

Marketing Campaign Analysis

Marshal Multani

March 2021

Introduction

In this project, a thorough analysis of a retail food company's marketing campaign is presented. It aims to understand the campaign's interactions with its target audience, find business opportunities and insights, and to propose any data-driven actions to maximize the optimal results of the campaign and generate value to the company.

Products from 5 major categories are sold: wines, rare meat products, exotic fruits, specially prepared fish, and sweet products. These can further be divided into "gold" and "regular" products. The customers can order and acquire products through 3 sales channels: physical stores, catalogs, and the company's website.

Objective(s)

The key objectives are:

1. *EDA*: explore the data to understand the characteristic features of the respondents to the previous marketing campaigns by the company, to make better execution of the forthcoming one.
2. *Regression analysis*: build a regression model to identify significant factors that influence the number of store purchases by the respondents. Also, compare the performance of the previous campaigns by their respective geographical regions.
3. *Visualization*: plot and visualize the performances of the campaigns and individual products.

Dataset

The dataset contains socio-demographic and firmographic features of 2,240 customers. Additionally, it contains binary flags for those customers that responded to the campaign by buying a product.

```
df <- read.csv("marketing_data.csv")
dim(df)
```

```
## [1] 2240 28
```

```
head(df)
```

##	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome
## 1	1826	1970	Graduation	Divorced	\$84,835.00	0	0
## 2	1	1961	Graduation	Single	\$57,091.00	0	0
## 3	10476	1958	Graduation	Married	\$67,267.00	0	1
## 4	1386	1967	Graduation	Together	\$32,474.00	1	1
## 5	5371	1989	Graduation	Single	\$21,474.00	1	0
## 6	7348	1958	PhD	Single	\$71,691.00	0	0

##	Dt_Customer	Recency	MntWines	MntFruits	MntMeatProducts	MntFishProducts
## 1	6/16/14	0	189	104	379	111
## 2	6/15/14	0	464	5	64	7
## 3	5/13/14	0	134	11	59	15
## 4	5/11/14	0	10	0	1	0
## 5	4/8/14	0	6	16	24	11
## 6	3/17/14	0	336	130	411	240

##	MntSweetProducts	MntGoldProds	NumDealsPurchases	NumWebPurchases
## 1	189	218	1	4
## 2	0	37	1	7
## 3	2	30	1	3
## 4	0	0	1	1
## 5	0	34	2	3
## 6	32	43	1	4

##	NumCatalogPurchases	NumStorePurchases	NumWebVisitsMonth	AcceptedCmp3
## 1	4	6	1	0
## 2	3	7	5	0
## 3	2	5	2	0
## 4	0	2	7	0
## 5	1	2	7	1
## 6	7	5	2	0

##	AcceptedCmp4	AcceptedCmp5	AcceptedCmp1	AcceptedCmp2	Response	Complain	Country
## 1	0	0	0	0	1	0	SP
## 2	0	0	0	1	1	0	CA
## 3	0	0	0	0	0	0	US
## 4	0	0	0	0	0	0	AUS
## 5	0	0	0	0	1	0	SP
## 6	0	0	0	0	1	0	SP

The “Income” column in the data frame is of “chr” data type containing commas and the Dollar (\$) sign. To apply any arithmetic operation on it for the analysis, it needs to be *coerced* to a numeric data type by performing string replacement.

The “Dt_Customer” column is also of “chr” data type. This needs to be coerced to “Date” type.

```
df$Income <- str_replace_all(df$Income,"([$,])", "")
df$Income <- as.numeric(df$Income)
df$Dt_Customer <- as.Date(df$Dt_Customer,format = '%m/%d/%Y')
head(df)
```

##	ID	Year_Birth	Education	Marital_Status	Income	Kidhome	Teenhome
## 1	1826	1970	Graduation	Divorced	84835	0	0
## 2	1	1961	Graduation	Single	57091	0	0
## 3	10476	1958	Graduation	Married	67267	0	1
## 4	1386	1967	Graduation	Together	32474	1	1
## 5	5371	1989	Graduation	Single	21474	1	0
## 6	7348	1958	PhD	Single	71691	0	0

##	Dt_Customer	Recency	MntWines	MntFruits	MntMeatProducts	MntFishProducts
## 1	0014-06-16	0	189	104	379	111
## 2	0014-06-15	0	464	5	64	7
## 3	0014-05-13	0	134	11	59	15
## 4	0014-05-11	0	10	0	1	0
## 5	0014-04-08	0	6	16	24	11
## 6	0014-03-17	0	336	130	411	240

```
##      MntSweetProducts MntGoldProds NumDealsPurchases NumWebPurchases
## 1             189           218                1                4
## 2              0            37                1                7
## 3              2            30                1                3
## 4              0             0                1                1
## 5              0            34                2                3
## 6             32            43                1                4
##      NumCatalogPurchases NumStorePurchases NumWebVisitsMonth AcceptedCmp3
## 1              4              6                1                0
## 2              3              7                5                0
## 3              2              5                2                0
## 4              0              2                7                0
## 5              1              2                7                1
## 6              7              5                2                0
##      AcceptedCmp4 AcceptedCmp5 AcceptedCmp1 AcceptedCmp2 Response Complain Country
## 1              0              0              0              0         1         0      SP
## 2              0              0              0              1         1         0      CA
## 3              0              0              0              0         0         0      US
## 4              0              0              0              0         0         0     AUS
## 5              0              0              0              0         1         0      SP
## 6              0              0              0              0         1         0      SP
```

Exploratory Data Analysis (EDA)

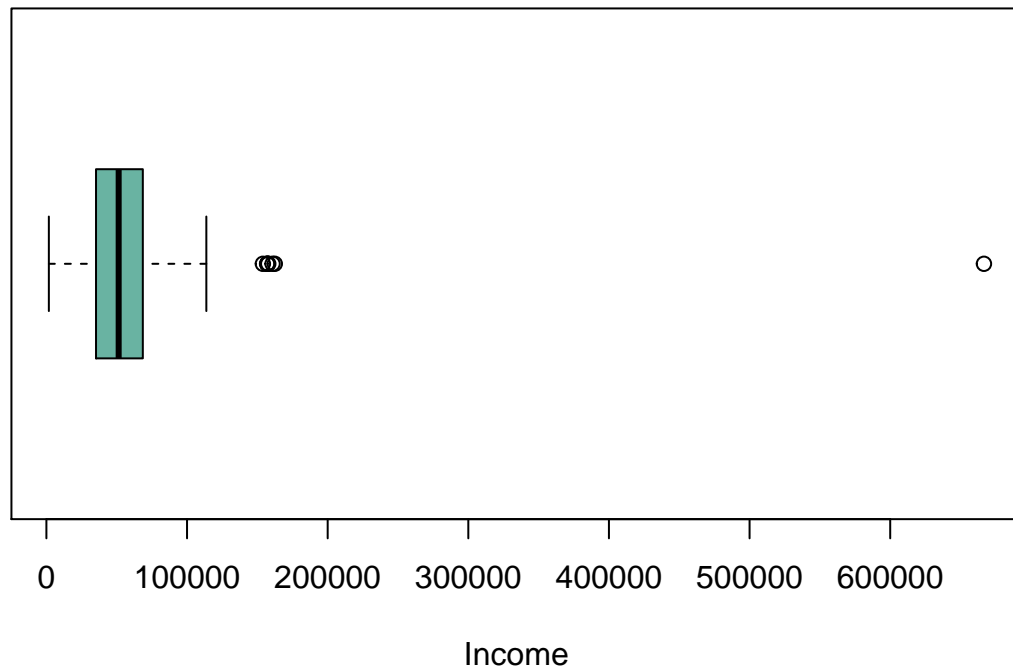
Duplicates, Outliers and Null Values Count the number of duplicate values that may be in the data frame. Also, Identify features that contain NULL values. Then, using the distribution of any such feature can help to replace the NULL value with the *median* value to avoid the effects of outliers on the imputation value.

```
sapply(df, function(df) sum(is.na(df)))
```

```
##              ID          Year_Birth          Education          Marital_Status
##              0              0              0              0
##              Income          Kidhome          Teenhome          Dt_Customer
##              24              0              0              0
##              Recency          MntWines          MntFruits          MntMeatProducts
##              0              0              0              0
##      MntFishProducts      MntSweetProducts      MntGoldProds      NumDealsPurchases
##              0              0              0              0
##      NumWebPurchases NumCatalogPurchases      NumStorePurchases      NumWebVisitsMonth
##              0              0              0              0
##      AcceptedCmp3      AcceptedCmp4      AcceptedCmp5      AcceptedCmp1
##              0              0              0              0
##      AcceptedCmp2      Response      Complain      Country
##              0              0              0              0
```

The 'Income' column contains 24 NULL values. It can be replaced by the median Income.

```
boxplot(df$Income , col="#69b3a2" , xlab="Income",horizontal = TRUE)
```



```
# Set the outlier, which may be a data entry error, in the Income column to N/A.
df <- df %>%
  mutate_at(vars(Income), na_if, 666666)
max(df$Income, na.rm = TRUE)
```

```
## [1] 162397
```

```
# Set the n/a entries in Income to the median income.
df$Income[is.na(df$Income)]<-median(df$Income, na.rm = TRUE)
# check for duplicate values
sum(anyDuplicated(df))
```

```
## [1] 0
```

There are no duplicate values in the data frame.

Feature Engineering Review a list of variables in the data frame which can be combined to create new useful variables for the analysis.

```
str(df)
```

```
## 'data.frame': 2240 obs. of 28 variables:
## $ ID : int 1826 1 10476 1386 5371 7348 4073 1991 4047 9477 ...
```

```
## $ Year_Birth      : int  1970 1961 1958 1967 1989 1958 1954 1967 1954 1954 ...
## $ Education      : chr   "Graduation" "Graduation" "Graduation" "Graduation" ...
## $ Marital_Status : chr   "Divorced" "Single" "Married" "Together" ...
## $ Income         : num   84835 57091 67267 32474 21474 ...
## $ Kidhome        : int    0 0 0 1 1 0 0 0 0 0 ...
## $ Teenhome       : int    0 0 1 1 0 0 0 1 1 1 ...
## $ Dt_Customer    : Date, format: "0014-06-16" "0014-06-15" ...
## $ Recency        : int    0 0 0 0 0 0 0 0 0 0 ...
## $ MntWines       : int   189 464 134 10 6 336 769 78 384 384 ...
## $ MntFruits      : int   104 5 11 0 16 130 80 0 0 0 ...
## $ MntMeatProducts : int   379 64 59 1 24 411 252 11 102 102 ...
## $ MntFishProducts : int   111 7 15 0 11 240 15 0 21 21 ...
## $ MntSweetProducts : int   189 0 2 0 0 32 34 0 32 32 ...
## $ MntGoldProds   : int   218 37 30 0 34 43 65 7 5 5 ...
## $ NumDealsPurchases : int    1 1 1 1 2 1 1 1 3 3 ...
## $ NumWebPurchases : int    4 7 3 1 3 4 10 2 6 6 ...
## $ NumCatalogPurchases : int   4 3 2 0 1 7 10 1 2 2 ...
## $ NumStorePurchases : int    6 7 5 2 2 5 7 3 9 9 ...
## $ NumWebVisitsMonth : int    1 5 2 7 7 2 6 5 4 4 ...
## $ AcceptedCmp3    : int    0 0 0 0 1 0 1 0 0 0 ...
## $ AcceptedCmp4    : int    0 0 0 0 0 0 0 0 0 0 ...
## $ AcceptedCmp5    : int    0 0 0 0 0 0 0 0 0 0 ...
## $ AcceptedCmp1    : int    0 0 0 0 0 0 0 0 0 0 ...
## $ AcceptedCmp2    : int    0 1 0 0 0 0 0 0 0 0 ...
## $ Response        : int    1 1 0 0 1 1 1 0 0 0 ...
## $ Complain        : int    0 0 0 0 0 0 0 0 0 0 ...
## $ Country         : chr    "SP" "CA" "US" "AUS" ...
```

‘Mnt...’ variables can be summed to create a ‘MntTotal’, representing the total amount spent by a customer in all the years as a customer. ‘Num...Purchases’ variables can be summed to create a ‘TotalPurchases’ Variable. A ‘TotDependents’ variable can be created by adding together ‘Kidhome’ and ‘Teenhome’. Customers with higher education and income of more than \$60,000 can also be used to create two new variables.

```
# Total amount spent by far
df <- mutate(df, MntTotal = MntWines + MntFruits + MntMeatProducts + MntFishProducts + MntSweetProducts)
df$MntTotal <- as.numeric(df$MntTotal)

# Total number of purchases by far
df <- mutate(df, TotalPurchases = NumDealsPurchases + NumWebPurchases + NumCatalogPurchases + NumStorePurchases)

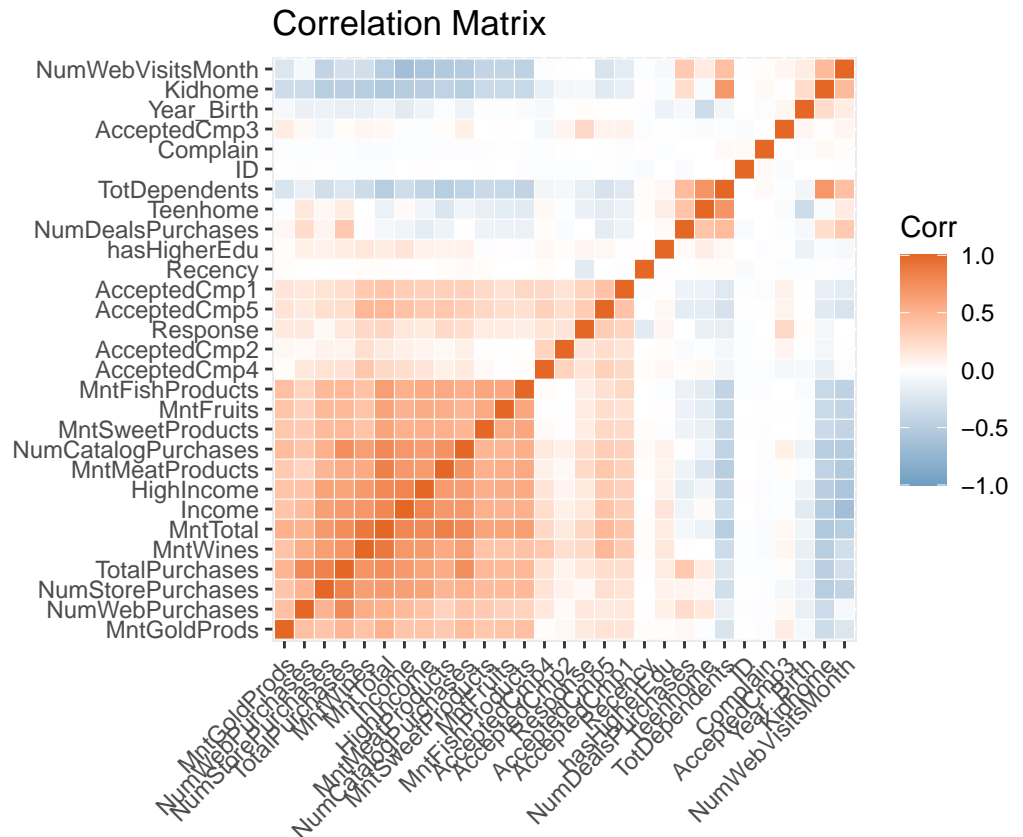
# Total number of dependents
df <- mutate(df, TotDependents = Kidhome + Teenhome)

# High income individuals
df <- mutate(df, HighIncome = Income > 60000)
df$HighIncome <- as.numeric(df$HighIncome)

# Customers with Higher Education
df <- mutate(df, hasHigherEdu = Education %in% c('Graduation', 'PhD', 'Master'))
df$hasHigherEdu <- as.numeric(df$hasHigherEdu)

# Subset of the data frame with only numeric variables
corrdf <- df[,sapply(df,is.numeric)]
```

```
# correlation matrix of the variables
corrMatrix <- cor(corrdf)
ggcorrplot(corrMatrix, title = "Correlation Matrix",
  hc.order = TRUE,
  outline.color = "white",
  ggtheme = ggplot2::theme_gray,
  colors = c("#6D9EC1", "white", "#E46726"))+
  theme(axis.text.x = element_text(angle = 45, vjust = 1,
    size = 9, hjust = 1),axis.text.y = element_text(size = 9))
```



Plots and Patterns

Plotting the correlation matrix of the features helps in identifying patterns or cluster in the data. Positive correlations between features appear orange, negative correlations appear blue, and no correlation appears white in the colored matrix above.

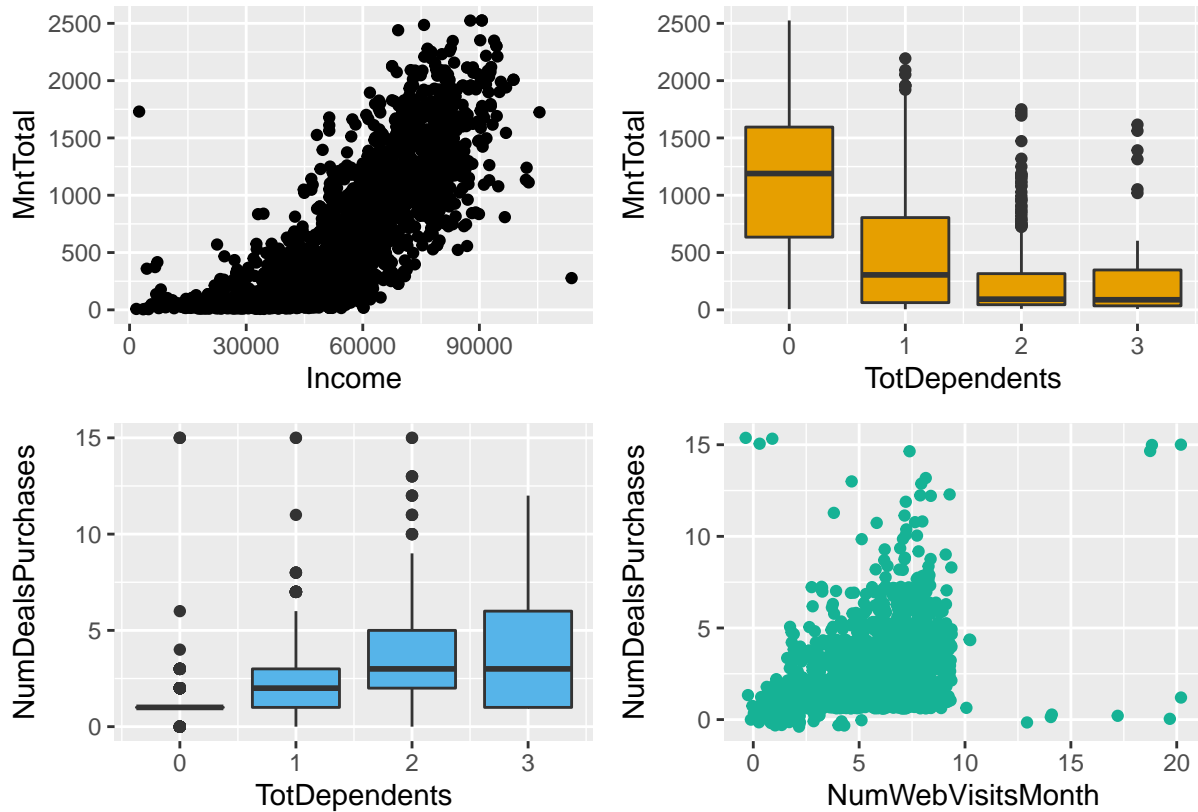
Findings:

- Total Amount and Total Purchases:
 - Total amount spent (MntTotal) and other 'Mnt' features, along with total purchases and other 'Purchases' features, are positively correlated with Income.
 - Total number of purchases in all three categories of ways to purchase - store, web and catalog - are also positively correlated with Income and *negatively* correlated with the 'TotDependents'.
- NumDealsPurchases correlation
 - 'NumDealsPurchases' is positively correlated with 'NumWebVisitsMonth', 'NumWebPurchases', and 'TotDependents'. *This suggests that customers with dependents prefer buying online with deals on products.*
- Anomalies:
 - 'Income' seems to suggest a positive, but weak, correlation with 'Response' to previous advertising campaigns.

```

plot1 <- ggplot(subset(df, Income<150000), aes(x=Income, y=MntTotal)) + geom_point()
plot2 <- ggplot(df, aes(x=TotDependents, y=MntTotal, group=TotDependents)) + geom_boxplot(fill = "#E69F00")
plot3 <- ggplot(df, aes(x=TotDependents, y=NumDealsPurchases, group=TotDependents)) + geom_boxplot(fill="#4682B4")
plot4 <- ggplot(df, aes(x=NumWebVisitsMonth, y=NumDealsPurchases, group=NumWebPurchases)) + geom_point(position = "jitter", fill="#40E0D0")
wrap_plots(plot1, plot2, plot3, plot4)

```



Regression Analysis

In order to gain further insight into what features explain the “Response” variable, denoting the response of the customers to the previous campaign, Logistic regression analysis is to be performed to help in classifying what factors lead the audience to respond to the advertising.

First, I will clean the data by dropping redundant columns in the dataset that are not needed in the regression model

```

df <- subset(df, select = -c(ID, Year_Birth, Dt_Customer))
view(df)

```

Next, split the data into two random subsets with a ratio of 70:30. The larger subset is to be used for training the model and the rest of the data is for evaluating model estimates.

```

set.seed(42)
sampleSplit <- sample.split(Y=df$Response, SplitRatio=0.7)
trainSet <- subset(x=df, sampleSplit==TRUE)
testSet <- subset(x=df, sampleSplit==FALSE)

model <- glm(Response ~ ., family=binomial(link='logit'), data=trainSet)
summary(model)

```

```

##
## Call:
## glm(formula = Response ~ ., family = binomial(link = "logit"),
##      data = trainSet)
##
## Deviance Residuals:
##      Min       1Q   Median       3Q      Max
## -3.3659  -0.3979  -0.2265  -0.1046   3.3429
##
## Coefficients: (4 not defined because of singularities)
##              Estimate Std. Error z value Pr(>|z|)
## (Intercept)    11.17285242  882.74395780   0.013   0.989901
## EducationBasic    -0.75581972   0.87102182  -0.868   0.385537
## EducationGraduation  0.52810811   0.40082098   1.318   0.187649
## EducationMaster    0.85737386   0.44802543   1.914   0.055662
## EducationPhD       1.19755581   0.43242113   2.769   0.005616
## Marital_StatusAlone -14.19380885  882.74463942  -0.016   0.987171
## Marital_StatusDivorced -14.26632194  882.74371435  -0.016   0.987106
## Marital_StatusMarried -15.48009310  882.74367210  -0.018   0.986009
## Marital_StatusSingle -14.20083149  882.74369232  -0.016   0.987165
## Marital_StatusTogether -15.24924606  882.74369483  -0.017   0.986217
## Marital_StatusWidow   -13.53658537  882.74376693  -0.015   0.987765
## Marital_StatusYOLO    -13.58966391  882.74495914  -0.015   0.987717
## Income            -0.00000905   0.00001118  -0.809   0.418237
## Kidhome            0.21032631   0.26994957   0.779   0.435902
## Teenhome          -1.12467190   0.24640170  -4.564 0.00000500963995910
## Recency           -0.02802246   0.00360531  -7.773 0.000000000000000769
## MntWines           0.00016101   0.00048425   0.333   0.739509
## MntFruits          0.00265933   0.00316870   0.839   0.401329
## MntMeatProducts    0.00293096   0.00061522   4.764 0.00000189676412172
## MntFishProducts    -0.00275549   0.00236363  -1.166   0.243701
## MntSweetProducts   0.00169818   0.00277596   0.612   0.540707
## MntGoldProds       0.00528676   0.00206416   2.561   0.010431
## NumDealsPurchases  0.22738186   0.06209036   3.662   0.000250
## NumWebPurchases    0.05210549   0.04105275   1.269   0.204358
## NumCatalogPurchases 0.09791708   0.05345431   1.832   0.066983
## NumStorePurchases  -0.17248960   0.04518564  -3.817   0.000135
## NumWebVisitsMonth  0.28670651   0.06140932   4.669 0.00000302995806872
## AcceptedCmp3       2.08635793   0.27248679   7.657 0.000000000000001907
## AcceptedCmp4       1.65339851   0.34157600   4.840 0.00000129513124812
## AcceptedCmp5       1.67395765   0.36139869   4.632 0.00000362349069849
## AcceptedCmp1       1.31947918   0.34226346   3.855   0.000116
## AcceptedCmp2       0.95186935   0.65637019   1.450   0.147002
## Complain          0.21812645   1.12216177   0.194   0.845878
## CountryCA         -0.06524827   0.43977413  -0.148   0.882053

```


## CountryGER	-0.07047488	0.53087673	-0.133	0.894390
## CountryIND	-0.76365126	0.53836530	-1.418	0.156056
## CountryME	15.09790720	882.74356920	0.017	0.986354
## CountrySA	-0.17701780	0.42684238	-0.415	0.678351
## CountrySP	-0.16051249	0.37773118	-0.425	0.670882
## CountryUS	-1.07290856	0.61334191	-1.749	0.080242
## MntTotal	NA	NA	NA	NA
## TotalPurchases	NA	NA	NA	NA
## TotDependents	NA	NA	NA	NA
## HighIncome	0.19176026	0.40380402	0.475	0.634869
## hasHigherEdu	NA	NA	NA	NA
##				
## (Intercept)				
## EducationBasic				
## EducationGraduation				
## EducationMaster	.			
## EducationPhD	**			
## Marital_StatusAlone				
## Marital_StatusDivorced				
## Marital_StatusMarried				
## Marital_StatusSingle				
## Marital_StatusTogether				
## Marital_StatusWidow				
## Marital_StatusYOLO				
## Income				
## Kidhome				
## Teenhome	***			
## Recency	***			
## MntWines				
## MntFruits				
## MntMeatProducts	***			
## MntFishProducts				
## MntSweetProducts				
## MntGoldProds	*			
## NumDealsPurchases	***			
## NumWebPurchases				
## NumCatalogPurchases	.			
## NumStorePurchases	***			
## NumWebVisitsMonth	***			
## AcceptedCmp3	***			
## AcceptedCmp4	***			
## AcceptedCmp5	***			
## AcceptedCmp1	***			
## AcceptedCmp2				
## Complain				
## CountryCA				
## CountryGER				
## CountryIND				
## CountryME				
## CountrySA				
## CountrySP				
## CountryUS	.			
## MntTotal				
## TotalPurchases				

```
## TotDependents
## HighIncome
## hasHigherEdu
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
## Null deviance: 1321.45 on 1567 degrees of freedom
## Residual deviance: 793.33 on 1527 degrees of freedom
## AIC: 875.33
##
## Number of Fisher Scoring iterations: 13
```

```
# Do Anova test
anova(model, test="Chisq")
```

```
## Analysis of Deviance Table
##
## Model: binomial, link: logit
##
## Response: Response
##
## Terms added sequentially (first to last)
##
##
```

	Df	Deviance	Resid. Df	Resid. Dev	Pr(>Chi)
## NULL			1567	1321.45	
## Education	4	13.331	1563	1308.11	0.0097686 **
## Marital_Status	7	35.877	1556	1272.24	0.000007646060066843 ***
## Income	1	20.482	1555	1251.76	0.000006020828321304 ***
## Kidhome	1	0.454	1554	1251.30	0.5005548
## Teenhome	1	42.214	1553	1209.09	0.000000000081807239 ***
## Recency	1	58.741	1552	1150.35	0.000000000000017982 ***
## MntWines	1	61.816	1551	1088.53	0.000000000000003772 ***
## MntFruits	1	0.082	1550	1088.45	0.7740870
## MntMeatProducts	1	18.535	1549	1069.91	0.000016679284382077 ***
## MntFishProducts	1	4.325	1548	1065.59	0.0375580 *
## MntSweetProducts	1	0.709	1547	1064.88	0.3998125
## MntGoldProds	1	18.008	1546	1046.87	0.000022002658370788 ***
## NumDealsPurchases	1	11.695	1545	1035.18	0.0006267 ***
## NumWebPurchases	1	1.470	1544	1033.71	0.2253019
## NumCatalogPurchases	1	6.669	1543	1027.04	0.0098080 **
## NumStorePurchases	1	35.442	1542	991.60	0.000000002627746591 ***
## NumWebVisitsMonth	1	28.230	1541	963.37	0.000000107708998320 ***
## AcceptedCmp3	1	62.259	1540	901.11	0.000000000000003011 ***
## AcceptedCmp4	1	53.153	1539	847.95	0.0000000000000308485 ***
## AcceptedCmp5	1	27.775	1538	820.18	0.000000136247514216 ***
## AcceptedCmp1	1	15.246	1537	804.93	0.000094368867506171 ***
## AcceptedCmp2	1	2.456	1536	802.47	0.1170593
## Complain	1	0.051	1535	802.42	0.8211929
## Country	7	8.864	1528	793.56	0.2625368
## MntTotal	0	0.000	1528	793.56	
## TotalPurchases	0	0.000	1528	793.56	

```
## TotDependents      0    0.000    1528    793.56
## HighIncome         1    0.226    1527    793.33      0.6348538
## hasHigherEdu       0    0.000    1527    793.33
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
```

Running the regression tells us that Education, Teenhome, and Recency have extremely low p-value at 5% threshold, suggesting a strong association of these features of affecting the log-probability of response of the individuals to the advertising. Furthermore, MntMeatProducts and AcceptedCmp3 are also considerable variable in the model.

```
# Predict the value of Response on the test set
fitted.results <- predict(model,newdata=testSet,type='response')
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

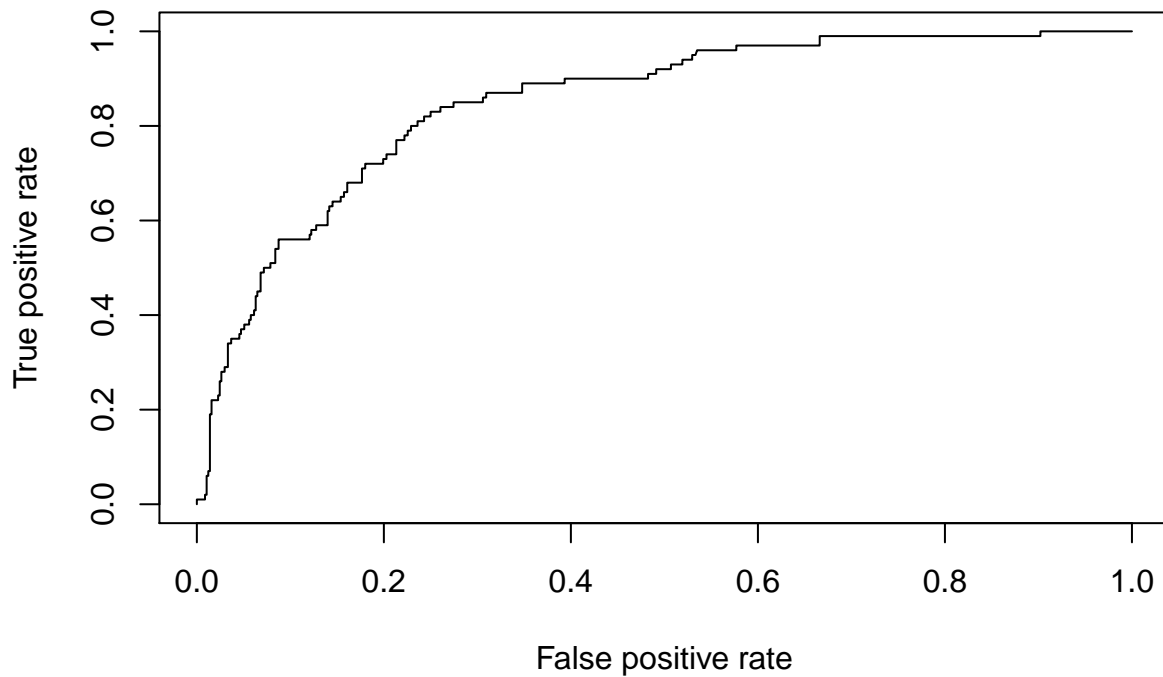
```
fitted.results <- ifelse(fitted.results > 0.5,1,0)
misClasificError <- mean(fitted.results != testSet$Response)
print(paste('Accuracy',1-misClasificError))
```

```
## [1] "Accuracy 0.864583333333333"
```

```
# Compute the AUC (Area Under the ROC Curve)
p <- predict(model, newdata=testSet, type="response")
```

```
## Warning in predict.lm(object, newdata, se.fit, scale = 1, type = if (type == :
## prediction from a rank-deficient fit may be misleading
```

```
pr <- prediction(p, testSet$Response)
prf <- performance(pr, measure = "tpr", x.measure = "fpr")
plot(prf)
```



```
auc <- performance(pr, measure = "auc")
auc <- auc@y.values[[1]]
auc
```

```
## [1] 0.8493794
```

The predicted values from the model gives us the accuracy of 0.86, which is quite good. Additionally, after plotting ROC, the evaluated AUC for the model is 0.84. As a rule of thumb, a model with good predictive ability should have an AUC closer to 1, so our model is fairly accurate.

Conclusion

- The response to the campaign is positively correlated with income and negatively correlated with kids/teens.
- The analysis showed that the best selling products are meat and wine. I suggest that the company invest in boosting the sales of rest of the product line.
- Number of dependents and number of web visits are positively correlated with the number of deals purchases, so I suggest targeting ads with online offers to homes with kids/teens.