CRM and Machine Learning

Giovanni Compian

Contents

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

1 Data

The data that we use is a development sample that includes records on 250,000 randomly selected customers from the data base of a company that sells kitchenware, housewares, and specialty food products. The data include a high-dimensional set of more than 150 customer features. The random sample represents only a small percentage of the whole data base of the company.

Customers are identified by a unique `customer_id`. The targeting status of a customers as of October 10, 2017 is indicated by `mailing_indicator`. The total dollar spend (online, phone and mail order, in-store) in the subsequent 15 week period is captured by `outcome_spend`. All other variables are customer summary data based on the whole transaction history of each customer. These summary data were retrieved one week before the catalog mailing.

The customer features are largely self-explanatory. For example, `orders_online_1yr` indicates the number of orders placed during the last year. Variables such as `spend_m_1yr` indicate spending in a specific department, labeled `m`. Due to privacy reasons, what this department exactly is cannot be revealed. Similarly, no details on specific product types (e.g. `clicks_product_type_104`) can be disclosed. Also, to preserve confidentiality, some variables had to be scaled. Hence, you may see an order count such as 2.4, even though originally orders can only be 0, 1, 2, ... Please note that the scaling has no impact on the predictive power of the statistical models that you estimate. Furthermore, the restricted interpretability of some of the variables has no bearing on our analysis. Ultimately, we use the features to predict spending levels or incremental spending levels, but we do not interpret these features as causal. In particular, we cannot *manipulate* variables such as `orders_online_1yr`, which explains why the corresponding estimates have no causal interpretation.

Load the `crm_DT` data.table.

```
data_folder    = "Data"
path = "Customer-Development-2017.RData"
load(paste0(data_folder, "/", path))
```

I recommend renaming the `mailing_indicator` to make it clear that this variable represents a targeting *treatment*, W_i.

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

```
# Check Available columns
names(crm_DT)
```

We split the sample into a 50% training and 50% validation sample. To ensure that we all have the same training and validation sample, use the following seed:

```
set.seed(1999)
# Indicate which rows are training and which are testing
crm_DT[, training_sample := rbinom(nrow(crm_DT), 1, 0.5)]
```

2 Data inspection [15 points]

Summarize and describe some key aspects of the `outcome_spend` variable. In particular, what is the purchase incidence (i.e., what fraction of customers make a purchase), what is the distribution of dollar spending, and what is the conditional distribution of dollar spending given that a customer made a purchase?

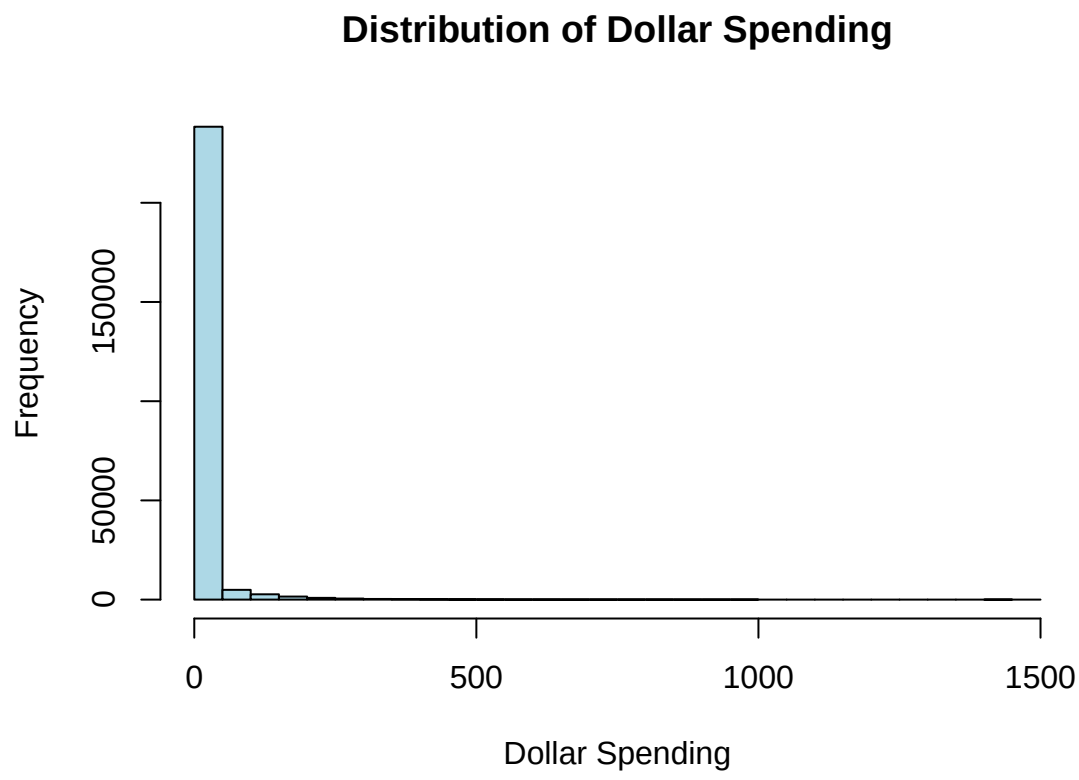
```
# Summarize Key Aspects of the `outcome_spend`
# 1. Purchase incidence
# - % of customers make a purchase
# - Proportion of cust that has > 0 outcome spend
# `outcome_spend` > 0 creates a logical vector where each entry is TRUE, else FALSE
purchase_incidence <- mean(crm_DT$outcome_spend > 0)
print(purchase_incidence)
```

```
[1] 0.06208
```

```
# 2. Distribution of dollar spending
# - how much customers are spending overall, including both customers who made purchases and those who
summary(crm_DT$outcome_spend)  # Provides summary statistics
```

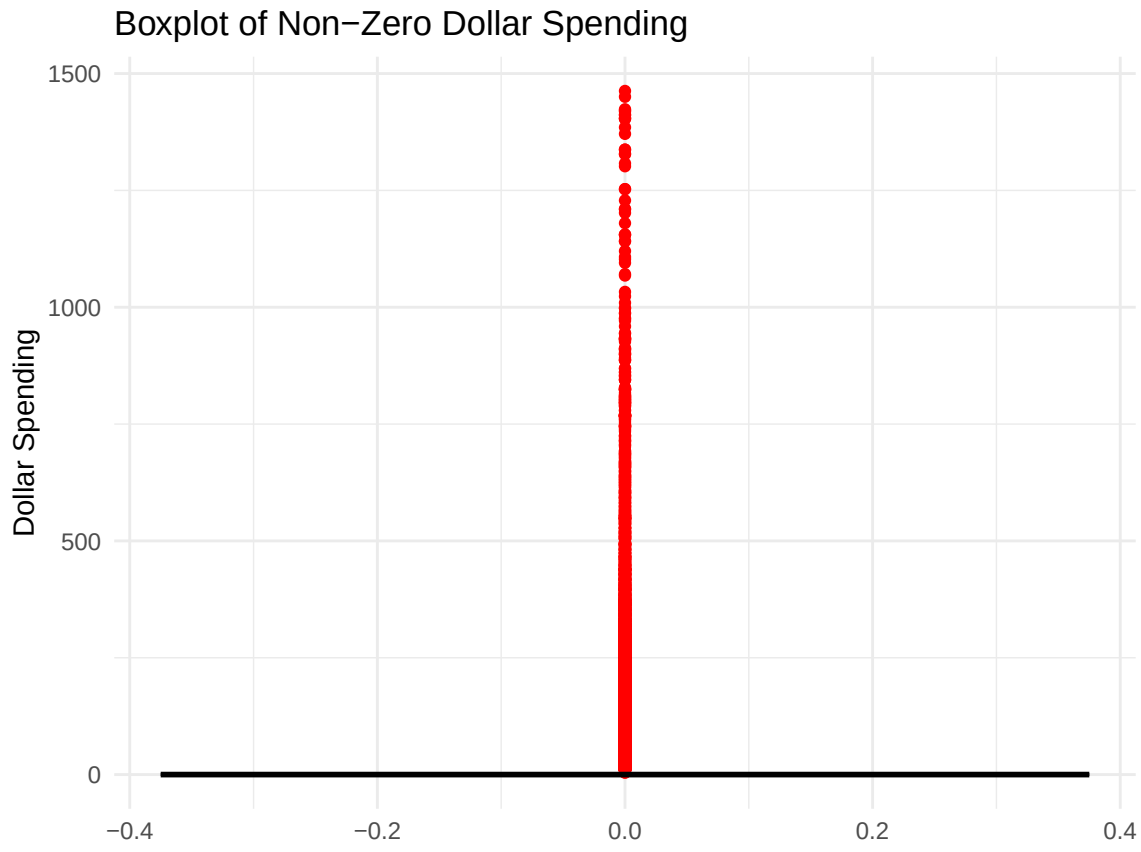
```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max.
   0.00    0.00    0.00    7.98    0.00 1462.84
```

```
hist(crm_DT$outcome_spend,
     main = "Distribution of Dollar Spending",  # Title of the plot
     xlab = "Dollar Spending",                  # Label for x-axis
     ylab = "Frequency",                        # Label for y-axis
     col = "lightblue",                         # Fill color
     breaks = 30)                               # Number of # Boxplot of outcome_spend
```




```
ggplot(crm_DT, aes(y = outcome_spend)) +
  geom_boxplot(
```

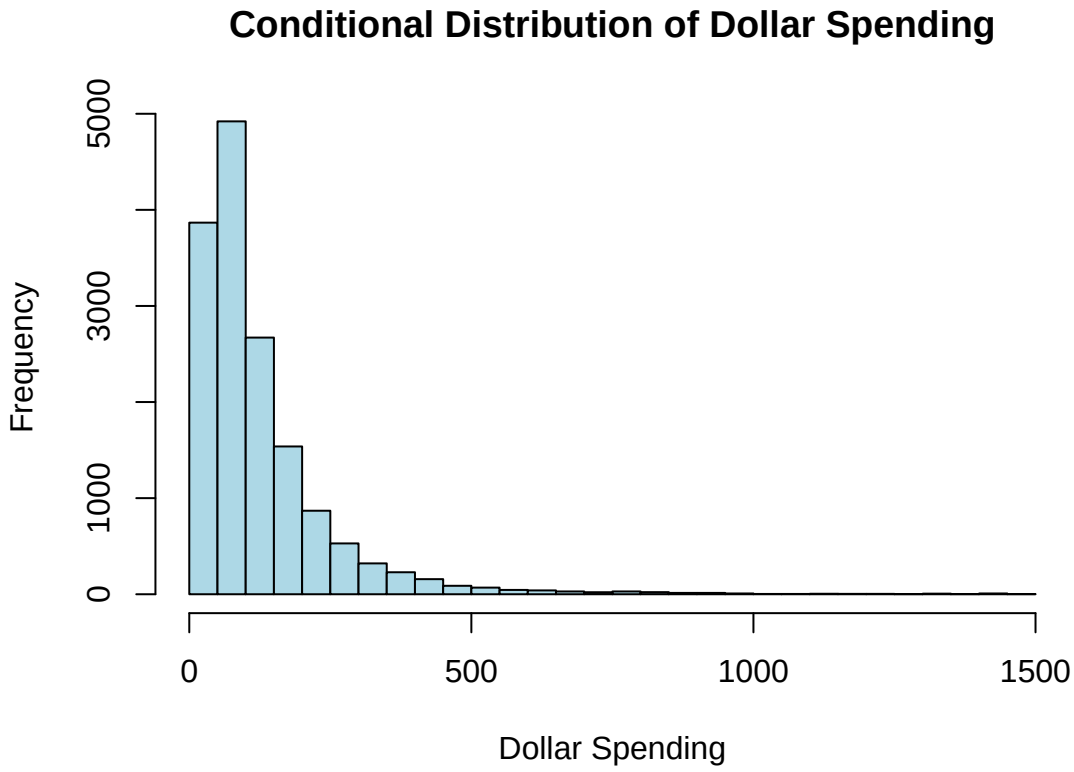
```
    fill = "lightgreen",
    color = "black",
    outlier.color = "red"
  ) +
  labs(
    title = "Boxplot of Non-Zero Dollar Spending",
    y = "Dollar Spending"
  ) +
  theme_minimal()
```



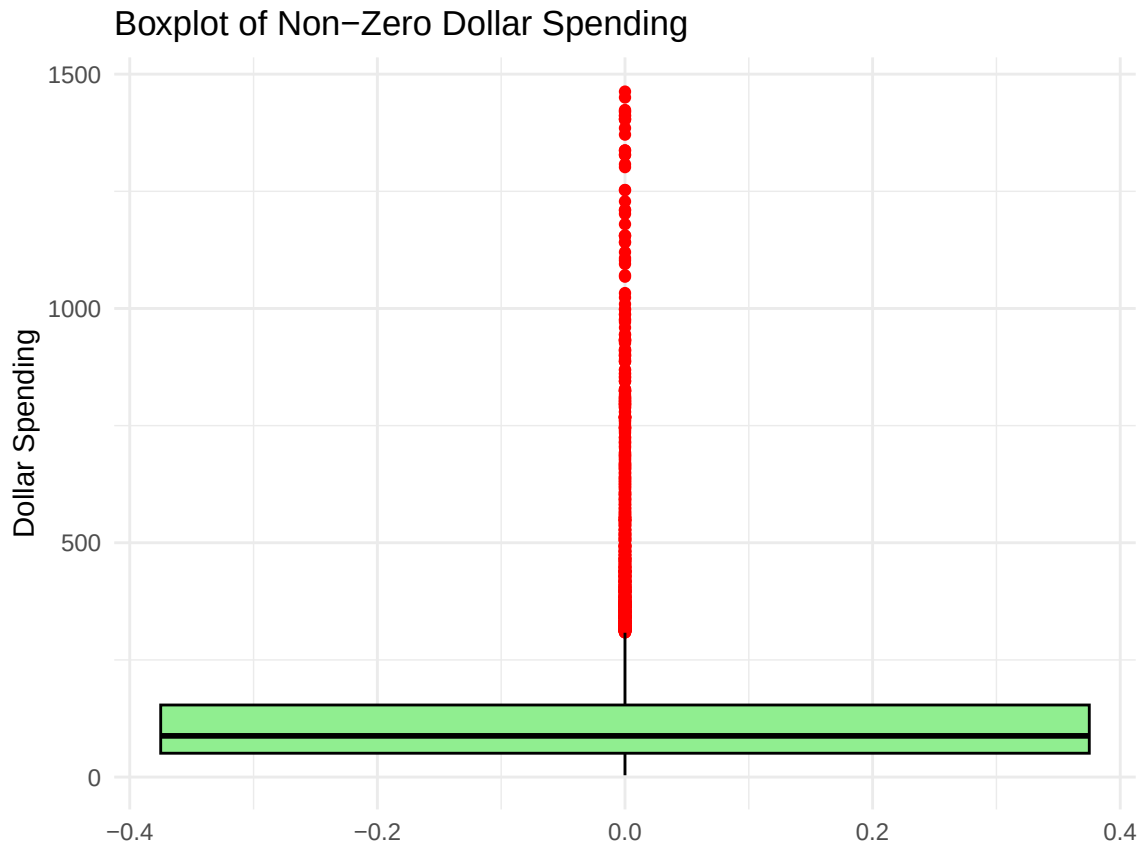

```
# 3. Conditional distribution of dollar spending
# - Focus on people who made the purchase outcome_spend > 0
spend_given_purchase <- crm_DT[outcome_spend > 0]
summary(spend_given_purchase$outcome_spend)  # Summary statistics for non-zero spending
```

```
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
   3.95   50.90   87.94  128.55  153.85 1462.84 
```

```
hist(spend_given_purchase$outcome_spend,
     main = "Conditional Distribution of Dollar Spending",  # Title of the plot
     xlab = "Dollar Spending",                 # Label for x-axis
     ylab = "Frequency",                       # Label for y-axis
     col = "lightblue",                        # Fill color
     breaks = 30)       # Histogram for conditional spending
```




```
ggplot(spend_given_purchase, aes(y = outcome_spend)) +
  geom_boxplot(
    fill = "lightgreen",
    color = "black",
    outlier.color = "red"
  ) +
  labs(
    title = "Boxplot of Non-Zero Dollar Spending",
    y = "Dollar Spending"
  ) +
  theme_minimal()
```




```
outlier_count <- sum(crm_DT$outcome_spend > 153.85)
outlier_count
```

```
[1] 3861
```

Analysis - Only 6.2% of customers made purchase - Out of those purchased, the distribution is average $128.55 per person, and the majority of people spend less than the mean. However, there are substantial amount of outlier. 3861 outlier customers out of 15520 customers, who purchased from $153 to $1462 per person.

3 Data pre-processing

Data sets with a large number of inputs (features) often contain highly correlated variables. The presence of such variables is not necessarily a problem if we employ an estimation method that uses regularization. However, if two variables are almost perfectly correlated then one of them captures virtually all the information contained in both variables. Also, OLS (or logistic regression) without regularization will be unfeasible with perfectly or near perfectly correlated inputs. Hence, it is helpful to eliminate some highly correlated variables from the data set.

Here is a helpful method to visualize the degree of correlation among all inputs in the data. Install the package `corrplot`. Then calculate a matrix of correlation coefficients among all inputs:

```
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. There is a huge amount of information in each graph, hence I recommend zooming in! Please see the `corrplot` documentation for a description of all the options.

```
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

# 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]
```

In the first statement above, we set the correlations in the upper triangle and on the diagonal of the correlation matrix to `NA`. 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.

Then we create a new data table, `cor_DT`, that includes all pairs of features and the respective correlation coefficient. Make sure you understand how this table is computed in the four lines of code above!

Now find all correlations larger than 0.95 in absolute value. Inspect these correlations, and then eliminate one of the virtually redundant variables in each highly correlated pair from the data set (to ensure that we end up with the same data, 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
in_database_months	acquisition_days_since	0.9997
emails_days_1yr	emails_days_2yr	0.9604
emailview_3m	emailview_6m	0.9505

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

Note that this last step eliminates from `crm_DT` all variables listed in `large_cor_DT$row`.

4 Predictive model estimation [25 points]

Use the training sample to estimate the conditional expectation of dollar spending, based on all available customer information (features). In particular, following the common approach in the industry that we have discussed in class, estimate the model only for customers who were targeted, such that $W_i = 1$. Hence, we estimate a model that predicts expected dollar spending, conditional on all customer features and conditional on being targeted. This is the same approach we have followed, e.g., in the JCPenney example in class.

Estimate and compare the following models:

1. OLS
2. LASSO
3. Elastic net

```
# Create training and testing database
training_data <- crm_DT[training_sample==1]
testing_data <- crm_DT[training_sample==0]
dim(training_data)
```

```
[1] 125030    150
```

```
dim(testing_data)
```

```
[1] 124970    150
```

```
# Conditional Expectation
# Expected value (mean) of a variable Y given some conditions ie. Targered (W)
# E[Y | features, W=1]
# STEPS:
# 1. Filter the training sample for W=1
# 2. Estimate a Model to Predict Spending
# 3. Follow the "Common Approach" Discussed in Class
# 4. Output Expected Dollar Spending

# 1. Filter the training sample for W=1
dim(training_data)
```

```
[1] 125030    150
```

```
training_data_targeted <- training_data[training_data$W == 1, ]
testing_data_targeted <- testing_data[testing_data$W == 1, ]
fit_OLS = lm(outcome_spend ~ ., data = training_data_targeted)
summary_OLS = summary(fit_OLS)
summary_OLS
```

```
Call:
lm(formula = outcome_spend ~ ., data = training_data_targeted)

Residuals:
    Min      1Q  Median      3Q     Max
-452.84   -9.17   -3.45   -0.53 1328.51

Coefficients: (2 not defined because of singularities)
                                 Estimate Std. Error t value Pr(>|t|)
(Intercept)                    -1.114e+01  1.031e+01  -1.080 0.280254
customer_type_L                 1.037e+00  4.704e-01   2.204 0.027558 *
```

```
customer_type_C                  -1.074e+00  4.685e-01  -2.292 0.021887 *
clicks_product_type_504_3m        4.549e-03  2.458e-02   0.185 0.853187
clicks_product_type_112_6m        2.671e-02  2.552e-02   1.047 0.295277
clicks_product_type_301_1m        7.642e-01  2.123e-01   3.599 0.000320 ***
clicks_product_type_001_12m       1.488e-02  3.176e-02   0.469 0.639360
clicks_product_type_201_1m       -5.405e-02  8.303e-02  -0.651 0.515073
clicks_product_type_201_6m        5.104e-02  2.359e-02   2.164 0.030503 *
web_activity_3m                   1.568e-02  5.611e-03   2.794 0.005204 **
clickthrough_1m                   5.811e-01  3.173e-01   1.831 0.067044 .
emailview_1m                      2.341e-01  5.397e-02   4.338 1.44e-05 ***
orders_online_1yr                 1.344e+00  2.684e-01   5.008 5.51e-07 ***
online_customer                  -3.140e+00  1.012e+00  -3.103 0.001918 **
orders_season_E                  -2.293e-01  5.811e-02  -3.947 7.93e-05 ***
orders_mail_dept_h_1yr            5.594e-01  5.559e-01   1.006 0.314349
spend_d_h_1yr                     4.098e-03  4.488e-03   0.913 0.361246
spend_online_attributed_target    3.253e-03  2.948e-04  11.035  < 2e-16 ***
dollars_season_C                  1.646e-03  3.323e-04   4.951 7.38e-07 ***
spend_season_C                    8.870e-04  3.521e-04   2.520 0.011752 *
spend_e                          -1.800e-03  1.658e-03  -1.086 0.277669
spend_z1                         -7.645e-04  5.574e-04  -1.372 0.170186
spend_period_1b                   1.176e-02  5.397e-03   2.179 0.029344 *
spend_period_2b                   3.882e-03  1.248e-03   3.111 0.001866 **
spend_h_3yr                      -8.230e-03  3.163e-03  -2.602 0.009266 **
spend_g_3yr                       1.146e-02  1.497e-03   7.655 1.96e-14 ***
last_years_since                 -2.235e+00  6.171e-01  -3.621 0.000293 ***
spend_instore_o                   1.687e-04  1.052e-03   0.160 0.872582
spend_instore_o_3yr               5.862e-03  2.771e-03   2.116 0.034359 *
spend_instore_h_3yr              -7.076e-03  6.133e-03  -1.154 0.248580
spend_instore_t                  -7.650e-04  2.923e-03  -0.262 0.793545
spend_online_sh                  -4.224e-03  5.586e-03  -0.756 0.449553
spend_online_o_1yr                1.333e-02  4.151e-03   3.211 0.001323 **
spend_online_n_3yr                1.221e-02  3.725e-03   3.277 0.001050 **
spend_online_o_3yr               -4.307e-04  2.152e-03  -0.200 0.841385
spend_online_a_1yr               -5.079e-03  3.193e-03  -1.591 0.111711
spend_online_a_3yr                5.990e-03  1.268e-03   4.722 2.34e-06 ***
orders_online_t_3yr              -5.891e-01  6.034e-01  -0.976 0.328913
spend_online_sg_1yr               1.954e-02  9.669e-03   2.021 0.043258 *
spend_online_g_1yr               -1.523e-02  5.449e-03  -2.795 0.005197 **
orders_online_s_1yr               1.398e+00  7.385e-01   1.893 0.058330 .
customer_type_7                  -9.471e+00  1.529e+00  -6.195 5.86e-10 ***
customer_type_A                   1.109e-01  4.727e-01   0.235 0.814568
orders_attributed_mail_type_B     2.008e-01  1.609e-01   1.248 0.212051
spend_attributed_mail_type_B      4.314e-03  1.648e-03   2.619 0.008828 **
spend_attributed_mail_type_C     -3.412e-04  8.072e-04  -0.423 0.672531
mean_spend_attributed_mail_type_C 2.240e-03  2.955e-03   0.758 0.448446
mean_spend_attributed_mail_type_A -1.554e-03 3.960e-03  -0.392 0.694757
clicks_product_type_104_2yr      -6.621e-02  1.108e-01  -0.598 0.549959
clicks_product_type_312_3yr      -3.511e-02  1.713e-02  -2.049 0.040437 *
orders_e                         -1.871e-01  2.460e-01  -0.760 0.446983
orders_mail_dept_c_1yr            1.791e+00  4.956e-01   3.615 0.000301 ***
spend_d_s_1yr                    -1.135e-02  5.385e-03  -2.108 0.035046 *
spend_d_k_1yr                     9.991e-03  5.309e-03   1.882 0.059872 .
spend_a_1yr                      -8.242e-03  4.656e-03  -1.770 0.076711 .
spend_h_1yr                       6.007e-03  4.785e-03   1.255 0.209305
```

```
spend_direct_1yr                -3.521e-04  3.993e-03  -0.088 0.929744
acquisition_months_since         3.641e-03  1.565e-03   2.327 0.019969 *
spend_online_c                   6.129e-03  2.641e-03   2.321 0.020298 *
spend_online_a_6m                6.397e-03  2.885e-03   2.217 0.026626 *
orders_online_o                 -1.942e-01  4.681e-02  -4.149 3.34e-05 ***
orders_online_c_1yr              2.016e-01  7.331e-01   0.275 0.783344
spend_online_b_1yr              -1.333e-02  5.090e-03  -2.620 0.008803 **
customer_type_2                 -1.031e+00  5.153e-01  -2.001 0.045355 *
customer_income                 -1.906e-06  3.488e-06  -0.546 0.584726
orders_attributed_mail_type_A    7.294e-02  1.887e-01   0.386 0.699151
spend_attributed_mail_type_A    -2.077e-03  2.086e-03  -0.996 0.319208
clicks_product_type_312_3m      -2.251e-01  7.373e-02  -3.053 0.002270 **
clicks_product_type_301_2yr      5.812e-02  3.715e-02   1.564 0.117714
clickthrough_6m                 -1.810e-01  1.398e-01  -1.295 0.195459
emails_days_2yr                 -2.407e-03  1.725e-03  -1.395 0.162885
emails_3m                        6.095e-03  1.348e-02   0.452 0.651155
emailreceived_months_since       1.792e-02  9.718e-03   1.844 0.065150 .
orders_3yr                       4.847e-02  7.548e-02   0.642 0.520818
orders_mail_dept_d_1yr           1.124e-01  3.533e-01   0.318 0.750349
spent_q                         -5.846e-03  3.771e-03  -1.550 0.121133
spend_instore_q                 -7.066e-03  1.242e-02  -0.569 0.569357
spend_online_s                   2.260e-03  1.384e-03   1.632 0.102647
clicks_product_type_112_3yr     -4.274e-02  1.553e-02  -2.752 0.005920 **
spend_instore_n_3yr             -6.651e-04  3.081e-03  -0.216 0.829081
spend_instore_p                  3.438e-03  6.751e-03   0.509 0.610621
spend_instore_p_1yr              6.812e-03  2.130e-02   0.320 0.749156
orders_online_n_1yr             -1.097e+00  5.875e-01  -1.867 0.061968 .
spend_instore_a_yr               7.281e-04  1.857e-03   0.392 0.695017
orders_attributed_mail_type_C    1.060e-01  9.089e-02   1.166 0.243559
clickthrough_3yr                 4.835e-03  2.601e-02   0.186 0.852525
spend_instore_g_1yr             -1.855e-03  1.179e-02  -0.157 0.874989
spend_period_3b                 -4.789e-03  1.236e-03  -3.876 0.000106 ***
spend_instore_a                  8.957e-05  4.027e-04   0.222 0.823997
spend_direct_g                  -2.807e-04  6.662e-04  -0.421 0.673528
emailview_24m                    7.330e-03  6.124e-03   1.197 0.231298
spend_online_h_3yr              -9.820e-03  2.927e-03  -3.355 0.000794 ***
emailview_months_since          -1.548e-02  9.991e-03  -1.550 0.121179
orders_instore_c                 2.451e-01  2.468e-01   0.993 0.320794
emails_1yr                       1.705e-03  5.548e-03   0.307 0.758644
clicks_product_type_001_1m      -3.765e-01  2.012e-01  -1.871 0.061306 .
clickthrough_3m                  3.532e-01  2.591e-01   1.363 0.172818
orders_d_1yr                    -1.177e+00  2.696e-01  -4.365 1.27e-05 ***
orders_h                        -4.078e-02  2.637e-02  -1.547 0.121939
clicks_product_type_104          1.242e-02  3.309e-02   0.375 0.707350
clicks_product_type_502_3m      -5.376e-02  4.596e-02  -1.170 0.242174
web_activity_1m                 -5.034e-03  1.884e-02  -0.267 0.789368
orders_instore_m                 9.888e-02  1.208e-01   0.819 0.413063
spend_notz_1yr                   4.666e-03  6.017e-03   0.775 0.438124
orders_online_sh                 1.060e-01  2.883e-01   0.368 0.713181
orders_instore_n                -1.817e-01  1.200e-01  -1.514 0.129958
clicks_product_type_502_1yr      2.986e-02  1.035e-02   2.884 0.003928 **
orders_hm                        7.806e-02  4.118e-02   1.896 0.058026 .
emailview                       -1.378e-03  1.803e-03  -0.764 0.444588
spend_m_1yr                      7.514e-03  9.599e-03   0.783 0.433759
```

```
clicks_product_type_301_3m         1.292e-01  1.021e-01   1.265 0.205708
orders_instore_a_yr               -3.129e-01  2.789e-01  -1.122 0.261852
clicks_product_type_001_6m        -1.487e-01  7.676e-02  -1.938 0.052684 .
clickthrough_months_since          1.141e-02  6.949e-03   1.642 0.100517
orders_online_h_1yr               -1.315e+00  4.705e-01  -2.795 0.005190 **
orders_instore_h_3yr               8.589e-01  3.914e-01   2.194 0.028208 *
spend_period_1a                    1.102e-03  6.727e-03   0.164 0.869923
orders_total                      -3.961e-03  1.629e-02  -0.243 0.807880
spend_online_s_3yr                 3.820e-03  3.563e-03   1.072 0.283752
spend_M_3yr                        1.275e-02  2.805e-03   4.545 5.50e-06 ***
orders_c                           1.307e-01  8.727e-02   1.498 0.134140
spend_l                           -5.696e-04  2.662e-03  -0.214 0.830597
spend_instore_n                    3.112e-03  3.352e-03   0.928 0.353240
orders_z                           3.649e-01  5.618e-01   0.649 0.516044
web_activity_24m                  -7.181e-03  2.449e-03  -2.932 0.003365 **
spend_instore_q_3yr                1.028e-02  1.898e-02   0.542 0.588126
clicks_product_type_504           -7.729e-03  1.872e-03  -4.128 3.67e-05 ***
emailview_6m                      -5.962e-02  2.168e-02  -2.750 0.005959 **
buy_instore_days_since             2.268e-04  2.019e-04   1.123 0.261326
spend_h                           -1.050e-04  3.979e-04  -0.264 0.791811
clicks_product_type_801_3yr        6.335e-02  2.268e-02   2.793 0.005221 **
spend_nott_1yr                     1.108e-02  1.068e-02   1.038 0.299411
orders_online_s_3yr               -4.248e-01  3.165e-01  -1.342 0.179572
spend_instore_h_1yr                2.167e-03  9.108e-03   0.238 0.811965
spend_online_b_3yr                -7.299e-04  1.773e-03  -0.412 0.680609
orders_instore_s_3yr              -1.806e-01  1.603e-01  -1.127 0.259954
orders_instore_r_3yr              -3.399e-01  8.659e-01  -0.393 0.694677
clicks_product_type_502_1m         8.076e-02  4.533e-02   1.782 0.074801 .
mean_spend_attributed_mail_type_B  3.759e-03  3.394e-03   1.108 0.267961
clicks_product_type_201_3yr        3.699e-02  1.662e-02   2.226 0.026019 *
store_trips                        2.096e-02  3.575e-02   0.586 0.557739
spend_online_t_1yr                -2.794e-02  1.032e-02  -2.708 0.006774 **
orders_online_o_1yr               -6.049e-01  5.032e-01  -1.202 0.229328
spend_online_k                    -4.045e-03  8.733e-04  -4.632 3.63e-06 ***
spend_online_s_1yr                -5.588e-03  7.297e-03  -0.766 0.443754
clicks_product_type_503            3.883e-03  1.104e-03   3.517 0.000437 ***
orders_instore_t_3yr              -3.362e-01  4.710e-01  -0.714 0.475395
W                                         NA         NA      NA       NA
customer_id                        1.413e-08  9.846e-09   1.435 0.151370
training_sample                           NA         NA      NA       NA
---
Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1

Residual standard error: 44.26 on 83502 degrees of freedom
Multiple R-squared:  0.1129,    Adjusted R-squared:  0.1113
F-statistic: 72.29 on 147 and 83502 DF,  p-value: < 2.2e-16
```

```r
# Interpretation : R is quite high and R^2 is low 0.1129
# suggesting that model is not capture much variation in y.
# Indicating true relationship between predictors and y is weak.
```

```r
# 2. Estimate a Model to Predict Spending
# Output in vector
pred_y_OLS = predict(fit_OLS, newdata = testing_data_targeted)
mse_OLS = mean((testing_data$outcome_spend - pred_y_OLS)^2)
```

```r
results = data.table(
  input = rownames(summary_OLS$coefficients),    # Extract row names (input variable names)
  est_OLS = summary_OLS$coefficients[, 1],      # Extract coefficient estimates
  p_OLS = summary_OLS$coefficients[, 4]         # Extract p-values
)


kable(results, digits=3)
```

input	est_OLS	p_OLS
(Intercept)	-11.137	0.280
customer_type_L	1.037	0.028
customer_type_C	-1.074	0.022
clicks_product_type_504_3m	0.005	0.853
clicks_product_type_112_6m	0.027	0.295
clicks_product_type_301_1m	0.764	0.000
clicks_product_type_001_12m	0.015	0.639
clicks_product_type_201_1m	-0.054	0.515
clicks_product_type_201_6m	0.051	0.031
web_activity_3m	0.016	0.005
clickthrough_1m	0.581	0.067
emailview_1m	0.234	0.000
orders_online_1yr	1.344	0.000
online_customer	-3.140	0.002
orders_season_E	-0.229	0.000
orders_mail_dept_h_1yr	0.559	0.314
spend_d_h_1yr	0.004	0.361
spend_online_attributed_target	0.003	0.000
dollars_season_C	0.002	0.000
spend_season_C	0.001	0.012
spend_e	-0.002	0.278
spend_z1	-0.001	0.170
spend_period_1b	0.012	0.029
spend_period_2b	0.004	0.002
spend_h_3yr	-0.008	0.009
spend_g_3yr	0.011	0.000
last_years_since	-2.235	0.000
spend_instore_o	0.000	0.873
spend_instore_o_3yr	0.006	0.034
spend_instore_h_3yr	-0.007	0.249
spend_instore_t	-0.001	0.794
spend_online_sh	-0.004	0.450
spend_online_o_1yr	0.013	0.001
spend_online_n_3yr	0.012	0.001
spend_online_o_3yr	0.000	0.841
spend_online_a_1yr	-0.005	0.112
spend_online_a_3yr	0.006	0.000
orders_online_t_3yr	-0.589	0.329
spend_online_sg_1yr	0.020	0.043
spend_online_g_1yr	-0.015	0.005
orders_online_s_1yr	1.398	0.058
customer_type_7	-9.471	0.000

input	est_OLS	p_OLS
customer_type_A	0.111	0.815
orders_attributed_mail_type_B	0.201	0.212
spend_attributed_mail_type_B	0.004	0.009
spend_attributed_mail_type_C	0.000	0.673
mean_spend_attributed_mail_type_C	0.002	0.448
mean_spend_attributed_mail_type_A	-0.002	0.695
clicks_product_type_104_2yr	-0.066	0.550
clicks_product_type_312_3yr	-0.035	0.040
orders_e	-0.187	0.447
orders_mail_dept_c_1yr	1.791	0.000
spend_d_s_1yr	-0.011	0.035
spend_d_k_1yr	0.010	0.060
spend_a_1yr	-0.008	0.077
spend_h_1yr	0.006	0.209
spend_direct_1yr	0.000	0.930
acquisition_months_since	0.004	0.020
spend_online_c	0.006	0.020
spend_online_a_6m	0.006	0.027
orders_online_o	-0.194	0.000
orders_online_c_1yr	0.202	0.783
spend_online_b_1yr	-0.013	0.009
customer_type_2	-1.031	0.045
customer_income	0.000	0.585
orders_attributed_mail_type_A	0.073	0.699
spend_attributed_mail_type_A	-0.002	0.319
clicks_product_type_312_3m	-0.225	0.002
clicks_product_type_301_2yr	0.058	0.118
clickthrough_6m	-0.181	0.195
emails_days_2yr	-0.002	0.163
emails_3m	0.006	0.651
emailreceived_months_since	0.018	0.065
orders_3yr	0.048	0.521
orders_mail_dept_d_1yr	0.112	0.750
spent_q	-0.006	0.121
spend_instore_q	-0.007	0.569
spend_online_s	0.002	0.103
clicks_product_type_112_3yr	-0.043	0.006
spend_instore_n_3yr	-0.001	0.829
spend_instore_p	0.003	0.611
spend_instore_p_1yr	0.007	0.749
orders_online_n_1yr	-1.097	0.062
spend_instore_a_yr	0.001	0.695
orders_attributed_mail_type_C	0.106	0.244
clickthrough_3yr	0.005	0.853
spend_instore_g_1yr	-0.002	0.875
spend_period_3b	-0.005	0.000
spend_instore_a	0.000	0.824
spend_direct_g	0.000	0.674
emailview_24m	0.007	0.231
spend_online_h_3yr	-0.010	0.001
emailview_months_since	-0.015	0.121
orders_instore_c	0.245	0.321

input	est_OLS	p_OLS
emails_1yr	0.002	0.759
clicks_product_type_001_1m	-0.376	0.061
clickthrough_3m	0.353	0.173
orders_d_1yr	-1.177	0.000
orders_h	-0.041	0.122
clicks_product_type_104	0.012	0.707
clicks_product_type_502_3m	-0.054	0.242
web_activity_1m	-0.005	0.789
orders_instore_m	0.099	0.413
spend_notz_1yr	0.005	0.438
orders_online_sh	0.106	0.713
orders_instore_n	-0.182	0.130
clicks_product_type_502_1yr	0.030	0.004
orders_hm	0.078	0.058
emailview	-0.001	0.445
spend_m_1yr	0.008	0.434
clicks_product_type_301_3m	0.129	0.206
orders_instore_a_yr	-0.313	0.262
clicks_product_type_001_6m	-0.149	0.053
clickthrough_months_since	0.011	0.101
orders_online_h_1yr	-1.315	0.005
orders_instore_h_3yr	0.859	0.028
spend_period_1a	0.001	0.870
orders_total	-0.004	0.808
spend_online_s_3yr	0.004	0.284
spend_M_3yr	0.013	0.000
orders_c	0.131	0.134
spend_l	-0.001	0.831
spend_instore_n	0.003	0.353
orders_z	0.365	0.516
web_activity_24m	-0.007	0.003
spend_instore_q_3yr	0.010	0.588
clicks_product_type_504	-0.008	0.000
emailview_6m	-0.060	0.006
buy_instore_days_since	0.000	0.261
spend_h	0.000	0.792
clicks_product_type_801_3yr	0.063	0.005
spend_nott_1yr	0.011	0.299
orders_online_s_3yr	-0.425	0.180
spend_instore_h_1yr	0.002	0.812
spend_online_b_3yr	-0.001	0.681
orders_instore_s_3yr	-0.181	0.260
orders_instore_r_3yr	-0.340	0.695
clicks_product_type_502_1m	0.081	0.075
mean_spend_attributed_mail_type_B	0.004	0.268
clicks_product_type_201_3yr	0.037	0.026
store_trips	0.021	0.558
spend_online_t_1yr	-0.028	0.007
orders_online_o_1yr	-0.605	0.229
spend_online_k	-0.004	0.000
spend_online_s_1yr	-0.006	0.444
clicks_product_type_503	0.004	0.000

input	est_OLS	p_OLS
orders_instore_t_3yr	-0.336	0.475
customer_id	0.000	0.151

```r
# 3. Follow the "Common Approach" Discussed in Class


# TODO : Evaluate the variables and discard those that are not influencing predictions

# Step 1: Identify columns with p > 0.05
insignificant_vars <- results[p_OLS > 0.05]$input  # Variables to drop

# Step 2: Drop these columns from the datasets
training_data_reduced <- training_data_targeted[, !(colnames(training_data_targeted) %in% insignificant
testing_data_reduced <- testing_data_targeted[, !(colnames(testing_data_targeted) %in% insignificant_va

# Step 3: Refit the model using the reduced dataset
fit_OLS_reduced <- lm(outcome_spend ~ ., data = training_data_reduced)

# Step 4: Predict on reduced dataset
pred_y_OLS_reduced <- predict(fit_OLS_reduced, newdata = testing_data_reduced)

# Step 5: Calculate Mean Squared Error for the reduced model
mse_OLS_reduced <- mean((testing_data_reduced$outcome_spend - pred_y_OLS_reduced)^2)

# Compare performance
cat("MSE (Full Model):", mse_OLS, "\n")
```

```
MSE (Full Model): 2201.572
```

```r
cat("MSE (Reduced Model):", mse_OLS_reduced, "\n")
```

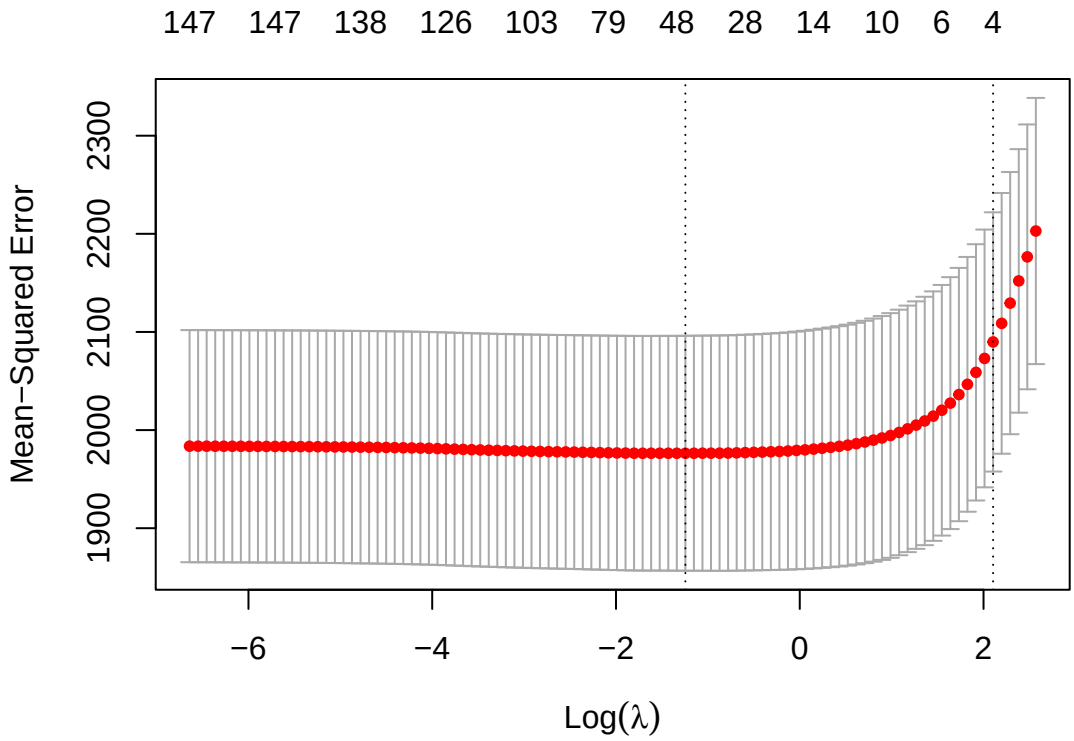
```
MSE (Reduced Model): 1956.938
```

```r
# TODO : Add the other models result for comparison

# LASSO

library(glmnet)

y_train <- training_data_targeted$outcome_spend
X_train <- model.matrix(outcome_spend ~ 0 + ., data = training_data_targeted)

# Set seed for reproducibility
set.seed(37105)

# Create custom folds for cross-validation
N_obs_training <- nrow(training_data_targeted)
folds <- sample(1:10, N_obs_training, replace = TRUE)

# Fit LASSO with custom folds
cv_lasso <- cv.glmnet(X_train, y_train, alpha = 1, foldid = folds)

# Identify the best lambda
```

```
best_lambda <- cv_lasso$lambda.min
cat("Best Lambda:", best_lambda, "\n")
```

```
Best Lambda: 0.287945
```

```
# Refit the model with the best lambda
lasso_model <- glmnet(X_train, y_train, alpha = 1, lambda = best_lambda)

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




```
# Display coefficients
length(coef(cv_lasso, s = "lambda.min")[, 1])
```

```
[1] 150
```

```
nrow(results)
```

```
[1] 148
```

```
coef_vals <- rep(coef(cv_lasso, s = "lambda.min")[, 1], length.out = nrow(results))
results[, est_LASSO := coef_vals]
# Verify matching dimensions
if (length(coef_vals) == nrow(results)) {
    results[, est_LASSO := coef_vals]
} else {
    stop("Dimensions do not match. Please check the data alignment.")
}
```

```
# Predict and calculate MSE
y_test <- testing_data_targeted$outcome_spend
X_test <- model.matrix(outcome_spend ~ 0 + ., data = testing_data_targeted)
```

```r
# Predict using the LASSO model
predictions_lasso <- predict(cv_lasso, newx = X_test, s = "lambda.min")

# Calculate Mean Squared Error (MSE) on the test dataset
mse_lasso <- mean((y_test - predictions_lasso)^2)
cat("LASSO Mean Squared Error on Test Set:", mse_lasso, "\n")
```

```
LASSO Mean Squared Error on Test Set: 1950.222
```

```r
# Elastic net

library(data.table)

# Step 1: Prepare the data
y <- training_data_targeted$outcome_spend  # Replace 'outcome_spend' with your actual target variable n
X <- model.matrix(outcome_spend ~ 0 + ., data = training_data_targeted)  # Design matrix (exclude inter

# Step 2: Set a fixed seed and create custom folds
set.seed(37105)
N_obs_training <- nrow(training_data_targeted)
folds <- sample(1:10, N_obs_training, replace = TRUE)

# Step 3: Set up a coarser grid for alpha values
alphas <- seq(0, 1, by = 0.05)  # Range of alpha values with a step of 0.05
mse_results <- data.table(alpha = alphas, mse = rep(NA, length(alphas)))

# Step 4: Cross-validate for each alpha value using custom folds
for (i in 1:length(alphas)) {
  alpha_value <- alphas[i]
  cv_model <- cv.glmnet(X, y, alpha = alpha_value, foldid = folds)
  mse_results[i, mse := min(cv_model$cvm)]  # Store the minimum MSE for each alpha
}

# Step 5: Identify the best alpha and corresponding lambda
best_alpha <- mse_results[which.min(mse), alpha]
cat("Best Alpha:", best_alpha, "\n")
```

```
Best Alpha: 0
```

```r
cv_best <- cv.glmnet(X, y, alpha = best_alpha, foldid = folds)
best_lambda <- cv_best$lambda.min
cat("Best Lambda:", best_lambda, "\n")
```

```
Best Lambda: 14.66832
```

```r
# Step 6: Refit the Elastic Net model with the best alpha and lambda
elastic_net_model <- glmnet(X, y, alpha = best_alpha, lambda = best_lambda)
```

```r
# Step 7: Display coefficients
length(coef(elastic_net_model, alpha = best_alpha, lambda = best_lambda)[, 1])
```

```
[1] 150
```

```r
nrow(results)
```

```
[1] 148
```

```
coef_vals <- rep(coef(elastic_net_model, alpha = best_alpha, lambda = best_lambda)[, 1], length.out = n
results[, est_elastic := coef_vals]

# Step 8: Evaluate on the testing data
y_test <- testing_data_targeted$outcome_spend
X_test <- model.matrix(outcome_spend ~ 0 + ., data = testing_data_targeted)

# Predict and calculate MSE
predictions_elastic <- predict(elastic_net_model, newx = X_test)
mse_elastic <- mean((y_test - predictions_elastic)^2)
cat("Mean Squared Error on Test Set:", mse_elastic, "\n")
```

```
Mean Squared Error on Test Set: 1949.174
```

```
# 4. Compare the models with results coeeficients
cat(mse_OLS, mse_lasso, mse_elastic)
```

```
2201.572 1950.222 1949.174
```

```
kable(results, digits=3)
```

input	est_OLS	p_OLS	est_LASSO	est_elastic
(Intercept)	-11.137	0.280	0.225	-10.997
customer_type_L	1.037	0.028	0.000	0.562
customer_type_C	-1.074	0.022	0.000	-0.367
clicks_product_type_504_3m	0.005	0.853	0.000	0.001
clicks_product_type_112_6m	0.027	0.295	0.000	0.015
clicks_product_type_301_1m	0.764	0.000	0.537	0.602
clicks_product_type_001_12m	0.015	0.639	0.000	-0.006
clicks_product_type_201_1m	-0.054	0.515	0.000	-0.010
clicks_product_type_201_6m	0.051	0.031	0.037	0.039
web_activity_3m	0.016	0.005	0.001	0.004
clickthrough_1m	0.581	0.067	0.455	0.532
emailview_1m	0.234	0.000	0.053	0.095
orders_online_1yr	1.344	0.000	0.426	0.278
online_customer	-3.140	0.002	0.000	1.344
orders_season_E	-0.229	0.000	-0.006	-0.058
orders_mail_dept_h_1yr	0.559	0.314	0.085	0.258
spend_d_h_1yr	0.004	0.361	0.002	0.003
spend_online_attributed_target	0.003	0.000	0.002	0.001
dollars_season_C	0.002	0.000	0.001	0.001
spend_season_C	0.001	0.012	0.000	0.000
spend_e	-0.002	0.278	0.000	-0.001
spend_z1	-0.001	0.170	0.000	0.000
spend_period_1b	0.012	0.029	0.007	0.007
spend_period_2b	0.004	0.002	0.005	0.006
spend_h_3yr	-0.008	0.009	0.000	0.002
spend_g_3yr	0.011	0.000	0.009	0.005
last_years_since	-2.235	0.000	-1.988	-1.914
spend_instore_o	0.000	0.873	0.000	0.000
spend_instore_o_3yr	0.006	0.034	0.001	0.003
spend_instore_h_3yr	-0.007	0.249	0.000	0.000
spend_instore_t	-0.001	0.794	0.000	0.000
spend_online_sh	-0.004	0.450	0.000	0.000

input	est_OLS	p_OLS	est_LASSO	est_elastic
spend_online_o_1yr	0.013	0.001	0.009	0.005
spend_online_n_3yr	0.012	0.001	0.000	0.002
spend_online_o_3yr	0.000	0.841	0.000	0.004
spend_online_a_1yr	-0.005	0.112	0.000	0.002
spend_online_a_3yr	0.006	0.000	0.005	0.003
orders_online_t_3yr	-0.589	0.329	0.000	0.094
spend_online_sg_1yr	0.020	0.043	0.001	0.017
spend_online_g_1yr	-0.015	0.005	0.000	0.002
orders_online_s_1yr	1.398	0.058	0.000	0.649
customer_type_7	-9.471	0.000	-5.655	-5.020
customer_type_A	0.111	0.815	0.000	0.225
orders_attributed_mail_type_B	0.201	0.212	0.000	0.144
spend_attributed_mail_type_B	0.004	0.009	0.005	0.003
spend_attributed_mail_type_C	0.000	0.673	0.000	0.000
mean_spend_attributed_mail_type_C	0.002	0.448	0.000	0.003
mean_spend_attributed_mail_type_A	-0.002	0.695	0.000	-0.002
clicks_product_type_104_2yr	-0.066	0.550	0.000	-0.029
clicks_product_type_312_3yr	-0.035	0.040	0.000	-0.023
orders_e	-0.187	0.447	-0.380	-0.359
orders_mail_dept_c_1yr	1.791	0.000	0.000	0.348
spend_d_s_1yr	-0.011	0.035	0.000	-0.006
spend_d_k_1yr	0.010	0.060	0.006	0.011
spend_a_1yr	-0.008	0.077	-0.003	-0.006
spend_h_1yr	0.006	0.209	0.014	0.007
spend_direct_1yr	0.000	0.930	0.000	0.002
acquisition_months_since	0.004	0.020	0.000	0.001
spend_online_c	0.006	0.020	0.006	0.005
spend_online_a_6m	0.006	0.027	0.000	0.001
orders_online_o	-0.194	0.000	-0.001	-0.012
orders_online_c_1yr	0.202	0.783	0.822	0.831
spend_online_b_1yr	-0.013	0.009	-0.004	-0.006
customer_type_2	-1.031	0.045	-0.082	-0.497
customer_income	0.000	0.585	0.000	0.000
orders_attributed_mail_type_A	0.073	0.699	0.000	-0.025
spend_attributed_mail_type_A	-0.002	0.319	0.000	0.001
clicks_product_type_312_3m	-0.225	0.002	0.000	-0.109
clicks_product_type_301_2yr	0.058	0.118	0.000	0.010
clickthrough_6m	-0.181	0.195	0.000	-0.045
emails_days_2yr	-0.002	0.163	0.000	-0.001
emails_3m	0.006	0.651	0.000	0.001
emailreceived_months_since	0.018	0.065	0.000	0.011
orders_3yr	0.048	0.521	0.000	0.022
orders_mail_dept_d_1yr	0.112	0.750	0.000	0.095
spent_q	-0.006	0.121	0.000	0.001
spend_instore_q	-0.007	0.569	0.000	-0.002
spend_online_s	0.002	0.103	0.001	0.003
clicks_product_type_112_3yr	-0.043	0.006	-0.011	-0.028
spend_instore_n_3yr	-0.001	0.829	0.000	0.000
spend_instore_p	0.003	0.611	0.000	0.002
spend_instore_p_1yr	0.007	0.749	0.000	0.004
orders_online_n_1yr	-1.097	0.062	0.000	-0.031
spend_instore_a_yr	0.001	0.695	0.000	0.000

input	est_OLS	p_OLS	est_LASSO	est_elastic
orders_attributed_mail_type_C	0.106	0.244	0.000	-0.044
clickthrough_3yr	0.005	0.853	0.000	-0.012
spend_instore_g_1yr	-0.002	0.875	0.000	-0.001
spend_period_3b	-0.005	0.000	0.000	0.000
spend_instore_a	0.000	0.824	0.000	0.000
spend_direct_g	0.000	0.674	0.000	0.001
emailview_24m	0.007	0.231	0.000	0.000
spend_online_h_3yr	-0.010	0.001	0.000	0.002
emailview_months_since	-0.015	0.121	0.000	-0.008
orders_instore_c	0.245	0.321	0.000	0.155
emails_1yr	0.002	0.759	0.000	0.000
clicks_product_type_001_1m	-0.376	0.061	-0.026	-0.252
clickthrough_3m	0.353	0.173	0.000	0.140
orders_d_1yr	-1.177	0.000	0.000	-0.265
orders_h	-0.041	0.122	-0.003	-0.003
clicks_product_type_104	0.012	0.707	0.000	0.029
clicks_product_type_502_3m	-0.054	0.242	0.000	0.000
web_activity_1m	-0.005	0.789	0.000	0.011
orders_instore_m	0.099	0.413	0.000	0.057
spend_notz_1yr	0.005	0.438	0.002	0.002
orders_online_sh	0.106	0.713	0.000	-0.020
orders_instore_n	-0.182	0.130	0.000	-0.067
clicks_product_type_502_1yr	0.030	0.004	0.000	0.005
orders_hm	0.078	0.058	0.000	0.058
emailview	-0.001	0.445	0.000	-0.001
spend_m_1yr	0.008	0.434	0.007	0.005
clicks_product_type_301_3m	0.129	0.206	0.000	0.085
orders_instore_a_yr	-0.313	0.262	0.000	-0.161
clicks_product_type_001_6m	-0.149	0.053	-0.010	-0.064
clickthrough_months_since	0.011	0.101	0.000	0.009
orders_online_h_1yr	-1.315	0.005	0.000	-0.064
orders_instore_h_3yr	0.859	0.028	0.006	0.373
spend_period_1a	0.001	0.870	0.000	0.001
orders_total	-0.004	0.808	0.000	-0.002
spend_online_s_3yr	0.004	0.284	0.000	0.003
spend_M_3yr	0.013	0.000	0.004	0.003
orders_c	0.131	0.134	0.000	0.042
spend_l	-0.001	0.831	0.001	0.002
spend_instore_n	0.003	0.353	0.000	0.001
orders_z	0.365	0.516	0.000	-0.151
web_activity_24m	-0.007	0.003	0.000	-0.001
spend_instore_q_3yr	0.010	0.588	0.000	0.003
clicks_product_type_504	-0.008	0.000	-0.002	-0.003
emailview_6m	-0.060	0.006	0.000	-0.006
buy_instore_days_since	0.000	0.261	0.000	0.000
spend_h	0.000	0.792	0.000	0.000
clicks_product_type_801_3yr	0.063	0.005	0.000	0.015
spend_nott_1yr	0.011	0.299	0.000	0.004
orders_online_s_3yr	-0.425	0.180	0.000	0.042
spend_instore_h_1yr	0.002	0.812	0.000	0.000
spend_online_b_3yr	-0.001	0.681	0.000	0.002
orders_instore_s_3yr	-0.181	0.260	0.000	-0.076

input	est_OLS	p_OLS	est_LASSO	est_elastic
orders_instore_r_3yr	-0.340	0.695	0.000	-0.260
clicks_product_type_502_1m	0.081	0.075	0.020	0.037
mean_spend_attributed_mail_type_B	0.004	0.268	0.003	0.007
clicks_product_type_201_3yr	0.037	0.026	0.000	0.025
store_trips	0.021	0.558	0.000	-0.004
spend_online_t_1yr	-0.028	0.007	0.000	-0.003
orders_online_o_1yr	-0.605	0.229	0.000	0.130
spend_online_k	-0.004	0.000	-0.001	-0.001
spend_online_s_1yr	-0.006	0.444	0.000	-0.001
clicks_product_type_503	0.004	0.000	0.000	0.001
orders_instore_t_3yr	-0.336	0.475	0.000	-0.159
customer_id	0.000	0.151	0.000	0.000

Compare the estimated coefficients for OLS, the LASSO, and the elastic net. How "sparse" is the prediction problem, i.e. how many inputs are selected by the LASSO and the elastic net?

```
# Directly store the 'results' object to a data frame
df_results <- as.data.frame(results)

# Now, 'df_results' contains the data in a data frame format
print(df_results)
```

```
                             input       est_OLS        p_OLS     est_LASSO
1                      (Intercept) -1.113661e+01 2.802543e-01  2.249714e-01
2                    customer_type_L  1.036664e+00 2.755758e-02  0.000000e+00
3                    customer_type_C -1.074023e+00 2.188651e-02  0.000000e+00
4          clicks_product_type_504_3m  4.548860e-03 8.531870e-01  0.000000e+00
5          clicks_product_type_112_6m  2.671487e-02 2.952767e-01  0.000000e+00
6          clicks_product_type_301_1m  7.641701e-01 3.195069e-04  5.369797e-01
7         clicks_product_type_001_12m  1.488354e-02 6.393599e-01  0.000000e+00
8          clicks_product_type_201_1m -5.405099e-02 5.150731e-01  0.000000e+00
9          clicks_product_type_201_6m  5.103847e-02 3.050250e-02  3.710646e-02
10                   web_activity_3m  1.567826e-02 5.203859e-03  1.290585e-03
11                   clickthrough_1m  5.810528e-01 6.704405e-02  4.554467e-01
12                      emailview_1m  2.341356e-01 1.438705e-05  5.302668e-02
13                 orders_online_1yr  1.343881e+00 5.514722e-07  4.259715e-01
14                   online_customer -3.139839e+00 1.917647e-03  0.000000e+00
15                   orders_season_E -2.293332e-01 7.926307e-05 -5.784775e-03
16             orders_mail_dept_h_1yr  5.593505e-01 3.143490e-01  8.527533e-02
17                      spend_d_h_1yr  4.097626e-03 3.612459e-01  1.816582e-03
18    spend_online_attributed_target  3.253234e-03 2.713982e-28  2.188949e-03
19                  dollars_season_C  1.645552e-03 7.380481e-07  1.316580e-03
20                    spend_season_C  8.870382e-04 1.175200e-02  0.000000e+00
21                           spend_e -1.800098e-03 2.776688e-01  0.000000e+00
22                          spend_z1 -7.644959e-04 1.701856e-01  0.000000e+00
23                   spend_period_1b  1.175846e-02 2.934443e-02  6.561506e-03
24                   spend_period_2b  3.882499e-03 1.866356e-03  4.916653e-03
25                       spend_h_3yr -8.230186e-03 9.265814e-03  0.000000e+00
26                       spend_g_3yr  1.145537e-02 1.958976e-14  8.842281e-03
27                 last_years_since -2.234771e+00 2.931695e-04 -1.988424e+00
28                   spend_instore_o  1.687183e-04 8.725816e-01  0.000000e+00
29               spend_instore_o_3yr  5.862421e-03 3.435915e-02  9.905177e-04
```

```
30                spend_instore_h_3yr -7.076214e-03 2.485796e-01  0.000000e+00
31                    spend_instore_t -7.650096e-04 7.935455e-01  0.000000e+00
32                    spend_online_sh -4.224207e-03 4.495533e-01  0.000000e+00
33                  spend_online_o_1yr  1.332981e-02 1.322808e-03  8.881672e-03
34                  spend_online_n_3yr  1.220651e-02 1.050320e-03  0.000000e+00
35                  spend_online_o_3yr -4.307139e-04 8.413851e-01  0.000000e+00
36                  spend_online_a_1yr -5.078993e-03 1.117114e-01  0.000000e+00
37                  spend_online_a_3yr  5.989656e-03 2.335848e-06  4.843537e-03
38                orders_online_t_3yr -5.890673e-01 3.289129e-01  0.000000e+00
39                spend_online_sg_1yr  1.954357e-02 4.325775e-02  1.249729e-03
40                 spend_online_g_1yr -1.522748e-02 5.197347e-03  0.000000e+00
41                orders_online_s_1yr  1.398220e+00 5.832995e-02  0.000000e+00
42                    customer_type_7 -9.471146e+00 5.860414e-10 -5.655110e+00
43                    customer_type_A  1.108730e-01 8.145680e-01  0.000000e+00
44      orders_attributed_mail_type_B  2.008478e-01 2.120512e-01  0.000000e+00
45       spend_attributed_mail_type_B  4.314408e-03 8.828361e-03  4.973848e-03
46       spend_attributed_mail_type_C -3.411905e-04 6.725311e-01  0.000000e+00
47  mean_spend_attributed_mail_type_C  2.240062e-03 4.484456e-01  0.000000e+00
48  mean_spend_attributed_mail_type_A -1.553781e-03 6.947573e-01  0.000000e+00
49        clicks_product_type_104_2yr -6.621291e-02 5.499585e-01  0.000000e+00
50        clicks_product_type_312_3yr -3.510561e-02 4.043748e-02  0.000000e+00
51                           orders_e -1.870707e-01 4.469829e-01 -3.799217e-01
52              orders_mail_dept_c_1yr  1.791288e+00 3.008452e-04  0.000000e+00
53                        spend_d_s_1yr -1.135099e-02 3.504580e-02  0.000000e+00
54                        spend_d_k_1yr  9.990890e-03 5.987205e-02  6.061441e-03
55                          spend_a_1yr -8.241957e-03 7.671051e-02 -2.794462e-03
56                          spend_h_1yr  6.007175e-03 2.093054e-01  1.398609e-02
57                     spend_direct_1yr -3.520577e-04 9.297435e-01  0.000000e+00
58            acquisition_months_since  3.640547e-03 1.996944e-02  0.000000e+00
59                       spend_online_c  6.129448e-03 2.029834e-02  5.580969e-03
60                    spend_online_a_6m  6.396803e-03 2.662584e-02  0.000000e+00
61                      orders_online_o -1.941999e-01 3.343627e-05 -1.287076e-03
62                  orders_online_c_1yr  2.015892e-01 7.833444e-01  8.222303e-01
63                   spend_online_b_1yr -1.333465e-02 8.802554e-03 -3.739668e-03
64                      customer_type_2 -1.031370e+00 4.535477e-02 -8.246518e-02
65                      customer_income -1.906378e-06 5.847263e-01  0.000000e+00
66      orders_attributed_mail_type_A  7.293819e-02 6.991513e-01  0.000000e+00
67       spend_attributed_mail_type_A -2.077473e-03 3.192079e-01  0.000000e+00
68         clicks_product_type_312_3m -2.250576e-01 2.269581e-03  0.000000e+00
69        clicks_product_type_301_2yr  5.812267e-02 1.177141e-01  0.000000e+00
70                      clickthrough_6m -1.809642e-01 1.954586e-01  0.000000e+00
71                      emails_days_2yr -2.406623e-03 1.628845e-01  0.000000e+00
72                            emails_3m  6.094845e-03 6.511548e-01  0.000000e+00
73          emailreceived_months_since  1.792158e-02 6.514997e-02  0.000000e+00
74                         orders_3yr  4.846640e-02 5.208181e-01  0.000000e+00
75              orders_mail_dept_d_1yr  1.123989e-01 7.503494e-01  0.000000e+00
76                           spent_q -5.845538e-03 1.211334e-01  0.000000e+00
77                    spend_instore_q -7.065838e-03 5.693574e-01  0.000000e+00
78                     spend_online_s  2.259620e-03 1.026470e-01  1.453288e-03
79        clicks_product_type_112_3yr -4.274395e-02 5.920062e-03 -1.057481e-02
80                spend_instore_n_3yr -6.651107e-04 8.290811e-01  0.000000e+00
81                    spend_instore_p  3.437764e-03 6.106214e-01  0.000000e+00
82                spend_instore_p_1yr  6.811924e-03 7.491560e-01  0.000000e+00
83                orders_online_n_1yr -1.096686e+00 6.196838e-02  0.000000e+00
```

```
84                spend_instore_a_yr  7.280898e-04 6.950172e-01  0.000000e+00
85     orders_attributed_mail_type_C  1.059929e-01 2.435592e-01  0.000000e+00
86                   clickthrough_3yr  4.835172e-03 8.525248e-01  0.000000e+00
87               spend_instore_g_1yr -1.854901e-03 8.749895e-01  0.000000e+00
88                   spend_period_3b -4.788827e-03 1.063270e-04  0.000000e+00
89                   spend_instore_a  8.957052e-05 8.239971e-01  0.000000e+00
90                    spend_direct_g -2.806977e-04 6.735281e-01  8.010951e-05
91                     emailview_24m  7.330105e-03 2.312984e-01  0.000000e+00
92                spend_online_h_3yr -9.819564e-03 7.937226e-04  0.000000e+00
93            emailview_months_since -1.548471e-02 1.211790e-01  0.000000e+00
94                  orders_instore_c  2.450786e-01 3.207944e-01  0.000000e+00
95                        emails_1yr  1.704639e-03 7.586441e-01  0.000000e+00
96       clicks_product_type_001_1m -3.764531e-01 6.130631e-02 -2.619685e-02
97                   clickthrough_3m  3.531858e-01 1.728185e-01  0.000000e+00
98                      orders_d_1yr -1.176731e+00 1.274338e-05  0.000000e+00
99                          orders_h -4.078051e-02 1.219386e-01 -2.737827e-03
100          clicks_product_type_104  1.242081e-02 7.073499e-01  0.000000e+00
101       clicks_product_type_502_3m -5.375752e-02 2.421737e-01  0.000000e+00
102                  web_activity_1m -5.033553e-03 7.893677e-01  0.000000e+00
103                 orders_instore_m  9.887628e-02 4.130628e-01  0.000000e+00
104                   spend_notz_1yr  4.665607e-03 4.381240e-01  1.806098e-03
105                 orders_online_sh  1.059657e-01 7.131812e-01  0.000000e+00
106                 orders_instore_n -1.817171e-01 1.299583e-01  0.000000e+00
107      clicks_product_type_502_1yr  2.985943e-02 3.927976e-03  0.000000e+00
108                        orders_hm  7.806311e-02 5.802573e-02  0.000000e+00
109                        emailview -1.377998e-03 4.445882e-01  0.000000e+00
110                      spend_m_1yr  7.513966e-03 4.337589e-01  6.920087e-03
111       clicks_product_type_301_3m  1.292103e-01 2.057077e-01  0.000000e+00
112              orders_instore_a_yr -3.129140e-01 2.618523e-01  0.000000e+00
113       clicks_product_type_001_6m -1.487256e-01 5.268419e-02 -9.987581e-03
114        clickthrough_months_since  1.141227e-02 1.005171e-01  0.000000e+00
115              orders_online_h_1yr -1.315076e+00 5.189663e-03  0.000000e+00
116             orders_instore_h_3yr  8.588711e-01 2.820836e-02  5.802940e-03
117                  spend_period_1a  1.101640e-03 8.699234e-01  0.000000e+00
118                     orders_total -3.960935e-03 8.078805e-01  0.000000e+00
119               spend_online_s_3yr  3.819754e-03 2.837518e-01  0.000000e+00
120                     spend_M_3yr  1.274994e-02 5.495593e-06  3.923075e-03
121                        orders_c  1.307310e-01 1.341397e-01  0.000000e+00
122                         spend_l -5.695906e-04 8.305966e-01  1.148198e-03
123                  spend_instore_n  3.111539e-03 3.532405e-01  0.000000e+00
124                        orders_z  3.648780e-01 5.160439e-01  0.000000e+00
125                web_activity_24m -7.181436e-03 3.364533e-03  0.000000e+00
126              spend_instore_q_3yr  1.027798e-02 5.881262e-01  0.000000e+00
127         clicks_product_type_504 -7.728557e-03 3.666807e-05 -2.144394e-03
128                    emailview_6m -5.961796e-02 5.958973e-03  0.000000e+00
129          buy_instore_days_since  2.267600e-04 2.613257e-01  0.000000e+00
130                         spend_h -1.050225e-04 7.918113e-01  0.000000e+00
131      clicks_product_type_801_3yr  6.335267e-02 5.221407e-03  0.000000e+00
132                  spend_nott_1yr  1.108363e-02 2.994113e-01  0.000000e+00
133              orders_online_s_3yr -4.247523e-01 1.795717e-01  0.000000e+00
134              spend_instore_h_1yr  2.166784e-03 8.119651e-01  0.000000e+00
135               spend_online_b_3yr -7.299314e-04 6.806088e-01  0.000000e+00
136             orders_instore_s_3yr -1.806120e-01 2.599535e-01  0.000000e+00
137             orders_instore_r_3yr -3.398634e-01 6.946769e-01  0.000000e+00
```

```
138        clicks_product_type_502_1m  8.075919e-02 7.480093e-02  1.999926e-02
139 mean_spend_attributed_mail_type_B  3.759257e-03 2.679608e-01  2.821571e-03
140       clicks_product_type_201_3yr  3.699492e-02 2.601911e-02  0.000000e+00
141                       store_trips  2.095979e-02 5.577386e-01  0.000000e+00
142                spend_online_t_1yr -2.793538e-02 6.774197e-03  0.000000e+00
143               orders_online_o_1yr -6.049498e-01 2.293283e-01  0.000000e+00
144                    spend_online_k -4.044994e-03 3.627811e-06 -1.108107e-03
145                spend_online_s_1yr -5.588247e-03 4.437544e-01  0.000000e+00
146           clicks_product_type_503  3.883423e-03 4.367617e-04  0.000000e+00
147              orders_instore_t_3yr -3.361870e-01 4.753949e-01  0.000000e+00
148                       customer_id  1.412568e-08 1.513705e-01  0.000000e+00
      est_elastic
1   -1.099685e+01
2    5.622041e-01
3   -3.671953e-01
4    1.335006e-03
5    1.501925e-02
6    6.019745e-01
7   -5.516266e-03
8   -9.604213e-03
9    3.882705e-02
10   3.987644e-03
11   5.319034e-01
12   9.458741e-02
13   2.776983e-01
14   1.344345e+00
15  -5.790046e-02
16   2.579773e-01
17   3.442460e-03
18   1.336635e-03
19   1.198042e-03
20   9.531149e-05
21  -6.703767e-04
22  -2.148257e-04
23   6.605708e-03
24   6.135322e-03
25   1.617230e-03
26   5.069527e-03
27  -1.913987e+00
28   4.801376e-04
29   3.008084e-03
30  -4.817568e-04
31   3.567789e-04
32  -4.792542e-04
33   5.477986e-03
34   1.582054e-03
35   4.134608e-03
36   2.360931e-03
37   2.508213e-03
38   9.379510e-02
39   1.693069e-02
40   1.698618e-03
41   6.490487e-01
42  -5.020064e+00
```

```
43   2.251783e-01
44   1.437364e-01
45   3.357357e-03
46   1.227813e-04
47   2.716067e-03
48  -1.522263e-03
49  -2.895283e-02
50  -2.299500e-02
51  -3.587998e-01
52   3.481997e-01
53  -5.778477e-03
54   1.070116e-02
55  -5.700420e-03
56   6.927119e-03
57   2.105746e-03
58   1.471280e-03
59   5.171491e-03
60   1.321817e-03
61  -1.223662e-02
62   8.309249e-01
63  -5.839275e-03
64  -4.973049e-01
65  -5.367756e-07
66  -2.530718e-02
67   6.688974e-04
68  -1.090432e-01
69   9.501414e-03
70  -4.521679e-02
71  -8.252407e-04
72   8.366013e-04
73   1.098658e-02
74   2.172919e-02
75   9.535138e-02
76   6.464171e-04
77  -1.592257e-03
78   3.186725e-03
79  -2.770624e-02
80  -1.108904e-04
81   1.918737e-03
82   4.285115e-03
83  -3.131969e-02
84   3.079854e-04
85  -4.445481e-02
86  -1.154757e-02
87  -7.964364e-04
88   2.575908e-04
89   1.221779e-04
90   8.376663e-04
91   3.672003e-04
92   1.674259e-03
93  -8.032774e-03
94   1.548434e-01
95  -1.589982e-04
96  -2.524007e-01
```

```
97   1.395401e-01
98  -2.650396e-01
99  -3.275212e-03
100  2.916171e-02
101 -1.572870e-04
102  1.149721e-02
103  5.746735e-02
104  2.375209e-03
105 -1.982766e-02
106 -6.665095e-02
107  4.689264e-03
108  5.817520e-02
109 -5.346856e-04
110  5.377585e-03
111  8.501436e-02
112 -1.605963e-01
113 -6.410335e-02
114  9.211690e-03
115 -6.428111e-02
116  3.726457e-01
117  5.350520e-04
118 -1.657727e-03
119  2.877605e-03
120  2.794144e-03
121  4.210187e-02
122  1.855743e-03
123  5.292499e-04
124 -1.512271e-01
125 -8.024014e-04
126  2.508078e-03
127 -3.365180e-03
128 -6.272200e-03
129  1.405650e-04
130  4.379270e-04
131  1.531386e-02
132  3.622967e-03
133  4.247323e-02
134  4.128811e-05
135  1.953997e-03
136 -7.581923e-02
137 -2.597561e-01
138  3.738452e-02
139  7.015295e-03
140  2.497584e-02
141 -3.975649e-03
142 -3.091704e-03
143  1.300268e-01
144 -5.750509e-04
145 -1.214518e-03
146  1.003094e-03
147 -1.594992e-01
148  0.000000e+00
```

```r
# Count the number of non-zero coefficients in the LASSO column
non_zero_lasso <- sum(df_results$est_LASSO != 0)
cat("Number of non-zero coefficients in LASSO:", non_zero_lasso-1, "\n")
```

```
Number of non-zero coefficients in LASSO: 45
```

```r
# Count the number of non-zero coefficients in the Elastic Net column
non_zero_elastic <- sum(df_results$est_elastic != 0)
cat("Number of non-zero coefficients in Elastic Net:", non_zero_elastic-1, "\n")
```

```
Number of non-zero coefficients in Elastic Net: 146
```

```r
# Count the number of variables in the ols model
num_ols_vars <- length(df_results$est_OLS) - 1  # Subtract 1 to exclude the intercept
cat("Number of variables in the OLS model:", num_ols_vars, "\n")
```

```
Number of variables in the OLS model: 147
```

```r
# Count the number of variables in the reduced ols model
num_ols_vars <- length(coef(fit_OLS_reduced)) - 1  # Subtract 1 to exclude the intercept
cat("Number of variables in the reduced OLS model:", num_ols_vars, "\n")
```

```
Number of variables in the reduced OLS model: 53
```

The OLS model includes 147 variables, which highlights that it does not perform any feature selection. This results in a full model with all predictors, which may lead to overfitting. The LASSO model selects 45 non-zero coefficients, significantly reducing the number of predictors compared to OLS. This demonstrates LASSO's capability for feature selection and inducing sparsity by shrinking less relevant coefficients to zero. The Elastic Net model retains 146 non-zero coefficients, which is close to the full set of predictors in OLS. The results highlight a clear contrast in sparsity between LASSO and Elastic Net. LASSO aggressively reduces the predictor set to a sparse subset, favoring simplicity and interpretability. On the other hand, Elastic Net provides a less sparse model that retains more predictors to capture relationships.

5 Model validation [30 points]

Take the validation sample and select only those customers who were targeted, i.e. $W_i = 1$. Using this sample, compare the observed and predicted sales outcomes.

First, compare the mean-squared error (MSE) based on the predictions of the three estimation methods.

```
cat(mse_OLS, mse_OLS_reduced, mse_lasso, mse_elastic)
```

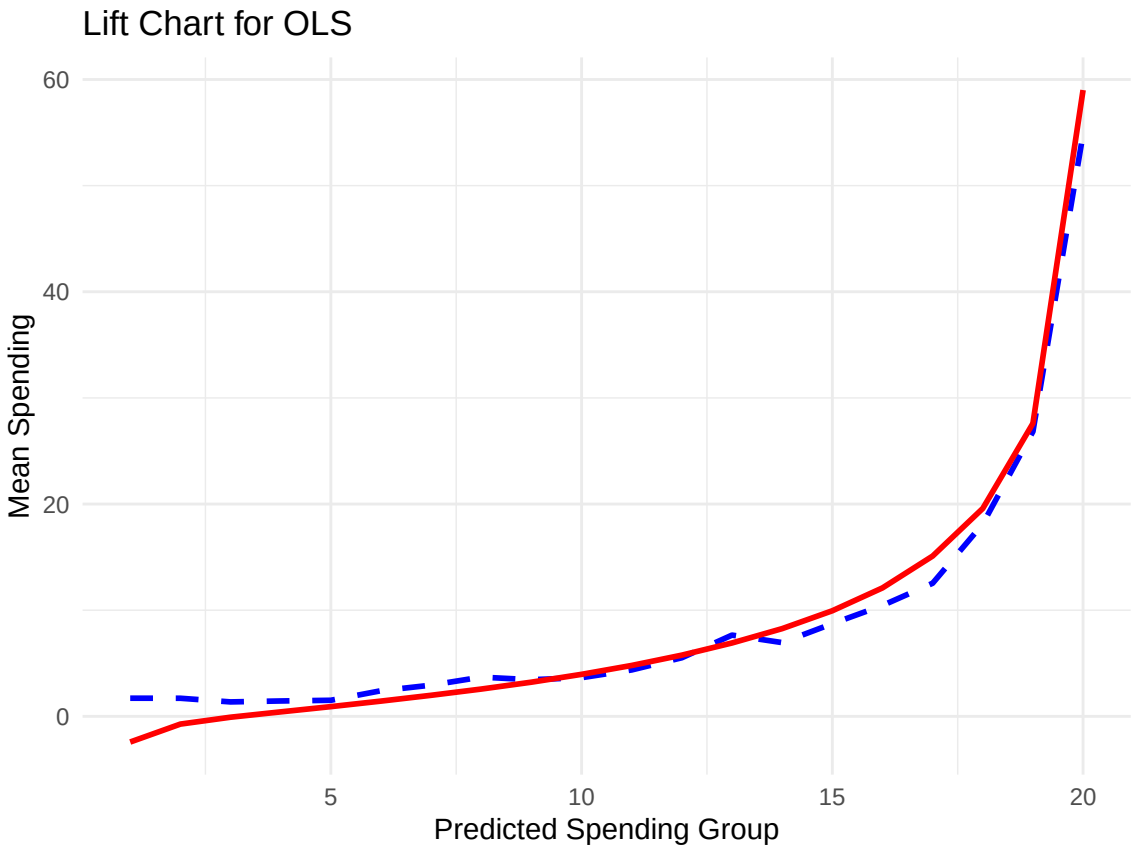
```
2201.572 1956.938 1950.222 1949.174
```

The mean squared error(MSE) is the largest for the original OLS model, showing that it has the worst predictive power among the models. The reduced OLS model improves MSE a bit by excluding irrelevant predictors. The elastic net model has the best predictive power since it has the lowest MSE value, as it achieves a balance between sparsity and capturing correlated predictors.

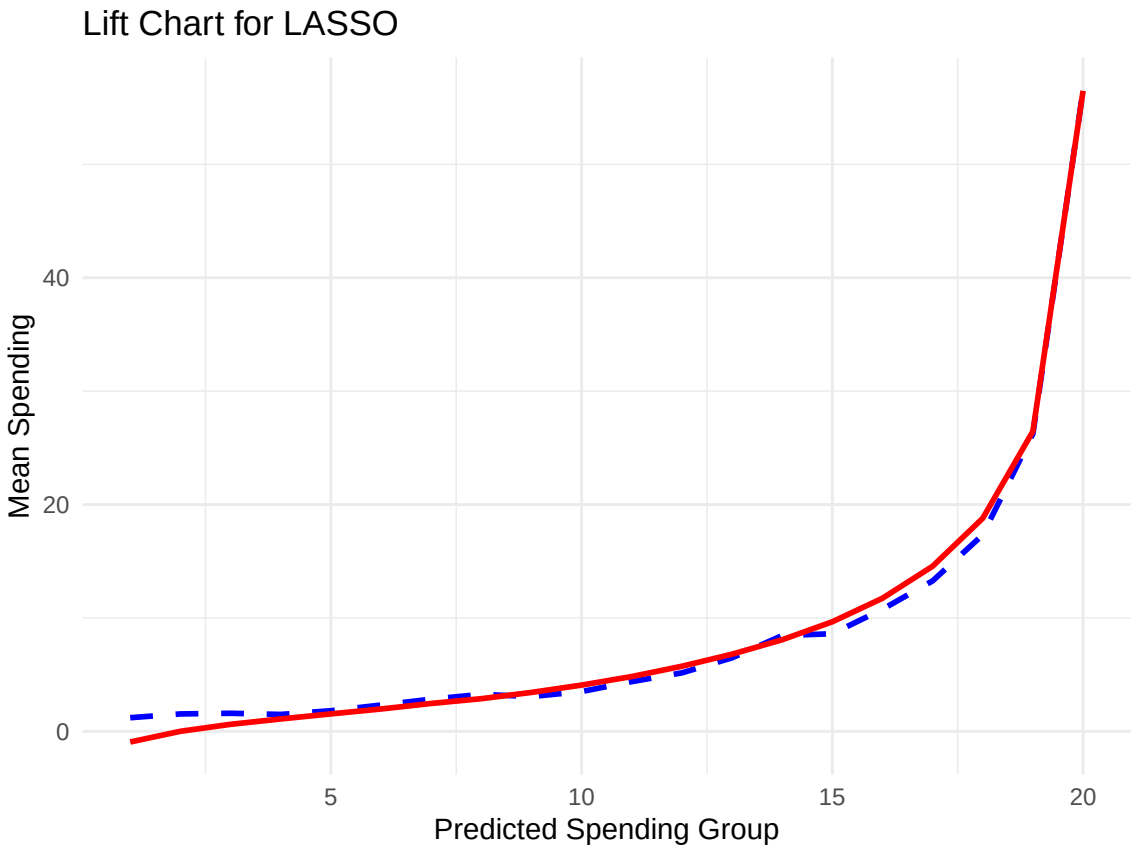
Second, create lift tables and charts (use 20 scores/groups), plot the lifts, and compare the lift tables. I recommend **not** normalizing the lifts by the average spending in the sample, so that we can directly assess the magnitude of predicted mean spending.

```
# Add predictions to testing data
lift_data <- copy(testing_data_targeted)  # Only targeted customers
lift_data[, predicted_OLS := pred_y_OLS]
lift_data[, predicted_lasso := predictions_lasso]
lift_data[, predicted_elastic := predictions_elastic]

# Function to create lift table
create_lift_table <- function(data, predicted_col, actual_col, n_groups = 20) {
  data_copy <- copy(data)  # Work with a copy to ensure original remains untouched
  data_copy[, predicted_group := ntile(get(predicted_col), n_groups)]  # Divide into groups
  lift_table <- data_copy[, .(
    mean_actual_spend = mean(get(actual_col)),    # Mean actual spending
    mean_predicted_spend = mean(get(predicted_col))  # Mean predicted spending
  ), by = predicted_group]
  return(lift_table)
}

# Create lift tables for each model
lift_table_OLS <- create_lift_table(lift_data, "predicted_OLS", "outcome_spend")
lift_table_LASSO <- create_lift_table(lift_data, "predicted_lasso", "outcome_spend")
lift_table_Elastic <- create_lift_table(lift_data, "predicted_elastic", "outcome_spend")

# Plot function for lift charts
plot_lift_chart <- function(lift_table, model_name) {
  ggplot(lift_table, aes(x = predicted_group)) +
    geom_line(aes(y = mean_actual_spend), color = "blue", linetype = "dashed", size = 1) +
    geom_line(aes(y = mean_predicted_spend), color = "red", size = 1) +
    labs(
      title = paste("Lift Chart for", model_name),
      x = "Predicted Spending Group",
      y = "Mean Spending"
    ) +
    theme_minimal()
}

# Plot lift charts
plot_OLS <- plot_lift_chart(lift_table_OLS, "OLS")
```

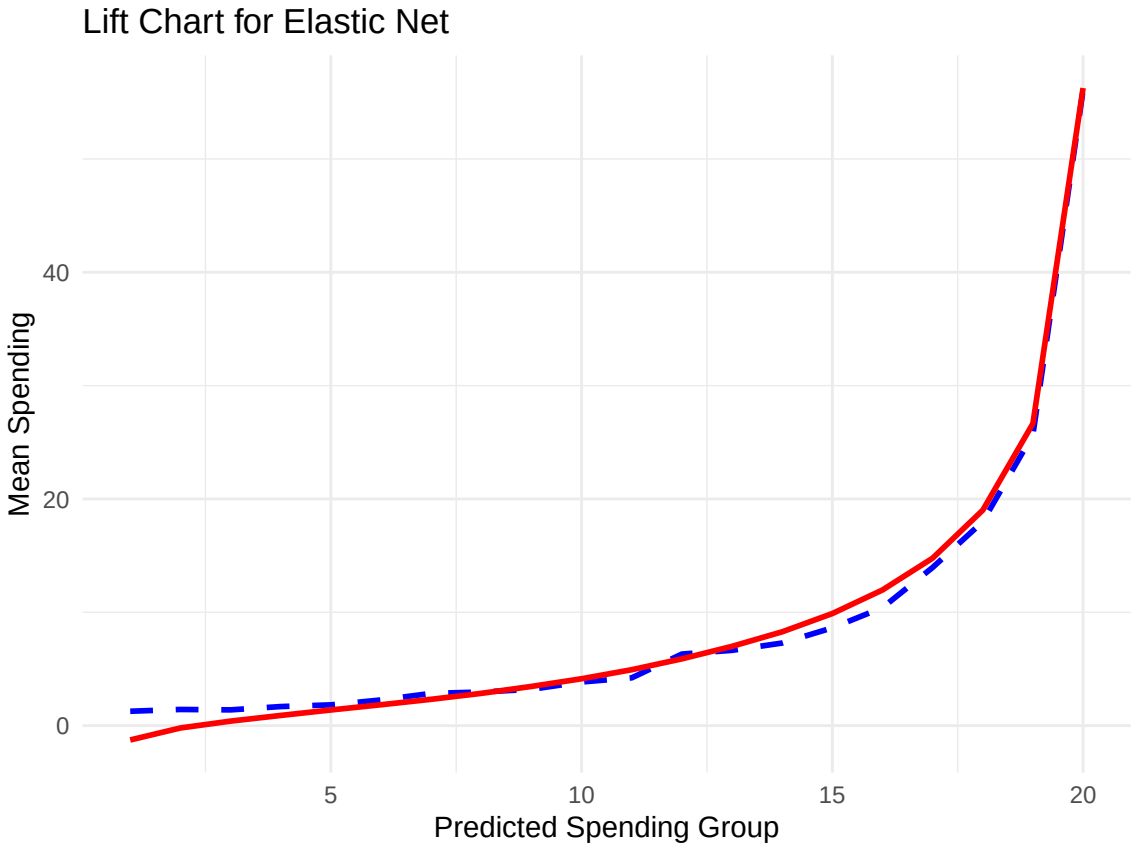
```r
plot_LASSO <- plot_lift_chart(lift_table_LASSO, "LASSO")
plot_Elastic <- plot_lift_chart(lift_table_Elastic, "Elastic Net")

# Display plots
print(plot_OLS)
```




```r
print(plot_LASSO)
```




```
print(plot_Elastic)
```



Overall, how well do the models fit?

The Elastic Net model fits the data best, with the lowest mean squared error (MSE = 1949.17), indicating it provides the most accurate predictions by effectively handling irrelevant features and multicollinearity. LASSO follows closely with an MSE of 1950.22, making it a strong alternative when simplicity is preferred. In contrast, OLS performs the worst (MSE = 2201.57), struggling to capture variance due to its lack of regularization. Therefore, Elastic Net is the recommended model for targeting customer spending in this context.

6 Traditional targeting profit prediction [20 points]

Now we work with the whole validation sample, including customers who were targeted and customers who were not targeted.

We use the preferred model that according to our previous analysis fits the data best. Using this model, we predict expected dollar spending for *all* customers in the validation sample. Consistent with standard marketing analytics practice (again think about the JCPenney example from class), we take these predictions to be indicative of what customers would spend if they were targeted, i.e.

$$\mathbb{E}(Y_i|\boldsymbol{x}_i, W_i = 1).$$

and, conversely, we assume that spending is zero whenever a customer is not targeted, $\mathbb{E}(Y_i|\boldsymbol{x}_i, W_i = 0) = 0$. (Here, we use $\boldsymbol{x}_i$ to denote the set of independent variables for customer i). Given this, we predict the expected targeting profit for each customer in the validation sample. The margin and targeting cost data are:

```
margin = 0.325          # 32.5%
cost   = 0.99           # 99 cents

# Use the best model to predict expected spending for all validation sample customers
y_validation <- testing_data$outcome_spend
X_validation <- model.matrix(outcome_spend ~ 0 + ., data = testing_data)

predicted_spending <- predict(elastic_net_model, newx = X_validation)

results_data <- data.table(
  customer_id = testing_data_reduced$customer_id,
  predicted_spending = predicted_spending
)

# Predict profit for all customers
results_data[, profit := (margin * predicted_spending) - cost]

# Determine customers to target (those with positive profit)
results_data[, target := profit > 0]

# Calculate the percentage of customers to target
percentage_to_target <- mean(results_data$target) * 100

# Print the result
print(paste("Percentage of customers to target:", round(percentage_to_target, 2), "%"))
```

```
[1] "Percentage of customers to target: 60.54 %"
```

What is the percentage of customers who should be targeted based on this analysis?

7 Incrementality [10 points]

Under what conditions is the analysis in the last section valid? In particular, what could invalidate the assumption that spending is zero whenever a customer is not targeted, i.e. $\mathbb{E}(Y_i|\boldsymbol{x}_i, W_i = 0) = 0$, in the context of our data? Discuss.

- Conditions for Validity

1. The assumption holds if the targeting intervention (e.g., catalog, ads) is the only factor influencing spending behavior. In such cases, non-targeted customers would not make purchases without being targeted.
2. The assumption is valid if customers who are not targeted share the same characteristics as those who are targeted and would behave identically in the absence of targeting.
3. If customers have no inherent propensity to spend without being targeted, $\mathbb{E}(Y_i|\boldsymbol{x}_i, W_i = 0) = 0$ is valid. This is often assumed in direct response campaigns where purchases are directly attributed to the targeting intervention.
4. The assumption is valid if the campaign's effects are confined to targeted individuals and do not influence non-targeted customers through social interactions or other indirect channels.

- Conditions That Could Invalidate the Assumption

1. If customers have a natural propensity to spend regardless of targeting (e.g., due to brand loyalty or regular purchasing habits), the assumption fails. For example, frequent buyers may continue spending even without a marketing intervention.
2. If targeting is based on customer characteristics that are also predictors of spending (e.g., targeting high-value customers), the assumption overestimates the impact of targeting. This is often the case in non-randomized targeting scenarios.
3. External influences such as holidays, economic conditions, or concurrent promotions could drive spending among non-targeted customers, invalidating the assumption.
4. If there is variation in customer behavior, with some customers spending independently of targeting while others require targeting to make purchases, the assumption does not universally hold.
5. Targeted campaigns may have indirect effects on non-targeted customers, such as through social influence, word-of-mouth, or shared household dynamics. This would result in non-zero spending for Wi= 0.