Assignment 7: Customer Targeting and Targeting Policy Evaluation

Günter J. Hitsch

May 27, 2020

Contents

1 Overview and data

In this assignment you will work with data from a company that interacts with its customers using multiple forms of targeting, including direct mail, e-mail, and display or Facebook advertising. The majority of sales occur through the company's online or direct mail/phone channels.

We will use data from a direct mail targeting campaign that is annually repeated in fall. The data are in the data frame `crm_df` in the file `Targeting-Data.RData`.

The data include a `customer_id`, a key that is used by the company to track customer online and offline activities, whether a customer qualifies for and is exposed to a targeting effort, and customer purchases across all sales channels. The data include twenty customer attributes (often called *features*) or behavior variables. These data record the past online and offline customer behavior, RFM-type variables including past purchases across various product categories, and some customer demographics. The privacy policy of the company does not allow us to reveal the exact identity of the variables, and hence they are named `x_1`, `x_2`, etc. Also, the variables are scaled (divided by their standard deviation). Note that scaling does not affect the predictive power of the estimated model.

The customer attributes are captured exactly seven days before a direct mail is sent to the customers. The purchase response during the months after the customer was targeted is recorded in `spend` (measured in dollars).

`target` is an indicator variable that equals 1 if a customer was targeted and 0 otherwise. In particular, `crm_df` includes data from a **randomized sample** of all customers who were eligible for the targeting campaign. In this sample, the company performed an **A/B test** where customers were randomly assigned to a treatment group that was targeted and a control group that was not targeted.

2 Data description

Summarize some key aspects of the data. In particular:

- The probability of targeting a customer in the A/B test was 2/3. Confirm that the observed targeting rate was indeed approximately 2/3.

```
load("Targeting-Data.Rdata")

prop.table(table(crm_df$target))
```

- Provide some basic summary statistics of the `spend` variable.

```
hist(crm_df$spend)

hist(crm_df[crm_df$spend > 0, ]$spend)

summary(crm_df$spend)
sd(crm_df$spend)
```

- Document the probability of a purchase, and summarize and visualize the distribution of `spend` *conditional* on a purchase, i.e. given `spend` > 0. You will see why it is more informative to separately document the purchase probability and spending conditional on a purchase compared to simply describing the overall, unconditional variation in `spend`.

```
prop.table(table(crm_df$spend > 0))

summary(crm_df[crm_df$spend > 0,]$spend)

sd(crm_df[crm_df$spend > 0,]$spend)
```

3 Estimation (model training)

First, set a seed for the random number generator and add a new column to `crm_df`, called `validation_sample`:

```
set.seed(5807)

library(tidyr)

crm_df = crm_df %>%
   mutate(validation_sample = rbinom(n(), size = 1, prob = 0.5))
```

Initializing the random number generator using the `set.seed` function ensures that you will get the same sequence of random numbers every time you re-run your script. In principle you can choose *any* number in `set.seed`. However, to ensure that we all get the *exact* same results and can easily compare our work, please use the number indicated above, 5807.

Note that we split the sample into two halves that are of roughly equal size (`prob = 0.5`).

Now estimate a regression model to predict the spending level (`spend`) given all customer attributes, `x_1`, ..., `x_20`.

```
train_df = crm_df %>%
   filter(validation_sample == 0) %>%
   filter(target == 1) %>%
   select(x_1:spend)

fit_lm = lm(spend ~ ., data = train_df)

summary(fit_lm)
```

Make sure that that you only use the training sample, i.e. the observations where `validation_sample` is 0.

Furthermore, following a widely used approach in the industry, only predict spending given that a customer was targeted (only use observations when `target` equals 1).

Inspect the estimated regression coefficients. Does the regression output provide any evidence that the model can predict spending, i.e. that there is a statistical association between the customer attributes and spending, conditional on being targeted?

4 Model validation

Provide evidence for the validity of the model.

(1) Use `predict` to predict expected customer-level spending, and use the `cut_number` function to assign each customer to one of 20 segments according to expected (predicted) spending. Note that in a linear regression the `predict` function does not require the `type = "response"` argument:

```
predicted_spend = predict(fit, newdata = <data set or placeholder>)
```

```
valid_df = crm_df %>%
   filter(validation_sample == 1)

#valid_df = crm_df %>%
#   filter(validation_sample == 1) %>%
#   filter(target == 0)


valid_df$predict = cut_number(predict(fit_lm, valid_df), 20, labels = FALSE)

valid_df$value = predict(fit_lm, valid_df)

head(valid_df)

table(valid_df$spend > 0, valid_df$value > 0)

summary(valid_df$spend)
summary(valid_df$value)
summary(valid_df$predict)

hist(valid_df$spend, breaks = 100)

hist(valid_df[valid_df$spend > 0,]$spend, breaks = 100)

hist(valid_df[valid_df$value > 0,]$spend, breaks = 100)

hist(valid_df$predict)
```

(2) Create a summary table that contains average spending and the lift for each of the 20 segments.

Note: In this application the outcome is dollar spending, a continuous variable, not a categorical (0/1) variable. The lift, however, is defined almost exactly as in the categorical case based on the ratio of segment-level average spending relative to average spending among all customers.

```
#lift = 100 * avg of segment / avg of all

p_v = mean(valid_df$spend)

summ_table = valid_df %>%
   group_by(predict) %>%
   summarize(average = mean(spend),
             lift = 100 * average/p_v)
```

(3) Provide a graph of segment-level average spending on the y-axis and the customer score (segment) on the x-axis.

```
ggplot(aes(x = predict, y = average), data = summ_table) + geom_line() + geom_point()+ theme(text = ele
```

(4) Provide a lift chart.

```
ggplot(aes(x = predict, y = lift), data = summ_table) + geom_line() + geom_point() + theme(text = eleme
```

(5) Display the data in (3) and (4) in the form of a table.

```
summ_table
```

Hint: Carefully study the solution to assignment 6 if any of the steps above are unclear. In particular, in step (1) you will create a `score` variable that is based on expected (predicted) spending. The `score` captures customer segments or *groups*.

Recall how to use the `cut_number` function:

```
score = cut_number(predicted_spend, n = <number of groups>, labels = FALSE)
```

Do the results provide evidence for the validity of the model?

```
#Yes
summ_table = valid_df %>%
   group_by(predict) %>%
   summarize(average_spend = mean(spend),
             average_predict = mean(value),
             average_non_zero = mean(ifelse(value > 0, value, 0)),
             lift = 100 * average_spend/p_v)

ggplot(data = summ_table) + geom_line(aes(x = predict, y = average_spend), col ="red") + geom_line(aes(
```

5 Profit evaluation: Traditional approach

Traditionally, marketers have evaluated the success of a CRM campaign based on the profit **level** of a targeting effort:

$$\mathbb{E}[\text{profit}_i|\boldsymbol{x}_i] = m \cdot \mathbb{E}[y_i|\boldsymbol{x}_i] - c$$

Here, $\boldsymbol{x}_i = (x_{i1}, x_{i2}, \ldots, x_{ip})$ includes all information (features) for customer i. m is the profit margin and c is the targeting cost.

This approach is correct if a customer who is not targeted does not make a purchase. Then, the profit level when being targeted is also the **incremental effect** of targeting, because baseline profits without targeting a customer are zero.

Use the percent margin and targeting cost data below to predict the expected profit level (given that the customer is targeted) for each customer based on the expected (predicted) spending level.

```
margin = 0.34          # Percent
cost   = 1.12          # Dollars
```

Evaluate the **total expected profit** if all customers with positive expected profit levels are targeted. The maintained assumption is that profits from customers who are not targeted are zero. To make the total profit number more easily interpretable, scale it to a customer base of one million (divide the total profit by the number of customers in the validation sample and multiply by one million).

What percentage of all customers should be targeted according to the expected profit level?

```
n_cust = length(valid_df$customer_id)
print(n_cust)

n_target = length(valid_df[(margin * valid_df$value - cost) > 0, ]$customer_id)
print(n_target)

profit = sum(margin * valid_df[(margin * valid_df$value - cost) > 0, ]$value - cost)
print(profit)

prfit_million = profit / n_cust * 1000000
#2691936 per 1 million customer
print(prfit_million)

n_target/n_cust
#61.7% are targeted
```

6 Profit evaluation using a randomized sample

The assumption that customers do not purchase unless targeted was often valid a quarter century ago, for example if making a purchase was impossible without a catalog. Today, this assumption is unlikely to hold, because most companies sell through an online channel, in addition to a direct mail or brick-and-mortar retail channel.

If baseline sales without targeting are not zero, how do we correctly evaluate the total targeting profit?

We will use the following approach:

(a) We no longer interpret the expected targeting profit that we previously calculated as the true incremental profit. Instead, we use the expected targeting profit as a *score* to rank customers according to their profitability. In particular, we assume (and then attempt to confirm) that this score variable is correlated with the true incremental profit from targeting a customer.

Note: The *score* that we use here is the expected customer-level targeting profit variable (call it `predicted_targeting_profit`) that we created in section 5. This variable is related to but not identical to the score variable indicating one of the customer segments in section 4 of this assignment. In particular, unlike the segment indicator in section 4, `predicted_targeting_profit` represents a non-discretized, continuous score.

```
valid_df$predicted_targeting_profit = margin * valid_df$value - cost
```

(b) If the score (`predicted_targeting_profit`) is correlated with the incremental targeting profit, we can pursue a targeting approach where we target the top n percent of all customers, where "top n percent" means the n percent of customers with the largest score values.

```
top_percent = seq(from = 0, to = 1, by = 0.01)
profit_df = predict_profit_top_n(top_percent, valid_df$predicted_targeting_profit,
valid_df$target, valid_df$spend, 2/3,
margin,
cost)

print(profit_df)
which.max(profit_df$profit)
#0.32   2772.247
```

(d) To evaluate the total profit when targeting the top n percent of customers we employ a **targeting policy evaluation approach using a randomized sample** (discussed in class). In particular, we choose a range of percentage values $n = 0, 0.01, 0.02, \ldots, 1$, and for each of these values we calculate the corresponding targeting profit.

(e) The relationship between the targeting percentage n and the corresponding targeting profit in step (c) will suggest the **optimal targeting percentage**, i.e. the percentage of customers n^* that maximizes total targeting profits.

To evaluate the total profit when targeting the top n percent of customers, we use the function **predict_profit_top_n**:

```
predict_profit_top_n <- function(top_percent, score, W, spend, treatment_Pr, margin, cost) {

   # Observed profits for treated and untreated units
   profit_0 = margin*spend
   profit_1 = margin*spend - cost

   # Observation-level inverse probability-weighted profit components by targeting status
   profit_component_0 = ((1 - W)/(1 - treatment_Pr))*profit_0
   profit_component_1 = (W/treatment_Pr)*profit_1

   # Output table
   K = length(top_percent)
   profits_df = data.frame(
      top_percent = top_percent,
      profit      = 0
   )

   # Profit scale factor ($1,000 per 1 million customers)
   scale_factor = 1000/length(W)

   for (k in 1:K) {
      if (top_percent[k] < 1e-12) {
         threshold = Inf
      } else if (top_percent[k] > 1 - 1e-12) {
         threshold = -Inf
      } else {
         threshold = quantile(score, probs = 1 - top_percent[k])
      }

      target = as.numeric(score >= threshold)
      profits_df$profit[k] = scale_factor*sum((1 - target)*profit_component_0 +
                                              target *profit_component_1)
   }

   return(profits_df)
}
```

Please keep in mind that you need to run the code that defines the function before you use it.

To use the function you need to provide the following inputs (make sure to use the correct order):

- `top_percent`: A range of values $n = 0, 0.01, 0.02, \ldots, 1$
- `score`: The score variable that we use to rank customers, expected targeting profits
- `W`: The treatment indicator (0/1) indicating if the customer was targeted in the data. In our application: `target`
- `spend`: The observed spending level in the data. In our application: `spend`
- `treatment_Pr`: The treatment probability. In our application: 2/3
- `margin`: Percent margin
- `cost`: Targeting cost

Note that the first four variables (`top_percent`, `score`, `W`, `spend`) are vectors (columns), whereas the other variables are numbers.

Example: If your data are in the data frame `df` that contains the columns `target`, `spend`, and `predicted_targeting_profit` (the score), then you can use the `predict_profit_top_n` function as follows:

Note that there is no need to put all the input data into separate rows—the way the code is written is for readability/clarity only.

The output of `predict_profit_top_n` is a table (data frame) with the percentages, n (`top_percent`), and the corresponding targeting profits (`profit`) as columns.

Note: The predicted targeting profits are expressed in $1,000 per one million customers.

Tasks

(1) Evaluate the targeting profits over a range of $n = 0, 0.01, 0.02, \ldots, 1$. Examine the results, and plot the relationship between n (on the x-axis) and profits (on the y-axis). Discuss the results.

```
top_percent = seq(from = 0, to = 1, by = 0.01)
profit_df = predict_profit_top_n(top_percent, valid_df$predicted_targeting_profit,
valid_df$target, valid_df$spend, 2/3,
margin,
cost)

ggplot(data = profit_df) + geom_line(aes(x = top_percent, y = profit), col ="red") + theme(text = eleme
```

(2) What is the targeting profit if none of the customers are targeted? What is the targeting profit under a blanket targeting strategy, when all customers are targeted?

```
profit_df %>%
   filter(top_percent == 0)

profit_df %>%
   filter(top_percent == 1)
```

(3) What is the optimal, profit-maximizing targeting percentage, n^*, and what is the corresponding targeting profit?

```
profit_df[which.max(profit_df$profit),]
#0.32   2772.247

profit_df[profit_df$top_percent == 0.62,]
#0.62 2645.101
```

(4) In light of your results, was the assumption that targeting profits in the absence of targeting are zero a good assumption?

```
#No
```

(5) By how much can profits be increased when targeting the top n^* percent of customers relative to the baseline profit if none of the customers are targeted? Express the difference in dollars and as a percentage of the baseline profit without targeting.

```
baseline = profit_df[profit_df$top_percent == 0, 2]
opt_target = profit_df[profit_df$top_percent == 0.32, 2]

opt_target - baseline

opt_target/baseline - 1
```