differences between the two data sets will remain because of the randomness in splitting a finite sample. For the training data, 70% is used to learn the model parameters and optimize the hyperparameters through cross-validation methods. The remaining 30% is used as the test data, which is used to evaluate the performance of the modelling techniques that are built while training the data. The 70%-30% split ratio is chosen to ensure that the training data is large enough to provide the model with sufficient information and that the test data is large enough to be able to properly evaluate the performance of the models.

First of all, personalized advertisements are constructed by using several machine learning models to predict outcome estimates (e.g. click-rates) and assign the treatment that generates the highest predicted outcome. The implementation of the logistic regression, lasso, classification and regression tree (CART), random forest and XGBoost algorithms are discussed in Section 4.2. The causal tree and causal forest algorithms are constructed by using an alternative approach, namely by generating estimates of the heterogeneous treatment effects, which are discussed in Section 4.3. In order to evaluate and compare the performance of the different personalized policies, the expected benefits of the personalized advertisements are evaluated by using the inverse propensity score (IPS) and doubly robust (DR) estimators. Doubly robust estimators use a propensity score model and an outcome model, which increases the robustness and potentially leads to lower variance estimates compared to the IPS estimator, which only uses a propensity score model. The off-policy evaluation methods are discussed in Section 4.4.

4.1 Theoretical framework for designing the personalized advertisements

The set of independent and identically distributed social media users is denoted by $i \in \{1, ..., K\}$, where every user is characterized by the pretreatment covariate vector $\mathbf{x_i} \in \mathbf{X}$ and $\mathbf{X} \in \mathbb{R}^{K \times p}$. Furthermore, every product $j \in \{1, ..., M\}$ is characterized by the pretreatment covariate vector $\mathbf{z_j} \in \mathbf{Z}$ and $\mathbf{Z} \in \mathbb{R}^{M \times q}$. Because all pretreatment variables represent distinct categories, each categorical variable is transformed into a set of dummy variables that indicates whether the specific category is present or not, also known as one-hot encoding. The covariate vector for each user-ad pair ij is defined as $\mathbf{f_{ij}} = [\mathbf{x_i} \ \ \mathbf{z_j}]^T \in \mathbf{F}$ and $\mathbf{F} \in \mathbb{R}^{(K+M) \times (p+q)}$, which includes all explanatory variables $\mathbf{x_i}$ and $\mathbf{z_j}$ for user i that is paired with product j. Every unique user-ad pair ij is defined as an observation, denoted by $ij \in \{1, ..., N\}$ with N indicating the total amount of observations. The treatment that user-ad pair ij receives is denoted by $W_{ij} \in \{0, 1, 2\}$, where $W_{ij} = 0$ indicates the control group without displayed likes, $W_{ij} = 1$ indicates treatment group 1 with a maximum of one displayed like and $W_{ij} = 2$ indicates treatment group 2 with multiple displayed likes. Eventually, the outcome for user-ad pair ij is denoted by the binary variable $Y_{ij}(\mathbf{f_{ij}}, W_{ij})$, which indicates whether user-ad pair ij with pretreatment variables $\mathbf{f_{ij}} = [\mathbf{x_i} \ \ \mathbf{z_j}]^T$ and allocated treatment W_{ij} clicks on the ad after seeing it for the first time.

The personalized treatment policy π is defined as the mapping between each user-ad pair ij and the assigned treatment W_{ij} such that each user-ad pair is allocated to one treatment. The

overall objective is to formulate the policy π such that the expectation of the outcomes and the reward function is maximized for every personalization modelling technique. For every policy π and corresponding outcome Y, the reward function is defined as $R(\pi, Y) = \frac{1}{N}\sum_{i=1}^{N}\mathbb{E}[Y(\mathbf{f_{ij}}, \pi(\mathbf{f_{ij}}))]$, which is maximized for every personalization technique and is used to determine which policy is ultimately optimal. Therefore, the optimal personalized policy is retrieved from

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

for every model, with Π indicating the total set of possible policies.

4.2 Design of personalized policies based on outcome estimators

In order to design the policies that are based on outcome estimates, the estimates of the expected outcome $\hat{f}(x = \mathbf{f_{ij}}, w)$ are obtained from the model $f(x, w) = \mathbb{E}[Y|\mathbf{f_{ij}} = x, W_{ij} = w]$ for every personalization technique. The optimal policy is then formulated by using the outcome estimator f to determine the maximum expected outcome for every observation ij and assigning

$$\pi_f(\mathbf{f_{ij}}) = w^{\star} \quad \text{with} \quad w^{\star} = \arg\max_{w \in W} \hat{f}(x = \mathbf{f_{ij}}, w) \tag{2}$$

and $W = \{0, 1, 2\}$, indicating the set of possible treatments.

4.2.1 Logistic regression model

Because the outcome variable $Y_{ij}(\mathbf{f_{ij}}, W_{ij})$ is binary, the logistic regression model is the first model that is analyzed in order to generate predictions for the probability that a user will click on a personalized advertisement. The logistic regression model is relatively simple and easy to implement, allowing for straightforward interpretations and efficient computations. For this research, a logistic regression model with first-order interaction effects is used to predict the observed outcome Y_{ij} for observation ij (e.g. user-ad pair) as a function of the pretreatment variables $\mathbf{f_{ij}}$, treatment variable W_{ij} and the interaction effects of both. Following the research of Yoganarasimhan et al. (2023), no higher-order interaction effects are included because this significantly increases model complexity and deteriorates the out-of-sample performance, which is critical for this research. The logistic regression model that is learned by using the training dataset is defined by the following specification:

$$Y_{ij} = \mathbf{f_{ij}}\beta_1 + W_{ij}\beta_2 + \mathbf{f_{ij}}W_{ij}\beta_3 + \epsilon_{ij} \tag{3}$$

where the coefficients β_1 capture the effect of the pretreatment variables $\mathbf{f_{ij}}$ on the outcome Y_{ij}, β_2 captures the effect of the different treatments on Y_{ij} and β_3 represents the interaction effects between the pretreatment variables $\mathbf{f_{ij}}$ and the treatments W_{ij}. The interaction effect measures the effectiveness of the different treatments across the combination of user and product characteristics for every user-ad pair ij. After the training dataset is used to learn the parameters of the logistic regression model, the estimates of the expected outcome $\hat{f}(x = \mathbf{f_{ij}}, w)$ are used to obtain the optimal treatment for every observation by formulating the policy that maximizes the expected click-rates.

4.2.2 Lasso model

In order to address potential overfitting and multicollinearity issues of the logistic regression model, the least absolute shrinkage and selection operator (lasso) model is investigated as it learns a simpler model that sets the coefficients of weak and highly correlated predictors to zero. The ad engagement outcome is once again modelled using $f(x, w) = \mathbb{E}[Y|\mathbf{f_{ij}} = x, W_{ij} = w]$, where $f(\cdot)$ is a lasso model that uses variables selection techniques to identify and exclude irrelevant predictors to minimize overfitting issues. Essentially, lasso estimates a logistic regression that minimizes the mean squared error by including an additional term that penalizes model complexity. The following lasso specification is used:

$$(\hat{\beta_1}, \hat{\beta_2}, \hat{\beta_3}) = \arg\min \sum_{ij=1}^{N} (Y_{ij} - \mathbf{f_{ij}}\beta_1 - W_{ij}\beta_2 - \mathbf{f_{ij}}W_{ij}\beta_3)^2 + \lambda(\|\beta_\mathbf{1}\|_1 + \|\beta_\mathbf{2}\|_1 + \|\beta_\mathbf{3}\|_1) \quad (4)$$

where $\|\beta_\mathbf{i}\|_1$ indicates the L1 norm of β_i that sums the absolute values of the coefficient vector. The regularization parameter λ is a hyperparameter that is learned from the training data, instead of fixing the value beforehand, and is used to control the strength of the penalty that is applied to the coefficients. A higher value of λ is related to a larger amount of shrinkage, which leads to an increased number of coefficients that are driven to zero. The optimal value for λ and further implementation details are discussed in Section B.1 of the Appendix. The estimates of the expected outcome $\hat{f}(x = \mathbf{f_{ij}}, w)$ that are generated by the lasso model are used to determine the optimal personalized policy $\pi_{lasso}(\mathbf{f_{ij}}) = w^\star$ for every observation ij, by assigning the treatment that maximizes the expected click-rate for every user-ad pair. In this way, either one, multiple or no likes of connections are allocated to the product advertisements that are presented to the users.

4.2.3 Classification and Regression Tree (CART) algorithm

The classification and regression tree (CART) algorithm is a tree-based model that aims to divide the covariate space into sub-regions and predict the outcome for the click-rates of the observations that are included in that specific sub-region. The objective of the algorithm is to find the optimal number of splits for creating the subsets, so the homogeneity of the target variable $Y_{ij}(\mathbf{f_{ij}}, W_{ij})$ is maximized within each subset. The partitioning technique is visualized by a decision tree structure, where each leaf of the tree symbolizes a sub-region that results in a specific output. The advantages of the CART model are portrayed by the fact that decision trees are easy to interpret and visualize and that the algorithm is capable of capturing non-linearity, which is in contrast with the previously discussed linear models. In addition to this, CART is insensitive to the scale of the variables and is capable of automatically selecting the variables that contribute to improving the predictive accuracy for splitting the nodes (Yoganarasimhan et al., 2023). The model specification for CART is represented by:

$$y = f(x, w) = \sum_{s=1}^{S} \rho_s I(x, w \in R_s) \quad (5)$$

where (x, w) indicates the pretreatment ($\mathbf{f_{ij}} = x$) and treatment ($W_{ij} = w$) variables, R_s represents the s^{th} subset of the S subsets that are used to separate the covariate space and ρ_s is the prediction of outcome y for subset s. The tree is built by minimizing a cost function, in this case the mean squared error, by using a greedy algorithm that grows the tree in steps. After the tree is built, a penalty term ξ is added in order to prevent overfitting by removing the nodes that do not increase the predictive accuracy of the tree. The optimal hyperparameter value is determined by using five-fold cross-validation and the implementation is further discussed in Section B.2 of the Appendix. After the optimal complexity parameter is incorporated into the model, which removes the nodes whose pruning yields the highest reduction in cost complexity, the pruned tree is used to generate predictions for the click-rate outcomes for the different treatment options. Similar to the previous methods, the optimal policy is formulated by assigning the treatments that maximize the expected outcomes for every user-ad pair.

Because the CART algorithm creates piecewise approximations and is therefore less suitable for continuous data and more prone to outliers, overfitting issues often arise and the out-of-sample performance is not always optimal. Therefore, bootstrap aggregating (bagging) and boosting techniques are explored by examining the random forest and XGBoost algorithms, with the aim of reducing the out-of-sample variance and maintaining a low bias for the in-sample fit.

4.2.4 Random Forest algorithm

The random forest algorithm is an ensemble method that builds and merges multiple decision trees to produce final predictions that are more accurate and less influenced by overfitting issues than predictions that are generated by single trees (CART algorithm). Bootstrap aggregating techniques are used to create multiple subsets of the training data, with the aim of training different decision trees for every subset and eventually aggregating the predictions of the different trees to enhance the overall final performance. When creating splits in the trees, a random subset of features of the covariate space is considered at each split, which produces diverse trees and reduces the level of correlation between the trees. Due to the ensemble nature of the method, which consists of combining and averaging the outputs of multiple decision trees, the final combined predictions are more robust, obtain lower variance levels and are more capable of handling outliers and high-dimensional data. Two hyper-parameters are used to construct the model, namely "n_{tree}" and "$mtry$", which are discussed in Section B.3 of the Appendix.

4.2.5 XGBoost algorithm

The extreme gradient boosting (XGBoost) algorithm is an ensemble learning method that is based on the gradient boosting framework, which sequentially combines multiple weak learners, in this case decision trees. The algorithm aims to improve the in-sample fit by reducing the bias, while maintaining low variance levels. Gradient descent is used to iteratively minimize the objective function, which consists of a loss function that measures the prediction errors and a regularization term that prevents overfitting by penalizing model complexity. The XGBoost algorithm recursively grows the trees by using a gradient-based algorithm to find the optimal splits for each node, where each new tree is trained and constructed with the aim of correcting the errors that were made by the previous tree. The maximum number of boosting iterations, e.g.

the sequential addition of decision trees to the ensemble with the aim of improving the overall performance, is set to 100 and the logarithmic loss metric is used to evaluate the predictive abilities of the model. The hyperparameters and their tuning process are discussed in Section B.4 of the Appendix.

4.3 Design of personalized policies based on heterogeneous treatment effects

An alternative approach for designing the optimal personalized policies is related to using estimates of the heterogeneous treatments effects for the three possible treatment pairs, instead of generating outcome estimates. After the most favourable treatment effect is determined, the corresponding treatment is assigned to every user-ad pair. The consistent estimates of the individual-level treatment effects are denoted by:

$$\tau_{w_m,w_{m'}}(x) = \mathbb{E}[Y(\mathbf{f_{ij}}, W_{ij} = W_m) - Y(\mathbf{f_{ij}}, W_{ij} = W_{m'})|\mathbf{f_{ij}} = x] \tag{6}$$

where $W_m, W_{m'} \in \{0, 1, 2\}$ and 3 pairwise models are analyzed for each treatment pair (control - treatment 1, control - treatment 2 and treatment 1 - treatment 2). The estimated treatments effects are then used to formulate the optimal policy by assigning the treatments according to:

$$\pi^\star(\mathbf{f_{ij}}) = w_m \quad \text{if and only if} \quad \forall m' \neq m \quad \tau_{w_m,w_{m'}}(x = \mathbf{f_{ij}}) \geq 0 \tag{7}$$

which maximizes the estimated treatment effect for every user-ad pair. In practice, instead of considering the treatment effect of each individual observation, the observations that have relatively similar features are jointly investigated by creating sub-populations. Therefore, this research estimates the conditional average treatment effect for specific sub-populations by using:

$$\tau_{w_m,w_{m'}}(x) = \frac{\sum_{(\mathbf{f_{ij}}) \in l(x), W_{ij} = W_m} Y_{ij}}{\sum 1[(\mathbf{f_{ij}}) \in l(x), W_{ij} = w_m]} - \frac{\sum_{(\mathbf{f_{ij}}) \in l(x), W_{ij} = W_{m'}} Y_{ij}}{\sum 1[(\mathbf{f_{ij}}) \in l(x), W_{ij} = w_{m'}]} \tag{8}$$

where $l(x)$ indicates the set of features for each sub-population. For every user-ad pair with covariate vector $\mathbf{f_{ij}}$, the observations in $l(x)$ are used to estimate the treatment effects of the possible treatment pairs. The causal tree and causal forest algorithms, introduced by Rzepakowski & Jaroszewicz (2010), are used to derive the optimal $l(x)$ for obtaining the personalized policies based on heterogeneous treatment effects. It is crucial that $l(x)$ includes sufficient observations so the noise in the estimates is minimized, while simultaneously maintaining the ability to capture the heterogeneity in the treatment effects by not incorporating too many different features.

4.3.1 Causal Tree algorithm

The idea behind causal tree is similar to the CART algorithm of Section 4.2.3, as it aims to create splits by identifying partitions with similar treatment effects, which is achieved by maximizing the variation of the estimated treatment effects (Athey & Imbens, 2016). The algorithm collects the observations with the same features into the same partition $l(x)$ by minimizing the variation in the post-split treatment effect, which therefore does not split observations that do

not have similar treatment effects. The treatment effect within each partition is determined using Equation 4.3. Following the approach of Athey & Imbens (2016), the optimal partitions are identified by maximizing the variation in the estimated treatment effects, using:

$$Var[\hat{\tau_{w_m,w_{m'}}}(x)] = \frac{1}{N}\sum_{ij=1}^{N}\hat{\tau^2_{w_m,w_{m'}}}(x) - (\frac{1}{N}\sum_{ij=1}^{N}\hat{\tau_{w_m,w_{m'}}}(x))^2 \tag{9}$$

which essentially consists of maximizing the squared estimated treatment effects, because the second term, the squared average estimated treatment effect, remains constant throughout the splitting process. For the final estimation, the algorithm prunes the tree and selects the split that maximizes $Var[\hat{\tau_{w_m,w_{m'}}}(x)] - \xi T$, which includes the complexity parameter ξ that is also used for the CART algorithm and the parameter T which refers to the number of terminal nodes. For every treatment pair, a different tree is built and pruned by following the previously described process, so in total three different trees are constructed to estimate the three heterogeneous treatment effects and assign the optimal treatments using Equation 7. The optimal values for ξ are listed in Section B.5 of the Appendix for every treatment pair.

4.3.2 Causal Forest - Generalized Random Forest algorithm

Following the approach of Wager & Athey (2018), the generalized random forest algorithm uses the intuition of random forests to estimate the heterogeneous treatment effects for the user-ad pairs. Similar to the previously discussed causal tree method, the objective of maximizing the variation in the estimated treatment effects is used to build the causal forest. In order to construct the trees in the forest, the nodes and resulting child nodes are recursively split by investigating a random sub-sample of the covariates. During the process of splitting the nodes, the R-learner objective function (Nie & Wager, 2021) is minimized for every treatment pair in order to estimate the treatment effect in every parent leaf P, which is specified by:

$$\hat{\tau}_P(\cdot) = \arg\min_\tau \left[\frac{1}{n_P}\sum_{ij=1}^{n_P}((Y_{ij} - \hat{m}^{(-ij)}(\mathbf{f_{ij}})) - (W_{ij} - \hat{e}^{(-ij)}(\mathbf{f_{ij}}))\hat{\tau}(\mathbf{f_{ij}}))^2 + \Lambda_n(\tau(\cdot))\right] \tag{10}$$

where n_P indicates the number of user-ad pairs that are considered in the parent partition and $\Lambda_n(\tau(\cdot))$ is a regularization element that determines the complexity of the model. The out-of-bag estimates for the outcome and propensity scores are denoted by $\hat{m}^{(-ij)}(\mathbf{f_{ij}})$ and $\hat{e}^{(-ij)}(\mathbf{f_{ij}})$, respectively, which are estimated by using random forest techniques. Every split is chosen by maximizing the following objective function:

$$\frac{n_L \cdot n_R}{(n_P)^2}(\hat{\tau}_L - \hat{\tau}_R)^2 \tag{11}$$

where n_L and n_R refer to the number of user-ad pairs that are included in the resulting left and right post-split partitions when comparing the two treatments w_m and $w_{m'}$. In contemplation of decreasing the computational burden, a gradient-based approximation of $\hat{\tau}$ is used for each child node instead of calculating the exact values for $\hat{\tau}_L$ and $\hat{\tau}_R$. Eventually, the algorithm uses the sub-populations $l(x)$ in the trees to estimate the treatment effects for each user-ad pair by

following:

$$\hat{\tau_{w_m,w_{m'}}}(x) = \frac{\sum_{i=l}^{N} \alpha_i(x)(Y_{ij} - \hat{m}^{(-ij)}(\mathbf{f_{ij}}))(W_{ij} - \hat{e}^{(-ij)}(\mathbf{f_{ij}}))}{\sum_{i=l}^{N} \alpha_i(x)(W_{ij} - \hat{e}^{(-ij)}(\mathbf{f_{ij}}))^2} \tag{12}$$

where $\alpha_i(x)$ reflects the frequency with which the i-th training sample is allocated to the same leaf as the set of features of the user-ad pair ($\mathbf{f_{ij}} = x$) and W_{ij} refers to the currently observed treatment w_m that is compared with $w_{m'}$. The success of the splits is related to how much the treatment effect estimates differ for each node, which results in the causal effect estimation and differs from the random forest objective. Similar to the causal tree method, three different casual forests are built for the three treatment pairs. The implementation details and the hyperparameters are further specified in Section B.6 of the Appendix.

4.3.3 Ensemble model

In order to investigate an additional approach for optimizing the personalized policies, an ensemble learning technique is used to combine the three best performing models (e.g. base models), with the aim of producing more advanced final predictions. Various methods can be used to integrate the base models into a new ensemble learning model, by using for example boosting, bagging or stacking techniques (Ganaie et al., 2022). For this research, a stacking technique is used to incorporate the predictions of the three best performing base models as inputs for a new meta-model. Stacking is suitable for this setting because it is a flexible method and allows for combining various types of models in order to take advantage of their mutual complementarity. In the stacking ensemble learning approach, the predictive abilities of the three best performing models are integrated by using several levels, in this case two levels, which are depicted in Figure 2 (Divina et al., 2018). By combining the predictive abilities of multiple models, the overall bias is reduced which results in more sophisticated final predictions (Ganaie et al., 2022). In the first step, the three best performing models, Model A, B & C, are trained and used to construct predictions for the different treatments, represented by Level 0 in Figure 2. The predictions of these individual base models are then used as new features, so-called meta-features, for constructing and training the new meta-model that is used to generate the final outcome predictions, denoted by "Generalizer" in Level 1 in Figure 2 below.

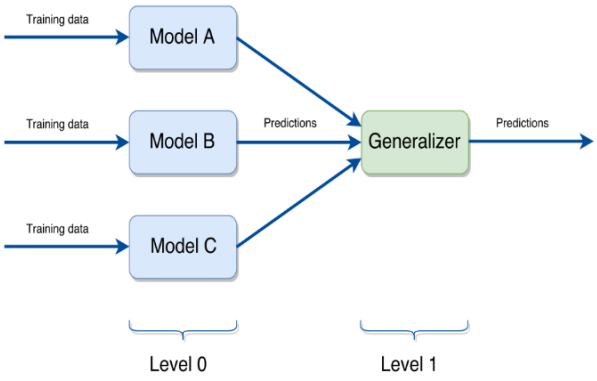

Figure 2: Graphical representation of the ensemble learning model that combines 3 best performing models (Divina et al., 2018)

In order to combine the base models in the most optimal way, an optimization problem is solved that assigns different weights w_v to the base models f_v for constituting the meta-model, similar to the approach of Gu et al. (2022). Because the actual outcome of the treatments remains unknown, five-fold cross-validation is used to minimize the average mean squared error of the folds by using the difference between the estimated outcome of the ensemble model $\hat{Y}_{E,ij}^{(k)}$ and the outcome $Y_{ij}^{(k)}$ of the validation set for the specific fold k that is extracted from the observed outcomes. The optimization problem is expressed by:

$$\min \quad \text{MSE}_k = \frac{1}{n_k} \sum_{ij \in val_k} (\hat{Y}_{E,ij}^{(k)} - Y_{ij}^{(k)})^2 \quad \text{with} \quad \hat{Y}_{E,ij}^{(k)} = \sum_{v=1}^{3} w_v \hat{f}_v^{(k)} \quad \text{for} \quad k \in \{1, ..., 5\} \tag{13}$$

$$\text{such that} \quad w_v \geq 0 \quad \text{and} \quad \sum_{v=1}^{3} w_v = 1 \quad \forall \quad v \in \{A, B, C\} \tag{14}$$

where n_k denotes the amount of observations in the corresponding validation set val_k and the restrictions in Equation 14 ensure that the weights of the three base models are non-negative and sum to 1 so the final predictions are stable and consistent. The weights are denoted by w_v and the base models are indicated by f_v for the 3 best performing models which are displayed by $v \in \{A, B, C\}$. Eventually, the set of weights that minimizes the average mean squared error of the five folds, e.g. minimizes $\frac{1}{k} \sum_{k=1}^{5} \text{MSE}_k$, is assigned to the base models and used to construct the meta-model. Similar to the approach for the models of Section 4.2, the outcome estimates of the meta-model are used to assign the treatments that maximizes the expected click-rates for the user-ad pairs. The three models that are chosen for the base models are determined by analyzing the off-policy evaluation methods that are illustrated in the following Section 4.4.

4.4 Off-policy evaluation methods

Off-policy evaluation methods are used to determine which personalization technique is ultimately optimal for designing the customized social advertisements, because the predicted click-rates cannot be compared with the actual click-rates if the policies would be implemented. This is due to the fact that the outcomes for the user-ad pairs are only observed for one specific treatment, which implies that the actual outcome for a user-ad pair remains unknown if the policy assigns a different treatment. For that reason, in order to be able to draw accurate conclusions about the most optimal personalization technique, the inverse propensity score (IPS) and doubly robust (DR) estimators are analyzed and compared for every model.

4.4.1 Inverse Propensity Score estimator

First, the IPS estimator calculates the propensity of receiving the treatment that is assigned to a specific user-ad pair for the different treatment options and uses it to construct a pseudo-population where every user-ad pair receives the treatment that is assigned by policy π. The average of the outcomes for the pseudo-population gives an unbiased estimate of the average expected click-rate for the full population, if the suggested policy would be implemented. Based on the approach of Yoganarasimhan et al. (2023), the IPS estimator is calculated by following:

$$\hat{R}_{IPS}(\pi, Y) = \frac{1}{N} \sum_{ij=1}^{N} \frac{\mathbb{1}[W_{ij} = \pi(\mathbf{f_{ij}})] Y_{ij}}{\hat{e}_{\pi(\mathbf{f_{ij}})}(W_{ij})} \quad \text{with} \quad \hat{e}_{\pi(\mathbf{f_{ij}})}(W_{ij}) = \sum_{t=1}^{N} \frac{\mathbb{1}[W_t = W_{ij}, \pi(\mathbf{f_t}) = \pi(\mathbf{f_{ij}})]}{\mathbb{1}[\pi(\mathbf{f_t}) = \pi(\mathbf{f_{ij}})]} \tag{15}$$

where Y_{ij} is the observed outcome for user-ad pair ij and $\hat{e}_{\pi(\mathbf{f_{ij}})}(W_{ij})$ is the empirical propensity that user-ad pair ij is given treatment $W_{ij} = w^\star$ if it is assigned $\pi(\mathbf{f_{ij}}) = w^\star$, with $\pi(\mathbf{f_{ij}})$ denoting the treatment that is assigned to user-ad pair ij by the policy π and $W_{ij} \in \{0, 1, 2\}$ indicating the control and treatment groups. By weighting the observations by the inverse of the propensity of actually receiving the treatment that was assigned, the IPS estimator balances the treatment groups and adjusts for the confounding bias that occurs when the allocation of the treatment groups is not assigned randomly but differs in such a way that it affects the overall outcome.

Although the IPS estimator is capable of providing insights on the possible benefits if the actual outcome is unknown, it relies solely on the correct specification of the propensity score model, resulting in potential bias issues if the propensity score model is misspecified. Therefore, the doubly robust estimator is also analyzed because it combines a propensity score model and an outcome regression model, resulting in a reduced risk of bias and higher efficiency (e.g. lower variance) because it merges the information of two models.

4.4.2 Doubly Robust estimator

The doubly robust estimator combines two concepts, namely an outcome regression model and an inverse probability weighting model, to estimate the effects of the treatments that are assigned by the personalization strategies. Therefore, the key advantage of the doubly robust estimator is related to the fact that the off-policy evaluation metric still generates consistent estimates if either the outcome model or the propensity score model is misspecified (Funk et al., 2011). The outcome model predicts the outcome (e.g. click-rate) based on the covariates and the treatment, while the propensity score model estimates the probability of receiving the treatment, which is used to assign weights to the outcomes and adjust for the fact that the treatments are not assigned randomly. The propensity score model is formulated in the same way as for the IPS estimator that is described in Section 4.4.1. Following the approach of Vermeulen & Vansteelandt (2015), the doubly robust reward estimator is calculated for all personalization strategies by using:

$$\hat{R}_{DR}(\pi, Y) = \frac{1}{N} \sum_{ij=1}^{N} \left[\frac{\mathbb{1}[W_{ij} = \pi(\mathbf{f_{ij}})] Y_{ij}}{\hat{e}_{\pi(\mathbf{f_{ij}})}(W_{ij})} + \left(1 - \frac{\mathbb{1}[W_{ij} = \pi(\mathbf{f_{ij}})]}{\hat{e}_{\pi(\mathbf{f_{ij}})}(W_{ij})} \right) \hat{Y}(\mathbf{f_{ij}}, \pi(\mathbf{f_{ij}})) \right] \tag{16}$$

where $\hat{e}_{\pi(\mathbf{f_{ij}})}(W_{ij})$ is the empirical propensity that a user-ad pair is given treatment $W_{ij} = w^\star$ if it is assigned $\pi_{(\mathbf{f_{ij}})} = w^\star$, defined in Equation 15, and $\hat{Y}(\mathbf{f_{ij}}, \pi(\mathbf{f_{ij}}))$ is the predicted outcome for user-ad pair ij with covariate vector $\mathbf{f_{ij}}$ and assigned treatment $\pi(\mathbf{f_{ij}})$. The predicted outcomes are determined by using logistic regression, lasso, CART, random forest and XGBoost models in order to contribute to the robustness of the results for the conclusions of the DR estimator.

5 Results

The main results are presented in Table 4, which reflects the predicted click-rate percentages that are calculated by the IPS and DR estimators for the different personalization strategies. The results for the training and test datasets are included in the table, as well as the percentage increase ($+\Delta$) or decrease ($-\Delta$) relative to the original number of observations who clicked on the advertisement after seeing it for the first time. Because the training and test datasets contain some minor differences for every time that they are reformulated, the average original click-rate is used to determine the resulting Δ. For the training dataset, the average observed click-rate for the original control/treatment assignment is 49.92% and for the test dataset the average is 49.80%. The click-rates that are higher than the original click-rates are underlined in Table 4, to provide an immediate overview of which personalized policies generate improvements. For the conditional average treatment effect estimators (CATE), causal tree and causal forest, it is not possible to calculate the doubly robust estimator because there are no predictions of the outcome variable for every treatment. Therefore, the corresponding elements in the table are left empty and the CATE models can only be evaluated by considering the IPS estimator.

	Training set		*Test set*		*Training set*		*Test set*	
Policy	**IPS**	**DR**	**IPS**	**DR**	ΔIPS	ΔDR	ΔIPS	ΔDR
π_{logreg}	57.20%	57.36%	58.93%	58.88%	+14.58%	+14.90%	+18.33%	+18.23%
π_{lasso}	48.05%	50.25%	51.62%	50.98%	-3.75%	+0.66%	+3.65%	+2.37%
π_{CART}	50.20%	50.32%	50.14%	50.11%	+0.56%	+0.80%	+0.68%	+0.62%
π_{RF}	49.31%	49.40%	48.35%	48.45%	-1.22%	-1.04%	-2.91%	-2.71%
π_{XGB}	50.40%	50.41%	54.95%	55.22%	+0.96%	+0.98%	+10.34%	+10.88%
$\pi_{ensemble}$	56.72%	56.84%	59.68%	59.62%	+13.62%	+13.86%	+19.84%	+19.72%
$\pi_{causaltree}$	49.82%		50.52%		-0.20%		+1.45%	
$\pi_{causalforest}$	49.04%		49.50%		-1.76%		-0.60%	

Table 4: Estimated click-rate % using IPS and DR estimators for the personalized policies

The DR estimators of Table 4 are constructed by using a logistic regression as outcome model, the DR estimates for the other outcome models are included in Section C of the Appendix. In order to examine the robustness of the results for the off-policy evaluation estimators and draw reliable conclusions about which model is optimal for designing personalized advertisements, the resulting DR estimators of the five different outcome models are compared. The results of the DR estimators in Section C of the Appendix do not differ substantially for the different outcome models, which contributes to the finding that the two off-policy evaluation metrics are relatively similar and result in identical conclusions about the performance of the personalization techniques.

The ensemble model is constructed using the logistic regression, lasso and XGBoost models as the base models, because those personalization techniques generate the largest improvements regarding the IPS and DR estimates presented in Table 4. The allocation of the weights for the base models is specified in Section B.7 of the Appendix. The logistic regression model is the

best performing model for both the training and test datasets, and the XGBoost algorithm is second-best according to the results of Table 4. Although the IPS and DR estimates for the CART algorithm are higher than the average click-rates for the training and test datasets, the lasso model is chosen as the third base model because the relative improvement of the estimated click-rate percentage is higher than for the CART model, especially for the test dataset where the increase is roughly 3.01% for lasso and only 0.65% for CART when averaging the IPS and DR estimates. These results are similar to the findings of Yoganarasimhan et al. (2023), as they conclude that the lasso, XGBoost and linear regression models are the most capable of assigning personalized free trial periods with the aim of maximizing the subscription rates for the software company.

The results for the ensemble learning model increase the expected click-rate percentages by roughly 13.5% and 19.8% for the training and test datasets, respectively, indicating that combining the predictive abilities of multiple well-performing models results in very favourable payoffs regarding the commercial potential of customized advertisements. For the test dataset, the estimated click-rates outperform the results for all of the other models, leading to the conclusion that the stacking technique is capable of capturing and combining the mutual complementarity of the three best performing models in order to produce customized advertisements with the highest expected profits. The superior performance of the ensemble model is slightly less dominant for the training dataset, as the increase for the IPS and DR estimator is slightly higher for the logistic regression model, namely 14.7%, compared to 13.7% for the ensemble model. This is possibly related to the limited performance of the lasso model for the training dataset, as it only assigns treatment 1 and treatment 2 for the training dataset and therefore does not exceed the original average click-rate for that dataset. If only two treatments are assigned by π_{lasso}, the number of assigned treatments that are equal to the original treatments is lower, which contributes to the IPS decrease of 3.75% for the training dataset.

A possible explanation for the advantageous performance of the logistic regression model compared to the other individual models could be related to the fact that the number of explanatory variables is relatively small. A small set of predictor variables often leads to overfitting issues for more complex models, because they require more information to be able to learn about the patterns that are present in the data. In addition to this, the logistic regression model is the only model that does not require hyperparameter tuning, which could also lead to potential problems for the machine learning models if the hyperparameters are not tuned optimally and are therefore not able to generate the most accurate predictions regarding the click-rates.

Furthermore, it is interesting to observe that the predictions for the CART model are more favourable than for the random forest model, whilst the random forest algorithm is constructed with the aim of using an ensemble approach to advance the CART technique, which only builds a single tree. The fact that the CART model produces more favourable policies is possibly also related to the relatively small dataset, which may lead to overfitting issues for the random forest model. In addition to this, the CART algorithm includes a hyperparameter that prunes the tree to prevent overfitting, whereas the tuning process might have been ineffective for the random forest model.

When analyzing the analogous random forest and XGBoost models, the results for the IPS and DR estimates in Table 4 indicate that the XGBoost model improves the predicted click-rates for all estimates of the training and test datasets, while the related random forest algorithm does not generates improvements at all. This is possibly related to the boosting element of the XGBoost algorithm, which builds the trees in the forest sequentially and constructs each new tree by correcting the errors of the previous ones. Therefore, the overall bias is reduced and the ultimate predictive power is enhanced by using the XGBoost algorithm. The random forest algorithm on the other hand averages the results of independently built trees, which reduces the variance but does not effectively address the overall bias. In addition to this, the implementation of the random forest algorithm used for this paper only incorporates two hyperparameters, while the XGBoost algorithm uses five parameters to fine-tune the model, possibly leading to more advanced prediction abilities that are more appropriate for this specific research setting. These conclusions are also harmonious with the results retrieved by Yoganarasimhan et al. (2023), as they also find that overfitting issues arise when using the random forest model for their dataset. In addition to this, Yoganarasimhan et al. (2023) also conclude that the XGBoost model is much more capable of capturing the heterogeneity that is present in the data, which results in more effective personalized policies.

When comparing the two different approaches for constructing the personalized policies, the results in Table 4 indicate that using outcome estimates is generally more effective than using estimates for the heterogeneous treatment effects, as the IPS estimates for the causal tree and causal forest algorithms are lower than for the models that use outcome estimates. A possible reason for this unfavourable result could be related to the fact that the models are not able to competently formulate the sets of covariates $l(x, z)$ that are used to estimate the treatment effects. This is once again in line with the conclusions of Yoganarasimhan et al. (2023), who also find that the causal tree and causal forest methods perform more poorly than the methods that are based on outcomes estimates.

Figure 3 below depicts the cumulative density functions (CDFs) of the difference of the treatment effects of treatment 2 and the control group for the test dataset of the ensemble, XGBoost and causal forest models. The largest treatment difference is captured by comparing the results for the personalized advertisements with multiple or zero likes of connections, which is why treatment 2 and the control group are portrayed here. The CDFs of the other treatment pairs are included in Section D of the Appendix. The shapes of the cumulative density functions that are presented in Figure 3 are harmonious with the results of Table 4, as the CDF for the ensemble model is the least vertical and is therefore the most capable of capturing the heterogeneity that is present in the data in order to assign the personalized policies. In contrast, the causal forest model is not capable of adequately capturing the heterogeneity that is present in the test data, which is reflected by the fact that the CDF function is an almost vertical line, indicating that the treatment effects are largely homogeneous for the observations in the test dataset. The heterogeneity of the treatment effects for the XGBoost model lies in between the CDFs of the ensemble and causal forest models, which is harmonious with the result that the XGBoost model improves the predictions for the click-rates if π_{XGB} would be assigned, but it does not surpass the outcome estimates for the ensemble model that integrates the strengths of

multiple well-performing models.

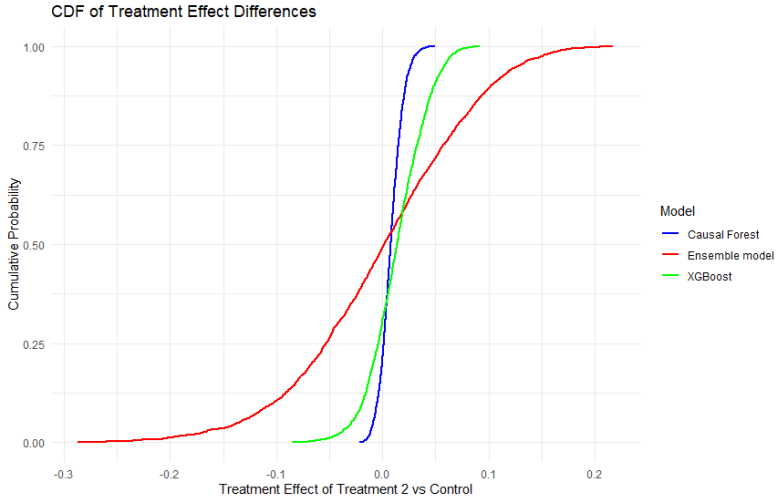

Figure 3: CDFs of the treatment effects of the test dataset for Treatment 2 vs Control for the ensemble, XGBoost and causal forest models

Table 5 below represents the fraction of the user-ad pairs that is assigned to each of the control and treatment groups for the personalized policies. This treatment allocation is relevant for the financial managers of the marketing campaigns, as the companies will most likely have to pay a higher fee if they wish to include likes of connections below the advertisements. Therefore, assigning treatment 2 will most likely be the most costly option and the control group will be the cheapest option, which must be taken into account when determining the ultimate profit of the personalized policies. The fractions that are retrieved from the test dataset are presented in Table 5 and the treatment allocations for the training dataset are included in Table 10 in Section E of the Appendix. It is interesting to note that the assignments of the treatments vary a lot between the different policies, and there is no obvious correlation between a potential increase in the estimated click-rate and the assignment of the user-ad pairs to specific control / treatment groups, although the financial perspective of the implementation should also be taken into consideration.

Policy-prescribed Treatment	π_{logreg}	π_{lasso}	π_{CART}	π_{RF}	π_{XGB}	$\pi_{causaltree}$	$\pi_{causalforest}$	$\pi_{ensemble}$
Control (*Test set*)	0.356	0.452	0.316	0.374	0.274	0.053	0.345	0.347
Treatment 1 (*Test set*)	0.342	0.133	0.296	0.346	0.083	0.185	0.060	0.343
Treatment 2 (*Test set*)	0.301	0.414	0.388	0.280	0.643	0.762	0.594	0.309

Table 5: Fraction of observations (user-ad pairs) assigned to the control / treatment groups by the personalized policies for the **test dataset**

6 Conclusion

In conclusion, this paper investigates the performance of several machine learning techniques for personalizing social media advertisements. The customization is based on the heterogeneous

features of the advertised products and of the users of the social media platform WeChat. The personalization element consists of assigning multiple, one or no likes of connections below the advertisement that is presented to the users, with the aim of using the social influence to increase the click-rate percentages. First, methods based on outcome estimates are analyzed, including the logistic regression, lasso, CART, random forest and XGBoost algorithms. For the heterogeneous treatment effects approach, the causal tree and causal forest algorithms are explored. In addition to this, an ensemble model that combines the logistic regression, lasso and XGBoost models is analyzed with the aim of producing improved final predictions. The performance of the models is evaluated by assessing the inverse propensity score (IPS) and doubly robust (DR) off-policy estimators. By analyzing both the IPS and DR estimators, the final conclusions for the performances of the models are more robust and less exposed to model misspecifications.

The main finding of this paper is that the ensemble model formulates the policy with the highest predicted click-rates for the IPS and DR estimators by combining the strengths of three well-performing base models. Therefore, the main research question is answered by concluding that the ensemble model is more optimal than the other models for designing personalized social media advertisements for this dataset. The ensemble model increases the estimates of the expected click-rates by roughly 13.7% and 19.8% for the training and test datasets, respectively, generating substantial improvements for the expected profits of personalized advertisements, compared to advertisements that are not tailored at the individual level. Although this result is favourable, it is important to note that the logistic regression model also generates substantial improvements, by increasing the expected click-rates with roughly 14.7% and 18.3% for the training and test datasets, respectively. Therefore, the expected profits of the two best performing models are comparable and the logistic regression model might be more suitable in situations where computational efficiency is valued highly.

A limitation of this paper is related to the fact that the number of explanatory variables is relatively small, which possibly contributes to the weak performance of the machine learning models that require a lot of information. Therefore, further research could focus on re-evaluating the performance of the models for a larger dataset with more predictor variables, which possibly enables the models to capture a larger part of the heterogeneity that is present in the data. However, the ethical side of this approach must also be taken into perspective, as including more features of social media users is not always justifiable and legitimate if they are unaware that their private information is stored and used. Another limitation stems from the short duration of the randomized field experiment, which only consisted of three weeks and therefore does not include any long-term information about whether the viewers of the advertisements turned into regular customers. Because of this, the actual sales potential of the personalized social advertisements also remains unknown, because no information is available about whether users who clicked on the ad also decided to purchase the advertised product. Therefore, an interesting extension could include researching the long-term effectiveness of personalized social media advertisements, which could lead to more encompassing and decisive conclusions about the commercial potential and profits.

References

Alalwan, A. A. (2018). Investigating the impact of social media advertising features on customer purchase intention. *International journal of information management*, *42*, 65–77.

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

Breiman, L. (2017). *Classification and regression trees*. Routledge.

Deligiannis, A., Argyriou, C. & Kourtesis, D. (2020). Predicting the optimal date and time to send personalized marketing messages to repeat buyers. *International Journal of Advanced Computer Science and Applications*, *11*(4).

Divina, F., Gilson, A., Goméz-Vela, F., García Torres, M. & Torres, J. F. (2018). Stacking ensemble learning for short-term electricity consumption forecasting. *Energies*, *11*(4), 949.

Funk, M. J., Westreich, D., Wiesen, C., Stürmer, T., Brookhart, M. A. & Davidian, M. (2011). Doubly robust estimation of causal effects. *American journal of epidemiology*, *173*(7), 761–767.

Ganaie, M. A., Hu, M., Malik, A. K., Tanveer, M. & Suganthan, P. N. (2022). Ensemble deep learning: A review. *Engineering Applications of Artificial Intelligence*, *115*, 105151.

Gu, J., Liu, S., Zhou, Z., Chalov, S. R. & Zhuang, Q. (2022). A stacking ensemble learning model for monthly rainfall prediction in the taihu basin, china. *Water*, *14*(3), 492.

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.

Huang, S., Aral, S., Hu, Y. J. & Brynjolfsson, E. (2020). Social advertising effectiveness across products: A large-scale field experiment. *Marketing Science*, *39*(6), 1142–1165.

Kiyohara, H., Nomura, M. & Saito, Y. (2024). Off-policy evaluation of slate bandit policies via optimizing abstraction. *arXiv preprint arXiv:2402.02171*.

Neugebauer, R. & van der Laan, M. (2005). Why prefer double robust estimators in causal inference? *Journal of statistical planning and inference*, *129*(1-2), 405–426.

Nie, X. & Wager, S. (2021). Quasi-oracle estimation of heterogeneous treatment effects. *Biometrika*, *108*(2), 299–319.

Oberlo. (2024). *Number of wechat users (2013–2023)*. `https://www.oberlo.com/statistics/number-of-wechat-users`.

Raudeliūnienė, J., Davidavičienė, V., Tvaronavičienė, M. & Jonuška, L. (2018). Evaluation of advertising campaigns on social media networks. *Sustainability*, *10*(4), 973.

Rzepakowski, P. & Jaroszewicz, S. (2010). Decision trees for uplift modeling. In *2010 ieee international conference on data mining* (pp. 441–450).

Song, W., Han, X. & Qi, J. (2023). Prediction of gas emission in the working face based on lasso-woa-xgboost. *Atmosphere*, *14*(11), 1628.

Sun, L., Zheng, X., Jin, Y., Jiang, M. & Wang, H. (2019). Estimating promotion effects using big data: A partially profiled lasso model with endogeneity correction. *Decision Sciences*, *50*(4), 816–846.

Van Doorn, J. & Hoekstra, J. C. (2013). Customization of online advertising: The role of intrusiveness. *Marketing Letters*, *24*, 339–351.

Van Wezel, M. & Potharst, R. (2007). Improved customer choice predictions using ensemble methods. *European Journal of Operational Research*, *181*(1), 436–452.

Vermeulen, K. & Vansteelandt, S. (2015). Bias-reduced doubly robust estimation. *Journal of the American Statistical Association*, *110*(511), 1024–1036.

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.

Yigit, A. T., Samak, B. & Kaya, T. (2020). An xgboost-lasso ensemble modeling approach to football player value assessment. *Journal of Intelligent & Fuzzy Systems*, *39*(5), 6303–6314.

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

A Data Section

Following the definitions of Huang et al. (2020), experience goods need to be experienced in order to be evaluated, such as vacations or healthcare services, while search goods can be assessed before making purchase decisions, referring to for instance laptops or credit card services. Regarding the difference between status and non-status goods, consumers wish to display social status or prestige when buying for instance status-conferring cars or hotels. Non-status goods on the other hand include for instance toothpaste or website services, which do not exhibit status symbols.

	Experience goods	**Search goods**
Status goods	2256 user-ad pairs	2479 user-ad pairs
Non-status goods	4633 user-ad pairs	2479 user-ad pairs

Table 6: Overview of the number of user-ad pairs per product type

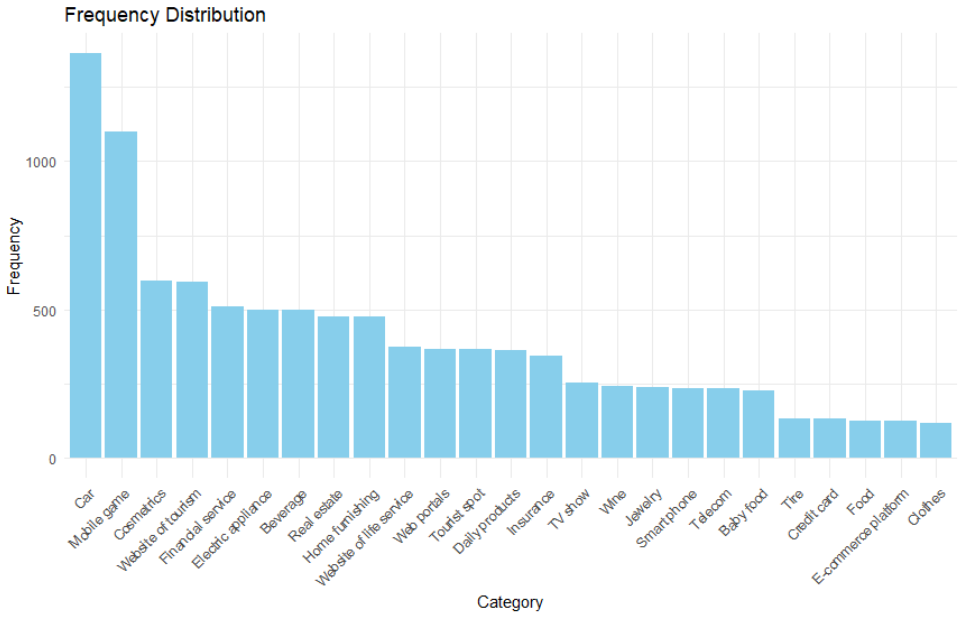

Figure 4: Ordered frequency distribution of the 25 possible product categories

Table 7 provides an overview of the average original treatment allocations for the training and test datasets, which is relatively equal.

	Training data	Test data
Control	0.338	0.332
Treatment 1	0.334	0.339
Treatment 2	0.328	0.328

Table 7: Average fraction of user-ad pairs in the control / treatment groups for the training and test datasets

B Hyperparameters

B.1 Lasso model

For the lasso model, the optimal value for λ is determined by implementing the cross-validation package of the "*glmnet*" package in R. In this way, the λ value is chosen that minimizes the mean squared error and therefore results in the best out-of-sample performance based on the five-fold cross-validation. Using this cross-validation approach, the optimal value for $\lambda = 0.01468$, indicating that a relatively large amount of coefficients are driven to zero and the complexity of the model is reduced with the aim of preventing overfitting issues.

B.2 CART

By using a cost complexity pruning technique, the CART algorithm selects the complexity parameter ξ that minimizes the cross-validated prediction error from the complexity parameter table. The model uses five-fold cross-validation to search over 3886 possible values for the complexity parameter ξ, ranging from $8.6 * 10^{-11}$ to $1.7 * 10^{-1}$. The optimal value is retrieved as $\xi = 0.00801$. In order to implement the model, the *"rpart"* package in R is used that implements a single tree, which is similar to the approach of Breiman (2017).

B.3 Random Forest

In order to implement the random forest model, the *"randomForest"* package in R is used. A grid search is used to determine the optimal values for the hyperparameters (1)n_{tree} and (2)m_{try}. First of all, "n_{tree}" indicates how many trees are used to build the ensemble forest. Higher values for "n_{tree}" generally lead to a reduction in the amount of overfitting and a decrease in the variance, because the predictions of the ensemble forest become more robust by averaging the individual trees to remove the random variations. The "*mtry*" parameter refers to the amount of randomly selected features that are considered from the total set of available features in order to determine the best split for each node of the decision trees. This concept incorporates the randomness element into the tree-building process, which is crucial for the random forest algorithm to perform well. By randomly selecting a subset of features at each split, the "*mtry*" parameter ensures that the individual trees in the forest are less correlated, which reduces the risk of overfitting and enhances the generalizability of the model. Lower values for "*mtry*" increase the amount of randomness because less features are considered at each split, which results in more diverse trees. A grid search is used to determine the optimal values for the hyperparameters, by exhaustively searching through the grid of possible hyperparameter values and selecting the combination that maximizes the performance of the model. The range over which the search is performed and the optimal values are listed in the following:

- $n_{tree} \in [10, 700]$ and $n^{\star}_{tree} = 36$
- $m_{try} \in [1, 16 \text{ (\# of features)}]$ and $m^{\star}_{try} = 11$

B.4 XGBoost

In order to implement the XGBoost algorithm, the *"xgboost"* package in R is used. By using five-fold cross-validation, the performance of the model is assessed and the hyper-parameters are tuned. The hyperparameters that need tuning are the learning rate (1)η, which shrinks the feature weights to make the boosting process more conservative, (2)δ_{max}, which indicates the maximum depth of the trees that are constructed in the model, (3)γ, which specifies the minimum loss reduction that is required to make a further partition on a leaf node and controls the complexity of the model, (4)$n_{feature}$, which controls the number of features that are used for training each tree, (5)w_{min}, which specifies the minimum sum of instance weight (hessian values) needed in a child node and (6)s_{min}, which indicates the minimum number of samples that are required to create a new leaf node in the tree. The hyperparameter tuning for the XGBoost model is similar to the random forest approach, as a grid search is used to determine the optimal values for the hyperparameters "η", "d_{max}", "γ", "$n_{feature}$", "w_{min}" and "s_{min}". Due to computational limitations and the fact that six hyperparameters need tuning, only a limited amount of values are considered in the grid search. The range over which the search is performed and the optimal values are listed in the following:

- $\eta \in \{0.01, 0.1, 0.3\}$ and $\eta^\star = 0.1$
- $d_{max} \in \{3, 6, 9\}$ and $d^\star_{max} = 3$
- $\gamma \in \{0, 0.1, 0.2\}$ and $\gamma^\star = 0.1$
- $n_{feature} \in \{0.5, 0.7, 1.0\}$ and $n^\star_{feature} = 0.7$
- $w_{min} \in \{1, 3, 5\}$ and $w^\star_{min} = 5$
- $s_{min} \in \{0.6, 0.8, 1.0\}$ and $s^\star_{min} = 0.8$

B.5 Causal Tree

The causal tree algorithm is implemented by using the *"grf"* and *"htetree"* packages in R. The optimal value for the complexity parameter ξ for every treatment duo is retrieved in a similar way as for the CART algorithm described in Section 4.2.3. Therefore, the complexity parameter ξ is chosen that minimizes the cross-validated prediction error from the complexity parameter table. Five-fold cross-validation is used to select the optimal value for ξ that prunes the tree to prevent overfitting, while simultaneously maintaining its predictive power. The optimal values for the three trees that are constructed for the three different treatment pairs are listed below.

- Tree (1) for control (no likes) and treatment 1 (one like): $\xi = 6.35e-05$
- Tree (2) for control (no likes) and treatment 2 (multiple likes): $\xi = 4.22e-05$
- Tree (3) for treatment 1 (one like) and treatment 2 (multiple likes): $\xi = 1.90e-06$

B.6 Causal Forest

The hyperparameters that are used to construct the causal forest for each treatment pair are (1)n_{frac}, which controls the fraction of examples that should be used in growing each tree, (2)n_{tree}, which indicates the number of trees that should be grown in the forest, (3)m_{try}, which determines the number of variables considered during each split, (4)s_{min}, which relates to the minimum size a leaf node is allowed to have, (5)α, which controls the maximum imbalance of a split and (6)p_{imb}, which controls how harshly imbalanced splits are penalized. The cross-validation method that is incorporated in the *"grf"* package in R is used to select the optimal values for the hyperparameters of the causal forest method. 100 distinct sets of parameter values are considered during the training procedure. The *"grf"* package incorporates a cross-validation procedure in the causalforest function that automatically tunes and selects the optimal values for the hyperparameters. The optimal parameter values are chosen by training a forest for each set of values and computing the out-of-bag error. Eventually, the optimal parameters are the ones that minimize the predicted smoothed error of the smoothing function for a new random draw of possible parameter values.

- Control - Treatment 1 pair: $n_{tree} = 144$, $m_{try} = 11$, $n_{frac} = 0.5$, $s_{min} = 77$, $\alpha = 0.107$, $p_{imb} = 0.473$
- Control - Treatment 2 pair: $n_{tree} = 221$, $m_{try} = 5$, $n_{frac} = 0.5$, $s_{min} = 3$, $\alpha = 0.006$, $p_{imb} = 0.155$
- Treatment 1 - Treatment 2 pair: $n_{tree} = 187$, $m_{try} = 12$, $n_{frac} = 0.5$, $s_{min} = 9$, $\alpha = 0.178$, $p_{imb} = 0.354$

B.7 Ensemble Learning model

By using the five-fold cross-validation approach described in Section 4.3.3, the initial weights that are assigned to the base models are:

- $w_A = 1$
- $w_B = 0$
- $w_C = 0$

for the base models; model A = logistic regression, model B = lasso and model C = XGBoost. This unfortunate allocation is possibly related to the relatively attractive performance of the logistic regression model, which increases the IPS and DR estimators for the expected click-rates substantially more than the lasso and XGBoost models. Because this allocation of weights would result in the same model as the logistic regression model of Section 4.2.1, additional restrictions are included for the weights to ensure that all three models are integrated in the final meta-model. The lower bound for the weights is set to 0.25 and the upper bound to 0.75, before standardizing the weights, in order to construct a more equal allocation of the features of the three base models. After incorporating these additional restrictions, the optimal weights are retrieved as:

- $w_A = 0.6$
- $w_B = 0.2$
- $w_C = 0.2$

for the base models; model A = logistic regression, model B = lasso and model C = XGBoost. Once again, the largest weight is assigned to the logistic regression model because it generates the most favourable predictions for the click-rates of the personalized treatments. However, the features of the lasso and XGBoost models are now also used as inputs for the ensemble model.

C Robustness Check Results

Table 8 and 9 below present the DR estimates that are retrieved by using different outcome models for the training and test datasets. Fortunately, the outcome estimates generate analogous results, which contributes to the robustness of the results for the estimated click-rates. For the training and test datasets, the ranking of the performance of the models in terms of their expected click-rates for all of the DR estimators is similar to the results for the IPS estimator, which validates the conclusions for the optimal customization technique for this dataset.

DR estimates for outcome model	π_{logreg}	π_{lasso}	π_{CART}	π_{RF}	π_{XGB}	$\pi_{ensemble}$
$\hat{y}_{logreg}$	57.36%	50.25%	50.32%	49.40%	50.41%	56.84%
$\hat{y}_{lasso}$	57.20%	50.19%	50.16%	49.33%	50.40%	56.74%
$\hat{y}_{CART}$	54.91%	50.24%	50.20%	49.31%	50.34%	56.66%
$\hat{y}_{RF}$	53.80%	50.17%	50.44%	49.49%	50.17%	52.52%
$\hat{y}_{XGB}$	57.12%	50.27%	50.36%	49.40%	50.47%	56.72%

Table 8: Doubly Robust estimators for the expected click-rates of the policies using different outcome models for the **training dataset**

DR estimates for outcome model	π_{logreg}	π_{lasso}	π_{CART}	π_{RF}	π_{XGB}	$\pi_{ensemble}$
$\hat{y}_{logreg}$	58.88%	50.98%	50.11%	48.45%	55.22%	59.62%
$\hat{y}_{lasso}$	58.90%	51.61%	50.11%	48.36%	55.01%	59.62%
$\hat{y}_{CART}$	58.52%	51.70%	50.14%	48.35%	55.06%	59.61%
$\hat{y}_{RF}$	54.03%	50.66%	49.72%	48.76%	51.85%	53.53%
$\hat{y}_{XGB}$	58.64%	51.01%	49.98%	48.37%	55.24%	59.34%

Table 9: Doubly Robust estimators for the expected click-rates of the policies using different outcome models for the **test dataset**

D Additional Figures

Figure 5 and 6 also portray the fact that the ensemble model is the most capable of capturing the heterogeneity that is present in the data for formulating the personalized treatments for the

other two treatment pairs. The causal forest is the least capable of capturing the heterogeneity, which is reflected by the almost vertical blue lines in the figures below and the poor evaluation of $\pi_{causalforest}$ in terms of the IPS and DR estimators.

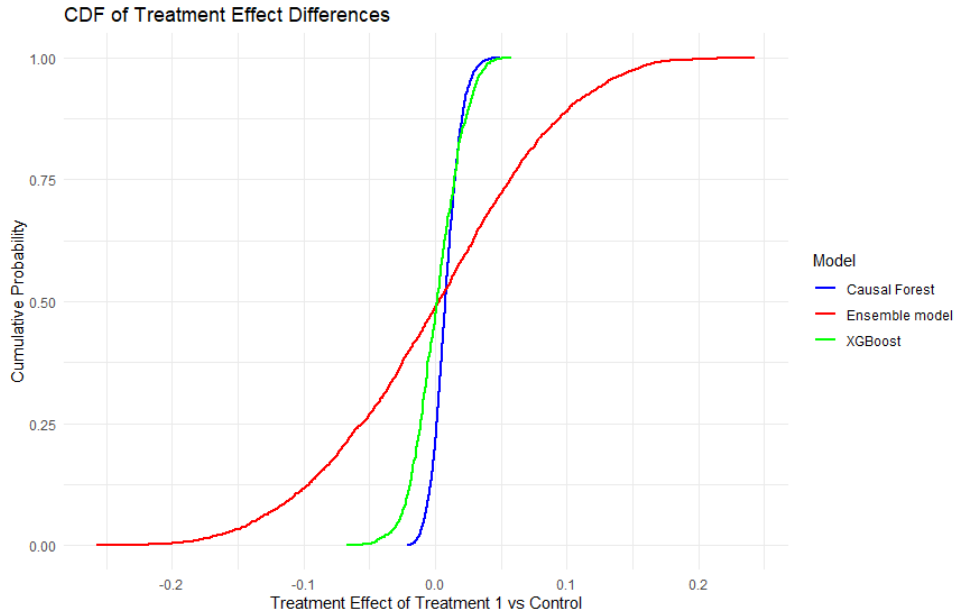

Figure 5: CDFs of the treatment effects of the test dataset for Treatment 1 vs Control for the ensemble, XGBoost and causal forest models

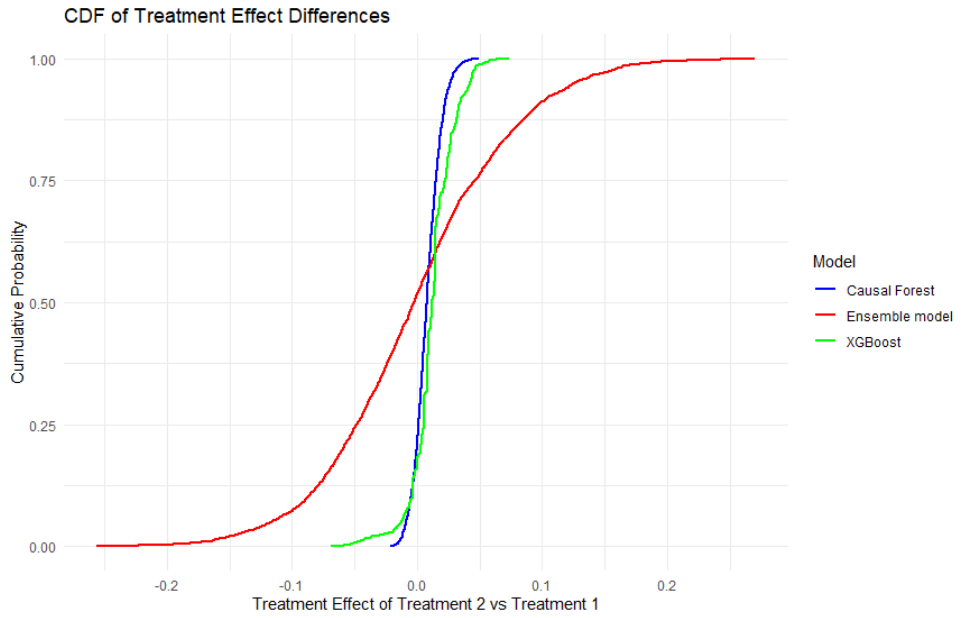

Figure 6: CDFs of the treatment effects of the test dataset for Treatment 2 vs Treatment 1 for the ensemble, XGBoost and causal forest models

E Personalized Treatment Allocations

Policy-prescribed Treatment	π_{logreg}	π_{lasso}	π_{CART}	π_{RF}	π_{XGB}	$\pi_{causaltree}$	$\pi_{causalforest}$	$\pi_{ensemble}$
Control (*Training set*)	0.322	0	0.380	0.446	0.479	0.105	0.167	0.325
Treatment 1 (*Training set*)	0.289	0.956	0.193	0.299	0.340	0.175	0.505	0.338
Treatment 2 (*Training set*)	0.392	0.044	0.427	0.255	0.182	0.719	0.328	0.337

Table 10: Fraction of observations (user-ad pairs) assigned to the control / treatment groups by the personalized policies for the **training dataset**

F Programming Code

The code package includes the files for replicating the results of this paper. All models are implemented and evaluated in R. For every individual model, a separate R file is included in the R package that designs and evaluates the personalized social media advertisements for the training and test datasets. The function of the additional files in the code package is further described in the *"README"* file.