I pledge my honor that I have not violated the Honor Code during this examination.

Nichada Wongrassamee December 11, 2024

(My typed name is an acknowledgment that my work did not violate the Booth School Honor Code)

Final Assignment: Incrementality and Optimal Personalized Targeting

Giovanni Compiani

Contents

1 Final assignment instructions

The final assignment is an individual assignment. Any discussion of the material with your classmates or any other person constitutes a violation of the Booth honor code.

Very important: Please do not start to work on the assignment one day before the due date. Give it some time, and work on the different parts systematically, step by step. Breaking the overall task into small pieces and writing clean code will make the analysis much easier.

1.1 Group peer evaluations

Typically, there are no problems to report, and you can skip this section.

However, if there were issues in your group, in particular if one or multiple group members did not contribute to the group work, you can report this by adding a cover sheet or appendix to your final assignment submission.

In particular, provide an **effort rating**. The effort rating ranges from 0–100%. For example, a 90% rating implies that the group member will get 90% of the group grade (for all assignments). The average rating across all members is taken as the final effort rating for a group member. If no effort rating is turned in, a default rating of 100% will be used.

Also, please **explain the reason for a rating below 100%** in a few words.

1.2 Submission instructions

The final assignment will be due on **Wednesday, December 11, 2024, by 11:59 p.m.**

Your submission will not be graded and you will get a score of zero if you don't submit the "Data and Class Materials Usage Restrictions" form by the deadline.

Submitting the final

- Submission on Canvas
- Please ensure that you submit a single document in pdf format.
- Indicate your name, section that you are enrolled in, and your student ID on the cover page.
- The cover page needs to include the Booth School Honor Code, your typed name (or alternatively a scanned signature), and an acknowledgment that your typed name indicates that you did not violate the Honor Code. Example:

  I pledge my honor that I have not violated the Honor Code during this examination.

  John Doe

  December 11, 2024

  (My typed name is an acknowledgment that my work did not violate the Booth School Honor Code)
- Your finals will be graded within about 7 days.
- All requests for re-grading need to be submitted in writing with a detailed explanation of why you would like me to re-consider your exam within two weeks of receipt of your exam. Regrading might results in an increase or decrease in the final grade.
- The material covered in class gives you the tools you need to complete this assignment. However, if you have clarification questions, please do not send an email but rather post a Canvas thread so that all students can benefit from the exchange.

2 Managing a computationally intensive data-analysis

Some of the analysis will take a nontrivial amount of time. To be efficient, save the results from the computationally intensive steps, such as the model `fit` objects or the predicted outputs, to a file.

Also, when you conduct a long, extensive analysis, it is useful to spread the code across multiple scripts or R Markdown files. For example, you may want to create three R Markdown files corresponding to the three steps in the analysis outlined below.

3 Overview

As in the previous assignment, we use customer-level data from the data base of a company that sells kitchenware, housewares, and specialty food products. We again use the October 2017 sample that contains records on 250,000 customers, and in addition we use a sample from a similar catalog mailing in October 2018 that contains 125,000 records. Please refer to the last assignment for more details on the variables in the data set.

Both data sets have a very important property: The catalog-mailing in the October 2017 and 2018 data was fully **randomized**—the treatment, `mailing_indicator`, was randomly assigned to the customers. This will allow us to estimate the causal effect of catalog mailing for every customer in the sample.

The goal of the analysis is to create a **fully personalized** targeting policy based on the **incremental effect** of targeting, as opposed to the *level* of profits (which may have been achieved even without any targeting efforts).

Detailed hints on how to conduct the analysis are provided below.

4 Summary of findings

In your write-up, first **summarize your key findings** and the main insights that you gained from the analysis. Please be concise and provide a summary that does not exceed one page. The more detailed analysis supporting your findings should follow this summary and should be organized in the four main steps described next.

5 Step 1: Estimation and prediction of conditional average treatment effects

First, we load the relevant packages.

```r
library(bit64)
library(data.table)
library(glmnet)
library(ggplot2)
library(knitr)
library(corrplot)
library(dplyr)
```

We use the 2017 data to estimate and validate several models to predict the incremental effects.

```r
data_folder = "Data"

load(paste0(data_folder, "/Customer-Development-2017-2.RData"))
```

Split the sample into a 50 percent training and 50 percent validation sample. Please make sure to **use the same seed** as in the code chunk below for comparability.

```r
set.seed(2001)
crm_DT[, training_sample := rbinom(nrow(crm_DT), 1, 0.5)]
```

To make the code more readable, I recommend renaming the `mailing_indicator` to make it clear that the randomized mailing is the treatment, W_i.

```
setnames(crm_DT, "mailing_indicator", "W")
```

5.1 Data pre-processing

As in the previous assignment, remove highly correlated features from the data set.

```
# Remove highly correlated feature from dataset

# inspect data
dim(crm_DT)
```

```
[1] 250000    156
```

```
cor_matrix = cor(crm_DT[, !c("customer_id", "W", "outcome_spend"),
                        with = FALSE])
```

Now use `corrplot` to create a pdf file that visualizes the correlation among all variables in two separate graphs.

```
pdf("Correlation-Matrix.pdf", height = 16, width = 16)
corrplot(cor_matrix, method = "color",
         type = "lower", diag = FALSE,
         tl.cex = 0.4, tl.col = "gray10")

corrplot(cor_matrix, method = "number", number.cex = 0.25, addgrid.col = NA,
         type = "lower", diag = FALSE,
         tl.cex = 0.4, tl.col = "gray10")
dev.off()
```

```
pdf
  2
```

Create a data table that contains the correlations for all variable pairs:

```
# The correlations on the diagonal are 1.0 and the correlations in the upper triangle
# are identical to the correlations in the lower triangle.
# Hence, we do not need to summarize these correlations.
cor_matrix[upper.tri(cor_matrix, diag = TRUE)] = NA

# Converts the lower triangular part of the correlation matrix into a long-format data table,
# where each row represents a correlation between two variables.
cor_DT = data.table(row = rep(rownames(cor_matrix), ncol(cor_matrix)),
                    col = rep(colnames(cor_matrix), each = ncol(cor_matrix)),
                    cor = as.vector(cor_matrix))
cor_DT = cor_DT[is.na(cor) == FALSE]
```

Now find all correlations larger than 0.95 in absolute value.

```
# Eliminate the redundant variables in the `row` column
large_cor_DT = cor_DT[abs(cor) > 0.95]
kable(large_cor_DT, digits = 4)
```

row	col	cor
customer_type_3	online_customer	-0.9582
orders_online_attributed_target	spend_online_attributed_target	0.9654
acquisition_days_since	acquisition_months_since	1.0000
in_database_months	acquisition_months_since	0.9997

row	col	cor
in_database_months	acquisition_days_since	0.9997
emails_days_1yr	emails_days_2yr	0.9604
emailview_3m	emailview_6m	0.9505

```r
# save for use in step2
# save(large_cor_DT, file = "large_cor_DT.RData")
```

```r
# Remove the large_cor_DT$row from crm_DT
crm_DT = crm_DT[, !large_cor_DT$row, with = FALSE]
```

5.2 Randomization checks

Inspect the data to estimate the probability of a mailing (also called propensity score),

$$p = \Pr\{W_i = 1\}.$$

Perform a quick check to assess if the treatment (catalog mailing) was indeed randomized in the whole data base and hence does not depend on the customer attributes, $\boldsymbol{x}_i$.

5.2.1 Method 1 : propensity model

If all p-values for covariates in the summary are insignificant and the overall model fit is poor (low predictive power), it suggests randomization holds.

```r
library(tableone)

# Logistic regression to estimate propensity score
propensity_model <- glm(W ~ ., data = crm_DT, family = binomial)
summary(propensity_model)
```

```

Call:
glm(formula = W ~ ., family = binomial, data = crm_DT)

Coefficients:
                                Estimate Std. Error z value Pr(>|z|)
(Intercept)                   -2.091e-01  2.872e-01  -0.728 0.466631
customer_type_L                5.142e-03  1.311e-02   0.392 0.694920
customer_type_C               -4.435e-06  1.304e-02   0.000 0.999729
clicks_product_type_504_3m     3.757e-04  7.570e-04   0.496 0.619676
clicks_product_type_112_6m    -4.203e-04  6.862e-04  -0.613 0.540202
clicks_product_type_301_1m     9.685e-03  5.551e-03   1.745 0.081014 .
clicks_product_type_001_12m    1.152e-03  9.790e-04   1.177 0.239156
clicks_product_type_201_1m     4.068e-03  2.453e-03   1.658 0.097229 .
clicks_product_type_201_6m    -6.466e-04  6.542e-04  -0.988 0.322977
web_activity_3m               -6.018e-05  1.258e-04  -0.478 0.632404
clickthrough_1m               -3.168e-03  8.912e-03  -0.355 0.722260
emailview_1m                  -2.355e-04  1.506e-03  -0.156 0.875742
orders_online_1yr              1.453e-02  7.574e-03   1.918 0.055053 .
online_customer               -1.993e-03  2.733e-02  -0.073 0.941867
orders_season_E                7.384e-04  1.635e-03   0.452 0.651516
orders_mail_dept_h_1yr        -1.969e-02  1.477e-02  -1.333 0.182538
spend_d_h_1yr                  8.717e-06  1.234e-04   0.071 0.943706
```

```
spend_online_attributed_target    -5.828e-06  8.278e-06  -0.704 0.481433
dollars_season_C                   9.989e-06  9.477e-06   1.054 0.291848
spend_season_C                    -9.600e-06  9.985e-06  -0.961 0.336310
spend_e                            4.107e-05  4.630e-05   0.887 0.375093
spend_z1                           4.422e-06  1.546e-05   0.286 0.774902
spend_period_1b                    4.339e-04  1.525e-04   2.845 0.004441 **
spend_period_2b                   -3.381e-05  3.432e-05  -0.985 0.324676
spend_h_3yr                        2.513e-05  8.939e-05   0.281 0.778604
spend_g_3yr                        6.306e-05  3.701e-05   1.704 0.088359 .
last_years_since                   1.539e-02  1.717e-02   0.896 0.370061
spend_instore_o                   -2.567e-06  2.807e-05  -0.091 0.927121
spend_instore_o_3yr                3.214e-05  7.351e-05   0.437 0.661962
spend_instore_h_3yr                3.315e-04  1.683e-04   1.970 0.048795 *
spend_instore_t                   -5.658e-06  7.965e-05  -0.071 0.943372
spend_online_sh                    1.317e-04  1.575e-04   0.836 0.402892
spend_online_o_1yr                 7.701e-05  1.168e-04   0.659 0.509808
spend_online_n_3yr                -2.160e-05  1.040e-04  -0.208 0.835374
spend_online_o_3yr                -1.149e-04  5.511e-05  -2.085 0.037095 *
spend_online_a_1yr                -7.673e-05  8.924e-05  -0.860 0.389932
spend_online_a_3yr                 5.135e-05  3.484e-05   1.474 0.140577
orders_online_t_3yr                2.326e-02  1.622e-02   1.434 0.151529
spend_online_sg_1yr               -5.139e-04  2.657e-04  -1.934 0.053063 .
spend_online_g_1yr                 6.746e-05  1.474e-04   0.458 0.647133
orders_online_s_1yr                2.569e-03  2.091e-02   0.123 0.902201
customer_type_7                    4.110e-02  4.206e-02   0.977 0.328465
customer_type_A                   -3.477e-03  1.315e-02  -0.264 0.791481
orders_attributed_mail_type_B      4.358e-03  4.401e-03   0.990 0.322170
spend_attributed_mail_type_B      -6.477e-05  4.424e-05  -1.464 0.143190
spend_attributed_mail_type_C       3.058e-05  2.300e-05   1.330 0.183633
mean_spend_attributed_mail_type_C  7.012e-05  8.369e-05   0.838 0.402148
mean_spend_attributed_mail_type_A -8.015e-05  1.095e-04  -0.732 0.464396
clicks_product_type_104_2yr        1.626e-03  3.201e-03   0.508 0.611406
clicks_product_type_312_3yr       -9.083e-05  4.808e-04  -0.189 0.850158
orders_e                          -5.164e-03  6.779e-03  -0.762 0.446243
orders_mail_dept_c_1yr            -1.736e-02  1.361e-02  -1.276 0.202005
spend_d_s_1yr                      2.019e-04  1.544e-04   1.308 0.190984
spend_d_k_1yr                      9.186e-05  1.501e-04   0.612 0.540487
spend_a_1yr                        8.551e-05  1.267e-04   0.675 0.499628
spend_h_1yr                        1.174e-05  1.305e-04   0.090 0.928317
spend_direct_1yr                  -1.114e-04  1.068e-04  -1.043 0.296773
acquisition_months_since           3.126e-05  4.364e-05   0.716 0.473812
spend_online_c                     1.579e-05  7.512e-05   0.210 0.833495
spend_online_a_6m                  6.903e-05  7.925e-05   0.871 0.383728
orders_online_o                    2.189e-03  1.272e-03   1.721 0.085259 .
orders_online_c_1yr               -1.371e-02  2.060e-02  -0.665 0.505837
spend_online_b_1yr                 3.125e-05  1.395e-04   0.224 0.822675
customer_type_2                   -2.740e-02  1.434e-02  -1.910 0.056111 .
customer_income                    6.543e-08  9.693e-08   0.675 0.499619
orders_attributed_mail_type_A      2.273e-03  5.138e-03   0.442 0.658194
spend_attributed_mail_type_A      -4.397e-06  5.599e-05  -0.079 0.937406
clicks_product_type_312_3m        -1.193e-04  1.883e-03  -0.063 0.949487
clicks_product_type_301_2yr       -1.815e-04  1.083e-03  -0.168 0.866898
clickthrough_6m                   -1.226e-03  3.908e-03  -0.314 0.753657
emails_days_2yr                    1.472e-05  4.811e-05   0.306 0.759681
```

```
emails_3m                          2.719e-04  3.746e-04   0.726 0.467916    
emailreceived_months_since         1.780e-04  2.704e-04   0.658 0.510420    
orders_3yr                        -2.766e-03  2.132e-03  -1.297 0.194501    
orders_mail_dept_d_1yr            -1.410e-02  9.628e-03  -1.465 0.142973    
spent_q                           -1.425e-04  1.066e-04  -1.337 0.181267    
spend_instore_q                    4.057e-04  3.984e-04   1.018 0.308601    
spend_online_s                    -6.902e-06  3.725e-05  -0.185 0.853000    
clicks_product_type_112_3yr        3.359e-04  4.318e-04   0.778 0.436617    
spend_instore_n_3yr                7.632e-05  8.504e-05   0.897 0.369491    
spend_instore_p                   -1.851e-04  1.862e-04  -0.994 0.320302    
spend_instore_p_1yr               -1.173e-03  5.589e-04  -2.099 0.035808 *  
orders_online_n_1yr                1.181e-02  1.640e-02   0.720 0.471490    
spend_instore_a_yr                -1.748e-05  4.985e-05  -0.351 0.725826    
orders_attributed_mail_type_C      2.900e-04  2.566e-03   0.113 0.909986    
clickthrough_3yr                   2.585e-04  6.950e-04   0.372 0.709917    
spend_instore_g_1yr               -7.045e-05  4.165e-04  -0.169 0.865676    
spend_period_3b                   -2.289e-05  3.347e-05  -0.684 0.494068    
spend_instore_a                   -5.178e-06  1.049e-05  -0.494 0.621568    
spend_direct_g                     7.240e-06  1.846e-05   0.392 0.694946    
emailview_24m                      6.902e-05  1.709e-04   0.404 0.686374    
spend_online_h_3yr                -8.878e-05  8.136e-05  -1.091 0.275195    
emailview_months_since             3.021e-04  2.752e-04   1.098 0.272250    
orders_instore_c                   7.299e-03  7.107e-03   1.027 0.304432    
emails_1yr                        -3.020e-05  1.542e-04  -0.196 0.844747    
clicks_product_type_001_1m         6.093e-03  6.066e-03   1.004 0.315170    
clickthrough_3m                    6.255e-03  7.207e-03   0.868 0.385443    
orders_d_1yr                       2.694e-02  7.387e-03   3.648 0.000265 ***
orders_h                          -1.573e-03  7.434e-04  -2.116 0.034311 *  
clicks_product_type_104            7.846e-04  8.943e-04   0.877 0.380335    
clicks_product_type_502_3m        -3.239e-04  1.316e-03  -0.246 0.805497    
web_activity_1m                   -1.524e-04  4.959e-04  -0.307 0.758650    
orders_instore_m                   4.117e-03  3.368e-03   1.222 0.221536    
spend_notz_1yr                    -5.116e-04  1.719e-04  -2.977 0.002908 ** 
orders_online_sh                  -9.471e-03  8.170e-03  -1.159 0.246363    
orders_instore_n                  -6.110e-04  3.244e-03  -0.188 0.850598    
clicks_product_type_502_1yr       -1.300e-04  2.789e-04  -0.466 0.641229    
orders_hm                         -1.389e-03  1.142e-03  -1.216 0.224109    
emailview                         -1.686e-06  5.011e-05  -0.034 0.973162    
spend_m_1yr                       -2.471e-04  2.751e-04  -0.898 0.369024    
clicks_product_type_301_3m        -3.419e-03  2.969e-03  -1.152 0.249462    
orders_instore_a_yr               -8.722e-04  7.505e-03  -0.116 0.907479    
clicks_product_type_001_6m        -2.140e-03  2.153e-03  -0.994 0.320053    
clickthrough_months_since          5.164e-05  1.934e-04   0.267 0.789484    
orders_online_h_1yr                2.120e-03  1.320e-02   0.161 0.872390    
orders_instore_h_3yr              -2.208e-02  1.070e-02  -2.064 0.039043 *  
spend_period_1a                    4.785e-04  1.903e-04   2.514 0.011926 *  
orders_total                       2.065e-04  4.583e-04   0.451 0.652285    
spend_online_s_3yr                -1.647e-04  9.672e-05  -1.703 0.088502 .  
spend_M_3yr                        4.868e-05  7.972e-05   0.611 0.541464    
orders_c                           2.786e-03  2.374e-03   1.174 0.240479    
spend_l                           -6.838e-05  7.134e-05  -0.959 0.337775    
spend_instore_n                    2.903e-05  8.818e-05   0.329 0.741955    
orders_z                           8.898e-03  1.729e-02   0.515 0.606896    
web_activity_24m                   5.454e-05  6.115e-05   0.892 0.372424    
```

```
spend_instore_q_3yr                -2.256e-04  5.494e-04  -0.411 0.681390    
clicks_product_type_504             4.219e-05  5.453e-05   0.774 0.439058    
emailview_6m                       -4.591e-04  6.035e-04  -0.761 0.446813    
buy_instore_days_since             -2.740e-06  5.639e-06  -0.486 0.627017    
spend_h                            -8.566e-06  1.109e-05  -0.772 0.439972    
clicks_product_type_801_3yr        -2.256e-04  5.926e-04  -0.381 0.703492    
spend_nott_1yr                      2.604e-04  3.023e-04   0.861 0.388989    
orders_online_s_3yr                 3.865e-03  8.909e-03   0.434 0.664427    
spend_instore_h_1yr                -2.378e-04  2.502e-04  -0.951 0.341825    
spend_online_b_3yr                  4.172e-05  5.042e-05   0.828 0.407905    
orders_instore_s_3yr                1.327e-03  4.350e-03   0.305 0.760345    
orders_instore_r_3yr               -9.215e-03  2.155e-02  -0.428 0.668964    
clicks_product_type_502_1m         -5.249e-04  1.340e-03  -0.392 0.695154    
mean_spend_attributed_mail_type_B   1.112e-04  9.423e-05   1.180 0.237830    
clicks_product_type_201_3yr        -4.567e-04  4.427e-04  -1.032 0.302277    
store_trips                        -1.013e-03  9.722e-04  -1.042 0.297466    
spend_online_t_1yr                 -2.508e-04  2.849e-04  -0.880 0.378710    
orders_online_o_1yr                 8.237e-04  1.400e-02   0.059 0.953092    
spend_online_k                     -2.304e-05  2.428e-05  -0.949 0.342755    
spend_online_s_1yr                  1.125e-04  2.063e-04   0.545 0.585517    
clicks_product_type_503            -3.039e-05  2.866e-05  -1.060 0.289103    
orders_instore_t_3yr                1.889e-02  1.306e-02   1.446 0.148222    
outcome_spend                       1.851e-03  1.221e-04  15.153  < 2e-16 ***
customer_id                         8.293e-10  2.741e-10   3.025 0.002482 ** 
training_sample                     1.660e-02  8.509e-03   1.951 0.051040 .  
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

(Dispersion parameter for binomial family taken to be 1)

    Null deviance: 317490  on 249999  degrees of freedom
Residual deviance: 317068  on 249850  degrees of freedom
AIC: 317368

Number of Fisher Scoring iterations: 4
```

- There are some variables that are statistically significant such as orders_d_1yr, spend_period_1b, etc.

5.2.2 Method 2 : SMD check

Check statistically by comparing SMD value. If SMD value < 0.1, it is considered good balance, it suggests randomization holds.

```
# Method 2 : check by covariate

# Check balance using standardized mean differences
library(tableone)
table_one <- CreateTableOne(vars = setdiff(names(crm_DT), "W"), strata = "W", data = crm_DT)

# SMD < 0.1 is considered evidence of good balance, regardless of statistical significance (p-value).

# Extract the SMD values
smd_matrix <- ExtractSmd(table_one)

# Convert to data frame
```

```
smd_df <- as.data.frame(smd_matrix)
smd_df$Covariate <- rownames(smd_df)
rownames(smd_df) <- NULL

# Filter for SMD > 0.1
imbalanced_covariates <- smd_df[smd_df$`1` > 0.1, ]  # Adjust column name if needed
print(imbalanced_covariates)
```

```
[1] 1 vs 2    Covariate
<0 rows> (or 0-length row.names)
```

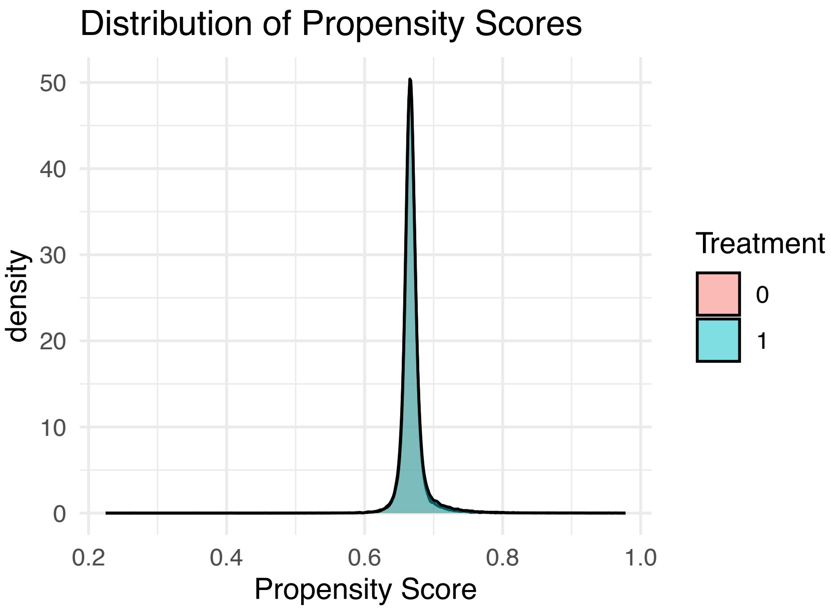
- There is no variable that has SMD > 0.1

5.2.3 Method 3 : Density plot

Density plot of the propensity scores from treatment (targeted) and control (non targeted) overlapped completely, indicating randomization.

```
# Predict propensity_score
crm_propensity <- copy(crm_DT)
crm_propensity[, propensity_score := predict(propensity_model, type = "response")]

# Show head
# crm_propensity[, .(propensity_score)][1:6]
```

```
# Method 3 : Visualize propensity score
ggplot(crm_propensity, aes(x = propensity_score, fill = factor(W))) +
  geom_density(alpha = 0.5) +
  labs(title = "Distribution of Propensity Scores", x = "Propensity Score", fill = "Treatment") +
  theme_minimal()
```



- Hence, it can be conclude from the propensity score and distribution that customer attributes are independent from the W. This indicates that the data is randomize.

5.3 Estimation of incremental effect of targeting (CATE)

We use the training sample to estimate the CATE (conditional average treatment effect) on dollar spending, τ_i, due to catalog targeting:

$$\tau_i = \mathbb{E}[Y_i(1) - Y_i(0)|\boldsymbol{x}_i].$$

In particular, we estimate the CATE using the *"two-in-one" regression* approach discussed in class and we consider the following models:

(a) OLS
(b) LASSO
(c) Elastic Net

You can include **interactions between variables in an R model formula** using either the syntax `x*z` or `x:z`. When you use `x*z`, R will include `x`, `z`, and the interaction (`x` multiplied with `z`) in the regression. Using `x:z`, only the interaction is included. Similarly, using `.*z`, all variables in the data set and all interactions of the variables with `z` will be included, while `.:z` only adds the interaction terms with `z`.

```
x*z : Include x,z (covariantes) and x $\cdot$ z (interaction)
x:z : Include x $\cdot$ z
.*z : Includes all variables and their interactions with z
."z : Includes only interaction terms with z
```

5.4 Estimation: Hints

To simplify your code I recommend **creating separate training and validation samples** from the full data set. Also, exclude the variables that are not used in the statistical analysis:

```
training_DT   = crm_DT[training_sample == 1,
                       !c("customer_id", "training_sample"), with = FALSE]
validation_DT = crm_DT[training_sample == 0,
                       !c("customer_id", "training_sample"), with = FALSE]

dim(training_DT) # 148 features including W , outcome_spend
```

```
[1] 125123    148
```

5.4.1 Two-in-One Regression

$$Y_i = \beta_0 + \beta_1 W + \beta_2 x_i + \beta_3 (W \cdot x_i) + \epsilon_i$$

- $\beta_0 + \beta_2 x_i$: expected outcome of W=0 - $\beta_1 + \beta_3 x_i$: treat ment effect τ_i CATE

The "two-in-one" regression combines the estimation of both the treatment and control outcomes into a single model. It estimates the Conditional Average Treatment Effect (CATE) directly by including an interaction term between treatment W and the covariates xi. Output treatment effects (CATE) is in coefficients associate with :

W (main treatment effect, β_1) Interaction Terms (W:x1, W:x2, . . .) capture how treatment effects vary with covariates β_3

Unlike the conditional expectation in previous exercise that we have to fit models and compute the difference treatment effect manually E[Y1|x] - E[Y0|x].

```
# Find CATE (T_i) with Two-in-One Regression

# Fit an OLS model, using all covariates xi to estimate Ti :
```

```
fit_ols <- lm(outcome_spend ~ W * .,
                data = training_DT)

# View results
summary_OLS = summary(fit_ols)

# save to result for comparison
results = data.table(input = rownames(summary_OLS$coefficients),
                     est_OLS = summary_OLS$coefficients[, 1],
                     p_OLS = summary_OLS$coefficients[, 4])
```

For replicability and comparability, **provide `cv.glmnet` with the folds used for cross-validation**. Set the seed below and then draw numbers indicating the fold $(1, 2, \ldots, 10)$ that an observation belongs to:

```
set.seed(961)

N_obs_training = nrow(training_DT)
folds = sample(1:10, N_obs_training, replace = TRUE)
```

You can now use these fold id's in `glmnet` as follows:

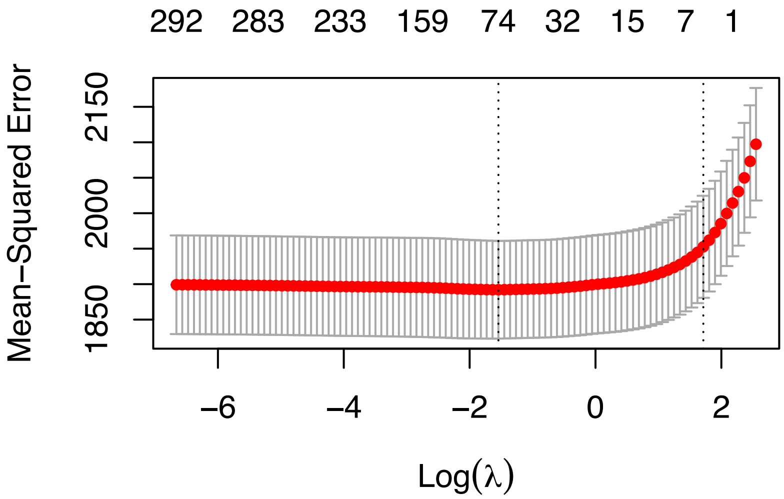
```
# fit_glmnet = cv.glmnet(..., foldid = folds)
```

5.4.2 Lasso Regression

```
# LASSO Regression

# Prepare data for glmnet (model.matrix creates interaction terms automatically)
X_train <- model.matrix(outcome_spend ~ W * ., data = training_DT)[, -1]
y_train <- training_DT$outcome_spend

# Fit a LASSO model (alpha = 1 for LASSO)
fit_LASSO <- cv.glmnet(x = X_train, y = y_train, alpha = 1.0, foldid = folds)

bestlambda_coef = coef(fit_LASSO, s = "lambda.min")[,1]
length(bestlambda_coef) #294
```

```
[1] 294
```

```
# Add est_LASSO to result table
# s= "lambda.min". Feature selection from the LASSO model at the
# optimal regularization parameter, determined by cross-validation
# behind the scene
# [,1] convert coef() to matrix

results[, est_LASSO := bestlambda_coef]

# Plot cross-validation results
plot(fit_LASSO)
```



Selecting best lambda model

X-Axis : Log(λ) - smaller lambda : less regularization - higher lambda : strong regularization. Model has selected less features and is more simple.

Y-Axis (MSE) - lower MSE indicates better performing model

Left Dashed Vertical Line - left dashed line : lambda min. The most accurate model has around 80 variable - right dashed line : optimize the speed of the model while compromise the accuracy

5.4.3 Elastic Net

```r
# TODO why lasso and elnet produce the same prediction ?

# Output table
alpha_seq = seq(0, 1, by = 0.05)
L = length(alpha_seq)
rmse_DT = data.table(alpha = alpha_seq, mean_cv_error = rep(0, L))

# Calculate cross-validation error for different alpha values
for (i in 1:L) {
  cv_i = cv.glmnet(x = X_train,
                   y = y_train,
                   alpha = rmse_DT[i, alpha],
                   foldid = folds)
  rmse_DT[i, mean_cv_error := min(cv_i$cvm)]
  cat("Iteration", i, "cv_error:", min(cv_i$cvm), "\n")
}
```

```
Iteration 1 cv_error: 1893.354
Iteration 2 cv_error: 1892.34
Iteration 3 cv_error: 1891.996
Iteration 4 cv_error: 1891.98
Iteration 5 cv_error: 1891.994
Iteration 6 cv_error: 1892.005
Iteration 7 cv_error: 1892.022
Iteration 8 cv_error: 1892.02
Iteration 9 cv_error: 1892.037
Iteration 10 cv_error: 1892.032
Iteration 11 cv_error: 1892.034
Iteration 12 cv_error: 1892.05
Iteration 13 cv_error: 1892.078
Iteration 14 cv_error: 1892.097
Iteration 15 cv_error: 1892.105
Iteration 16 cv_error: 1892.094
Iteration 17 cv_error: 1892.08
Iteration 18 cv_error: 1892.08
Iteration 19 cv_error: 1892.097
Iteration 20 cv_error: 1892.102
Iteration 21 cv_error: 1892.105
```

```r
# Optimal alpha:
index_min = which.min(rmse_DT$mean_cv_error)
opt_alpha = rmse_DT[index_min, alpha]

cat("Optimal alpha:", opt_alpha, "\n")
```

```
Optimal alpha: 0.15
```

```r
fit_elnet <- cv.glmnet(x = X_train, y = y_train, alpha = opt_alpha, foldid = folds)
results[, est_elnet := coef(fit_elnet, s = "lambda.min")[,1]]
```

5.4.4 Estimation results

```r
kable(results, digits = 4)
```

input	est_OLS	p_OLS	est_LASSO	est_elnet
(Intercept)	-10.4238	0.0000	-2.6007	-2.7579
W	15.2711	0.0000	0.4445	0.4861
customer_type_L	1.5324	0.0200	0.9183	0.9301
customer_type_C	-0.7818	0.2322	-0.1329	-0.1283
clicks_product_type_504_3m	-0.0008	0.9838	0.0000	0.0008
clicks_product_type_112_6m	-0.0005	0.9872	-0.0026	-0.0043
clicks_product_type_301_1m	0.6063	0.0375	0.2648	0.2797
clicks_product_type_001_12m	0.1881	0.0029	0.0000	0.0000
clicks_product_type_201_1m	0.5617	0.0000	0.0119	0.0618
clicks_product_type_201_6m	0.0487	0.1328	0.0865	0.0840
web_activity_3m	0.0405	0.0000	0.0093	0.0095
clickthrough_1m	1.0096	0.0273	0.5472	0.5488
emailview_1m	-0.0578	0.4418	0.0299	0.0349
orders_online_1yr	-0.8464	0.0277	0.0000	0.0000
online_customer	8.9686	0.0000	2.7317	2.8460
orders_season_E	-0.0922	0.2672	-0.0417	-0.0473
orders_mail_dept_h_1yr	0.5158	0.4482	0.0000	0.0250
spend_d_h_1yr	0.0144	0.0179	0.0010	0.0011
spend_online_attributed_target	0.0025	0.0000	0.0026	0.0025
dollars_season_C	-0.0003	0.5367	0.0006	0.0006
spend_season_C	-0.0010	0.0566	0.0000	0.0000
spend_e	-0.0009	0.6988	0.0000	0.0000
spend_z1	0.0005	0.5564	0.0000	0.0000
spend_period_1b	-0.0143	0.0699	0.0055	0.0055
spend_period_2b	0.0011	0.5472	0.0000	0.0000
spend_h_3yr	-0.0029	0.5142	0.0009	0.0008
spend_g_3yr	0.0021	0.2402	0.0000	0.0000

input	est_OLS	p_OLS	est_LASSO	est_elnet
last_years_since	-1.1486	0.1886	-2.2488	-2.2459
spend_instore_o	-0.0006	0.6392	0.0000	0.0000
spend_instore_o_3yr	0.0102	0.0071	0.0011	0.0012
spend_instore_h_3yr	-0.0068	0.4107	0.0000	0.0000
spend_instore_t	0.0079	0.0473	0.0012	0.0014
spend_online_sh	-0.0279	0.0002	-0.0005	-0.0005
spend_online_o_1yr	0.0163	0.0056	0.0204	0.0178
spend_online_n_3yr	0.0176	0.0007	0.0026	0.0032
spend_online_o_3yr	0.0078	0.0029	0.0048	0.0052
spend_online_a_1yr	0.0291	0.0000	0.0097	0.0093
spend_online_a_3yr	0.0033	0.0597	0.0045	0.0043
orders_online_t_3yr	0.9233	0.2266	0.0000	0.0000
spend_online_sg_1yr	0.0515	0.0000	0.0176	0.0201
spend_online_g_1yr	0.0065	0.3869	0.0005	0.0026
orders_online_s_1yr	5.7816	0.0000	0.5434	0.6552
customer_type_7	-0.9693	0.6425	-3.4669	-3.2623
customer_type_A	0.3546	0.5908	0.0000	0.0000
orders_attributed_mail_type_B	0.1631	0.4542	0.0035	0.0000
spend_attributed_mail_type_B	0.0005	0.8273	0.0001	0.0005
spend_attributed_mail_type_C	0.0027	0.0193	0.0000	0.0000
mean_spend_attributed_mail_type_C	-0.0065	0.1218	0.0000	0.0000
mean_spend_attributed_mail_type_A	0.0060	0.2907	0.0000	0.0000
clicks_product_type_104_2yr	-0.1575	0.3313	0.0000	0.0000
clicks_product_type_312_3yr	0.1216	0.0000	0.0000	0.0000
orders_e	0.3244	0.3330	0.0000	0.0000
orders_mail_dept_c_1yr	1.6206	0.0186	0.0000	0.0000
spend_d_s_1yr	0.0032	0.6816	0.0000	0.0000
spend_d_k_1yr	0.0146	0.0532	0.0000	0.0000
spend_a_1yr	0.0096	0.1199	0.0000	0.0000
spend_h_1yr	-0.0046	0.4724	0.0000	0.0000
spend_direct_1yr	0.0122	0.0297	0.0000	0.0000
acquisition_months_since	0.0013	0.5282	0.0000	0.0000
spend_online_c	-0.0077	0.0413	0.0000	0.0000
spend_online_a_6m	-0.0117	0.0030	0.0000	0.0000
orders_online_o	-0.0460	0.4834	0.0000	0.0000
orders_online_c_1yr	-4.4836	0.0000	0.0000	0.0000
spend_online_b_1yr	0.0326	0.0000	0.0078	0.0087
customer_type_2	0.4242	0.5534	0.0000	0.0000
customer_income	0.0000	0.9320	0.0000	0.0000
orders_attributed_mail_type_A	0.4050	0.1234	0.0000	0.0000
spend_attributed_mail_type_A	-0.0078	0.0074	0.0000	0.0000
clicks_product_type_312_3m	-0.5010	0.0000	-0.0222	-0.0364
clicks_product_type_301_2yr	0.0582	0.3145	0.0000	0.0000
clickthrough_6m	-0.2926	0.1260	0.0000	0.0000
emails_days_2yr	0.0027	0.2676	0.0000	0.0000
emails_3m	0.0462	0.0143	0.0000	0.0000
emailreceived_months_since	0.0134	0.3185	0.0000	0.0000
orders_3yr	-0.1773	0.1061	0.0000	0.0000
orders_mail_dept_d_1yr	-1.1280	0.0213	0.0000	0.0000
spent_q	-0.0062	0.2570	0.0000	0.0000
spend_instore_q	-0.0363	0.0963	0.0000	0.0000
spend_online_s	-0.0034	0.0311	0.0000	0.0000

input	est_OLS	p_OLS	est_LASSO	est_elnet
clicks_product_type_112_3yr	-0.0457	0.0335	0.0000	0.0000
spend_instore_n_3yr	-0.0067	0.1280	0.0000	0.0000
spend_instore_p	0.0003	0.9764	0.0020	0.0025
spend_instore_p_1yr	-0.0377	0.1649	0.0000	0.0000
orders_online_n_1yr	-3.1511	0.0001	0.0000	0.0000
spend_instore_a_yr	-0.0013	0.6072	0.0000	0.0000
orders_attributed_mail_type_C	-0.2688	0.0372	0.0000	0.0000
clickthrough_3yr	0.0775	0.0175	0.0000	0.0000
spend_instore_g_1yr	-0.0163	0.5963	0.0000	0.0000
spend_period_3b	0.0011	0.5260	0.0007	0.0010
spend_instore_a	-0.0001	0.8638	0.0000	0.0000
spend_direct_g	-0.0002	0.8009	0.0000	0.0000
emailview_24m	0.0081	0.3423	0.0000	0.0000
spend_online_h_3yr	-0.0024	0.5548	0.0000	0.0000
emailview_months_since	-0.0009	0.9462	0.0000	0.0000
orders_instore_c	-0.3906	0.2816	0.0000	0.0000
emails_1yr	-0.0136	0.0775	0.0000	0.0000
clicks_product_type_001_1m	-0.6178	0.0461	0.0000	0.0000
clickthrough_3m	0.1477	0.6899	0.0000	0.0000
orders_d_1yr	0.1615	0.6600	0.0000	0.0000
orders_h	0.0646	0.0858	0.0000	0.0000
clicks_product_type_104	0.0581	0.2155	0.0000	0.0004
clicks_product_type_502_3m	0.0170	0.8111	0.0000	0.0000
web_activity_1m	0.0043	0.8735	0.0000	0.0000
orders_instore_m	-0.3569	0.0305	0.0000	0.0000
spend_notz_1yr	-0.0027	0.7560	0.0000	0.0000
orders_online_sh	1.4226	0.0003	0.0000	0.0000
orders_instore_n	-0.5270	0.0006	0.0000	0.0000
clicks_product_type_502_1yr	-0.0007	0.9641	0.0000	0.0000
orders_hm	0.1150	0.0415	0.0000	0.0000
emailview	-0.0021	0.4053	0.0000	0.0000
spend_m_1yr	-0.0084	0.5574	0.0000	0.0000
clicks_product_type_301_3m	0.2178	0.1798	0.0000	0.0000
orders_instore_a_yr	0.3839	0.3202	0.0000	0.0000
clicks_product_type_001_6m	-0.2407	0.0505	0.0000	0.0000
clickthrough_months_since	-0.0033	0.7317	0.0000	0.0000
orders_online_h_1yr	2.7312	0.0000	0.0000	0.0000
orders_instore_h_3yr	0.5801	0.2543	0.0000	0.0000
spend_period_1a	-0.0180	0.0697	0.0000	0.0000
orders_total	0.0016	0.9434	0.0000	0.0000
spend_online_s_3yr	0.0054	0.2726	0.0000	0.0000
spend_M_3yr	0.0014	0.7221	0.0000	0.0000
orders_c	-0.2166	0.0627	0.0000	-0.0034
spend_l	0.0091	0.0050	0.0000	0.0000
spend_instore_n	0.0205	0.0000	0.0019	0.0020
orders_z	1.2292	0.1704	0.0000	0.0000
web_activity_24m	-0.0176	0.0000	0.0000	0.0000
spend_instore_q_3yr	0.0395	0.1826	0.0000	0.0000
clicks_product_type_504	-0.0015	0.6061	0.0000	0.0000
emailview_6m	0.0187	0.5386	0.0000	0.0000
buy_instore_days_since	-0.0001	0.6400	0.0000	0.0000
spend_h	0.0012	0.0374	0.0000	0.0001

input	est_OLS	p_OLS	est_LASSO	est_elnet
clicks_product_type_801_3yr	-0.0898	0.0007	0.0000	-0.0006
spend_nott_1yr	0.0239	0.1266	0.0000	0.0000
orders_online_s_3yr	-0.0023	0.9961	0.0000	0.0000
spend_instore_h_1yr	0.0112	0.3611	0.0000	0.0000
spend_online_b_3yr	-0.0077	0.0021	0.0000	0.0000
orders_instore_s_3yr	-0.3951	0.0717	0.0000	0.0000
orders_instore_r_3yr	0.7846	0.3948	0.0000	0.0000
clicks_product_type_502_1m	-0.0401	0.5695	0.0000	0.0000
mean_spend_attributed_mail_type_B	-0.0054	0.2559	0.0000	0.0000
clicks_product_type_201_3yr	0.0250	0.2618	0.0000	0.0017
store_trips	0.0546	0.2537	0.0000	0.0000
spend_online_t_1yr	-0.0091	0.5533	0.0000	0.0000
orders_online_o_1yr	1.2013	0.0887	0.0000	0.0000
spend_online_k	-0.0012	0.3141	-0.0026	-0.0023
spend_online_s_1yr	-0.0101	0.3434	0.0000	0.0000
clicks_product_type_503	0.0033	0.0368	0.0000	0.0000
orders_instore_t_3yr	0.3557	0.5711	0.0000	0.0000
W:customer_type_L	1.1900	0.1379	0.9995	0.9906
W:customer_type_C	-0.2652	0.7394	0.0000	0.0000
W:clicks_product_type_504_3m	0.0319	0.5294	0.0000	0.0000
W:clicks_product_type_112_6m	-0.0669	0.0949	0.0000	0.0000
W:clicks_product_type_301_1m	-0.0101	0.9759	0.0000	0.0000
W:clicks_product_type_001_12m	-0.1896	0.0055	0.0000	0.0000
W:clicks_product_type_201_1m	-0.7148	0.0000	-0.0099	-0.0693
W:clicks_product_type_201_6m	0.0625	0.1165	0.0000	0.0000
W:web_activity_3m	-0.0201	0.0262	0.0000	0.0000
W:clickthrough_1m	0.1417	0.7961	0.0000	0.0000
W:emailview_1m	0.2057	0.0250	0.0000	0.0000
W:orders_online_1yr	1.7645	0.0002	0.0000	0.0000
W:online_customer	-10.3207	0.0000	0.0000	0.0000
W:orders_season_E	-0.0959	0.3397	0.0000	0.0000
W:orders_mail_dept_h_1yr	-0.8190	0.3419	0.0862	0.0492
W:spend_d_h_1yr	-0.0007	0.9222	0.0000	0.0000
W:spend_online_attributed_target	0.0011	0.0311	0.0000	0.0000
W:dollars_season_C	0.0015	0.0118	0.0003	0.0004
W:spend_season_C	0.0009	0.1588	0.0000	0.0000
W:spend_e	-0.0008	0.7673	0.0000	0.0000
W:spend_z1	0.0005	0.5820	0.0000	0.0000
W:spend_period_1b	0.0261	0.0052	0.0000	0.0000
W:spend_period_2b	0.0001	0.9458	0.0025	0.0026
W:spend_h_3yr	-0.0049	0.3658	0.0000	0.0000
W:spend_g_3yr	0.0070	0.0022	0.0022	0.0022
W:last_years_since	-2.2109	0.0372	0.0000	0.0000
W:spend_instore_o	0.0003	0.8688	0.0000	0.0000
W:spend_instore_o_3yr	-0.0053	0.2446	0.0000	0.0000
W:spend_instore_h_3yr	-0.0051	0.6052	0.0000	0.0000
W:spend_instore_t	-0.0052	0.2802	0.0000	0.0000
W:spend_online_sh	0.0201	0.0302	0.0000	0.0000
W:spend_online_o_1yr	0.0061	0.3948	0.0000	0.0000
W:spend_online_n_3yr	-0.0088	0.1691	0.0000	0.0000
W:spend_online_o_3yr	-0.0151	0.0000	-0.0035	-0.0034
W:spend_online_a_1yr	-0.0258	0.0000	0.0000	0.0000

input	est_OLS	p_OLS	est_LASSO	est_elnet
W:spend_online_a_3yr	0.0020	0.3536	0.0000	0.0000
W:orders_online_t_3yr	0.2210	0.8176	0.1483	0.1954
W:spend_online_sg_1yr	-0.0646	0.0000	-0.0207	-0.0220
W:spend_online_g_1yr	-0.0204	0.0255	0.0000	0.0000
W:orders_online_s_1yr	-4.4249	0.0006	0.0000	0.0000
W:customer_type_7	-9.9106	0.0001	-0.2902	-0.6093
W:customer_type_A	-0.2034	0.8004	0.5411	0.5235
W:orders_attributed_mail_type_B	0.2233	0.4062	0.0821	0.1384
W:spend_attributed_mail_type_B	0.0027	0.3364	0.0059	0.0047
W:spend_attributed_mail_type_C	-0.0027	0.0596	0.0001	0.0001
W:mean_spend_attributed_mail_type_C	0.0095	0.0623	0.0040	0.0039
W:mean_spend_attributed_mail_type_A	0.0049	0.4776	0.0035	0.0036
W:clicks_product_type_104_2yr	0.2889	0.1511	0.0006	0.0174
W:clicks_product_type_312_3yr	-0.1552	0.0000	0.0000	0.0000
W:orders_e	-0.6884	0.0920	-0.2611	-0.2839
W:orders_mail_dept_c_1yr	-2.3847	0.0045	0.0000	0.0000
W:spend_d_s_1yr	0.0100	0.2851	0.0000	0.0000
W:spend_d_k_1yr	-0.0032	0.7272	0.0000	0.0000
W:spend_a_1yr	-0.0066	0.3886	-0.0060	-0.0064
W:spend_h_1yr	0.0225	0.0047	0.0065	0.0066
W:spend_direct_1yr	-0.0182	0.0068	0.0052	0.0052
W:acquisition_months_since	-0.0007	0.7776	0.0005	0.0006
W:spend_online_c	0.0136	0.0032	0.0042	0.0047
W:spend_online_a_6m	0.0088	0.0702	-0.0042	-0.0041
W:orders_online_o	-0.0681	0.3915	-0.0954	-0.0923
W:orders_online_c_1yr	4.0166	0.0014	0.0000	0.0000
W:spend_online_b_1yr	-0.0482	0.0000	-0.0124	-0.0120
W:customer_type_2	-2.3585	0.0069	-0.5792	-0.6055
W:customer_income	0.0000	0.9201	0.0000	0.0000
W:orders_attributed_mail_type_A	-0.2204	0.4868	0.0000	0.0000
W:spend_attributed_mail_type_A	0.0026	0.4448	0.0000	0.0000
W:clicks_product_type_312_3m	0.3724	0.0055	0.0000	0.0000
W:clicks_product_type_301_2yr	-0.0874	0.2023	0.0000	-0.0019
W:clickthrough_6m	0.2318	0.3216	0.0000	0.0000
W:emails_days_2yr	-0.0037	0.2031	0.0000	0.0000
W:emails_3m	-0.0626	0.0063	0.0000	0.0000
W:emailreceived_months_since	-0.0041	0.8007	0.0000	0.0000
W:orders_3yr	0.0085	0.9493	0.0000	0.0000
W:orders_mail_dept_d_1yr	1.2494	0.0368	0.0000	0.0000
W:spent_q	0.0039	0.5514	0.0000	0.0000
W:spend_instore_q	0.0297	0.2402	0.0000	0.0000
W:spend_online_s	0.0034	0.0980	0.0000	0.0000
W:clicks_product_type_112_3yr	0.0825	0.0014	0.0000	0.0005
W:spend_instore_n_3yr	0.0060	0.2600	0.0000	0.0000
W:spend_instore_p	0.0115	0.3122	0.0000	0.0000
W:spend_instore_p_1yr	0.0490	0.1462	0.0000	0.0000
W:orders_online_n_1yr	3.8470	0.0001	0.0000	0.0067
W:spend_instore_a_yr	0.0050	0.0998	0.0000	0.0000
W:orders_attributed_mail_type_C	0.3864	0.0134	0.0000	0.0000
W:clickthrough_3yr	-0.0493	0.2240	0.0000	0.0000
W:spend_instore_g_1yr	0.0140	0.7021	0.0000	0.0000
W:spend_period_3b	-0.0009	0.6496	0.0000	0.0000

input	est_OLS	p_OLS	est_LASSO	est_elnet
W:spend_instore_a	-0.0006	0.3308	0.0000	0.0000
W:spend_direct_g	-0.0003	0.8075	0.0000	0.0000
W:emailview_24m	0.0030	0.7719	0.0000	0.0000
W:spend_online_h_3yr	-0.0093	0.0623	0.0000	0.0000
W:emailview_months_since	-0.0211	0.2035	0.0000	0.0000
W:orders_instore_c	-0.0599	0.8908	-0.1655	-0.2032
W:emails_1yr	0.0168	0.0738	0.0000	0.0000
W:clicks_product_type_001_1m	0.6609	0.0826	0.0000	0.0000
W:clickthrough_3m	-0.5460	0.2204	0.0000	0.0000
W:orders_d_1yr	-0.4326	0.3311	0.0000	0.0000
W:orders_h	-0.1171	0.0106	0.0000	0.0000
W:clicks_product_type_104	-0.0467	0.3994	0.0000	0.0000
W:clicks_product_type_502_3m	-0.0411	0.6314	0.0000	0.0000
W:web_activity_1m	-0.0479	0.1299	0.0000	0.0000
W:orders_instore_m	0.4428	0.0351	0.0000	0.0000
W:spend_notz_1yr	-0.0035	0.7382	0.0000	0.0000
W:orders_online_sh	-0.9881	0.0412	0.0000	0.0000
W:orders_instore_n	0.2908	0.1322	0.0000	0.0000
W:clicks_product_type_502_1yr	-0.0106	0.5640	0.0000	0.0000
W:orders_hm	0.0351	0.6105	0.0539	0.0580
W:emailview	-0.0039	0.2007	-0.0002	-0.0006
W:spend_m_1yr	0.0190	0.2659	0.0037	0.0045
W:clicks_product_type_301_3m	-0.2193	0.2501	0.0000	0.0000
W:orders_instore_a_yr	-0.8051	0.0847	0.0000	0.0000
W:clicks_product_type_001_6m	0.1003	0.4692	-0.0051	-0.0110
W:clickthrough_months_since	0.0086	0.4683	0.0000	0.0000
W:orders_online_h_1yr	-3.1609	0.0001	0.0000	0.0000
W:orders_instore_h_3yr	0.3872	0.5421	0.0000	0.0000
W:spend_period_1a	0.0240	0.0419	0.0000	0.0000
W:orders_total	0.0115	0.6810	0.0000	0.0000
W:spend_online_s_3yr	-0.0034	0.5839	0.0000	0.0000
W:spend_M_3yr	0.0122	0.0118	0.0000	0.0000
W:orders_c	-0.1094	0.4457	0.0000	0.0000
W:spend_l	0.0011	0.7991	0.0017	0.0017
W:spend_instore_n	-0.0131	0.0131	0.0000	0.0000
W:orders_z	-0.7220	0.5000	0.0000	0.0000
W:web_activity_24m	0.0249	0.0000	0.0020	0.0021
W:spend_instore_q_3yr	-0.0137	0.6954	0.0000	0.0000
W:clicks_product_type_504	-0.0029	0.4300	0.0000	0.0000
W:emailview_6m	-0.0535	0.1482	0.0000	0.0000
W:buy_instore_days_since	0.0002	0.6432	0.0000	0.0000
W:spend_h	-0.0008	0.2395	0.0000	0.0000
W:clicks_product_type_801_3yr	0.0825	0.0131	0.0000	0.0000
W:spend_nott_1yr	0.0020	0.9129	0.0000	0.0000
W:orders_online_s_3yr	0.2615	0.6435	0.0000	0.0000
W:spend_instore_h_1yr	-0.0068	0.6513	0.0000	0.0000
W:spend_online_b_3yr	0.0062	0.0395	0.0000	0.0000
W:orders_instore_s_3yr	0.0967	0.7187	0.0000	-0.0054
W:orders_instore_r_3yr	-2.9190	0.0188	-0.7370	-0.7956
W:clicks_product_type_502_1m	0.0801	0.3572	0.0000	0.0000
W:mean_spend_attributed_mail_type_B	0.0065	0.2601	0.0000	0.0010
W:clicks_product_type_201_3yr	-0.0097	0.7228	0.0000	0.0000

input	est_OLS	p_OLS	est_LASSO	est_elnet
W:store_trips	0.0040	0.9462	0.0000	0.0000
W:spend_online_t_1yr	-0.0342	0.0637	0.0000	0.0000
W:orders_online_o_1yr	-1.7509	0.0422	0.0000	0.0000
W:spend_online_k	-0.0048	0.0012	0.0000	0.0000
W:spend_online_s_1yr	-0.0102	0.4317	0.0000	0.0000
W:clicks_product_type_503	-0.0031	0.1091	0.0000	0.0000
W:orders_instore_t_3yr	-1.1246	0.1474	0.0000	0.0000

Number of selected coefficient in different models :

```
sum(abs(results$est_OLS) > 0)
```

```
[1] 294
```

```
sum(abs(results$est_LASSO) > 0)
```

```
[1] 75
```

```
sum(abs(results$est_elnet) > 0)
```

```
[1] 86
```

- LASSO and elastic net only select a relatively small fraction of all features, 75 and 86 features. Hence, a significant amount of regularization is used.

Once you have estimated the models, save all the output objects in a file for later use.

```
# Save the fitted model to a file
# save(fit_ols, fit_LASSO, fit_elnet, results, file = "models.RData")
```

5.5 Predict treatment effects

To validate the models you need to predict the CATE. You will make your life much easier if you first create a new data.table,

```
predict_DT = crm_DT[training_sample == 0,
                    c("customer_id", "outcome_spend", "W"), with = FALSE]
```

Add the model predictions to this table.

Hint: CATE prediction

The predicted CATE is based on the difference between predicted spending conditional on $W_i = 1$ and $W_i = 0$. For example, using the OLS estimates, you would first predict:

```
# Y_1 = predict(fit_OLS, newdata = validation_DT[, W := 1])
# Y_0 = predict(fit_OLS, newdata = validation_DT[, W := 0])
```

Then take the difference between `Y_1` and `Y_0` to predict the CATE, τ_i.

You will use a similar approach for the LASSO, where you will have to appropriately create the matrix of inputs (independent variables) depending on the value of `W`.

Warning: Remember that `validation_DT[, W := 1]` *permanently* changes all values in the `W` column of the table `validation_DT`. This is OK unless you plan to later use the changed `validation_DT` table for other purposes that require the actual values of `W`.

Hence, to be completely on the safe side, you can predict Y conditional on W using a copy of the table:

```
temp_DT = copy(validation_DT)

# Predict using OLS
Y_1_OLS = predict(fit_ols, newdata = temp_DT[, W := 1])
Y_0_OLS = predict(fit_ols, newdata = temp_DT[, W := 0])
predict_DT[, CATE_OLS := Y_1_OLS - Y_0_OLS]

# Predict using LASSO

# -> Assume target everyone
temp_DT[, W := 1]  # Set W = 1 for all data rows
Y_1_LASSO <- predict(
  fit_LASSO,
  newx = model.matrix(outcome_spend ~ W * ., data = temp_DT)[,-1], s = "lambda.min")

# -> Assume target no one
temp_DT[, W := 0]  # Set W = 0 for all data rows
Y_0_LASSO <- predict(
  fit_LASSO,
  newx = model.matrix(outcome_spend ~ W * ., data = temp_DT)[,-1], s = "lambda.min")
predict_DT[, CATE_LASSO := Y_1_LASSO - Y_0_LASSO]

# Predict using Elastic Net
temp_DT[, W := 1]
Y_1_EN <- predict(
  fit_elnet,
  newx = model.matrix(outcome_spend ~ W * ., data = temp_DT)[,-1], s = "lambda.min")

temp_DT[, W := 0]  # Set W = 0 for prediction
Y_0_EN <- predict(
  fit_elnet,
  newx = model.matrix(outcome_spend ~ W * ., data = temp_DT)[,-1], s = "lambda.min")
predict_DT[, CATE_EN := Y_1_EN - Y_0_EN]
```

```
# Save CATE predictions
fwrite(predict_DT, "CATE_predict_DT.csv")
```

Once all model predictions are added, save the table to a file for use in step 2.

Final Assignment: Incrementality and Optimal Personalized Targeting

Nichada Wongrassamee

Contents

```r
library(bit64)
library(data.table)
library(glmnet)
library(ggplot2)
library(knitr)
library(corrplot)
library(dplyr)
```

1 Step 2: Model fit in 2017 validation sample

Reload models and predict_DT here : - If you want to start from step 2, you can download the saved model in here.

```r
# Load CATE predictions
predict_DT <- fread("CATE_predict_DT.csv")
load("models.RData")
load("large_cor_DT.RData")
```

1.1 Descriptive analysis of predicted treatment effects

Document the ATE (average treatment effect in the data). Summarize and graph the distribution of the predicted incremental effects, τ_i, from the different estimation methods. How much variation is there in the CATEs compared to the ATE?

```r
ATE_OLS <- mean(predict_DT$CATE_OLS)
ATE_LASSO <- mean(predict_DT$CATE_LASSO)
ATE_EN <- mean(predict_DT$CATE_EN)

# Print results
cat("ATE (OLS):", ATE_OLS, "\n")
```

```
ATE (OLS): 2.735326
```

```r
cat("ATE (LASSO):", ATE_LASSO, "\n")
```

```
ATE (LASSO): 2.249949
```

```r
cat("ATE (Elastic Net):", ATE_EN, "\n")
```

```
ATE (Elastic Net): 2.278373
```

```r
# MSE in each model
mse_OLS = mean((predict_DT$CATE_OLS - ATE_OLS)^2)
mse_LASSO = mean((predict_DT$CATE_LASSO - ATE_LASSO)^2)
mse_elnet = mean((predict_DT$CATE_EN - ATE_EN)^2)
cat(mse_OLS, mse_LASSO, mse_elnet, "\n")
```

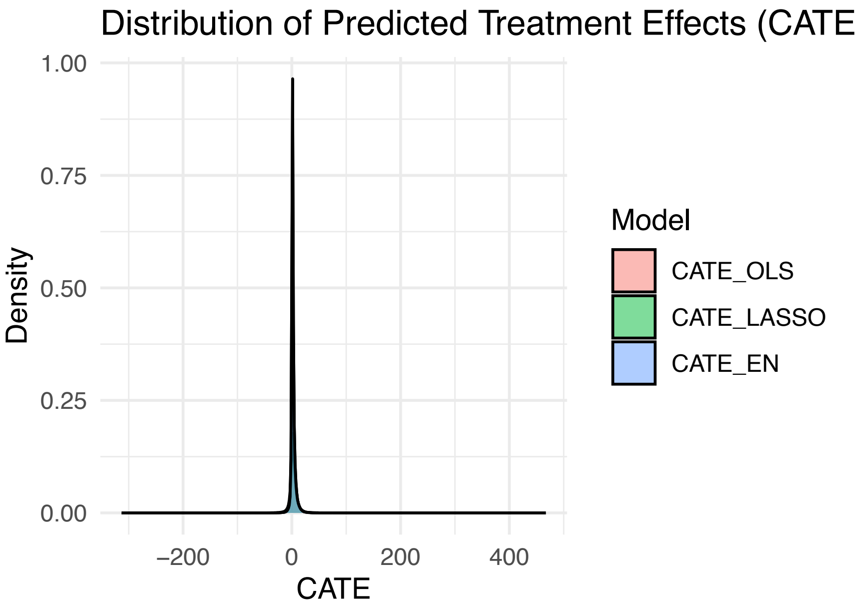
```
69.57642 18.76407 18.22866
```

```r
# Combine data for plotting
CATE_melted <- melt(predict_DT[, .(CATE_OLS, CATE_LASSO, CATE_EN)],
                    variable.name = "Model", value.name = "CATE")
```

```
Warning in melt.data.table(predict_DT[, .(CATE_OLS, CATE_LASSO, CATE_EN)], :
id.vars and measure.vars are internally guessed when both are 'NULL'. All
non-numeric/integer/logical type columns are considered id.vars, which in this
case are columns []. Consider providing at least one of 'id' or 'measure' vars
```

```
in future.
```

```r
# Plot the distribution of CATEs
ggplot(CATE_melted, aes(x = CATE, fill = Model)) +
  geom_density(alpha = 0.5) +
  labs(title = "Distribution of Predicted Treatment Effects (CATE)",
       x = "CATE", y = "Density") +
  theme_minimal()
```



```r
# TODO : variance ?

# Variance of CATE
var_CATE_OLS <- var(predict_DT$CATE_OLS)
var_CATE_LASSO <- var(predict_DT$CATE_LASSO)
var_CATE_EN <- var(predict_DT$CATE_EN)

# Print variance
cat("Variance (OLS):", var_CATE_OLS, "\n")
cat("Variance (LASSO):", var_CATE_LASSO, "\n")
cat("Variance (Elastic Net):", var_CATE_EN, "\n")
```

Also compare and comment on the scale difference between the incremental effects and the level of sales.

```r
# Compare scale
# TODO overall mean cate is the same as ate ?
scale_comparison <- predict_DT[, .(
  MEAN_CATE_OLS = mean(CATE_OLS),
  MIN_CATE_OLS = min(CATE_OLS),
  MAX_CATE_OLS = max(CATE_OLS),
  MEAN_CATE_LASSO = mean(CATE_LASSO),
  MIN_CATE_LASSO = min(CATE_LASSO),
  MAX_CATE_LASSO = max(CATE_LASSO),
  MEAN_CATE_EN = mean(CATE_EN),
  MIN_CATE_EN = min(CATE_EN),
  MAX_CATE_EN = max(CATE_EN),
  Mean_Sales = mean(outcome_spend)
)]

print(scale_comparison)
```

```
   MEAN_CATE_OLS MIN_CATE_OLS MAX_CATE_OLS MEAN_CATE_LASSO MIN_CATE_LASSO
           <num>        <num>        <num>           <num>          <num>
1:      2.735326    -313.1438     467.1557        2.249949      -101.5065
   MAX_CATE_LASSO MEAN_CATE_EN MIN_CATE_EN MAX_CATE_EN Mean_Sales
            <num>        <num>       <num>       <num>      <num>
1:       126.5286     2.278373   -104.1481    118.3283   7.855817
```

The incremental effects is referring to the change in sales (+-) on consumer as a result of targeting. However the mean sales is the actual ave value of sale, regardless of targeting. The scale of the incremental effects (CATE/ATE) is much smaller than the overall sales value (Mean_Sales = 7.86), indicating that the treatment has a relatively modest effect on individual sales. The range of sales outcome is also wider ranging from 0 - 1462.84. While CATE_OLS ranges from -313.14383 to 467.15565, -101 to 126 for CATE lasso and -104 to 118 for CATE elasticnet.

Having negative CATE_i means that the targeting has a negative impact on that particular customer_i. Having negative ATE means that the affects of targeting to everyone has ,on average, a reduce sales.

Having positve CATE_i means that the targeting has a positive impact on that particular customer_i. Having positive ATE means that the affects of targeting to everyone has ,on average, a increase sales.

Therefore, we should target only certain groups for optimal targeting result.

1.2 Model validation: Lifts

Evaluate the model fit using lift charts and a lift table. The focus is on evaluating how well the models predict the CATE, τ_i. Correspondingly, in a lift chart we create scores based on the estimated (predicted) CATE, and we calculate the average treatment effect in each score (group) based on the observed outcomes in the validation sample.

I recommend using 20 scores.

1.2.1 Create lift table

- I group the score_group based on the predicted CATE score per each model.
- Calculate ATE in each subgroup, using the outcome_spending in the validation data.
- Plot ATE of each subgroup.

```
# try 2 :
# score_group create
N_groups = 20

predict_DT[, score_group_OLS := as.integer(cut_number(predict_DT$CATE_OLS, n = N_groups))]
predict_DT[, score_group_LASSO := as.integer(cut_number(predict_DT$CATE_LASSO, n = N_groups))]
predict_DT[, score_group_EN := as.integer(cut_number(predict_DT$CATE_EN, n = N_groups))]
```

```
# OLS
lift_table_OLS <- function(data, N_groups = 20) {
  # Ensure the input data is a data.table
  DT <- copy(data)

  # Calculate the lift table based on the CATE_OLS column
  lift_DT <- DT[, .(
    model = "OLS",
    CATE = mean(CATE_OLS, na.rm = TRUE),
    y = mean(outcome_spend, na.rm = TRUE),
    N = .N,
    # Calculate standard error for ATE
    std_error = sqrt(var(outcome_spend[W == 1], na.rm = TRUE) / sum(W == 1) +
                     var(outcome_spend[W == 0], na.rm = TRUE) / sum(W == 0)),
    treatment_mean = mean(outcome_spend[W == 1], na.rm = TRUE),
    nontreatment_mean = mean(outcome_spend[W == 0], na.rm = TRUE),
    ATE = mean(outcome_spend[W == 1], na.rm = TRUE) -
          mean(outcome_spend[W == 0], na.rm = TRUE)
  ), keyby = score_group_OLS] # Use score_group_OLS for grouping

  # Add confidence intervals for ATE
  lift_DT[, `:=`(
    lower = ATE + qt(0.025, df = N - 1) * std_error,
    upper = ATE + qt(0.975, df = N - 1) * std_error
  )]

  # Remove unnecessary columns and calculate lift
  lift_DT[, c("std_error", "N") := NULL]
  lift_DT[, lift := 100 * ATE / mean(ATE, na.rm = TRUE)] # Lift based on ATE

  setnames(lift_DT, "score_group_OLS", "score_group")

  return(lift_DT)
}

lift_OLS = lift_table_OLS(predict_DT)
```

Lasso lift

```
# LASSO
lift_table_LASSO <- function(data, N_groups = 20) {
  # Ensure the input data is a data.table
  DT <- copy(data)

  # Calculate the lift table based on the CATE_LASSO column
  lift_DT <- DT[, .(
    model = "LASSO",
    CATE = mean(CATE_LASSO, na.rm = TRUE),
    y = mean(outcome_spend, na.rm = TRUE),
    N = .N,
    # Calculate standard error for ATE
    std_error = sqrt(var(outcome_spend[W == 1], na.rm = TRUE) / sum(W == 1) +
                     var(outcome_spend[W == 0], na.rm = TRUE) / sum(W == 0)),
    treatment_mean = mean(outcome_spend[W == 1], na.rm = TRUE),
    nontreatment_mean = mean(outcome_spend[W == 0], na.rm = TRUE),
    ATE = mean(outcome_spend[W == 1], na.rm = TRUE) -
          mean(outcome_spend[W == 0], na.rm = TRUE)
  ), keyby = score_group_LASSO] # Use score_group_OLS for grouping

  # Add confidence intervals for ATE
  lift_DT[, `:=`(
    lower = ATE + qt(0.025, df = N - 1) * std_error,
```

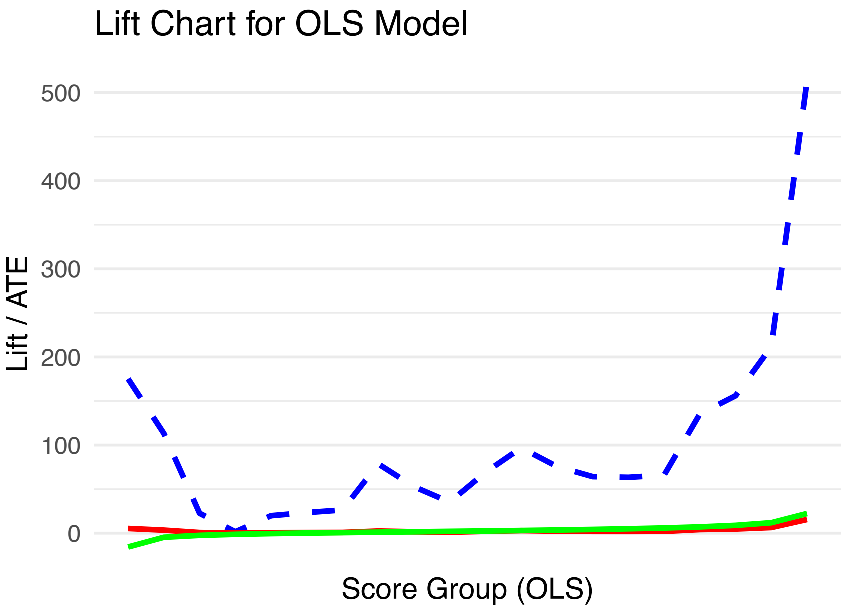
```r
    upper = ATE + qt(0.975, df = N - 1) * std_error
  )]

  # Remove unnecessary columns and calculate lift
  lift_DT[, c("std_error", "N") := NULL]
  lift_DT[, lift := 100 * ATE / mean(ATE, na.rm = TRUE)] # Lift based on ATE

  setnames(lift_DT, "score_group_LASSO", "score_group")

  return(lift_DT)
}

lift_LASSO = lift_table_LASSO(predict_DT)
```

Elastic Net Lift

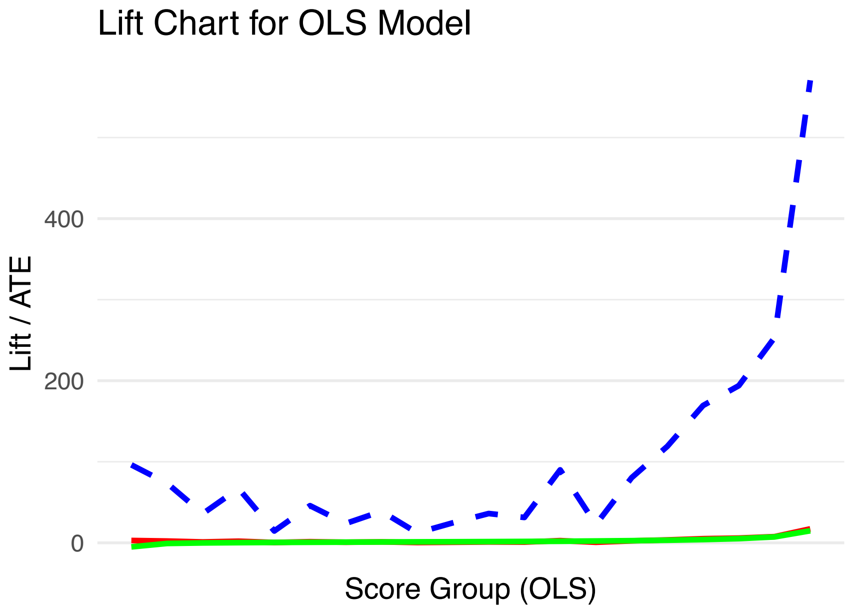
```r
# Elastic Net
lift_table_EN <- function(data, N_groups = 20) {
  # Ensure the input data is a data.table
  DT <- copy(data)

  # Calculate the lift table based on the CATE_EN column
  lift_DT <- DT[, .(
    model = "Elastic",
    CATE = mean(CATE_EN, na.rm = TRUE),
    y = mean(outcome_spend, na.rm = TRUE),
    N = .N,
    # Calculate standard error for ATE
    std_error = sqrt(var(outcome_spend[W == 1], na.rm = TRUE) / sum(W == 1) +
                     var(outcome_spend[W == 0], na.rm = TRUE) / sum(W == 0)),
    treatment_mean = mean(outcome_spend[W == 1], na.rm = TRUE),
    nontreatment_mean = mean(outcome_spend[W == 0], na.rm = TRUE),
    ATE = mean(outcome_spend[W == 1], na.rm = TRUE) -
          mean(outcome_spend[W == 0], na.rm = TRUE)
  ), keyby = score_group_EN] # Use score_group_OLS for grouping

  # Add confidence intervals for ATE
  lift_DT[, `:=`(
    lower = ATE + qt(0.025, df = N - 1) * std_error,
    upper = ATE + qt(0.975, df = N - 1) * std_error
  )]

  # Remove unnecessary columns and calculate lift
  lift_DT[, c("std_error", "N") := NULL]
  lift_DT[, lift := 100 * ATE / mean(ATE, na.rm = TRUE)] # Lift based on ATE

  setnames(lift_DT, "score_group_EN", "score_group")

  return(lift_DT)
}

lift_EN = lift_table_EN(predict_DT)
```

```r
# Function to plot lift chart for OLS model
plot_lift_chart_OLS <- function(lift_table) {
  ggplot(lift_table, aes(x = score_group)) +
    geom_line(aes(y = lift), color = "blue", linetype = "dashed", size = 1) +
    geom_line(aes(y = ATE), color = "red", size = 1) +
    geom_line(aes(y = CATE), color = "green", size = 1) +
    labs(
      title = "Lift Chart for OLS Model",
      x = "Score Group (OLS)",
      y = "Lift / ATE"
    ) +
    scale_x_continuous(breaks = unique(lift_table$score_group_OLS)) +
    theme_minimal()
}

plot_lift_chart_OLS(lift_OLS)
```

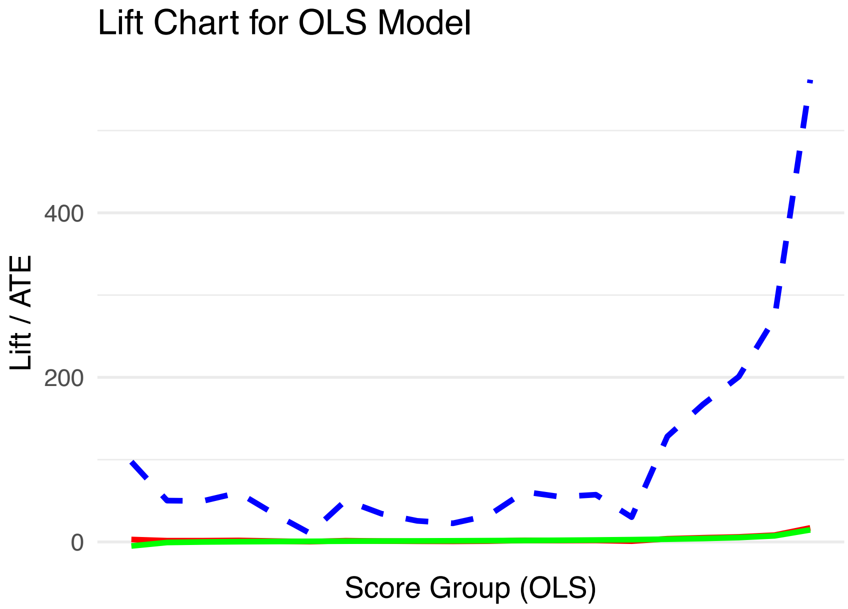
```
Warning: Using `size` aesthetic for lines was deprecated in ggplot2 3.4.0.
i Please use `linewidth` instead.
This warning is displayed once every 8 hours.
Call `lifecycle::last_lifecycle_warnings()` to see where this warning was
generated.
```



```r
plot_lift_chart_OLS(lift_LASSO)
```



```
plot_lift_chart_OLS(lift_EN)
```



Summary Lift

```
combined_ATE <- data.table()

# Select relevant columns and rename ATE to indicate the model
lift_OLS_clean <- lift_OLS[, .(score_group, ATE_OLS = ATE)]
lift_LASSO_clean <- lift_LASSO[, .(score_group, ATE_LASSO = ATE)]
lift_EN_clean <- lift_EN[, .(score_group, ATE_EN = ATE)]

# Combine the ATE columns from all tables into one table by score_group
combined_ATE <- merge(lift_OLS_clean, lift_LASSO_clean, by = "score_group", all = TRUE)
combined_ATE <- merge(combined_ATE, lift_EN_clean, by = "score_group", all = TRUE)

# Display the combined table with kable
```

```r
kable(combined_ATE, digits = 2, col.names = c("Score Group", "ATE (OLS)", "ATE (LASSO)", "ATE (EN)"))
```

Score Group	ATE (OLS)	ATE (LASSO)	ATE (EN)
1	5.29	2.91	2.96
2	3.42	2.23	1.53
3	0.68	1.09	1.51
4	0.05	2.03	1.82
5	0.60	0.44	1.03
6	0.71	1.39	0.32
7	0.80	0.72	1.53
8	2.37	1.18	1.04
9	1.59	0.37	0.78
10	1.06	0.75	0.69
11	2.08	1.09	0.98
12	2.93	0.95	1.85
13	2.29	2.73	1.67
14	1.94	0.68	1.74
15	1.91	2.43	0.91
16	1.99	3.60	3.90
17	4.08	5.13	5.08
18	4.71	5.87	6.10
19	6.39	7.66	8.26
20	15.51	17.27	17.07

```r
combined_CATE <- data.table()

# Select relevant columns and rename ATE to indicate the model
lift_OLS_clean <- lift_OLS[, .(score_group, CATE)]
lift_LASSO_clean <- lift_LASSO[, .(score_group, CATE)]
lift_EN_clean <- lift_EN[, .(score_group, CATE)]

# Combine the ATE columns from all tables into one table by score_group
combined_CATE <- merge(lift_OLS_clean, lift_LASSO_clean, by = "score_group", all = TRUE)
combined_CATE <- merge(combined_CATE, lift_EN_clean, by = "score_group", all = TRUE)

# Display the combined table with kable
kable(combined_CATE, digits = 2, col.names = c("Score Group", "CATE (OLS)", "CATE (LASSO)", "CATE (EN)"
```

Score Group	CATE (OLS)	CATE (LASSO)	CATE (EN)
1	-15.72	-5.00	-4.93
2	-4.68	-0.74	-0.69
3	-2.52	-0.09	-0.05
4	-1.33	0.25	0.28
5	-0.52	0.49	0.52
6	0.09	0.70	0.72
7	0.61	0.88	0.90
8	1.09	1.06	1.06
9	1.54	1.22	1.21
10	2.01	1.37	1.37
11	2.48	1.52	1.54
12	3.00	1.72	1.75

Score Group	CATE (OLS)	CATE (LASSO)	CATE (EN)
13	3.56	1.96	2.00
14	4.20	2.27	2.33
15	4.96	2.68	2.76
16	5.88	3.25	3.34
17	7.10	4.04	4.14
18	8.85	5.22	5.31
19	11.82	7.32	7.36
20	22.29	14.88	14.65

2 Analysis :

Since I group low CATE in group 1 and higher CATE in higher group numbers. The expected lift graph should reflect high treatment effect ATE in group 20.

The current plot is showing unsmoothed result due to the model may be overfitting, having too many unnecessary features that confused the treatment effect.

Future improvement, reduce overfitting my reduce the feature further. Only include necessary features to predict treatment effect.

3 Step 3: How well do the models predict in a different year?

So far we have assessed the goodness of the model using data from the 2017 catalog mailing campaign. Since we also have data from the following year, we can leverage that to perform an *even more* out-of-sample validation exercise. If the model estimated on 2017 data predicts incremental spending well in 2018, that means the model has the ability to really help when applied to future decisions.

Optionally, you can download `model.RData` here if you want to start from step 3.

```r
load(paste0(data_folder, "/Randomized-Implementation-Sample-2018.RData"))
setnames(crm_DT, "mailing_indicator", "W")
```

Warning: The 2018 data table is also named `crm_DT`, just as the 2017 data.

The approach:

1. Use the model predictions based **only on the 2017 data**.
2. Predict the CATE for the customers in the October 2018 data.
3. Evaluate the model predictions via lift charts using the 2018 data.

The structure of the 2018 data is identical to the structure of the 2017 data. In particular, the data contains a randomly selected sample of all customers in the data base, and the treatment assignment W_i is randomized.

Note: Ideally you would perform step 1 by re-estimating all models using *all* of the 2017 data (no training/validation sample split). However, to preserve time, it is perfectly fine if you keep the estimates from the training sample in step 1 to predict the 2018 treatment effects.

```r
# crm_DT is now 2018 data
temp_DT = copy(crm_DT)
# Remove the large_cor_DT$row from crm_DT
temp_DT = temp_DT[, !c(large_cor_DT$row, "customer_id"), with = FALSE]
```

```r
predict_DT_2018 = crm_DT[, .(customer_id, outcome_spend, W)]
```

```
# Predict using OLS
temp_DT[, W := 1]  # Set W = 1 for all data rows
Y_1_OLS = predict(fit_ols, newdata = temp_DT)

temp_DT[, W := 0]  # Set W = 1 for all data rows
Y_0_OLS = predict(fit_ols, newdata = temp_DT)

predict_DT_2018[, CATE_OLS := Y_1_OLS - Y_0_OLS]

# Predict using LASSO

# -> Assume target everyone
temp_DT[, W := 1]  # Set W = 1 for all data rows
Y_1_LASSO <- predict(
  fit_LASSO,
  newx = model.matrix(outcome_spend ~ W * .,
                      data = temp_DT)[,-1], s = "lambda.min")

# -> Assume target no one
temp_DT[, W := 0]  # Set W = 0 for all data rows
Y_0_LASSO <- predict(
  fit_LASSO,
  newx = model.matrix(outcome_spend ~ W * .,
                      data = temp_DT)[,-1], s = "lambda.min")
predict_DT_2018[, CATE_LASSO := Y_1_LASSO - Y_0_LASSO]

# Predict using Elastic Net
temp_DT[, W := 1]
Y_1_EN <- predict(
  fit_elnet,
  newx = model.matrix(outcome_spend ~ W * .,
                      data = temp_DT)[,-1], s = "lambda.min")

temp_DT[, W := 0]  # Set W = 0 for prediction
Y_0_EN <- predict(
  fit_elnet,
  newx = model.matrix(outcome_spend ~ W * .,
                      data = temp_DT)[,-1], s = "lambda.min")
predict_DT_2018[, CATE_EN := Y_1_EN - Y_0_EN]
```

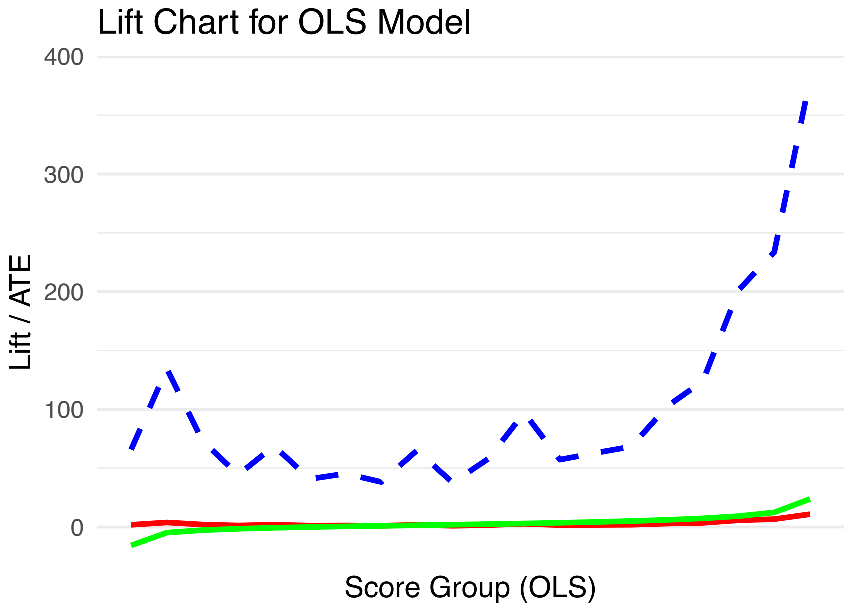
3.0.1 Lift 2018

```
# try 2 2018 :
# score_group create
N_groups = 20

predict_DT_2018[, score_group_OLS := as.integer(cut_number(predict_DT_2018$CATE_OLS, n = N_groups))]
predict_DT_2018[, score_group_LASSO := as.integer(cut_number(predict_DT_2018$CATE_LASSO, n = N_groups))]
predict_DT_2018[, score_group_EN := as.integer(cut_number(predict_DT_2018$CATE_EN, n = N_groups))]
```

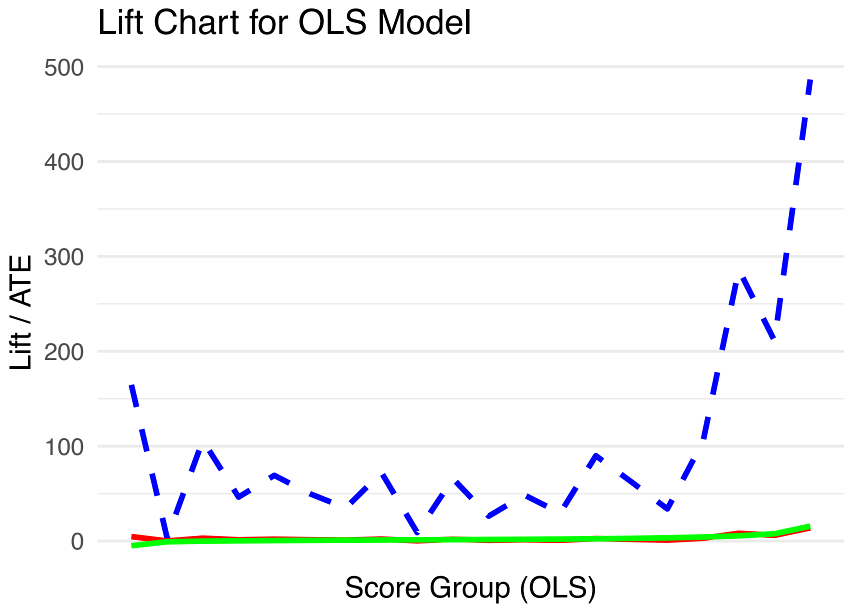
```
lift_OLS_2018 = lift_table_OLS(predict_DT_2018)
lift_LASSO_2018 = lift_table_LASSO(predict_DT_2018)
lift_EN_2018 = lift_table_EN(predict_DT_2018)
```

Plot loft

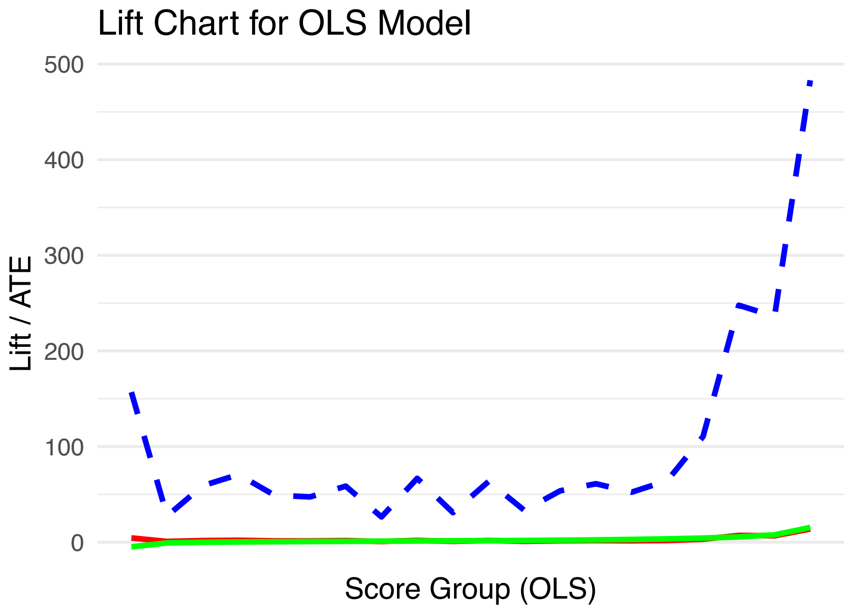
```
plot_lift_chart_OLS(lift_OLS_2018)
```



```
plot_lift_chart_OLS(lift_LASSO_2018)
```



```
plot_lift_chart_OLS(lift_EN_2018)
```



Summary Lift

```
# expected result
combined_ATE <- data.table()

# Select relevant columns and rename ATE to indicate the model
lift_OLS_clean <- lift_OLS_2018[, .(score_group, ATE_OLS = ATE)]
lift_LASSO_clean <- lift_LASSO_2018[, .(score_group, ATE_LASSO = ATE)]
lift_EN_clean <- lift_EN_2018[, .(score_group, ATE_EN = ATE)]

# Combine the ATE columns from all tables into one table by score_group
combined_ATE <- merge(lift_OLS_clean, lift_LASSO_clean, by = "score_group", all = TRUE)
combined_ATE <- merge(combined_ATE, lift_EN_clean, by = "score_group", all = TRUE)

# Display the combined table with kable
kable(combined_ATE, digits = 2, col.names = c("Score Group", "ATE (OLS)", "ATE (LASSO)", "ATE (EN)"))
```

Score Group	ATE (OLS)	ATE (LASSO)	ATE (EN)
1	1.88	4.72	4.50
2	3.84	0.09	0.79
3	2.09	3.04	1.70
4	1.27	1.33	2.03
5	1.97	1.98	1.41
6	1.17	1.43	1.36
7	1.30	1.00	1.68
8	1.09	2.07	0.76
9	1.86	0.26	1.91
10	1.08	1.89	0.89
11	1.66	0.75	1.82
12	2.77	1.39	0.95
13	1.64	0.85	1.54
14	1.80	2.58	1.75
15	1.95	1.78	1.50
16	2.93	0.97	1.85

Score Group	ATE (OLS)	ATE (LASSO)	ATE (EN)
17	3.54	3.01	3.17
18	5.75	8.20	7.10
19	6.68	6.05	6.77
20	10.87	13.94	13.85

```r
# predicted result

combined_CATE <- data.table()

# Select relevant columns and rename ATE to indicate the model
lift_OLS_clean <- lift_OLS_2018[, .(score_group, CATE)]
lift_LASSO_clean <- lift_LASSO_2018[, .(score_group, CATE)]
lift_EN_clean <- lift_EN_2018[, .(score_group, CATE)]

# Combine the ATE columns from all tables into one table by score_group
combined_CATE <- merge(lift_OLS_clean, lift_LASSO_clean, by = "score_group", all = TRUE)
combined_CATE <- merge(combined_CATE, lift_EN_clean, by = "score_group", all = TRUE)

# Display the combined table with kable
kable(combined_CATE, digits = 2, col.names = c("Score Group", "CATE (OLS)", "CATE (LASSO)", "CATE (EN)"
```

Score Group	CATE (OLS)	CATE (LASSO)	CATE (EN)
1	-15.67	-4.93	-4.88
2	-4.79	-0.75	-0.71
3	-2.65	-0.09	-0.06
4	-1.48	0.27	0.29
5	-0.65	0.52	0.54
6	0.00	0.74	0.76
7	0.53	0.94	0.95
8	1.01	1.12	1.13
9	1.47	1.31	1.29
10	1.95	1.47	1.45
11	2.46	1.61	1.62
12	3.00	1.81	1.83
13	3.59	2.06	2.09
14	4.27	2.37	2.43
15	5.07	2.82	2.89
16	6.07	3.41	3.50
17	7.38	4.24	4.33
18	9.25	5.49	5.57
19	12.40	7.67	7.70
20	23.81	15.70	15.41

3.0.2 Evaluate the model predictions via lift charts using the 2018 data.

The model shows some lift but the graph is unsmooth. This could be due to some overfitting in the original model, since the distribution of the CATE is concentrated in the middle with wide variance. However, the model seems to show lift despite having small impact on actual ATE (small rise magnitude).

4 Step 4: Develop a Targeting Policy

Based on the results in Steps 2 and 3, choose the best-performing model and use it to decide who should be targeted in the 2018 sample. For these calculations, assume that the cost of targeting a customer is $0.99, that the profit margin is 32.5 percent and that you have a large (unlimited) marketing budget for this campaign.

Target a customer if the incremental value from targeting exceeds the targeting cost (in expectation)

$$m \cdot \tau > c$$

Target if :

$$\tau > \frac{c}{m}$$

Calculate threshold for targeting

```
margin = 0.325 # 32.5 percent
cost = 0.99 # 99 cents
threshold = cost / margin
```

I found that the LASSO and the elastic net fit about equally well. I will pick LASSO as the preferred model to perform cost analysis.

```
# Assuming `predict_DT` contains customer-level CATE predictions
predict_DT_2018[, target := ifelse(CATE_LASSO > threshold, 1, 0)]
```

Percentage of customer that we should target :

```
# Summarize targeting results
summary_results <- predict_DT_2018[target == 1, .(
  total_customers_targeted = .N,
  avg_CATE = mean(CATE_LASSO, na.rm = TRUE),
  percent_targetted = .N / nrow(predict_DT_2018),
  total_cost = .N * cost,
  total_profit = sum(CATE_LASSO * margin, na.rm = TRUE)
)]

# Calculate net profit
summary_results[, net_profit := total_profit - total_cost]

# View summary
print(summary_results)
```

```
   total_customers_targeted avg_CATE percent_targetted total_cost total_profit
                      <int>    <num>             <num>      <num>        <num>
1:                    31706 7.240607          0.253648   31388.94     74610.47
   net_profit
        <num>
1:   43221.53
```

In the targeting policy, I used Lasso model to preduct CATE in 2018. Then, I filtered the customers that we should target based on CATE_i exceeding the threshold (cost/ margin). With this, there are 31706 total customers targeted that will add to the profit if we target them.

This accounts to 25% of total customers in 2018. We will be spedning 31,388 dollars on cost and gain 74610.47 total profit.

5 WriteUp

Summary of Findings: Model Performance and Lift Chart Analysis

I conducted a two-in-one regression analysis to compare the effects of targeting versus non-targeting at the individual customer level (CATE). After fitting the models, the OLS, LASSO, and Elastic Net retained 294, 75, and 86 features, respectively, after regularization.

These three models were then used to predict CATE on the validation dataset. To evaluate their performance, I validated the predictions using a lift chart. The lift chart illustrates the targeting effect across score groups. The methodology involves predicting CATE with each model, segmenting customers into score groups based on their CATE values, and calculating the ATE for each group. ATE is computed as the mean outcome spending for targeted customers (W=1) minus the mean outcome spending for non-targeted customers (W=0) within each segment. The plotted ATE is expected to show increasing targeting effects as the score group values rise.

The models evaluated (OLS, LASSO, Elastic Net) perform some lift, however with some inconsistency in lift curve and the ATE in treatment effects across score groups according to the lift graphs produced on the validation data set. Variations in lift patterns suggest issues such as potential overfitting or model instability. The lack of a clear lift trend diminishes the effectiveness of the targeting strategy.

In this analysis, I used CATE (Conditional Average Treatment Effect) to provide customer-level insights into the incremental impact of targeting. This granularity enables refined targeting policies compared to using ATE (Average Treatment Effect) alone. However, the variance in CATE predictions across models highlights the need for improved feature selection and model tuning to enhance prediction stability and accuracy.

To create a targeting Policy, I picked the most optimal model to predict CATE in 2018. Since the three models that I produced did not demonstrate lift as the score group increase, I chose LASSO model for the purpose of the analysis. In cost-benefit analysis, I identified a threshold to target customers where incremental profit exceeds the targeting cost. As a result, I can target 25.4% of the customer base (31,706 customers) yielded an estimated net profit of $43,221, highlighting the potential financial benefit of personalized targeting despite model limitations.

During the analysis, I encounter challenges including model overfitting (due to high concentration of CATE and wide variance), variability of results and limited lift outcome. Model Overfitting, The models may include extraneous features, leading to overfitting and reduced generalizability to new data. Variability in Results: Differences in treatment effect predictions across models make it challenging to select the most reliable model. The limited lift outcome limits the strategic advantage of personalized targeting.

There are some potential improvement that I want to include :

1. Model Refinement: Simplify feature sets to focus on the most impactful predictors of treatment effect, reducing noise and overfitting. Eventhough lasso and elasticnet already have regularization technique to reduce the model features. However, there is still need for further feature selection. Consider ensemble methods to improve prediction robustness.
2. Incorporate Business Context: Collaborate with domain experts to incorporate contextual insights, improving model interpretability and alignment with business goals.

While the models did not yield the desired lift, the use of CATE remains a valuable approach for identifying profitable customer segments. By addressing overfitting and variability challenges, the targeting policy can be significantly enhanced to maximize financial outcomes and campaign efficiency.