Targeting Based on Heterogeneous Treatment Effects

Kanyao Han

February 23, 2018

Step 1: Estimation and prediction of conditional average treatment effects

```
library(bit64)
library(data.table)
```

```
## Warning: package 'data.table' was built under R version 3.4.3
```

```
library(glmnet)
```

```
## Warning: package 'glmnet' was built under R version 3.4.3
```

```
library(causalTree)
```

```
## Warning: package 'rpart.plot' was built under R version 3.4.3
```

```
library(tidyverse)
```

```
## Warning: package 'stringr' was built under R version 3.4.3
```

```
library(knitr)
library(broom)
library(corrplot)
```

```
## Warning: package 'corrplot' was built under R version 3.4.3
```

```
library(psych)
library(ranger)
```

```
## Warning: package 'ranger' was built under R version 3.4.3
```

```
load("Customer-Development-2015.RData")
```

Data pre-processing

```
set.seed(2001)
crm_DT[, training_sample := rbinom(nrow(crm_DT), 1, 0.5)]
setnames(crm_DT, "mailing_indicator", "W")

#Craete correlation matrix
cor_matrix = cor(crm_DT[, !c("customer_id", "W",
                             "outcome_spend"),with = FALSE])
# Create pdf format visulation matrix
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
cor_matrix[upper.tri(cor_matrix, diag = TRUE)] = NA
```

```
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]

# Eliminate highly correlated variables
large_cor_DT = cor_DT[abs(cor) > 0.95]
crm_DT_rem = crm_DT[, !large_cor_DT$row, with = FALSE]

# Show the table of elimiated varibales
large_cor_DT
```

```
##                                row                            col
## 1:                 customer_type_3                 online_customer
## 2: orders_online_attributed_target spend_online_attributed_target
## 3:          acquisition_days_since       acquisition_months_since
## 4:              in_database_months       acquisition_months_since
## 5:              in_database_months         acquisition_days_since
## 6:                 emails_days_1yr                emails_days_2yr
## 7:                    emailview_3m                   emailview_6m
##           cor
## 1: -0.9581664
## 2:  0.9653779
## 3:  1.0000000
## 4:  0.9996683
## 5:  0.9996683
## 6:  0.9604113
## 7:  0.9504640
```

Randomization checks

```
cat("The probability of a mailing is", nrow(crm_DT[W == 1]) / nrow(crm_DT))
```

```
## The probability of a mailing is 0.668864
```

```
assess <- cor(crm_DT)

assess %>%
  as.data.frame() %>%
  mutate(variables = row.names(assess),
         abs_coff = abs(W)) %>%
  select(abs_coff, variables) %>%
  filter(variables != 'W') %>%
  arrange(desc(abs_coff)) %>%
  head(10)
```

```
##       abs_coff                    variables
## 1  0.030687792                outcome_spend
## 2  0.006588372            spend_online_b_3yr
## 3  0.005455547            spend_online_a_1yr
## 4  0.005425714            spend_online_b_1yr
## 5  0.005352244  spend_attributed_mail_type_C
## 6  0.005329942 orders_attributed_mail_type_C
## 7  0.005203031                   orders_3yr
## 8  0.005191545              spend_period_1b
## 9  0.005047109            spend_online_a_3yr
## 10 0.004914750           spend_instore_p_1yr
```

The table shows the absolute value of top ten strongest correlation between mailing indicator and other variables. Except the variable of outcome_spend that are supposed to be correlated with mailing indicator, the pearson correlation scores of all variables are very small. Therefore, mailing is unlikely to depend on them.

Estimation of heterogeneous treatment effects

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

OLS and LASSO

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

```
set.seed(2001)
train.x <- model.matrix(outcome_spend ~ 0 + . * W, data = training_DT)
train.y <- training_DT$outcome_spend
cv.lasso = cv.glmnet(train.x, train.y, alpha = 1.0)
fit_lasso = as.numeric(coef(cv.lasso, s = "lambda.min"))
```

```
coefs <- cbind(coef(fit_OLS), as.data.frame(fit_lasso))
colnames(coefs) = c("OLS", "LASSO")
kable(coefs, caption = "The coefficient of OLS and LASSO")
```

Table 1: The coefficient of OLS and LASSO

	OLS	LASSO
(Intercept)	-10.4237687	-2.2859521
customer_type_L	1.5323734	0.5400523
customer_type_C	-0.7818111	0.0000000
clicks_product_type_504_3m	-0.0008496	0.0000000
clicks_product_type_112_6m	-0.0005086	0.0000000
clicks_product_type_301_1m	0.6062685	0.1913960
clicks_product_type_001_12m	0.1881354	0.0000000
clicks_product_type_201_1m	0.5616708	0.0000000
clicks_product_type_201_6m	0.0486686	0.0847655
web_activity_3m	0.0404776	0.0090689
clickthrough_1m	1.0096415	0.5074061
emailview_1m	-0.0577769	0.0190553
orders_online_1yr	-0.8464205	0.0000000
online_customer	8.9685765	2.5155505
orders_season_E	-0.0921761	0.0000000
orders_mail_dept_h_1yr	0.5158121	0.0000000
spend_d_h_1yr	0.0144129	0.0010191
spend_online_attributed_target	0.0024684	0.0024449
dollars_season_C	-0.0003045	0.0006510
spend_season_C	-0.0009630	0.0000000
spend_e	-0.0008912	0.0000000
spend_z1	0.0004670	0.0000000
spend_period_1b	-0.0142584	0.0062241
spend_period_2b	0.0010506	0.0000000
spend_h_3yr	-0.0029276	0.0008544
spend_g_3yr	0.0020990	0.0001308
last_years_since	-1.1485791	-2.1080051
spend_instore_o	-0.0006195	0.0000000
spend_instore_o_3yr	0.0102093	0.0005857
spend_instore_h_3yr	-0.0068046	0.0000000
spend_instore_t	0.0079197	0.0005971
spend_online_sh	-0.0278957	0.0000000
spend_online_o_1yr	0.0162517	0.0200363
spend_online_n_3yr	0.0176486	0.0023202
spend_online_o_3yr	0.0077808	0.0036317
spend_online_a_1yr	0.0290649	0.0093199
spend_online_a_3yr	0.0033422	0.0048312
orders_online_t_3yr	0.9233484	0.0000000

	OLS	LASSO
spend_online_sg_1yr	0.0514794	0.0023661
spend_online_g_1yr	0.0065082	0.0000000
orders_online_s_1yr	5.7816258	0.3430809
customer_type_7	-0.9693192	-2.9255084
customer_type_A	0.3546467	0.0000000
orders_attributed_mail_type_B	0.1630720	0.0000000
spend_attributed_mail_type_B	0.0004885	0.0000000
spend_attributed_mail_type_C	0.0027243	0.0000000
mean_spend_attributed_mail_type_C	-0.0064892	0.0000000
mean_spend_attributed_mail_type_A	0.0060175	0.0000000
clicks_product_type_104_2yr	-0.1575330	0.0000000
clicks_product_type_312_3yr	0.1216075	0.0000000
orders_e	0.3243718	0.0000000
orders_mail_dept_c_1yr	1.6206004	0.0000000
spend_d_s_1yr	0.0031505	0.0000000
spend_d_k_1yr	0.0146115	0.0000000
spend_a_1yr	0.0095989	0.0000000
spend_h_1yr	-0.0046038	0.0000000
spend_direct_1yr	0.0122114	0.0000000
acquisition_months_since	0.0013148	0.0000000
spend_online_c	-0.0077187	0.0000000
spend_online_a_6m	-0.0117197	0.0000000
orders_online_o	-0.0459762	0.0000000
orders_online_c_1yr	-4.4835998	0.0000000
spend_online_b_1yr	0.0326285	0.0021512
customer_type_2	0.4242386	0.0000000
customer_income	-0.0000004	0.0000000
orders_attributed_mail_type_A	0.4049904	0.0000000
spend_attributed_mail_type_A	-0.0077656	0.0000000
clicks_product_type_312_3m	-0.5009797	0.0000000
clicks_product_type_301_2yr	0.0581843	0.0000000
clickthrough_6m	-0.2926115	0.0000000
emails_days_2yr	0.0026553	0.0000000
emails_3m	0.0461985	0.0000000
emailreceived_months_since	0.0133921	0.0000000
orders_3yr	-0.1773285	0.0000000
orders_mail_dept_d_1yr	-1.1280331	0.0000000
spent_q	-0.0062125	0.0000000
spend_instore_q	-0.0362621	0.0000000
spend_online_s	-0.0034040	0.0000000
clicks_product_type_112_3yr	-0.0456652	0.0000000
spend_instore_n_3yr	-0.0066837	0.0000000
spend_instore_p	0.0002721	0.0000000
spend_instore_p_1yr	-0.0377236	0.0000000
orders_online_n_1yr	-3.1511319	0.0000000
spend_instore_a_yr	-0.0012543	0.0000000
orders_attributed_mail_type_C	-0.2687712	0.0000000
clickthrough_3yr	0.0775048	0.0000000
spend_instore_g_1yr	-0.0162895	0.0000000
spend_period_3b	0.0010880	0.0006682
spend_instore_a	-0.0000815	0.0000000
spend_direct_g	-0.0002392	0.0000000
emailview_24m	0.0081034	0.0000000
spend_online_h_3yr	-0.0024386	0.0000000
emailview_months_since	-0.0009149	0.0000000
orders_instore_c	-0.3906197	0.0000000
emails_1yr	-0.0136281	0.0000000
clicks_product_type_001_1m	-0.6178256	0.0000000
clickthrough_3m	0.1476846	0.0000000
orders_d_1yr	0.1614821	0.0000000

	OLS	LASSO
orders_h	0.0645826	0.0000000
clicks_product_type_104	0.0581256	0.0000000
clicks_product_type_502_3m	0.0169925	0.0000000
web_activity_1m	0.0043018	0.0000000
orders_instore_m	-0.3569307	0.0000000
spend_notz_1yr	-0.0027291	0.0000000
orders_online_sh	1.4226457	0.0000000
orders_instore_n	-0.5269629	0.0000000
clicks_product_type_502_1yr	-0.0006738	0.0000000
orders_hm	0.1150392	0.0000000
emailview	-0.0020853	0.0000000
spend_m_1yr	-0.0083884	0.0000000
clicks_product_type_301_3m	0.2178148	0.0000000
orders_instore_a_yr	0.3839173	0.0000000
clicks_product_type_001_6m	-0.2406725	0.0000000
clickthrough_months_since	-0.0033417	0.0000000
orders_online_h_1yr	2.7312291	0.0000000
orders_instore_h_3yr	0.5800739	0.0000000
spend_period_1a	-0.0180295	0.0000000
orders_total	0.0016445	0.0000000
spend_online_s_3yr	0.0054331	0.0000000
spend_M_3yr	0.0014057	0.0000000
orders_c	-0.2165899	0.0000000
spend_l	0.0090723	0.0000000
spend_instore_n	0.0205373	0.0013101
orders_z	1.2291890	0.0000000
web_activity_24m	-0.0175863	0.0000000
spend_instore_q_3yr	0.0394836	0.0000000
clicks_product_type_504	-0.0015381	0.0000000
emailview_6m	0.0186816	0.0000000
buy_instore_days_since	-0.0001322	0.0000000
spend_h	0.0011828	0.0000000
clicks_product_type_801_3yr	-0.0898191	0.0000000
spend_nott_1yr	0.0238686	0.0000000
orders_online_s_3yr	-0.0022567	0.0000000
spend_instore_h_1yr	0.0111784	0.0000000
spend_online_b_3yr	-0.0077022	0.0000000
orders_instore_s_3yr	-0.3950668	0.0000000
orders_instore_r_3yr	0.7845586	0.0000000
clicks_product_type_502_1m	-0.0400536	0.0000000
mean_spend_attributed_mail_type_B	-0.0053832	0.0000000
clicks_product_type_201_3yr	0.0250070	0.0000000
store_trips	0.0545995	0.0000000
spend_online_t_1yr	-0.0090850	0.0000000
orders_online_o_1yr	1.2012777	0.0000000
spend_online_k	-0.0012439	-0.0021712
spend_online_s_1yr	-0.0100672	0.0000000
clicks_product_type_503	0.0033073	0.0000000
orders_instore_t_3yr	0.3556849	0.0000000
W	15.2710823	0.0000000
customer_type_L:W	1.1899577	1.1806360
customer_type_C:W	-0.2651638	0.0000000
clicks_product_type_504_3m:W	0.0319454	0.0000000
clicks_product_type_112_6m:W	-0.0669111	0.0000000
clicks_product_type_301_1m:W	-0.0101398	0.0000000
clicks_product_type_001_12m:W	-0.1896029	0.0000000
clicks_product_type_201_1m:W	-0.7148368	0.0000000
clicks_product_type_201_6m:W	0.0624698	0.0000000
web_activity_3m:W	-0.0201362	0.0000000
clickthrough_1m:W	0.1416537	0.0000000

	OLS	LASSO
emailview_1m:W	0.2057091	0.0000000
orders_online_1yr:W	1.7644585	0.0000000
online_customer:W	-10.3206883	0.0000000
orders_season_E:W	-0.0958728	0.0000000
orders_mail_dept_h_1yr:W	-0.8189520	0.1361295
spend_d_h_1yr:W	-0.0007325	0.0000000
spend_online_attributed_target:W	0.0010955	0.0000000
dollars_season_C:W	0.0014674	0.0001641
spend_season_C:W	0.0008609	0.0000000
spend_e:W	-0.0008296	0.0000000
spend_z1:W	0.0005284	0.0000000
spend_period_1b:W	0.0261264	0.0000000
spend_period_2b:W	0.0001430	0.0024769
spend_h_3yr:W	-0.0049345	0.0000000
spend_g_3yr:W	0.0069833	0.0000879
last_years_since:W	-2.2108949	0.0000000
spend_instore_o:W	0.0002742	0.0000000
spend_instore_o_3yr:W	-0.0052852	0.0000000
spend_instore_h_3yr:W	-0.0051356	0.0000000
spend_instore_t:W	-0.0052011	0.0000000
spend_online_sh:W	0.0201109	0.0000000
spend_online_o_1yr:W	0.0060725	0.0000000
spend_online_n_3yr:W	-0.0087776	0.0000000
spend_online_o_3yr:W	-0.0150642	-0.0000424
spend_online_a_1yr:W	-0.0258027	0.0000000
spend_online_a_3yr:W	0.0019852	0.0000000
orders_online_t_3yr:W	0.2209908	0.0000000
spend_online_sg_1yr:W	-0.0646020	-0.0015906
spend_online_g_1yr:W	-0.0204276	0.0000000
orders_online_s_1yr:W	-4.4248925	0.0000000
customer_type_7:W	-9.9105781	0.0000000
customer_type_A:W	-0.2033647	0.4902376
orders_attributed_mail_type_B:W	0.2232580	0.0440158
spend_attributed_mail_type_B:W	0.0026500	0.0066611
spend_attributed_mail_type_C:W	-0.0026528	0.0000000
mean_spend_attributed_mail_type_C:W	0.0095328	0.0030695
mean_spend_attributed_mail_type_A:W	0.0048605	0.0026654
clicks_product_type_104_2yr:W	0.2889429	0.0000000
clicks_product_type_312_3yr:W	-0.1551708	0.0000000
orders_e:W	-0.6884143	-0.1343353
orders_mail_dept_c_1yr:W	-2.3847079	0.0000000
spend_d_s_1yr:W	0.0100345	0.0000000
spend_d_k_1yr:W	-0.0031762	0.0000000
spend_a_1yr:W	-0.0065793	-0.0051520
spend_h_1yr:W	0.0224750	0.0054767
spend_direct_1yr:W	-0.0182008	0.0058726
acquisition_months_since:W	-0.0007165	0.0002534
spend_online_c:W	0.0136083	0.0031187
spend_online_a_6m:W	0.0087809	-0.0027937
orders_online_o:W	-0.0680719	-0.0714311
orders_online_c_1yr:W	4.0165727	0.0000000
spend_online_b_1yr:W	-0.0481974	-0.0066387
customer_type_2:W	-2.3584917	-0.2044320
customer_income:W	0.0000006	0.0000000
orders_attributed_mail_type_A:W	-0.2203815	0.0000000
spend_attributed_mail_type_A:W	0.0026413	0.0000000
clicks_product_type_312_3m:W	0.3724077	0.0000000
clicks_product_type_301_2yr:W	-0.0873933	0.0000000
clickthrough_6m:W	0.2318008	0.0000000
emails_days_2yr:W	-0.0037236	0.0000000

	OLS	LASSO
emails_3m:W	-0.0626076	0.0000000
emailreceived_months_since:W	-0.0041402	0.0000000
orders_3yr:W	0.0084634	0.0000000
orders_mail_dept_d_1yr:W	1.2493939	0.0000000
spent_q:W	0.0039344	0.0000000
spend_instore_q:W	0.0297124	0.0000000
spend_online_s:W	0.0033940	0.0000000
clicks_product_type_112_3yr:W	0.0824775	0.0000000
spend_instore_n_3yr:W	0.0059700	0.0000000
spend_instore_p:W	0.0114761	0.0000000
spend_instore_p_1yr:W	0.0490062	0.0000000
orders_online_n_1yr:W	3.8470183	0.0000000
spend_instore_a_yr:W	0.0050088	0.0000000
orders_attributed_mail_type_C:W	0.3863748	0.0000000
clickthrough_3yr:W	-0.0493164	0.0000000
spend_instore_g_1yr:W	0.0139925	0.0000000
spend_period_3b:W	-0.0009450	0.0000000
spend_instore_a:W	-0.0005975	0.0000000
spend_direct_g:W	-0.0002767	0.0000000
emailview_24m:W	0.0030151	0.0000000
spend_online_h_3yr:W	-0.0093480	0.0000000
emailview_months_since:W	-0.0210851	0.0000000
orders_instore_c:W	-0.0598968	0.0000000
emails_1yr:W	0.0168292	0.0000000
clicks_product_type_001_1m:W	0.6609258	0.0000000
clickthrough_3m:W	-0.5459961	0.0000000
orders_d_1yr:W	-0.4326373	0.0000000
orders_h:W	-0.1171220	0.0000000
clicks_product_type_104:W	-0.0466831	0.0000000
clicks_product_type_502_3m:W	-0.0410559	0.0000000
web_activity_1m:W	-0.0479486	0.0000000
orders_instore_m:W	0.4427757	0.0000000
spend_notz_1yr:W	-0.0035120	0.0000000
orders_online_sh:W	-0.9881208	0.0000000
orders_instore_n:W	0.2908038	0.0000000
clicks_product_type_502_1yr:W	-0.0105534	0.0000000
orders_hm:W	0.0350867	0.0141908
emailview:W	-0.0039135	0.0000000
spend_m_1yr:W	0.0190318	0.0010924
clicks_product_type_301_3m:W	-0.2193320	0.0000000
orders_instore_a_yr:W	-0.8050795	0.0000000
clicks_product_type_001_6m:W	0.1003104	0.0000000
clickthrough_months_since:W	0.0085977	0.0000000
orders_online_h_1yr:W	-3.1609353	0.0000000
orders_instore_h_3yr:W	0.3872466	0.0000000
spend_period_1a:W	0.0239679	0.0000000
orders_total:W	0.0115420	0.0000000
spend_online_s_3yr:W	-0.0033651	0.0000000
spend_M_3yr:W	0.0121509	0.0000000
orders_c:W	-0.1093556	0.0000000
spend_l:W	0.0010503	0.0013754
spend_instore_n:W	-0.0130712	0.0000000
orders_z:W	-0.7219641	0.0000000
web_activity_24m:W	0.0248935	0.0009906
spend_instore_q_3yr:W	-0.0137286	0.0000000
clicks_product_type_504:W	-0.0028503	0.0000000
emailview_6m:W	-0.0534695	0.0000000
buy_instore_days_since:W	0.0001589	0.0000000
spend_h:W	-0.0008096	0.0000000
clicks_product_type_801_3yr:W	0.0825331	0.0000000

	OLS	LASSO
spend_nott_1yr:W	0.0020439	0.0000000
orders_online_s_3yr:W	0.2615383	0.0000000
spend_instore_h_1yr:W	-0.0067987	0.0000000
spend_online_b_3yr:W	0.0062119	0.0000000
orders_instore_s_3yr:W	0.0967249	0.0000000
orders_instore_r_3yr:W	-2.9189592	-0.2199327
clicks_product_type_502_1m:W	0.0800654	0.0000000
mean_spend_attributed_mail_type_B:W	0.0065177	0.0000000
clicks_product_type_201_3yr:W	-0.0096898	0.0000000
store_trips:W	0.0040052	0.0000000
spend_online_t_1yr:W	-0.0341662	0.0000000
orders_online_o_1yr:W	-1.7509273	0.0000000
spend_online_k:W	-0.0048122	0.0000000
spend_online_s_1yr:W	-0.0101581	0.0000000
clicks_product_type_503:W	-0.0031201	0.0000000
orders_instore_t_3yr:W	-1.1245886	0.0000000

Causal Forest

```
set.seed(2001)
fit_CF = causalForest(outcome_spend ~ . - W,
                      treatment = training_DT$W,
                      data = training_DT,
                      num.trees = 1000,
                      verbose = TRUE)

pred_tau_CF = predict(fit_CF, validation_DT)
```

Predict treatment effects

Since $\hat{\tau}_i = \delta_0 + \sum_{k=1}^{p} \delta_k x_{ik}$, and in our linear models with interaction terms, the δ_0 is the coefficient of W, the δ_k is the cofficient of the interaction term $x_k w$, we can compute the predicted CATE through the following function.

```
# Function for computing the predicted CATE of linear models with interaction terms
interaction_tau <- function(coef, data){
  colSums(coef[149:294] * t(data[, !c('outcome_spend', 'W')])) + coef[148]
}
```

```
# Compute and add the predicted CATE of the three models and the predicted spend of the linear models
crm_pred <- validation_DT %>%
  select(W, outcome_spend) %>%
  mutate(pred_tau_OLS = interaction_tau(coef(fit_OLS), validation_DT),
         pred_tau_lasso = interaction_tau(fit_lasso, validation_DT),
         pred_tau_CF = pred_tau_CF) %>%
  as.data.table()
```

```
head(crm_pred, 10)
```

```
##     W outcome_spend pred_tau_OLS pred_tau_lasso pred_tau_CF
##  1: 1             0    6.8922820      1.6297777    2.359051
##  2: 0             0   -2.3147083      1.1062204    2.896885
##  3: 0             0    0.6294586      1.6140947    1.628892
##  4: 0             0   -0.7357745      2.4605086    3.161940
##  5: 0             0    3.6830536      0.2293263    2.349746
##  6: 0             0   -0.9584761      0.2620470    1.113217
##  7: 1             0    5.7054312      2.2863212    3.433194
##  8: 0             0   -5.1012020      0.1365489    2.243982
##  9: 1             0    2.9704863      1.0901839    2.006842
## 10: 1             0  -16.8198947     -4.0111005    2.852758
```

Step 2: Model fit and profit evaluation in 2015 validation sample

Descriptive analysis of predicted treatment effects

Since we predict the CATE in the validation data, I also only caculate the ATE of the validation data.

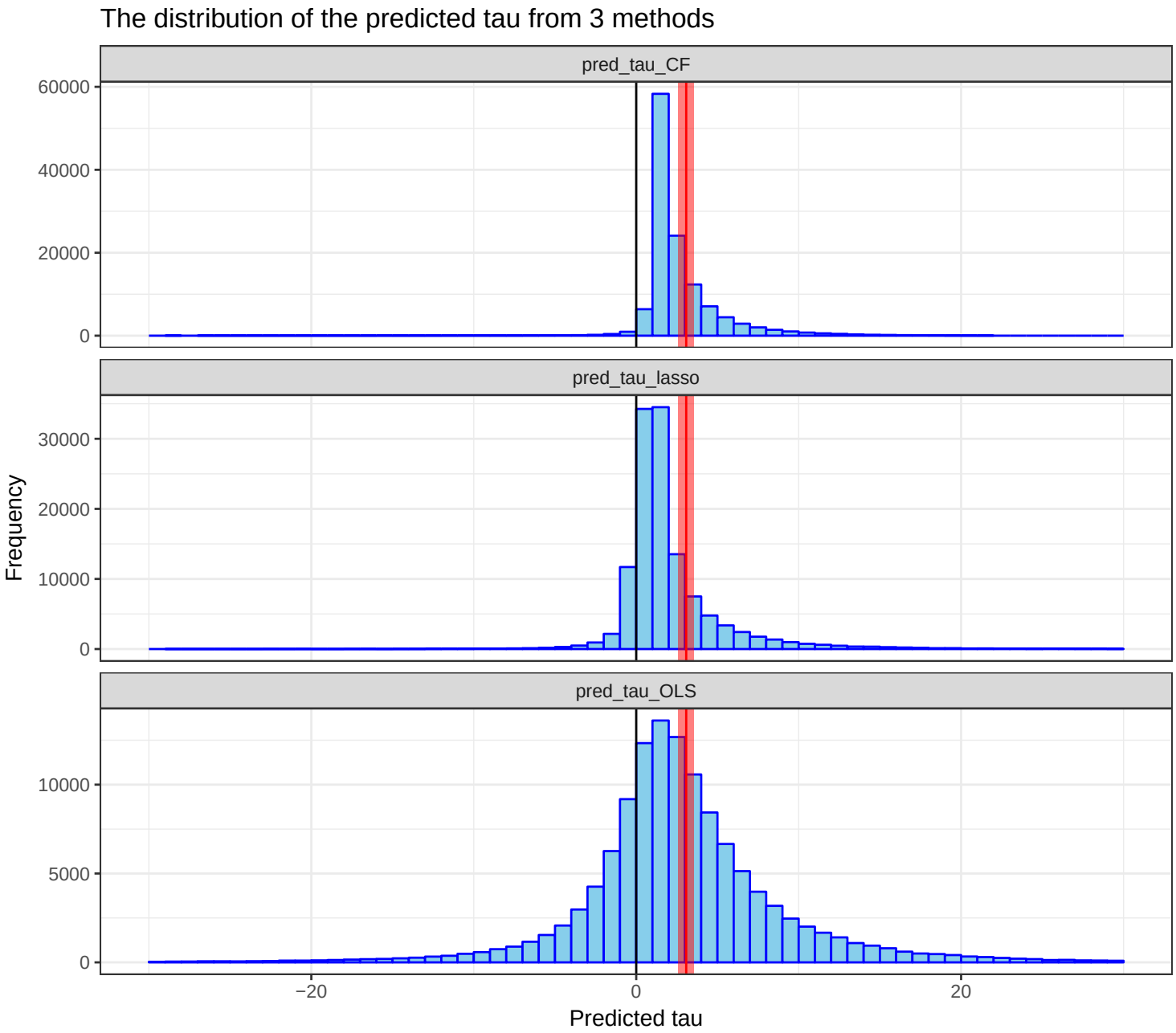
```
T.test <- t.test(outcome_spend ~ W, data = crm_pred,
                 conf.level = 0.95) %>%
  tidy() %>%
  mutate(ATE = -estimate, conf.low = -conf.low,
         conf.high = -conf.high) %>%
  select(-estimate, -statistic, - alternative, - method, -parameter)
bind_rows('Causal Forest' = as.tibble(describe(crm_pred$pred_tau_CF)),
          'LASSO' = as.tibble(describe(crm_pred$pred_tau_lasso)),
          'OLS' = as.tibble(describe(crm_pred$pred_tau_OLS)), .id = 'Model') %>%
  select(-vars, - kurtosis, - mad) %>%
  kable(caption = "Summary Statistics")
```

Table 2: Summary Statistics

Model	n	mean	sd	median	trimmed	min	max	range	skew	se
Causal Forest	124877	2.646259	2.545653	1.883543	2.243664	-41.21170	23.18033	64.39203	0.6711957	0.0072037
LASSO	124877	2.091489	3.644776	1.258325	1.551082	-65.42153	115.38805	180.80959	3.6607221	0.0103141
OLS	124877	2.735326	8.341281	2.247492	2.562889	-313.14383	467.15565	780.29949	-0.5008921	0.0236043

```
crm_pred %>%
  gather('pred_tau_CF', 'pred_tau_OLS', 'pred_tau_lasso',
         key = method, value = pred) %>%
  ggplot(aes(pred)) +
  geom_histogram(fill = "sky blue", binwidth = 1, boundary = 1, color = 'blue') +
  annotate("rect", xmin = T.test$conf.low, xmax = T.test$conf.high,
           ymin = -Inf, ymax = Inf, fill = "red", alpha = 0.5) +
  geom_vline(aes(xintercept=T.test$ATE), color = 'red') +
  geom_vline(aes(xintercept=0), color = 'black') +
  facet_wrap(~ method, scales = 'free_y', ncol = 1) +
  xlim(-30, 30) +
  theme_bw() +
  labs(title = 'The distribution of the predicted tau from 3 methods',
       x = 'Predicted tau',
       y = 'Frequency')
```

```
## Warning: Removed 1585 rows containing non-finite values (stat_bin).
```



The red line and the red area in the graph is the ATE and its 95% confidence interval of the validation data. According to the table and the graph, the OLS model has the largest variation and the casual forest model thas the smallest one. Besides, the mean of the result in the LASSO model is out of the 95% confidence interval of the ATE.

The CATE in the OLS model is least plausible since the CATE for most people, according to the theory and experience, should be positive. Since the mean of the LASSO result is out of the interval of the ATE, the result of the causal forest is most plausible. Finally, the OLS model are more close to normal distribution and the other two are obviously heavily right skewed. Therefore, the results of the LASSO and the causal tree models are more similar.

```
select(crm_pred, pred_tau_CF, pred_tau_OLS, pred_tau_lasso) %>%
  cor() %>%
  kable(caption = "The correlation between three methods")
```

Table 3: The correlation between three methods

	pred_tau_CF	pred_tau_OLS	pred_tau_lasso
pred_tau_CF	1.0000000	0.5043603	0.6887127
pred_tau_OLS	0.5043603	1.0000000	0.6261089
pred_tau_lasso	0.6887127	0.6261089	1.0000000

The table show their degrees of correlation. The LASSO and causal forest models are most similar, and the causal forest and the OLS models are least similar.

```
describe(crm_pred$outcome_spend) %>%
  kable(caption = "The summary statistics of the sales")
```

Table 4: The summary statistics of the sales

	vars	n	mean	sd	median	trimmed	mad	min	max	range	skew	kurtosis	se
X1	1	124877	7.855817	43.76681	0	0	0	0	1462.84	1462.84	10.65534	179.7962	0.1238522

Comparing to the scale of sales that most values are zero and the rest are positive, that of CATEs are usually non-zero and can be negative. In addition, the former has a wider range and skew.

Model validation: Lifts

The easiest way to compute the average treatment effect and the confidence intervals is directly using t.test function in R. So I write a for loop for t tests in the lift table function.

```
# Lift table function
liftTable <- function(outcome, pred, W, N_groups){

  DT = data.table(outcome = outcome,
                  pred = pred,
                  W = W)
  # Create groups
  DT[, score_group := as.integer(cut_number(pred, n = N_groups))]

  # The ATE for the whole dataset
  ATE = mean(DT[W == 1]$outcome) - mean(DT[W == 0]$outcome)

  # Compute the ATEs and the confidence intervals using t.test by groups
  TE_DT <- NA
  for (i in 1:20){
    group_DT <- filter(DT, score_group == i)
    TE <- t.test(outcome ~ W, data = group_DT, conf.level = 0.95) %>%
      tidy() %>%
      select(estimate, conf.low, conf.high) %>%
      mutate(score_group = i, estimate = -estimate,
             conf.low = -conf.low, conf.high = -conf.high)

    TE_DT <- rbind(TE_DT, TE)
    }

  # Compute the lifts and their confidence intervals
  TE_DT <- TE_DT %>%
    drop_na() %>%
    mutate(lift = 100 * estimate / ATE,
           lift_lower = 100 * conf.low / ATE,
           lift_upper = 100 * conf.high / ATE)

  return(TE_DT)
}
```

```
lift_OLS <- liftTable(crm_pred$outcome_spend, crm_pred$pred_tau_OLS, crm_pred$W, 20)
lift_lasso <- liftTable(crm_pred$outcome_spend, crm_pred$pred_tau_lasso, crm_pred$W, 20)
lift_CF <- liftTable(crm_pred$outcome_spend, crm_pred$pred_tau_CF, crm_pred$W, 20)

lift_table <-  bind_rows("OLS" = lift_OLS, "LASSO" = lift_lasso,
                         "Causal Forest" = lift_CF, .id = "Method")
```

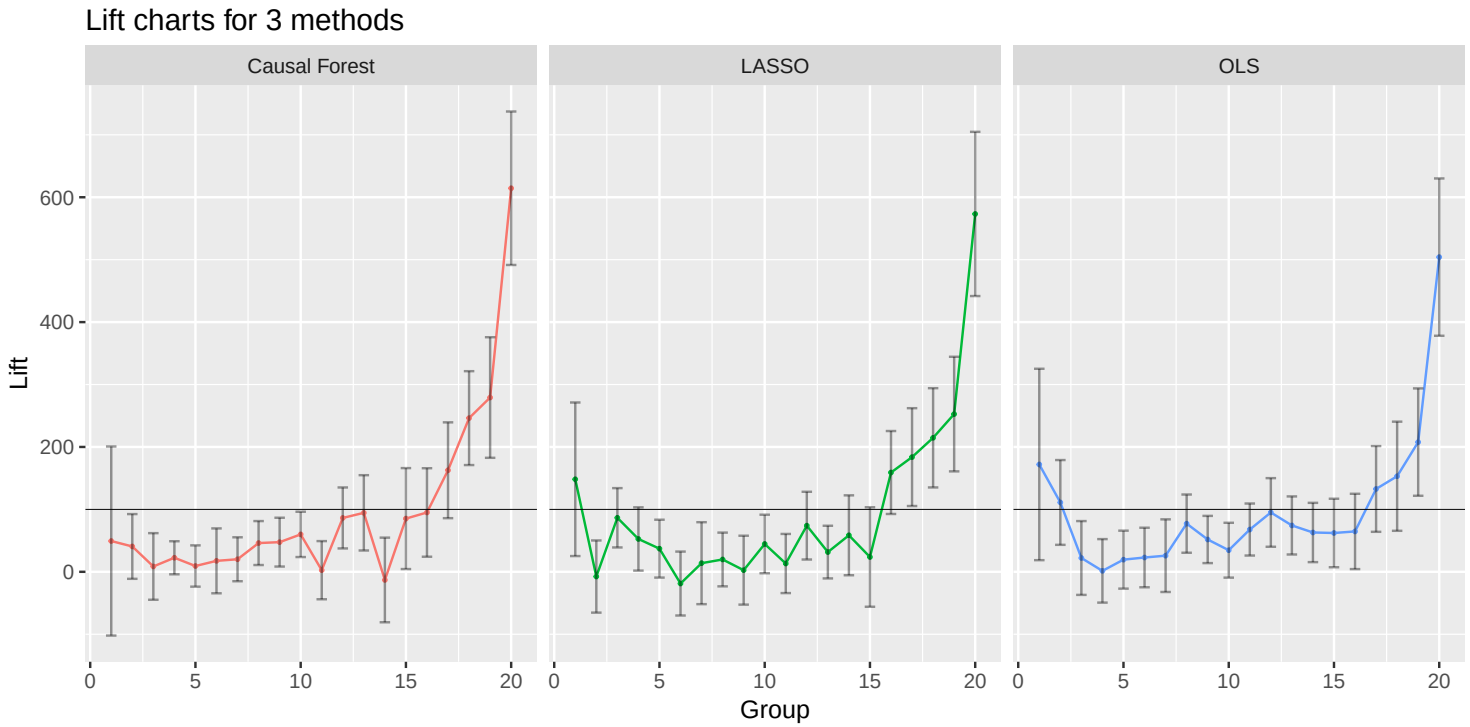
```
kable(lift_table, caption = "Lift table for four methods")
```

Table 5: Lift table for four methods

Method	estimate	conf.low	conf.high	score_group	lift	lift_lower	lift_upper
OLS	5.2904181	10.0065023	0.5743340	1	171.957397	325.24690	18.667897
OLS	3.4186614	5.5077140	1.3296088	2	111.118650	179.02029	43.217014
OLS	0.6813647	2.5000732	-1.1373439	3	22.146774	81.26127	-36.967720
OLS	0.0467584	1.6115048	-1.5179881	4	1.519814	52.37964	-49.340008
OLS	0.5988138	2.0269098	-0.8292821	5	19.463579	65.88177	-26.954615
OLS	0.7052523	2.1728798	-0.7623751	6	22.923207	70.62632	-24.779902
OLS	0.7953085	2.5853104	-0.9946935	7	25.850353	84.03178	-32.331074
OLS	2.3739914	3.8089839	0.9389990	8	77.163162	123.80552	30.520805
OLS	1.5923768	2.7557902	0.4289633	9	51.757906	89.57298	13.942833
OLS	1.0642888	2.4161983	-0.2876206	10	34.593170	78.53503	-9.348692
OLS	2.0840474	3.3656938	0.8024010	11	67.738949	109.39701	26.080884
OLS	2.9256596	4.6170713	1.2342480	12	95.094339	150.07123	40.117447
OLS	2.2851666	3.7162719	0.8540613	13	74.276039	120.79205	27.760029
OLS	1.9380858	3.3984740	0.4776975	14	62.994677	110.46249	15.526867
OLS	1.9131384	3.6012163	0.2250604	15	62.183799	117.05233	7.315264
OLS	1.9899143	3.8456395	0.1341892	16	64.679289	124.99695	4.361625
OLS	4.0835194	6.1963940	1.9706448	17	132.728897	201.40483	64.052963
OLS	4.7105706	7.4004672	2.0206740	18	153.110290	240.54149	65.679087
OLS	6.3929235	9.0388633	3.7469836	19	207.792739	293.79519	121.790289
OLS	15.5092841	19.3858147	11.6327535	20	504.106869	630.10790	378.105842
LASSO	4.5606886	8.3447368	0.7766405	1	148.238594	271.23361	25.243576
LASSO	-0.2327716	1.5457161	-2.0112592	2	-7.565903	50.24127	-65.373075
LASSO	2.6626841	4.1244184	1.2009497	3	86.546697	134.05826	39.035136
LASSO	1.6170971	3.1758560	0.0583382	4	52.561404	103.22661	1.896198
LASSO	1.1412469	2.5669543	-0.2844604	5	37.094583	83.43514	-9.245976
LASSO	-0.5780907	0.9967645	-2.1529458	6	-18.790002	32.39839	-69.978392
LASSO	0.4263547	2.4458877	-1.5931783	7	13.858043	79.50005	-51.783960
LASSO	0.6080782	1.9319565	-0.7158001	8	19.764703	62.79545	-23.266049
LASSO	0.0805957	1.7789655	-1.6177741	9	2.619646	57.82270	-52.583411
LASSO	1.3723836	2.8145573	-0.0697902	10	44.607345	91.48312	-2.268430
LASSO	0.4090828	1.8664903	-1.0483248	11	13.296644	60.66757	-34.074283
LASSO	2.2753024	3.9448020	0.6058029	12	73.955417	128.22009	19.690747
LASSO	0.9720626	2.2699296	-0.3258044	13	31.595491	73.78078	-10.589801
LASSO	1.8001154	3.7692461	-0.1690153	14	58.510150	122.51390	-5.493597
LASSO	0.7289096	3.1800182	-1.7221991	15	23.692152	103.36190	-55.977594
LASSO	4.8940202	6.9383175	2.8497229	16	159.073055	225.51999	92.626124
LASSO	5.6533888	8.0641225	3.2426550	17	183.755233	262.11265	105.397815
LASSO	6.6042832	9.0498011	4.1587654	18	214.662684	294.15071	135.174659
LASSO	7.7737674	10.5974443	4.9500906	19	252.675077	344.45461	160.895542
LASSO	17.6367874	21.6835024	13.5900724	20	573.258289	704.79091	441.725665
Causal Forest	1.5171543	6.1756205	-3.1413119	1	49.312908	200.72962	-102.103804
Causal Forest	1.2495977	2.8452831	-0.3460878	2	40.616367	92.48182	-11.249083
Causal Forest	0.2640900	1.9054903	-1.3773102	3	8.583865	61.93521	-44.767478
Causal Forest	0.6925192	1.5073024	-0.1222639	4	22.509337	48.99269	-3.974013
Causal Forest	0.2856689	1.3029896	-0.7316519	5	9.285254	42.35180	-23.781288
Causal Forest	0.5397969	2.1384258	-1.0588320	6	17.545319	69.50644	-34.415804
Causal Forest	0.6183264	1.7004712	-0.4638183	7	20.097806	55.27136	-15.075745
Causal Forest	1.4175141	2.5009411	0.3340870	8	46.074246	81.28948	10.859015
Causal Forest	1.4623133	2.6653942	0.2592323	9	47.530379	86.63479	8.425973
Causal Forest	1.8443147	2.9613221	0.7273073	10	59.946784	96.25350	23.640071
Causal Forest	0.0805355	1.5135625	-1.3524915	11	2.617689	49.19616	-43.960782
Causal Forest	2.6545151	4.1571728	1.1518574	12	86.281177	135.12289	37.439461
Causal Forest	2.9069978	4.7626466	1.0513490	13	94.487764	154.80295	34.172581
Causal Forest	-0.4015639	1.6838985	-2.4870262	14	-13.052254	54.73269	-80.837194
Causal Forest	2.6231279	5.1114038	0.1348521	15	85.260983	166.13879	4.383172

Method	estimate	conf.low	conf.high	score_group	lift	lift_lower	lift_upper
Causal Forest	2.9244302	5.1056380	0.7432225	16	95.054379	165.95138	24.157374
Causal Forest	5.0051873	7.3662826	2.6440921	17	162.686381	239.43037	85.942392
Causal Forest	7.5743070	9.8877687	5.2608454	18	246.191905	321.38763	170.996177
Causal Forest	8.5897148	11.5612486	5.6181810	19	279.196269	375.78168	182.610856
Causal Forest	18.9037034	22.6850650	15.1223418	20	614.437563	737.34526	491.529867

```
ggplot(lift_table, aes(score_group, lift, color = Method)) +
  geom_line(show.legend = FALSE) +
  geom_point(size = 0.5, show.legend = FALSE) +
  geom_errorbar(aes(ymin=lift_lower, ymax=lift_upper),
                width = 0.5, color = "black", alpha = 0.4) +
  geom_hline(aes(yintercept = 100), size = 0.2) +
  facet_wrap(~ Method) +
  labs(title = "Lift charts for 3 methods",
       x = "Group",
       y = "Lift")
```



The lift table and the chart show that the OLS model performs obviously worst. The causal forest model performs a bit better than the LASSO model.

Profit predictions

$$\hat{\Pi}(T) = \sum_{i=1}^{n} \left[\left(\frac{1 - W_i}{1 - e} \right) (1 - T_i) \cdot mY_i + \left(\frac{W_i}{e} \right) T_i \cdot (mY_i - c) \right]$$

For the actual randomized targeting, the targeting condition can be divided into four parts: should be targeted and was targeted (G_1); should be targeted but was not targeted; should not be targeted and was not be targeted(G_1); should not be targeted but was targed. In the equaition of $\hat{\Pi}(T)$, if the condition is "should be targeted but was not targeted" or "should not be targeted but was targeted" for customer i, $\hat{\Pi}_i(T)$ will be zero. It means these two kinds of condition have been omitted and we get the sub-sample that only contains G_0 and G_1. In this case, we must scale our calculation to substitute the observations not in the sub-sample. Since the estimated profit is based on a probability of the random sample of the profits from customers who should be targeted and 1/e of the random sample of the profits from customers who should not be targeted, we must respectively scale them by 1/e and 1/(1-e).

```r
# Optimal profit strategy function
opt_strategy <- function(outcome, tau, W, c, m){
  DT = data.table(tau = tau,
                  W = W,
                  outcome = outcome)

  DT[, t := if_else(m * tau > c, 1, 0)]
  e = nrow(DT[W == 1]) / nrow(DT)

  DT[, profit := ((1-W)/(1-e))*(1-t)*m*outcome + (W/e)*t*(m*outcome-c)]
  return(DT)

}
```

```r
c = 0.99
m = 0.325

opt_profit_OLS <- opt_strategy(crm_pred$outcome_spend, crm_pred$pred_tau_OLS,
                               crm_pred$W, c, m)
opt_profit_lasso <- opt_strategy(crm_pred$outcome_spend, crm_pred$pred_tau_lasso,
                                 crm_pred$W, c, m)
opt_profit_CF <- opt_strategy(crm_pred$outcome_spend, crm_pred$pred_tau_CF,
                              crm_pred$W, c, m)
```

```r
scale = 1000000 / 1000

OLS_opt <- mean(opt_profit_OLS$profit) * scale
OLS_per <- mean(opt_profit_OLS$t)
lasso_opt <- mean(opt_profit_lasso$profit) * scale
lasso_per <- mean(opt_profit_lasso$t)
CF_opt <- mean(opt_profit_CF$profit) * scale
CF_per <- mean(opt_profit_CF$t)
all <- mean(crm_pred[W == 1]$outcome_spend * m - c) * scale
none <- mean(crm_pred[W == 0]$outcome_spend * m) * scale
```

```r
cat('The scale of the profit is 1 thousand dollors per 1 million customers:')
```

```
## The scale of the profit is 1 thousand dollors per 1 million customers:
```

```r
cat('\n\nOLS:', 'percent is          ', OLS_per, ', profit is', OLS_opt)
```

```
##
##
## OLS: percent is           0.4200533 , profit is 2124.407
```

```r
cat('\nLASSO:', 'percent is        ', lasso_per, ', profit is', lasso_opt)
```

```
##
## LASSO: percent is         0.2069476 , profit is 2312.83
```

```r
cat('\nCausal Forest:', 'percent is', CF_per, ', profit is', CF_opt)
```

```
##
## Causal Forest: percent is 0.2658296 , profit is 2346.44
```

```r
cat('\nThe profit for targeting all is                ', all)
```

```
##
## The profit for targeting all is                 1896.12
```

```r
cat('\nThe profit for targeting none is               ', none)
```

```
##
## The profit for targeting none is                1886.229
```

The causal forest model performs best, with the targeting percent of 0.266 and the profit of 2346. The LASSO model are slightly

worse than the caudal forest model, which has the percent of 0.207 and the profit 2313. The OLS model is obviously worse than the former two because it target too many customers. However, even the worst OLS model is much better than targeting all customers or no customers.

Profits from targeting the top n percent of customers

```
# Function for targeting top n
TopPercent <- function(model_name, top_percent,
                       score, W, spend, margin, cost) {

  comb <- as.data.table(cbind(score,W,spend))
  L=length(top_percent)
  profits_DT=data.table(Model=rep(model_name,times=L),
                        Percent =rep(top_percent,times=1),
                        Profit = rep(0,times=L))

  # Perform profit calculations for all Top Percent
  for(i in 1:nrow(profits_DT)) {
    n = profits_DT[i, Percent]
    threshold = quantile(score, probs = 1-n)

     # Create dummy:1 if in top n of score
    comb$T <- as.numeric(comb$score > threshold)

    #Calcualte observed profit for targeted and non-targeted consumer
    comb = comb[, profit := ifelse(W == 1, comb$spend * margin - cost,
                                   comb$spend*margin)]

    #Both W and T are 1
    mean_profit_T = mean(comb$profit[comb$T+comb$W==2])
    #Both W and T are 0
    mean_profit_NT = mean(comb$profit[comb$T+comb$W==0])
    if (is.nan(mean_profit_T)) mean_profit_T = 0
    if (is.nan(mean_profit_NT)) mean_profit_NT = 0

    profit_T_scale = mean_profit_T*sum(comb$T)
    profit_NT_scale = mean_profit_NT*(length(W)-sum(comb$T))
    scale_factor = 1000/length(W)
    total_profit = (profit_T_scale + profit_NT_scale)*scale_factor
    profits_DT[i, Profit := total_profit]
    }

  return(profits_DT)
}
```

```
top_percent = seq(from = 0, to = 1, by = 0.01)
W = crm_pred$W
spend = crm_pred$outcome_spend

profitsDT_OLS = TopPercent('OLS',top_percent, crm_pred$pred_tau_OLS,
                                        W, spend, m, c)

profitsDT_lasso = TopPercent('LASSO',top_percent,
                                          crm_pred$pred_tau_lasso, W, spend, m, c)

profitsDT_CF = TopPercent('Forest', top_percent,
                                       crm_pred$pred_tau_CF, W, spend, m, c)


rbind(profitsDT_OLS,profitsDT_lasso, profitsDT_CF) %>%
  group_by(Model) %>%
```

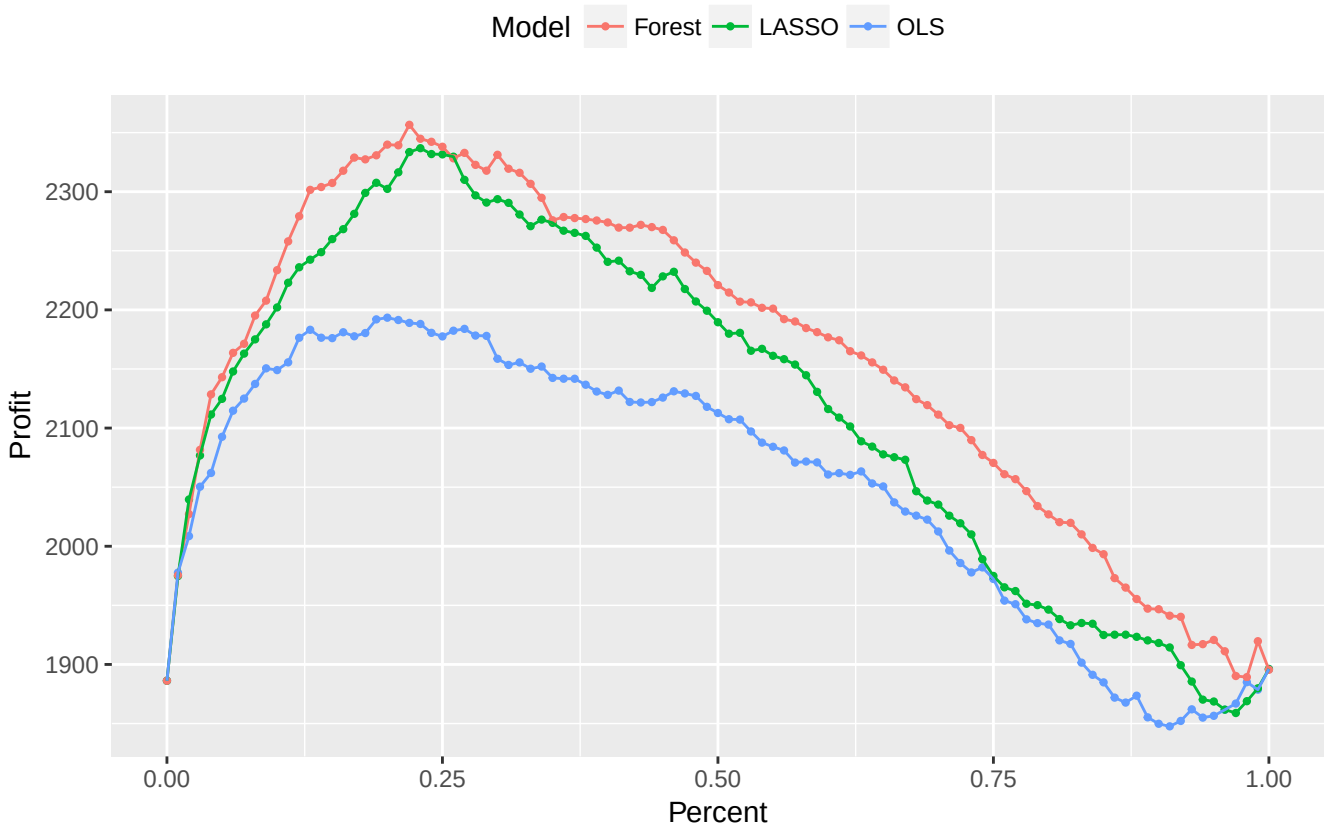
```
  arrange(desc(Profit)) %>%
  slice(1) %>%
  kable(caption = "The optimal profit of the top n stragety")
```

Table 6: The optimal profit of the top n stragety

Model	Percent	Profit
Forest	0.22	2356.613
LASSO	0.23	2336.804
OLS	0.20	2193.396

For the top n stragety, the profit based on the causal model is slightly better than that based on the LASSO model. Their targeting percents are also very similar. The OLS model is still the worst one. Besides,

```
rbind(profitsDT_OLS, profitsDT_lasso, profitsDT_CF) %>%
  ggplot(aes(Percent, Profit, color = Model)) +
  geom_point(size = 0.8) +
  geom_line() +
  theme(legend.position= 'top')
```



The results remain the same in the graph. The causal forest model can almost always get the best profit across all percents. On the contrary, the OLS model almost always performs the worst.

Step 3: How well do the models predict out-of-sample? — Profits and external model validity

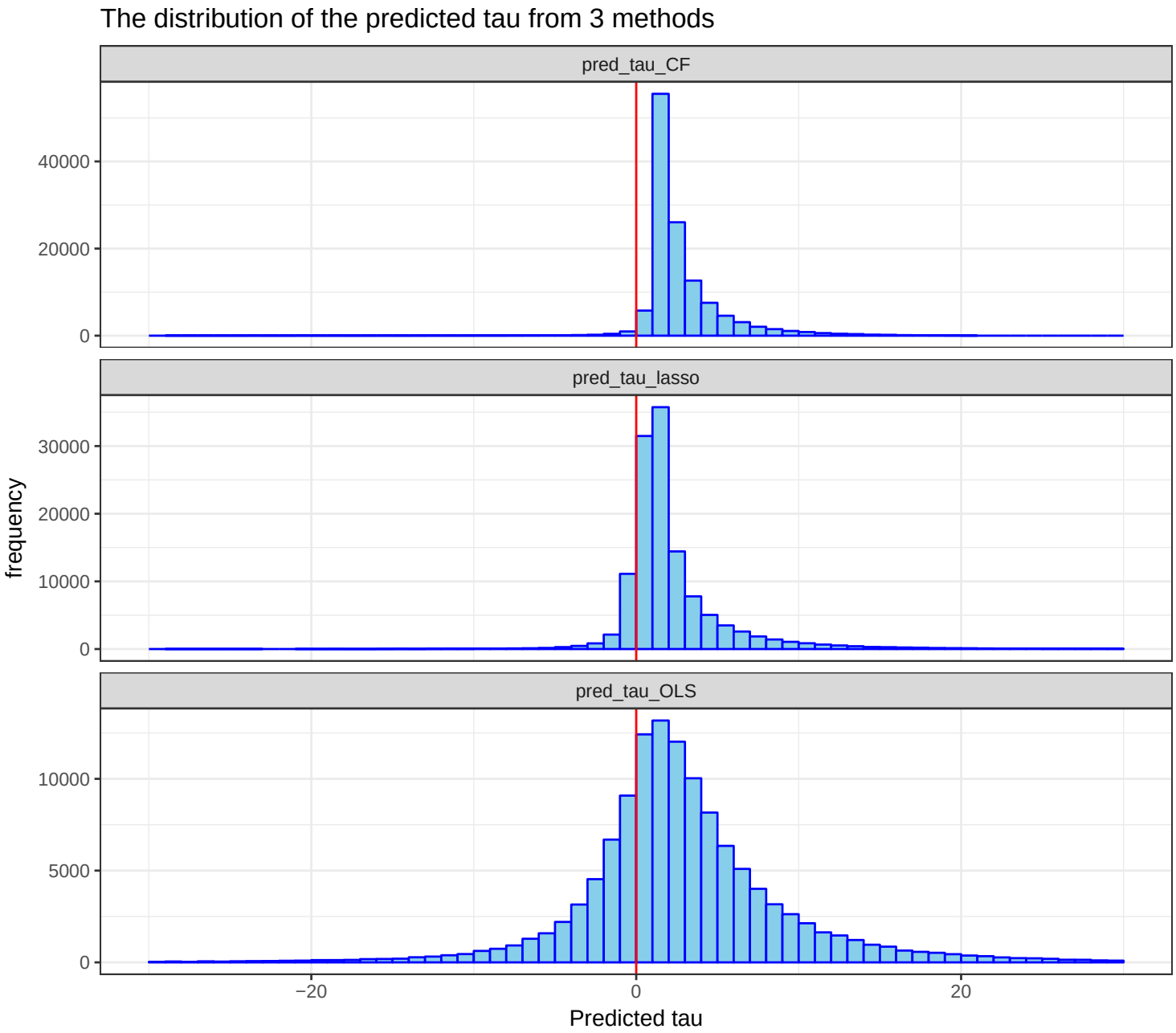
```
load("Randomized-Implementation-Sample-2016.RData")
```

```
crm_2016 = crm_DT[, !large_cor_DT$row, with = FALSE]
crm_2016 = crm_2016[, !c("customer_id"), with = FALSE]
setnames(crm_2016, "mailing_indicator", "W")
test.x16 <- model.matrix(outcome_spend ~ 0 + . * W, data = crm_2016)
```

```
crm_pred16 <- crm_2016 %>%
  select(W, outcome_spend) %>%
  mutate(pred_tau_OLS = interaction_tau(coef(fit_OLS), crm_2016),
         pred_tau_lasso = interaction_tau(fit_lasso, crm_2016),
         pred_tau_CF = predict(fit_CF, crm_2016)) %>%
  as.data.table()
```

```
crm_pred16 %>%
  gather('pred_tau_CF', 'pred_tau_OLS', 'pred_tau_lasso',
         key = method, value = pred) %>%
  ggplot(aes(pred)) +
  geom_histogram(fill = "sky blue", binwidth = 1, boundary = 1, color = 'blue') +
  geom_vline(aes(xintercept=0), color = 'red') +
  facet_wrap(~ method, scales = 'free_y', ncol = 1) +
  xlim(-30, 30) +
  theme_bw() +
  labs(title = 'The distribution of the predicted tau from 3 methods',
       x = 'Predicted tau',
       y = 'frequency')
```

```
## Warning: Removed 1850 rows containing non-finite values (stat_bin).
```




```
lift_OLS16 <- liftTable(crm_pred16$outcome_spend, crm_pred16$pred_tau_OLS, crm_pred16$W, 20)
lift_lasso16 <- liftTable(crm_pred16$outcome_spend, crm_pred16$pred_tau_lasso, crm_pred16$W, 20)
lift_CF16 <- liftTable(crm_pred16$outcome_spend, crm_pred16$pred_tau_CF, crm_pred16$W, 20)

lift_table16 <-  bind_rows("OLS" = lift_OLS16, "LASSO" = lift_lasso16,
                           "Causal Forest" = lift_CF16, .id = "Method")

kable(lift_table16, caption = "Lift table for four methods")
```

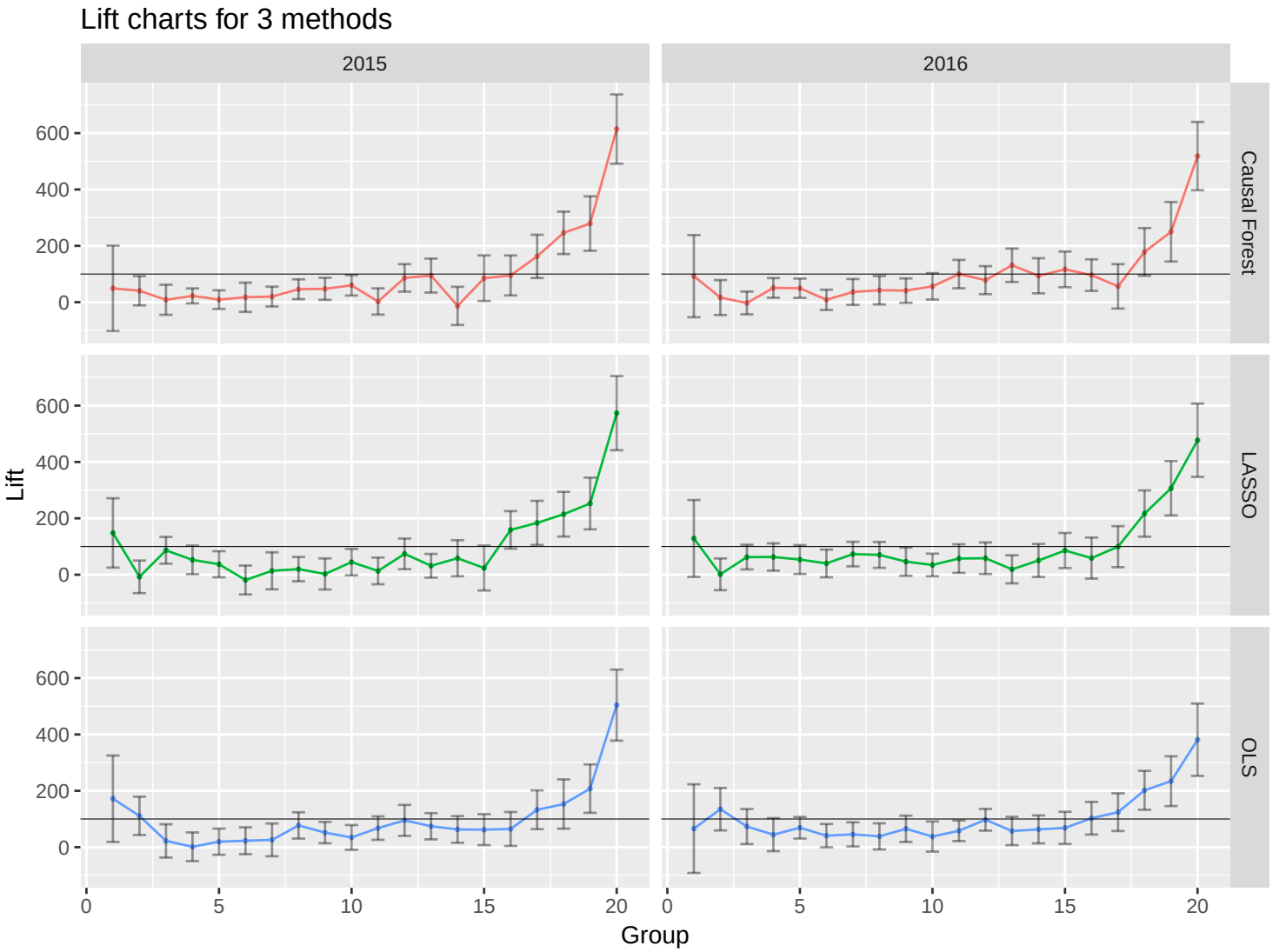
Table 7: Lift table for four methods

Method	estimate	conf.low	conf.high	score_group	lift	lift_lower	lift_upper
OLS	1.8783428	6.362173	-2.6054870	1	65.888165	223.17113	-91.3947955
OLS	3.8429147	5.991335	1.6944939	2	134.801056	210.16297	59.4391463
OLS	2.0873975	3.859919	0.3148759	3	73.221347	135.39754	11.0451586
OLS	1.2690476	2.939535	-0.4014399	4	44.515420	103.11247	-14.0816341
OLS	1.9673940	3.060465	0.8743235	5	69.011888	107.35442	30.6693595
OLS	1.1658516	2.345877	-0.0141734	6	40.895530	82.28823	-0.4971712
OLS	1.2957058	2.518355	0.0730562	7	45.450529	88.33841	2.5626530
OLS	1.0937021	2.415650	-0.2282457	8	38.364683	84.73572	-8.0063593

Method	estimate	conf.low	conf.high	score_group	lift	lift_lower	lift_upper
OLS	1.8584746	3.188257	0.5286921	9	65.191232	111.83710	18.5453642
OLS	1.0765627	2.602408	-0.4492829	10	37.763470	91.28680	-15.7598639
OLS	1.6591405	2.700532	0.6177490	11	58.199025	94.72877	21.6692847
OLS	2.7743722	3.876614	1.6721300	12	97.318921	135.98317	58.6546714
OLS	1.6357717	3.068535	0.2030088	13	57.379301	107.63750	7.1211038
OLS	1.7992383	3.213907	0.3845693	14	63.113353	112.73685	13.4898504
OLS	1.9518846	3.580622	0.3231471	15	68.467851	125.60041	11.3352963
OLS	2.9285941	4.582600	1.2745884	16	102.728687	160.74759	44.7097799
OLS	3.5402361	5.444949	1.6355232	17	124.183752	190.99692	57.3705842
OLS	5.7517833	7.718104	3.7854627	18	201.760002	270.73424	132.7857681
OLS	6.6769520	9.196382	4.1575220	19	234.212898	322.58900	145.8367951
OLS	10.8705473	14.529891	7.2112041	20	381.315067	509.67684	252.9532959
LASSO	3.6663556	7.556826	-0.2241147	1	128.607750	265.07695	-7.8614535
LASSO	0.0416993	1.646292	-1.5628937	2	1.462719	57.74834	-54.8228982
LASSO	1.7797020	3.026222	0.5331822	3	62.428060	106.15325	18.7028690
LASSO	1.7902579	3.171440	0.4090754	4	62.798339	111.24721	14.3494704
LASSO	1.5329004	2.993972	0.0718288	5	53.770801	105.02200	2.5195974
LASSO	1.1410399	2.543997	-0.2619167	6	40.025189	89.23785	-9.1874671
LASSO	2.0848926	3.328315	0.8414698	7	73.133481	116.75004	29.5169226
LASSO	2.0046401	3.312591	0.6966896	8	70.318399	116.19845	24.4383497
LASSO	1.3193438	2.749084	-0.1103963	9	46.279699	96.43186	-3.8724626
LASSO	0.9857342	2.129273	-0.1578047	10	34.577403	74.69025	-5.5354444
LASSO	1.6361395	3.082167	0.1901125	11	57.392202	108.11567	6.6687329
LASSO	1.6629444	3.259453	0.0664355	12	58.332457	114.33450	2.3304107
LASSO	0.5419882	1.961896	-0.8779198	13	19.011761	68.81903	-30.7955084
LASSO	1.4393683	3.107735	-0.2289981	14	50.489896	109.01254	-8.0327521
LASSO	2.4485082	4.226214	0.6708030	15	85.888323	148.24634	23.5303030
LASSO	1.6804184	3.757983	-0.3971457	16	58.945410	131.82182	-13.9310054
LASSO	2.8349097	4.912174	0.7576450	17	99.442444	172.30835	26.5765337
LASSO	6.1844748	8.525350	3.8435993	18	216.937874	299.05068	134.8250703
LASSO	8.7422107	11.489269	5.9951526	19	306.657666	403.01847	210.2968658
LASSO	13.6018063	17.308866	9.8947470	20	477.121669	607.15721	347.0861229
Causal Forest	2.6370481	6.791727	-1.5176311	1	92.501890	238.23897	-53.2351871
Causal Forest	0.4737018	2.240678	-1.2932741	2	16.616424	78.59808	-45.3652326
Causal Forest	-0.0764502	1.075517	-1.2284173	3	-2.681705	37.72679	-43.0901974
Causal Forest	1.4473238	2.445058	0.4495900	4	50.768960	85.76729	15.7706340
Causal Forest	1.4168420	2.395281	0.4384027	5	49.699725	84.02124	15.3782105
Causal Forest	0.2434599	1.268524	-0.7816040	6	8.540043	44.49705	-27.4169612
Causal Forest	1.0379934	2.347404	-0.2714171	7	36.410544	82.34181	-9.5207189
Causal Forest	1.2112017	2.648206	-0.2258022	8	42.486312	92.89327	-7.9206489
Causal Forest	1.1774334	2.411601	-0.0567343	9	41.301795	84.59370	-1.9901136
Causal Forest	1.6024726	2.939940	0.2650048	10	56.211241	103.12669	9.2957912
Causal Forest	2.8468042	4.281542	1.4120668	11	99.859675	150.18713	49.5322189
Causal Forest	2.2316615	3.647729	0.8155937	12	78.281814	127.95438	28.6092459
Causal Forest	3.7370859	5.430844	2.0433278	13	131.088813	190.50215	71.6754766
Causal Forest	2.6726022	4.448404	0.8968003	14	93.749051	156.04030	31.4577983
Causal Forest	3.3230031	5.128118	1.5178878	15	116.563694	179.88320	53.2441896
Causal Forest	2.7416990	4.338178	1.1452202	16	96.172814	152.17380	40.1718250
Causal Forest	1.6056141	3.852244	-0.6410157	17	56.321438	135.12831	-22.4854309
Causal Forest	5.0905037	7.501883	2.6791245	18	178.563756	263.14967	93.9778398
Causal Forest	7.1290262	10.139337	4.1187152	19	250.070676	355.66582	144.4755368
Causal Forest	14.7851722	18.239373	11.3309718	20	518.631562	639.79737	397.4657511

```
bind_rows('2015' = lift_table, '2016' = lift_table16, .id = "year") %>%
  ggplot(aes(score_group, lift, color = Method)) +
  geom_line(show.legend = FALSE) +
  geom_point(size = 0.5, show.legend = FALSE) +
  geom_errorbar(aes(ymin=lift_lower, ymax=lift_upper),
                width = 0.5, color = "black", alpha = 0.4) +
```

```
  geom_hline(aes(yintercept = 100), size = 0.2) +
  facet_grid(Method ~ year) +
  labs(title = "Lift charts for 3 methods",
       x = "Group",
       y = "Lift")
```



For the lift table and chart, although the lift level in the year 2016 is lower than that in the validation set in 2015, we can still get a generally increasing chart. It partly proves the predictive power of the models.

```
opt_profit_OLS16 <- opt_strategy(crm_pred16$outcome_spend, crm_pred16$pred_tau_OLS,
                                 crm_pred16$W, c, m)
opt_profit_lasso16 <- opt_strategy(crm_pred16$outcome_spend, crm_pred16$pred_tau_lasso,
                                   crm_pred16$W, c, m)
opt_profit_CF16 <- opt_strategy(crm_pred16$outcome_spend, crm_pred16$pred_tau_CF,
                                crm_pred16$W, c, m)
```

```
scale = 1000000 / 1000

OLS_opt16 <- mean(opt_profit_OLS16$profit) * scale
lasso_opt16 <- mean(opt_profit_lasso16$profit) * scale
CF_opt16 <- mean(opt_profit_CF16$profit) * scale
OLS_per16 <- mean(opt_profit_OLS16$t)
lasso_per16 <- mean(opt_profit_lasso16$t)
CF_per16 <- mean(opt_profit_CF16$t)

all16 <- mean(crm_pred16[W == 1]$outcome_spend * m - c) * scale
```

```
none16 <- mean(crm_pred16[W == 0]$outcome_spend * m) * scale
```

```
cat('The scale of the profit is 1 thousand dollors per 1 million customers:')
```

```
## The scale of the profit is 1 thousand dollors per 1 million customers:
```

```
cat('\n\nOLS:', 'percent is          ', OLS_per16, ', profit is', OLS_opt16)
```

```
##
##
## OLS: percent is           0.420168 , profit is 1862.352
```

```
cat('\nLASSO:', 'percent is        ', lasso_per16, ', profit is', lasso_opt16)
```

```
##
## LASSO: percent is         0.218848 , profit is 1999.916
```

```
cat('\nCausal Forest:', 'percent is', CF_per16, ', profit is ', CF_opt16)
```

```
##
## Causal Forest: percent is 0.27744 , profit is  1956.982
```

```
cat('\nThe profit for targeting all is              ', all16)
```

```
##
## The profit for targeting all is                1620.326
```

```
cat('\nThe profit for targeting none is              ', none16)
```

```
##
## The profit for targeting none is               1683.814
```

In the 2016 dataset, the LASSO is a little better than the causal forest model. And the profit results from LASSO and causal forest are respectively lower than those in the 2015 validation data.

```
W16 = crm_pred16$W
spend16 = crm_pred16$outcome_spend

profits_OLS16 = TopPercent('OLS',top_percent, crm_pred16$pred_tau_OLS,
                           W16, spend16, m, c)

profits_lasso16 = TopPercent('LASSO',top_percent,
                             crm_pred16$pred_tau_lasso, W16, spend16, m, c)

profits_CF16 = TopPercent('Forest', top_percent,
                          crm_pred16$pred_tau_CF, W16, spend16, m, c)
```

```
rbind(profits_OLS16, profits_lasso16, profits_CF16) %>%
  group_by(Model) %>%
  arrange(desc(Profit)) %>%
  slice(1) %>%
  kable(caption = "The optimal profit of the top n stragety")
```

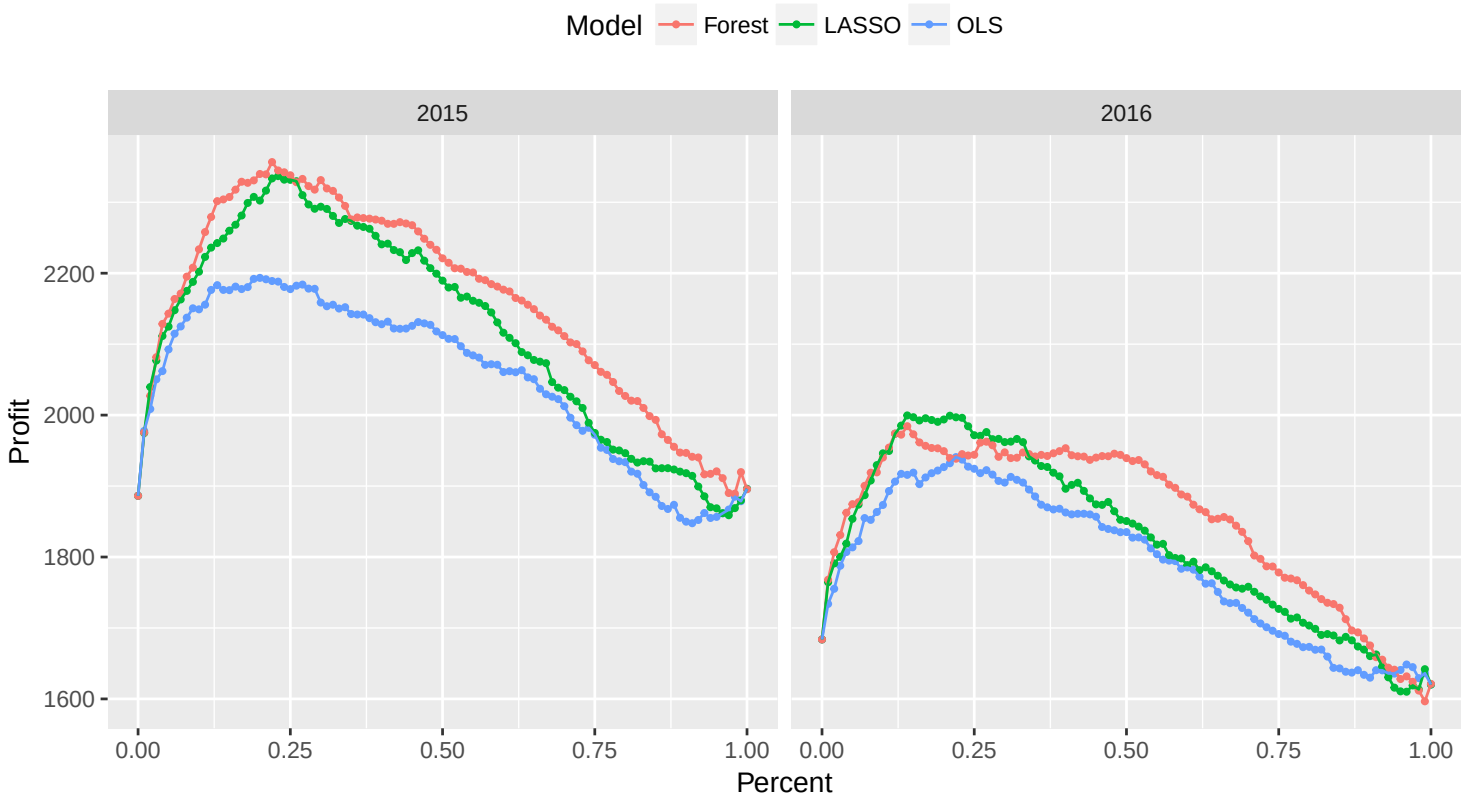
Table 8: The optimal profit of the top n stragety

Model	Percent	Profit
Forest	0.14	1984.377
LASSO	0.14	1999.543
OLS	0.22	1940.823

For the top n stratety, the highest profits that the causal forest and LASSO models can get are very similar (LASSO wins slightly), and the top percents they will target are the same. The OLS model is still the worst.

```
bind_rows('2015' = rbind(profitsDT_OLS, profitsDT_lasso, profitsDT_CF),
          '2016' = rbind(profits_OLS16, profits_lasso16, profits_CF16),
```

```
          .id = 'year') %>%
  ggplot(aes(Percent, Profit, color = Model)) +
  geom_point(size = 0.8) +
  geom_line() +
  theme(legend.position= 'top') +
  facet_wrap(~ year)
```



When we look at the curves, it is obviously that the profits in 2016 are always lower than those in the 2015 validation data. Since the profits of targeting all or no customers in 2016 are also lower than those in 2015, the lower profits may partly caused by non-model factors. Besides, although there is a gap of profits between the 2015 validation data and the 2016 data, their increasing and decreasing trends are not identical but share some similarities, and thus we are able to find a optimal profit point.

All in all, we can really predict customer behavior one year after the estimation data were collected, but the models would not be as powerful as those in the 2015 validation data, and the best model would also be different. Besides, although the causal forest model outperforms the LASSO model in the 2015 validation data, they look similar, in terms of optimal profit, in the 2016 data.

Optional analysis (bonus)

Elastic Net

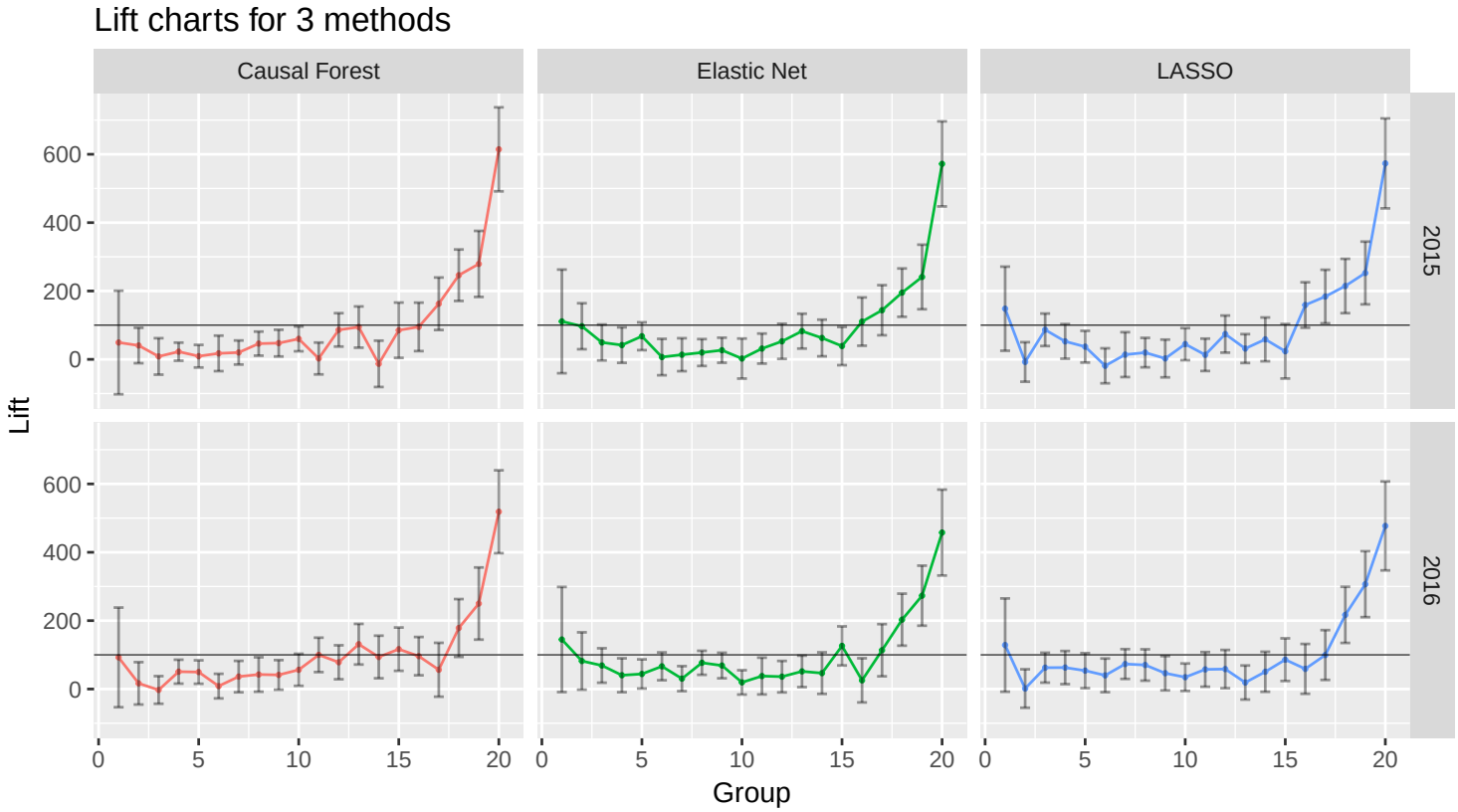
```
set.seed(2001)
alpha_seq =seq(0, 1, by = 0.05)
L =length(alpha_seq)
rmse_DT = data.table(alpha = alpha_seq,
                     mean_cv_error = rep(0, L))
folds = sample(1:10, nrow(training_DT), replace = TRUE)

for(i in 1:L) {
  cv_i = cv.glmnet(x = train.x, y = train.y, alpha = rmse_DT[i, alpha], foldid = folds)
  rmse_DT[i, mean_cv_error:= min(cv_i$cvm)]
}

index_min = which.min(rmse_DT$mean_cv_error)
```

```
opt_alpha = rmse_DT[index_min, alpha]

fit_elnet = cv.glmnet(x = train.x, y = train.y, alpha = opt_alpha)
```

```
coef_elnet = as.numeric(coef(fit_elnet, s = "lambda.min"))
```

```
crm_pred[, pred_elnet_tau := interaction_tau(coef_elnet, validation_DT)]
crm_pred16[, pred_elnet_tau := interaction_tau(coef_elnet, crm_2016)]
```

```
lift_EN16 <- liftTable(crm_pred16$outcome_spend, crm_pred16$pred_elnet_tau, crm_pred16$W, 20)
lift_EN <- liftTable(crm_pred$outcome_spend, crm_pred$pred_elnet_tau, crm_pred$W, 20)
lift_tableEN16 <-  bind_rows("Elastic Net" = lift_EN16, "LASSO" = lift_lasso16,
                             "Causal Forest" = lift_CF16, .id = "Method")
lift_tableEN15 <-  bind_rows("Elastic Net" = lift_EN, "LASSO" = lift_lasso,
                             "Causal Forest" = lift_CF, .id = "Method")
bind_rows('2015' = lift_tableEN15, '2016' = lift_tableEN16, .id = "year") %>%
  ggplot(aes(score_group, lift, color = Method)) +
  geom_line(show.legend = FALSE) +
  geom_point(size = 0.5, show.legend = FALSE) +
  geom_errorbar(aes(ymin=lift_lower, ymax=lift_upper),
                width = 0.5, color = "black", alpha = 0.4) +
  geom_hline(aes(yintercept = 100), size = 0.2) +
  facet_grid(year ~ Method) +
  labs(title = "Lift charts for 3 methods",
       x = "Group",
       y = "Lift")
```



For the lift charts, the elastic net model is similar to the LASSO model in the 2015 validation data and worse than it in the 2016 data. Besides, in both datasets, it performs worse than the causal forest model.

```
opt_profit_EN15 <- opt_strategy(crm_pred$outcome_spend, crm_pred$pred_elnet_tau,
                                crm_pred$W, c, m)
opt_profit_EN16 <- opt_strategy(crm_pred16$outcome_spend, crm_pred16$pred_elnet_tau,
                                crm_pred16$W, c, m)
```

```
EN_opt15 <- mean(opt_profit_EN15$profit) * scale
EN_opt16 <- mean(opt_profit_EN16$profit) * scale
EN15_per <- mean(opt_profit_EN15$t)
EN16_per <- mean(opt_profit_EN16$t)

cat('Elastic Net 2015:', 'percent is', EN15_per, ', profit is', EN_opt15)
```

```
## Elastic Net 2015: percent is 0.2669106 , profit is 2254.722
```

```
cat('\nElastic Net 2016:', 'percent is', EN16_per, ', profit is', EN_opt16)
```

```
##
## Elastic Net 2016: percent is 0.278328 , profit is 1940.98
```

In this caculation, the elastic net model is worse than the LASSO and causal forest models.

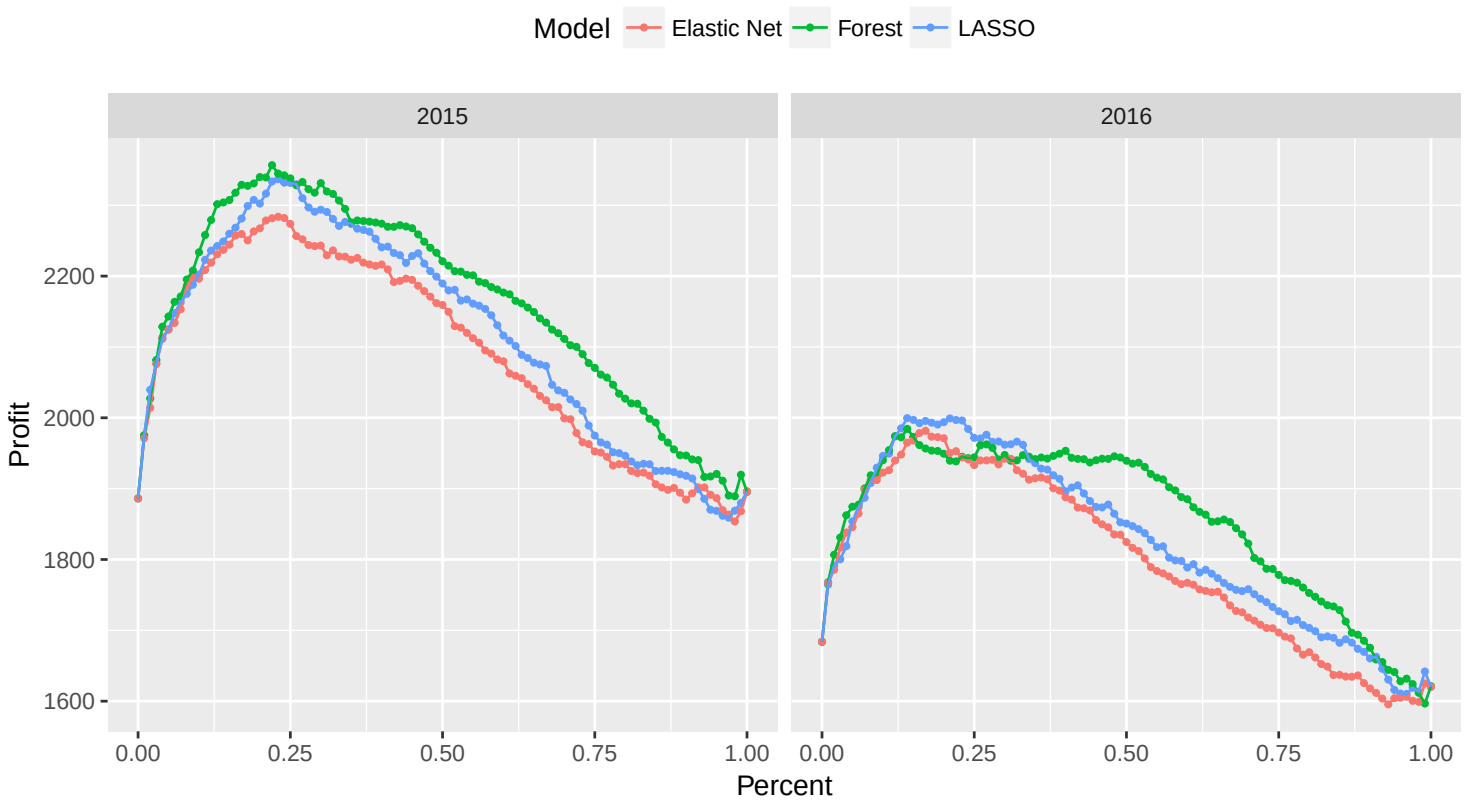
```
profits_EN16 = TopPercent('Elastic Net', top_percent,
                          crm_pred16$pred_elnet_tau, W16, spend16, m, c)

profits_EN15 = TopPercent('Elastic Net', top_percent,
                          crm_pred$pred_elnet_tau, W, spend, m, c)
```

```
bind_rows('2015' = profits_EN15, '2016' = profits_EN16, .id = 'year') %>%
  group_by(year) %>%
  arrange(desc(Profit)) %>%
  slice(1) %>%
  kable(caption = "The optimal profit of the top n stragety")
```

Table 9: The optimal profit of the top n stragety

year	Model	Percent	Profit
2015	Elastic Net	0.23	2283.787
2016	Elastic Net	0.17	1981.631

```
bind_rows('2015' = rbind(profitsDT_lasso, profitsDT_CF, profits_EN15),
          '2016' = rbind(profits_lasso16, profits_CF16, profits_EN16),
          .id = 'year') %>%
  ggplot(aes(Percent, Profit, color = Model)) +
  geom_point(size = 0.8) +
  geom_line() +
  theme(legend.position= 'top') +
  facet_wrap(~ year)
```



For the top n strategy, the elastic model does also not outperform the LASSO and causal forest models in both 2015 and 2016 data.

Traditional Method

I build some models based on the traditional methods. Since causal forest method only generates the coefficient about the CATE, I use random forest model as its counterpart.

```
training_T <- filter(training_DT, W == 1)
train.xt <- model.matrix(outcome_spend ~ 0 + . - W, data = training_T)
train.yt <- training_T$outcome_spend
test.xt <- model.matrix(outcome_spend ~ 0 + . - W, data = validation_DT)
test.xt16 <- model.matrix(outcome_spend ~ 0 + . - W, data = crm_2016)
```

```
fit_OLSt <- lm(outcome_spend ~ . - W, data = training_T)
```

```
set.seed(2001)
cv.lassot = cv.glmnet(train.xt, train.yt, alpha = 1.0)
lasso.lambdat = cv.lassot$lambda.min
fit_lassot = as.numeric(coef(cv.lassot, s = "lambda.min"))
```

```
set.seed(2001)
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))
folds = sample(1:10, nrow(training_T), replace = TRUE)

for(i in 1:L) {
  cv_i = cv.glmnet(x = train.xt, y = train.yt, alpha = rmse_DT[i, alpha], foldid = folds)
  rmse_DT[i, mean_cv_error:= min(cv_i$cvm)]
}

index_mint = which.min(rmse_DT$mean_cv_error)
opt_alphat = rmse_DT[index_mint, alpha]
```

```
fit_elnett = cv.glmnet(x = train.xt, y = train.yt, alpha = opt_alphat)
```

```
set.seed(2001)
fit.forest = ranger(outcome_spend ~. - W, data=training_T,
                    num.trees = 1000, importance="impurity", verbose=T)
```

```
spend_16 <- crm_pred16 %>%
  mutate(spend_OLS = predict(fit_OLSt, crm_2016),
         spend_lasso = as.vector(predict(cv.lassot, newx = test.xt16, s = "lambda.min")),
         spend_EN = as.vector(predict(fit_elnett, newx = test.xt16, s = "lambda.min")),
         spend_RF = predict(fit.forest, crm_2016)$prediction)
```

```
## Predicting.. Progress: 57%. Estimated remaining time: 23 seconds.
```

```
spend_15 <- crm_pred %>%
  mutate(spend_OLS = predict(fit_OLSt, validation_DT),
         spend_lasso = as.vector(predict(cv.lassot, newx = test.xt, s = "lambda.min")),
         spend_EN = as.vector(predict(fit_elnett, newx = test.xt, s = "lambda.min")),
         spend_RF = predict(fit.forest, validation_DT)$prediction)
```

```
## Predicting.. Progress: 71%. Estimated remaining time: 12 seconds.
```

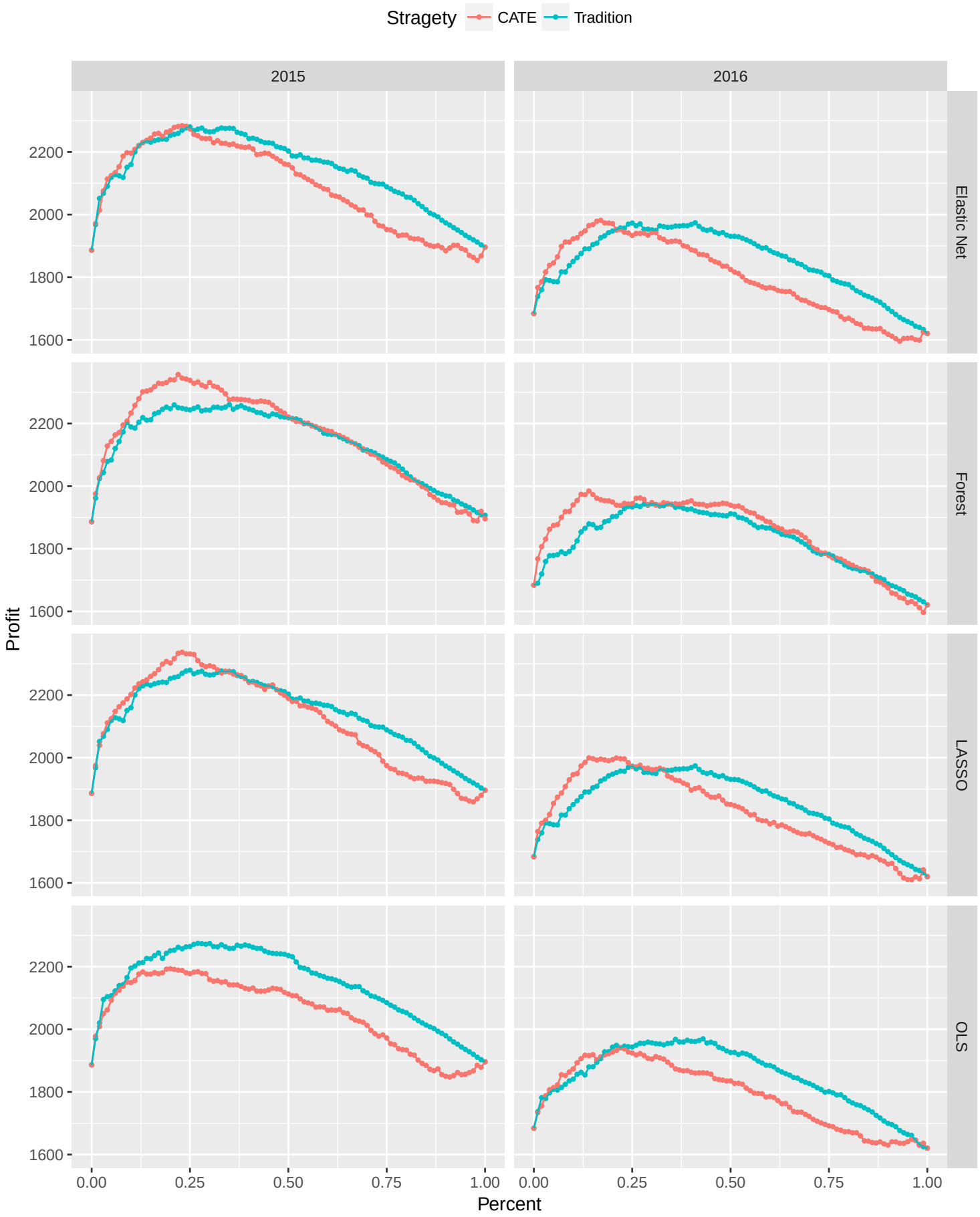
```
profits15_OLS = TopPercent('OLS', top_percent, spend_15$spend_OLS,
                           W, spend, m, c)
profits15_lasso = TopPercent('LASSO', top_percent,
                             spend_15$spend_lasso, W, spend, m, c)
profits15_EN = TopPercent('Elastic Net', top_percent,
                          spend_15$spend_EN, W, spend, m, c)
profits15_RF = TopPercent('Forest', top_percent,
                          spend_15$spend_RF, W, spend, m, c)

profits16_OLS = TopPercent('OLS', top_percent, spend_16$spend_OLS,
                           W16, spend16, m, c)
profits16_lasso = TopPercent('LASSO', top_percent,
                             spend_16$spend_lasso, W16, spend16, m, c)
profits16_EN = TopPercent('Elastic Net', top_percent,
                          spend_16$spend_EN, W16, spend16, m, c)
profits16_RF = TopPercent('Forest', top_percent,
                          spend_16$spend_RF, W16, spend16, m, c)

CATE <- bind_rows('2015' = rbind(profitsDT_OLS, profitsDT_lasso, profitsDT_CF, profits_EN15),
                  '2016' = rbind(profits_OLS16, profits_lasso16, profits_CF16, profits_EN16),
                  .id = 'year')

tra <- bind_rows('2015' = rbind(profits15_OLS, profits15_lasso, profits15_RF, profits15_EN),
                 '2016' = rbind(profits16_OLS, profits16_lasso, profits16_RF, profits16_EN),
                 .id = 'year')

bind_rows("Tradition" = tra, "CATE" = CATE, .id = 'Stragety') %>%
  group_by(Model, Stragety, year) %>%
  arrange(desc(Profit)) %>%
  slice(1) %>%
  arrange(desc(Profit)) %>%
  kable(caption = "The optimal profit of the top n stragety")
```

Table 10: The optimal profit of the top n stragety

Stragety	year	Model	Percent	Profit
CATE	2015	Forest	0.22	2356.613
CATE	2015	LASSO	0.23	2336.804
CATE	2015	Elastic Net	0.23	2283.787
Tradition	2015	Elastic Net	0.25	2279.921

Stragety	year	Model	Percent	Profit
Tradition	2015	LASSO	0.25	2279.921
Tradition	2015	OLS	0.27	2274.584
Tradition	2015	Forest	0.35	2260.476
CATE	2015	OLS	0.20	2193.396
CATE	2016	LASSO	0.14	1999.543
CATE	2016	Forest	0.14	1984.377
CATE	2016	Elastic Net	0.17	1981.631
Tradition	2016	Elastic Net	0.41	1973.968
Tradition	2016	LASSO	0.41	1973.968
Tradition	2016	OLS	0.43	1969.451
Tradition	2016	Forest	0.30	1943.951
CATE	2016	OLS	0.22	1940.823

```
bind_rows("Tradition" = tra, "CATE" = CATE, .id = 'Stragety') %>%
  ggplot(aes(Percent, Profit, color = Stragety)) +
  geom_point(size = 0.8) +
  geom_line() +
  theme(legend.position= 'top') +
  facet_grid(Model ~ year)
```



I compare the profits computed by models in traditional and CATE methods. Since the causal forest model does not provide the predicted spend, I build a random forest model as its counterpart.

Except the OLS models, any model of the CATE method outperforms any other model in the traditional method, respectively in the 2015 validation data and in the 2016 data. Therefore, the CATE method can help us get higher profits as long as we don't use some notoriously bad models in this field, such as OLS models.

A brief summary

The detailed explanation of the findings and insigts have been written above, I only briefly list them here.

1. In the 2015 validation data, the causal forest model outperforms OLS, LASSO and elastic net models, and can achieve the highest profit no matter we use top n or $\hat{\Pi}(T)$ to caculate it.
2. Using $\hat{\Pi}(T)$ and top n method to predict profit can get similar results.
3. The models can still predict the profits of the 2016 data. However, the profits they predict are lower than that in the 2015 validation data. It might be because of non-model factor since the profits of targeting all and no customers in the 2016 also become lower. However, we cannot deny the reasoning that the model build in one sample might lose some predictive power.
4. Besides, the best model in the validation data and that in the 2016 data is different also signifies that the results will be different in out of sample data, and hence we'd better use this kind of data to test our model.
5. The CATE method can get higher profit compared to the traditional method.