Marketing Analysis Project Report

Session 2 Group 7

Anita Banne Sui Ying Crystal Law Qian Qao

Introduction

The dataset evaluated in this project belongs to an online superstore that sells wine, fruit, meat, fish, sweets and gold products. The company has previously conducted five marketing campaigns. But the accepting rate to the campaigns have been low, which only 20% of the customers accepted marketing campaigns before.

Due to the low acceptance rate of marketing campaigns, the business question that we are try to answer is: how to increase the next marketing campaign's efficiency. In this report, we are going to show how we come up with strategies to find our target customers.We believe that targeting the right customers not only will provide a significant increase in customer engagement towards marketing campaigns, but it will also ultimately leads to an increment in profit.

The business question will be addressed using the Marketing Segmentation and Clustering Analysis, as well as Targeting and Binary Logit model.

Data Description

Table 1: A Glimpse of the dataset

Age	Education	Marital_Status	Income	Kid	Teen	Length	Recency
57	Graduation	Single	58138	0	0	664	58
60	Graduation	Single	46344	1	1	114	38
49	Graduation	Together	71613	0	0	313	26
30	Graduation	Together	26646	1	0	140	26
33	PhD	Married	58293	1	0	162	94
47	Master	Together	62513	0	1	294	16
43	Graduation	Divorced	55635	0	1	594	34
29	PhD	Married	33454	1	0	418	32
40	PhD	Together	30351	1	0	389	19
64	PhD	Together	5648	1	1	109	68
38	Basic	Married	7500	0	0	594	59
55	Graduation	Divorced	63033	0	0	227	82
62	Master	Divorced	59354	1	1	227	53
27	Graduation	Married	17323	0	0	628	38
68	PhD	Single	82800	0	0	583	23

MntWines	MntFruits	MntMeat	MntFish	MntSweet	MntGold	NumDeals_P	NumP	NumWebVisit	Accepted
635	88	546	172	88	88	3	22	7	0
11	1	6	2	1	6	2	4	5	0
426	49	127	111	21	42	1	20	4	0
11	4	20	10	3	5	2	6	6	0
173	43	118	46	27	15	5	14	5	0
520	42	98	0	42	14	2	20	6	0
235	65	164	50	49	27	4	17	6	0
76	10	56	3	1	23	2	8	8	0
14	0	24	3	3	2	1	5	9	0
28	0	6	1	1	13	1	1	20	1
6	16	11	11	1	16	1	5	8	0
194	61	480	225	112	30	1	15	2	0
233	2	53	3	5	14	3	12	6	0
3	14	17	6	1	5	1	4	8	0
1006	22	115	59	68	45	1	25	3	1

Note: The above only shows the first 15 lines of the dataset.

Variable Desciption

Variable	Variable Type	Variable Description
ID	Discrete	Unique ID number of the customer
Age	Discrete	Age of the customer
Education	Categorical	Education level of the customer
Marital_Status	Categorical	Marital status of the customer
Kid	Discrete	Number of young children in customer's household
Teen	Discrete	Number of teenagers in customer's household
Length	Discrete	Number of days since the customer's first purchase
Recency	Discrete	Number of days since the customer's last purchase
MntWines	Discrete	Amount spent on Wine products by the customer in the last 2 years
MntFruits	Discrete	Amount spent on Fruit products by the customer in the last 2 years
MntMeat	Discrete	Amount spent on Meat products by the customer in the last 2 years
MntFish	Discrete	Amount spent on Fish products by the customer in the last 2 years
MntSweet	Discrete	Amount spent on Sweet products by the customer in the last 2 years
MntGold	Discrete	Amount spent on Gold products by the customer in the last 2 years
NumDeals_P	Discrete	Total number of purchases made by customer with discount in last 2 years
NumP	Discrete	Total number of purchases made by the customer in the last 2 years
NumWebVisit	Discrete	Number of visits to company's website in the previous month
Accepted	Discrete	Binary indicator if the customer accepted marketing campaign before

Basic Descriptive Statistics of the dataset

Size of the dataset:

```
## [1] 2214   32
```

The dataset consists of 2214 rows and 20 columns.

Structure of the dataset:

```
## Classes 'data.table' and 'data.frame':   2214 obs. of  32 variables:
##  $ V1                     : int  0 1 2 3 4 5 6 7 8 9 ...
##  $ ID                     : int  5524 2174 4141 6182 5324 7446 965 6177 4855 5899 ...
##  $ Age                    : int  57 60 49 30 33 47 43 29 40 64 ...
##  $ Education              : chr  "Graduation" "Graduation" "Graduation" "Graduation" ...
##  $ Marital_Status         : chr  "Single" "Single" "Together" "Together" ...
##  $ Income                 : num  58138 46344 71613 26646 58293 ...
##  $ Kid                    : int  0 1 0 1 1 0 0 1 1 1 ...
##  $ Teen                   : int  0 1 0 0 0 1 1 0 0 1 ...
##  $ Length                 : int  664 114 313 140 162 294 594 418 389 109 ...
##  $ Recency                : int  58 38 26 26 94 16 34 32 19 68 ...
##  $ MntWines               : int  635 11 426 11 173 520 235 76 14 28 ...
##  $ MntFruits              : int  88 1 49 4 43 42 65 10 0 0 ...
##  $ MntMeat                : int  546 6 127 20 118 98 164 56 24 6 ...
##  $ MntFish                : int  172 2 111 10 46 0 50 3 3 1 ...
##  $ MntSweet               : int  88 1 21 3 27 42 49 1 3 1 ...
##  $ MntGold                : int  88 6 42 5 15 14 27 23 2 13 ...
##  $ NumDeals_P             : int  3 2 1 2 5 2 4 2 1 1 ...
##  $ NumP                   : int  22 4 20 6 14 20 17 8 5 1 ...
##  $ NumWebVisit            : int  7 5 4 6 5 6 6 8 9 20 ...
##  $ Accepted               : int  0 0 0 0 0 0 0 0 0 1 ...
##  $ Education_Basic        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Education_Graduation   : int  1 1 1 1 0 0 1 0 0 0 ...
##  $ Education_Master       : int  0 0 0 0 0 1 0 0 0 0 ...
##  $ Education_PhD          : int  0 0 0 0 1 0 0 1 1 1 ...
##  $ Marital_Status_Divorced: int  0 0 0 0 0 0 1 0 0 0 ...
##  $ Marital_Status_Married : int  0 0 0 0 1 0 0 1 0 0 ...
##  $ Marital_Status_Single  : int  1 1 0 0 0 0 0 0 0 0 ...
##  $ Marital_Status_Together: int  0 0 1 1 0 1 0 0 1 1 ...
##  $ Marital_Status_Widow   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Income_High            : int  0 0 1 0 0 0 0 0 0 0 ...
##  $ Income_Low             : int  0 0 0 1 0 0 0 1 1 1 ...
##  $ Income_Medium          : int  1 1 0 0 1 1 1 0 0 0 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

Marketing Methodologies:

In order to develop strategies to target the right customers, the below methodologies are used to solve the business question:

- First, Clustering Analysis: Group customers into segments so that all those in the same segment are similar, whereas those in different segments are different.
- Second, Logistic Regression: Create a model that can identify statistically significant demographics variables of customers accepting marketing campaigns, and thus to calculate propensity score to find the target customers.

Data Preprocessing

Problems we may find with the data:

- For a given category, it is possible that everyone spent the same amount, ie. no variation. No variation would cause problem to customer segmentation as we will not be able to segment people using a variable that indicates everyone is the same.
- Some categories might have greater variations in spending than others. The statistical method tend to give greater weight to those categories with larger variations, even though they may not be more informative about consumer preferences than categories with small variation.

  Thus, checking the variance of the variables and normalization are necessary before we move onto the analysis.
- Categorical variables do not have numerical values. Therefore, dummies have to be generated before conducting analysis.

Variance of the spending variables:

```
options(digits=4)
spend_names = names(df_spend)
diag( var( df_spend[, ..spend_names]) )
```

```
##   MntWines  MntFruits    MntMeat    MntFish   MntSweet    MntGold NumDeals_P       NumP
##  1.139e+05  1.584e+03  5.034e+04  2.999e+03  1.688e+03  2.687e+03  3.698e+00  5.195e+01
```

From the above table, there are significant differences in the column values, ie. the variances of the variables are all bigger than zero.

Since the variance of some columns are significantly bigger than the other columns. In order to avoid statistical method from giving greater weight to those categories with larger variations, all the behavioral variables need to be normalized before conducting Clustering Analysis. Normalization turns all variables to have an average value of zero and a variance of one. Thus, the varibles will contribute to the segment of customers equally after normalization.

Normalize all the spending variables:

The below shows the first 15 rows after performing normalization:

```
spend_names_z = paste0(spend_names, "_z")
df_spend[, (spend_names_z) := lapply(.SD, function(x) (x- mean(x))/sd(x)),
          .SDcols=spend_names ]
df_spend_z = df_spend %>% select(9:16)
head(df_spend_z,15)
```

```
##     MntWines_z MntFruits_z MntMeat_z MntFish_z MntSweet_z MntGold_z NumDeals_P_z   NumP_z
##  1:     0.9776      1.5481   1.68882    2.4528   1.483523   0.84942       0.3530  1.31026
##  2:    -0.8714     -0.6375  -0.71806   -0.6513  -0.634077  -0.73241      -0.1670 -1.18706
##  3:     0.3583      0.5683  -0.17874    1.3390  -0.147273  -0.03795      -0.6871  1.03278
##  4:    -0.8714     -0.5622  -0.65566   -0.5052  -0.585397  -0.75170      -0.1670 -0.90958
##  5:    -0.3914      0.4176  -0.21885    0.1521  -0.001231  -0.55879       1.3931  0.20034
##  6:     0.6368      0.3925  -0.30800   -0.6878   0.363872  -0.57808      -0.1670  1.03278
##  7:    -0.2076      0.9703  -0.01382    0.2252   0.534254  -0.32730       0.8731  0.61656
##  8:    -0.6788     -0.4114  -0.49520   -0.6330  -0.634077  -0.40447      -0.1670 -0.63210
##  9:    -0.8625     -0.6626  -0.63783   -0.6330  -0.585397  -0.80957      -0.6871 -1.04832
## 10:    -0.8210     -0.6626  -0.71806   -0.6695  -0.634077  -0.59737      -0.6871 -1.60328
## 11:    -0.8862     -0.2607  -0.69577   -0.4869  -0.634077  -0.53950      -0.6871 -1.04832
## 12:    -0.3291      0.8698   1.39464    3.4205   2.067688  -0.26943      -0.6871  0.33908
## 13:    -0.2136     -0.6124  -0.50857   -0.6330  -0.536716  -0.57808       0.3530 -0.07714
## 14:    -0.8951     -0.3109  -0.66903   -0.5782  -0.634077  -0.75170      -0.6871 -1.18706
## 15:     2.0769     -0.1100  -0.23223    0.3895   0.996718   0.01993      -0.6871  1.72648
```

Check if normalization was correctly conducted:

- The column means are computed as follow:

```
colMeans( df_spend[, ..spend_names_z] )
```

```
##   MntWines_z  MntFruits_z    MntMeat_z    MntFish_z   MntSweet_z    MntGold_z NumDeals_P_z       Num
##    5.742e-17    1.925e-17    1.656e-17   -3.224e-17   -4.758e-17   -2.097e-17   -5.797e-17   -7.254e-
```

From the above, we can see that all the column means are very small, which are considered as zero. Note that computer does not have absolute zero. A number that has 10 to the power -17 is small enough to be considered as zero in statistical software.

- The column variances are computed as follow:

```
diag( var( df_spend[, ..spend_names_z] ) )
```

```
##   MntWines_z  MntFruits_z    MntMeat_z    MntFish_z   MntSweet_z    MntGold_z NumDeals_P_z       Num
##            1            1            1            1            1            1            1
```

The variance of all behavioral variables equal to 1 after normalization.

Clustering Analysis

Basis Variables:

- The amounts spent on different category of products, as well as the number of deal products and total number of products purchased are the basis variables since they provide a good measure of consumer behavior. Thus, a group of customers who are similar based on the fact that they have similar spending habits and product preferences.

K-Means with 3 segments:

```
set.seed(42)
km <- kmeans( df_spend[, ..spend_names_z], 3)
df_spend[, seg := km$cluster]
cbind(km$size,km$centers)
```

```
##        MntWines_z MntFruits_z MntMeat_z MntFish_z MntSweet_z MntGold_z NumDeals_P_z  NumP_z
## 1 1065    -0.7562     -0.5222  -0.63411   -0.5456    -0.5264   -0.5560      -0.2129 -0.8864
## 2  573     0.8152      1.1820   1.23207    1.2401     1.1857    0.7303      -0.4710  1.0054
## 3  576     0.5873     -0.2104  -0.05321   -0.2247    -0.2063    0.3015       0.8623  0.6387
```

The results of the 3 segments K-means shows that

- The first segment is the largest, which consists of customers who are not very active in any of the 6 product categories, including wine, fruits, meat, fish, sweets and gold. In addition, this segment of customers purchased the least amount of products from the company in the last 2 years.
- The second segment has comparable size as the third segment. This segment consists of customers who purchase the most out of all product categories considering amount spending and total number of products bought. They are also the least interested in making deals purchases out of the three segments of customers.
- The third segment has size about half of the first segment, ie. about 500 customers. Having a value of `NumP_z` that ranks the second out of the three segments illustrates that customers in this segment are fairly active in product purchase. This group of customers are particularly interested in making deals purchases among all the segments. Also, they are more interested in wine and gold products than fruits, meat, fish and sweet products.

K-Means with 4 segments:

```
set.seed(42)
km <- kmeans( df_spend[, ..spend_names_z], 4)
df_spend[, seg := km$cluster]
cbind(km$size,km$centers)
```

```
##        MntWines_z MntFruits_z MntMeat_z MntFish_z MntSweet_z MntGold_z NumDeals_P_z  NumP_z
## 1 1097    -0.7318     -0.5164   -0.6279   -0.5393    -0.5171   -0.5414      -0.2120 -0.8480
## 2  517     1.0631      0.1702    0.6020    0.1937     0.1233    0.4419      -0.2404  0.9842
## 3  252     0.1472     -0.3385   -0.1941   -0.3326    -0.3099    0.2347       2.0287  0.3572
## 4  348     0.6210      1.6202    1.2253    1.6532     1.6711    0.8803      -0.4435  0.9522
```

The results of the 4 segments K-means shows that

- The first segment is the largest, which consists of customers who are not very active in any of the 6 product categories, including wine, fruits, meat, fish, sweets and gold. In addition, this segment of customers purchased the least amount of products from the company in the last 2 years.
- The second segment consists of customers who purchase the most total number of products purchased out of the 4 segments. Customers in this segment are fairly active and are more interested in wine products than fruits, meat, fish, sweet and gold.
- The third segment is the smallest. Having a value of `NumP_z` that ranks the third out of the four segments illustrates that customers in this segment are fairly active in product purchase. This group of customers are particularly interested in making deals purchases among all the segments. Also, they are more interested in wine and gold products than fruits, meat, fish and sweet products.
- The forth segment is the second smallest, which consists of customers who are very active in 5 out of the 6 product categories. The amount that this segment spent on fruits, meat, fish, sweets and gold are the highest among the four segments. Also, the total number of purchases that this segment made are comparable to that of the second segment, in which the customer made the most number of purchases in last 2 years. Customers in this segment are also the least interested in making deals purchases out of the four segments of customers.

Choose the optimize K:

Comparison between 3 and 4 segments of K-means segmentation

Customers in the second and forth segments make the most number of purchases and are the least interested making deals purchases among the 4 segments. On the other hand, the amount that customers in these two segments spent on all of the 6 product categories is the highest among all segments. Therefore, K-means with 3 segments is adequate to differentiate customers spending behaviors.

Demographics information of the segments

```
##    seg    N
## 1:   1 1065
## 2:   2  573
## 3:   3  576
```

The below are the 3 segments according to their spending behavior:

```
b=df[, lapply(.SD, mean), .SDcols = behavior_names_z, seg][order(seg)]
print(t(b))
```

```
##                 [,1]    [,2]     [,3]
## seg           1.0000  2.0000  3.00000
## MntWines_z   -0.7562  0.8152  0.58732
## MntFruits_z  -0.5222  1.1820 -0.21036
## MntMeat_z    -0.6341  1.2321 -0.05321
## MntFish_z    -0.5456  1.2401 -0.22474
## MntSweet_z   -0.5264  1.1857 -0.20631
## MntGold_z    -0.5560  0.7303  0.30155
## NumDeals_P_z -0.2129 -0.4710  0.86225
## NumP_z       -0.8864  1.0054  0.63872
```

The demographic information of the 3 segments are as follow:

```
b=df[, lapply(.SD, mean), .SDcols = demo_names, seg][order(seg)]
print(t(b))
```

```
##             [,1]      [,2]      [,3]
## seg    1.000e+00 2.000e+00 3.000e+00
## Age    4.318e+01 4.590e+01 4.819e+01
## Income 3.673e+04 7.467e+04 5.865e+04
## Kid    7.305e-01 4.887e-02 3.003e-01
## Teen   4.770e-01 2.426e-01 8.177e-01
```

Considering all the numerical demographics variables, we can see that there is no big difference in their average `Age` across customers in 3 segments, with an average age between 43 to 48 years old.

Among the 3 segments, customers in segment 2 has the highest average annual income, ie. about 70,000 dollars. Customers in segment 1 earns the least out of the 3 segments, with average annual income about 36000 dollars.

On the other hand, in terms of number of kids and teens in the household. Customers in segment 2 has the least average number of kids and teens among the three segments, where as customers in segment 1 have the highest average number of kid and teens.

From the above numerical demographics information of the 3 segments, we can see that the 3 segments are most different in Income, Kid and Teen.

Target and Binary Logit Model

After conducting customer segmentation, we conducted logistic regression with customers' demographics information to see what are the statistically significant predictors of them accepting marketing campaigns. Other than `Age`, `Income`, `Kid` and `Teen` that we just discussed, we also added `Education` and `Marital_Status` in the model.

In order to include the categorical variables into the model, we have turned the values of each categorical columns into dummy variables. For example, `Education_Basic`, `Education_Graduation`, `Marital_Status_Married`, which take values of 0 and 1.

The below shows a glimpse of the demographics variables after changing the categorical columns into dummy variables.

```
str(df_dummies)
```

```
## Classes 'data.table' and 'data.frame':   2214 obs. of  14 variables:
##  $ Age                    : int  57 60 49 30 33 47 43 29 40 64 ...
##  $ Income                 : num  58138 46344 71613 26646 58293 ...
##  $ Kid                    : int  0 1 0 1 1 0 0 1 1 1 ...
##  $ Teen                   : int  0 1 0 0 0 1 1 0 0 1 ...
##  $ Education_Basic        : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Education_Graduation   : int  1 1 1 1 0 0 1 0 0 0 ...
##  $ Education_Master       : int  0 0 0 0 0 1 0 0 0 0 ...
##  $ Education_PhD          : int  0 0 0 0 1 0 0 1 1 1 ...
##  $ Marital_Status_Divorced: int  0 0 0 0 0 0 1 0 0 0 ...
##  $ Marital_Status_Married : int  0 0 0 0 1 0 0 1 0 0 ...
##  $ Marital_Status_Single  : int  1 1 0 0 0 0 0 0 0 0 ...
##  $ Marital_Status_Together: int  0 0 1 1 0 1 0 0 1 1 ...
##  $ Marital_Status_Widow   : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Accepted               : int  0 0 0 0 0 0 0 0 0 1 ...
##  - attr(*, ".internal.selfref")=<externalptr>
```

Results:

```
d = data.frame(y=df_dummies[,13],df_dummies[,1:14])
bl_result = glm(Accepted ~ Age + Income + Kid + Teen + Education_Basic +
                Education_Graduation    + Education_Master +
                  Marital_Status_Divorced + Marital_Status_Married +
                  Marital_Status_Single + Marital_Status_Together,
                data=d, family="binomial")
summary(bl_result)
```

```
##
## Call:
## glm(formula = Accepted ~ Age + Income + Kid + Teen + Education_Basic +
##     Education_Graduation + Education_Master + Marital_Status_Divorced +
##     Marital_Status_Married + Marital_Status_Single + Marital_Status_Together,
##     family = "binomial", data = d)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -5.488  -0.689  -0.501  -0.366   2.557
```

```
## 
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)    
## (Intercept)             -2.30e+00   4.54e-01   -5.07  4.0e-07 ***
## Age                     -2.16e-04   4.85e-03   -0.04  0.96445    
## Income                   2.72e-05   3.23e-06    8.41  < 2e-16 ***
## Kid                     -5.18e-01   1.40e-01   -3.71  0.00021 ***
## Teen                    -4.58e-01   1.11e-01   -4.14  3.5e-05 ***
## Education_Basic          5.11e-02   4.70e-01    0.11  0.91352    
## Education_Graduation    -1.41e-01   1.39e-01   -1.01  0.31079    
## Education_Master        -1.42e-01   1.60e-01   -0.89  0.37516    
## Marital_Status_Divorced -9.60e-02   3.33e-01   -0.29  0.77300    
## Marital_Status_Married  -6.34e-04   2.99e-01    0.00  0.99831    
## Marital_Status_Single   -8.34e-02   3.12e-01   -0.27  0.78950    
## Marital_Status_Together -9.30e-02   3.06e-01   -0.30  0.76085    
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
## 
## (Dispersion parameter for binomial family taken to be 1)
## 
##     Null deviance: 2260.0  on 2213  degrees of freedom
## Residual deviance: 2040.6  on 2202  degrees of freedom
## AIC: 2065
## 
## Number of Fisher Scoring iterations: 5
```

```
logLik(bl_result)
```

```
## 'log Lik.' -1020 (df=12)
```

Alternative model 1:

In this alternative model, the customers' income levels are split into 3 categories, low, medium and high, which are obtained by breaking the range of `Income` into 3 group. Low corresponds to customers who have income ranging from 0 to 33.33 percentile; medium 33.33 to 66.7 percentile, and last but not least, high 66.7 to 100 percentile. Each of the income level column takes binary indicator of 0 and 1, with 1 representing their income is within the specified range of the column.

After breaking Income in 3 categories, the result is as follow:

```
bl_result2 = glm(Accepted ~ Age + Income_High + Income_Medium + Kid + Teen + Education_Basic +
                   Education_Graduation + Education_Master + Marital_Status_Divorced +
                   Marital_Status_Married + Marital_Status_Single + Marital_Status_Together,
                   data=df, family="binomial")
summary(bl_result2)
```

```
## 
## Call:
## glm(formula = Accepted ~ Age + Income_High + Income_Medium + 
##     Kid + Teen + Education_Basic + Education_Graduation + Education_Master + 
##     Marital_Status_Divorced + Marital_Status_Married + Marital_Status_Single + 
##     Marital_Status_Together, family = "binomial", data = df)
## 
```

```
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.128  -0.645  -0.486  -0.342   2.440
##
## Coefficients:
##                           Estimate Std. Error z value Pr(>|z|)
## (Intercept)              -1.751984   0.431482   -4.06  4.9e-05 ***
## Age                      -0.000133   0.004862   -0.03   0.9781
## Income_High               1.585711   0.183762    8.63  < 2e-16 ***
## Income_Medium             0.796978   0.188269    4.23  2.3e-05 ***
## Kid                      -0.428313   0.140164   -3.06   0.0022 **
## Teen                     -0.503704   0.118198   -4.26  2.0e-05 ***
## Education_Basic          -0.036236   0.470194   -0.08   0.9386
## Education_Graduation     -0.171754   0.139031   -1.24   0.2167
## Education_Master         -0.164431   0.160140   -1.03   0.3045
## Marital_Status_Divorced -0.061758   0.334225   -0.18   0.8534
## Marital_Status_Married   0.052600   0.301030    0.17   0.8613
## Marital_Status_Single   -0.008707   0.314086   -0.03   0.9779
## Marital_Status_Together -0.032799   0.307199   -0.11   0.9150
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2260.0  on 2213  degrees of freedom
## Residual deviance: 2034.5  on 2201  degrees of freedom
## AIC: 2060
##
## Number of Fisher Scoring iterations: 5
```

```r
logLik(bl_result2)
```

```
## 'log Lik.' -1017 (df=13)
```

Alternative model 2:

In this alternative model, the customers' age are further split into 4 categories - `Ageless25`, `Age25_39`, `Age39_64`, `Age64plus`. Each of the age level column takes binary indicator of 0 and 1, with 1 representing their age is within the specified range of the column.

```r
bl_result3 = glm(Accepted ~ Ageless25 + Age25_39 + Age39_64 + Income_High +  Income_Medium +
                 Kid + Teen + Education_Basic +  Education_Graduation   + Education_Master +
                Marital_Status_Divorced +   Marital_Status_Married +  Marital_Status_Single +
                  Marital_Status_Together, data=df3, family="binomial")
summary(bl_result3)
```

```
##
## Call:
## glm(formula = Accepted ~ Ageless25 + Age25_39 + Age39_64 + Income_High +
##     Income_Medium + Kid + Teen + Education_Basic + Education_Graduation +
##     Education_Master + Marital_Status_Divorced + Marital_Status_Married +
##     Marital_Status_Single + Marital_Status_Together, family = "binomial",
##     data = df3)
##
## Deviance Residuals:
##    Min      1Q  Median      3Q     Max
## -1.216  -0.645  -0.489  -0.345   2.424
##
## Coefficients:
##                          Estimate Std. Error z value Pr(>|z|)
## (Intercept)             -1.619457   0.367124   -4.41  1.0e-05 ***
## Ageless25                0.038057   0.353652    0.11   0.9143
## Age25_39                -0.218397   0.197776   -1.10   0.2695
## Age39_64                -0.206826   0.191519   -1.08   0.2802
## Income_High              1.581449   0.183494    8.62  < 2e-16 ***
## Income_Medium            0.793140   0.188034    4.22  2.5e-05 ***
## Kid                     -0.405421   0.139802   -2.90   0.0037 **
## Teen                    -0.481930   0.121154   -3.98  7.0e-05 ***
## Education_Basic         -0.049850   0.471826   -0.11   0.9159
## Education_Graduation    -0.157662   0.139877   -1.13   0.2597
## Education_Master        -0.154992   0.160667   -0.96   0.3347
## Marital_Status_Divorced -0.011365   0.335570   -0.03   0.9730
## Marital_Status_Married   0.090889   0.301294    0.30   0.7629
## Marital_Status_Single    0.020072   0.314562    0.06   0.9491
## Marital_Status_Together -0.000699   0.307786    0.00   0.9982
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 2260.0  on 2213  degrees of freedom
## Residual deviance: 2032.5  on 2199  degrees of freedom
## AIC: 2063
##
## Number of Fisher Scoring iterations: 5
```

```r
logLik(bl_result3)
```

```
## 'log Lik.' -1016 (df=15)
```

Interpretation:

We would like to to calculate the marginal impact of each variable. Holding all other variables constant, adding 1 to the designated variable, the probability changes are as follow: (only those statistically significant demographics variables are being evaluated below)

- Having high income increases the probability of accepting the marketing campaign by 32.52% compare to those who have low income.
- Having medium income increases the probability of accepting the marketing campaign by 13.91% compare to those that have low income.
- For every number increase in `Kid`, the probability of accepting the marketing campaign decreases by 4.866%.
- For every number increase in `Teen`, the probability of accepting the marketing campaign decreases by 5.63%

Out-of-sample fit of original data

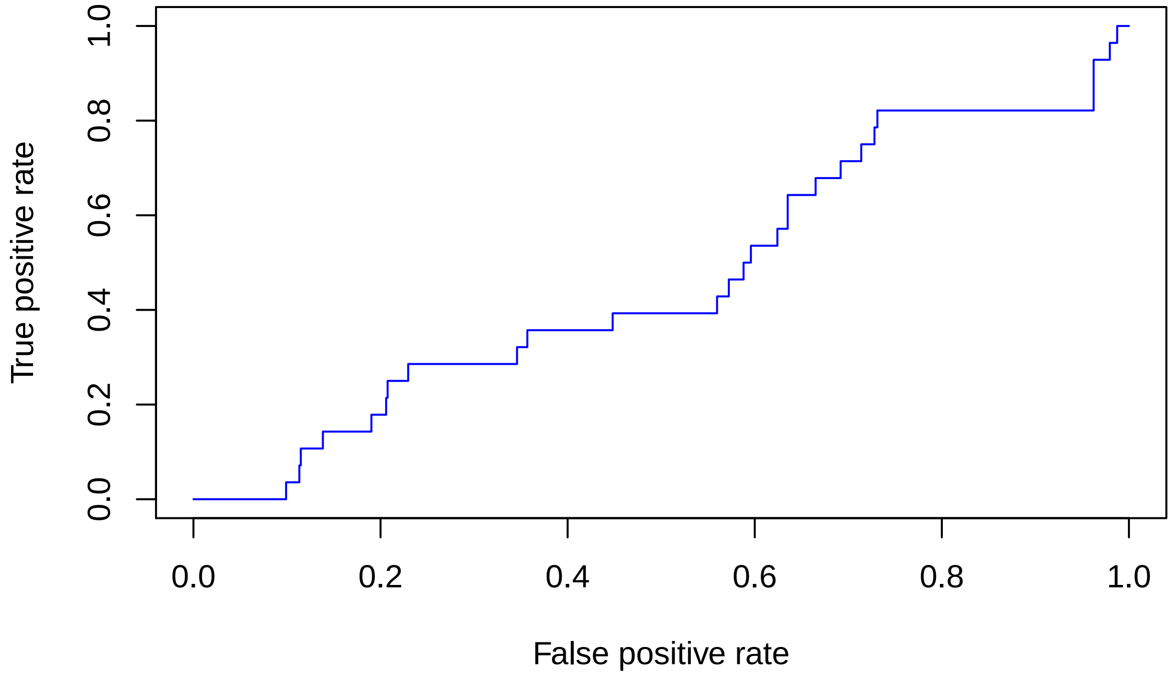
```
df_dummies[,fittedval:=bl_result3$fitted.values]
a3=df_dummies[,.(mnfit=mean(fittedval)),by=Accepted]
a3
```

```
##    Accepted  mnfit
## 1:        0 0.1852
## 2:        1 0.2920
```

We used our best logit model to calculate the propensity score of the customers. Propensity score measures the probability a customer is going to accept the marketing campaign. Among those who accepted, the model predicted average acceptance probability: 0.2920, which is only a little higher than the predicted average probability of those who did not: 0.1852. It is not a very good result.

AUC Score

```
pred <- prediction(predTst,yActual)
perf <- performance(pred,"tpr","fpr")
plot(perf,col="blue")
```



```
perf <- performance(pred,measure="auc")
print(paste("AUC= ", perf@y.values[[1]]))
```

```
## [1] "AUC=  0.46271338724169"
```

The above graph shows the accuracy predicting the model using ROC curve. With an AUC = 0.46, the model is not predicting values very well. The x-axis of the graph shows false positive rate and the y-axis shows true positive rate. In this case, our model is predicting true positives and false positives at almost the same rate, which indicates our model has no predicting power at all.

Imbalanced data

The dataset contains 2 variables of a total of 2214 data points, which 1550 of them will be used as training data. Accepted is the response variable that takes value of 0 and 1. The below shows that the severity of imbalanced data occurs in this dataset.

```
table(df_dummies_train$Accepted)
```

```
##
##    0    1
## 1230  320
```

```
prop.table(table(df_dummies_train$Accepted))
```

```
##
##      0      1
## 0.7935 0.2065
```

With only about 20% of the customers accepted marketing campaigns before verses about 80% who have never accepted. The algorithm doesn't get necessary information about those who have accepted marketing campaign before for accurate prediction. Therefore, it is necessary to balance the data before applying the algorithm.

Applying over-sampling

```
balanced_over <- ovun.sample(Accepted ~ ., data=df_dummies_train, method = "over" ,
                             N=2460, seed=1)$data
table(balanced_over$Accepted)
```

```
##
##    0    1
## 1230 1230
```

The solution that we used to deal with the imbalance data in our dataset is a package called "ROSE" in R. It helps with balancing the data using oversampling method, which duplicates the sample from the minority class. In this case, 1. After performing oversampling, the count of our 1s increases from 320 to 1230 and making our outcomes 1 and 0 having equal proportion.

Propensity score of balanced data

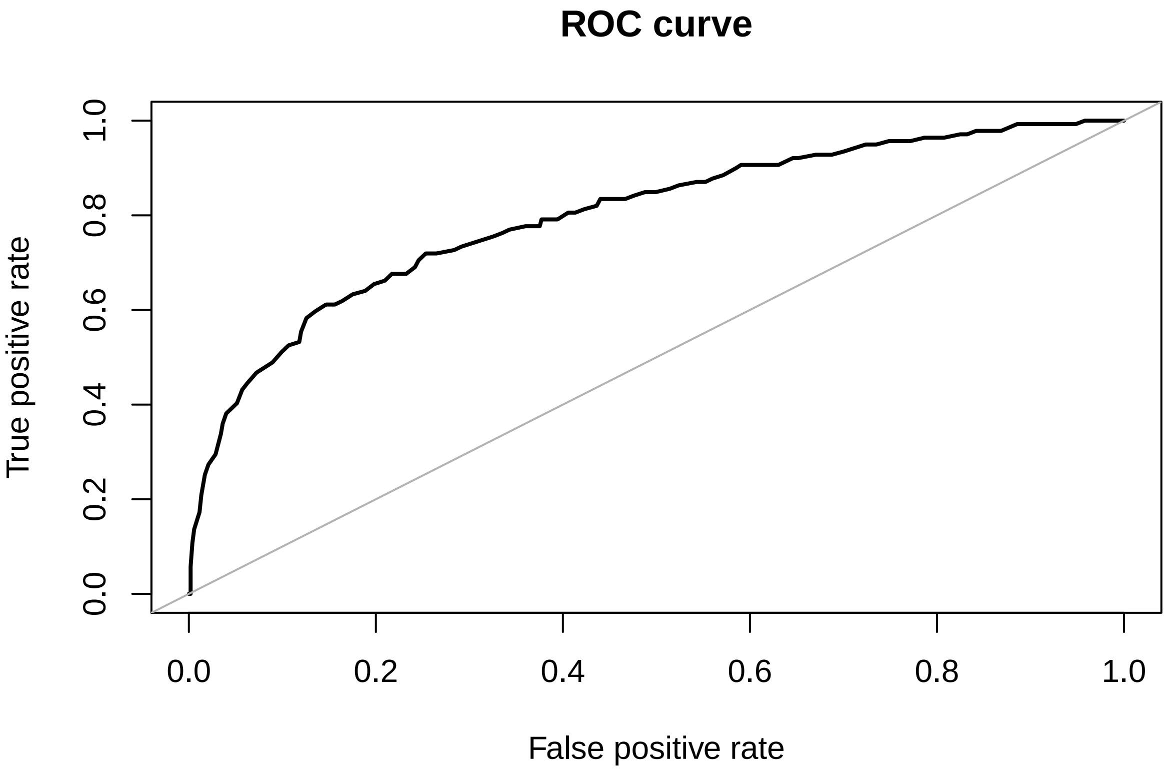
```
bl_over = glm(Accepted~ .,, family="binomial",data=balanced_over)
balanced_over<- as.data.table(balanced_over)
balanced_over[,fittedval :=bl_over$fitted.values]
a2 = balanced_over[,.(mnfit=mean(fittedval)),by=Accepted]
a2
```

```
##    Accepted  mnfit
## 1:        0 0.3451
## 2:        1 0.6549
```

After balancing the data, we used our logit model again to calculate the propensity score of the customers. This time, among those who accepted, the model predicted average acceptance probability: 0.6549, which is about 2 times than the predicted average probability of those who did not.

Out-of-sample fit of balanced data

```r
pred.over <-predict(bl_over, newdata=df_dummies_test)
roc.curve(df_dummies_test$Accepted,pred.over)
```



```
## Area under the curve (AUC): 0.799
```

The above graph shows the accuracy predicting the model using ROC curve after balancing the data. With an AUC of about 0.8, the model is predicting values better than before we balanced the data. The x-axis of the graph shows false positive rate and the y-axis shows true positive rate. In this case, our model is predicting true positives at a faster rate than that of false positives.

Target the right customers

After that we have set a threshold for propensity score of 0.5. Customers with propensity score of higher than 0.5 will be our target customers.

The below are the first 10 individuals' Propensity Score:

```
bl_over$fitted.values[1:10]
```

```
##      1      2      3      4      5      6      7      8      9     10 
## 0.7170 0.2248 0.5540 0.2581 0.2135 0.5905 0.1532 0.3695 0.3731 0.2805
```

Number of target customers in the dataset:

```
z = (bl_over$fitted.values > 0.5)
length(z[z== TRUE])
```

```
## [1] 1090
```

After computing all customers' propensity scores in the dataset, we found out that about 45 percentage of the customers have propensity score of higher than 0.5, and therefore are our target customers. Previously, we only had about 20% of the customers accepting marketing campaigns. We believe that with the help of propensity score, it will help identifying target customers, further increase marketing campaign efficiency and ultimately increasing profit.

Conclusion

We have conducted customer segmentation to identify 3 group of customers - active customers, fairly active customers and least active customers. Each of the segment is different in their spending habits and product preferences.

Other than that, we have created a logit model to calculate propensity score to target the right customers. We believe that with the help of propensity score, it will help identifying target customers, further increase marketing campaign efficiency and ultimately increasing profit.