Analysis of the Relationship between Facebook Advertising Campaign and Customer Conversion Rate

CEE 5930 Data Analysis Team Project 2

Presented to:
Dr. LINDA NOZICK

Date Performed: December 14, 2021
Date Submitted: December 14, 2021

Team 3

Chengyu Zhang
Dawei Wang
Ruixin Zhang
Steven He

TABLE OF CONTENTS

Background & Introduction

Nowadays, with the unprecedented development of the advertising industry, advertising has become an important source for people to obtain all kinds of information, especially for consumers. According to Ogilvy’s survey on consumer attitudes and evaluations of advertising, 86% of respondents in Taiwan, 74% in Hong Kong and 76% in the United States consider advertising to be a significant source of information about product features or service content. Domestic data show that 60% of people in the US believe that shopping is influenced by advertising, especially internet advertising. Most of the customer's information about the market and commodities comes directly or indirectly from advertising. The channels for people to obtain commodity information are gradually shifting from relying on their own contact and interpersonal communication in the past to relying on advertising. Advertising has become the main channel for people to gain commodity or product information.

In this project, we are interested in investigating how advertisements will affect customers' decisions, and what factors are more likely to influence a new customer. Besides, we will predict new customer’s involvement rate and test for accuracy. To fulfill our interest, three models related to class are used, which were discriminate analysis logit, ordered logit modeling, and classification tree building.

Dataset Description & Explanation

In order to get the most accurate, the group choose advertisement conversion data from organization’s social media ad campaign as dataset. This dataset gave the group comprehensive information to perform the model. There are 11 variables which are “ad_id”, “xyz*campaign*id”, “fb*campaign*id”, “age”, “gender”, “interest”, “Impressions”, “Clicks”, “Spent”, “Total conversion”, “Approved conversion”. One thing should be noticed that the group used sensitivity analysis at the beginning to eliminate irrelevant variables such as “gender”, “age”, and etc…

Sensitivity Analysis

The first step of our project is to analyze amongst all features, what would be the most important when affecting the user's decision on whether to purchase the advertised product. We designed a sensitivity analysis, to measure what feature has a positive-like correlation with our independent variable.
The result we get from the plot of ACC and clicks, ACC refers to the approved conversion where the user really purchased the advertised item, and the total conversion would be transferred to TCC, if the user only consulted the item of advertisement 0 to 1 time, we would categorize it to 0 in TCC, if the idea is consulted multiple times, it goes to 1 in the TCC category, referring more visible interest on the item. Based on our observation here from the plot, the higher number of clicks a user does on the advertisement would direct to a higher chance of making a purchase.

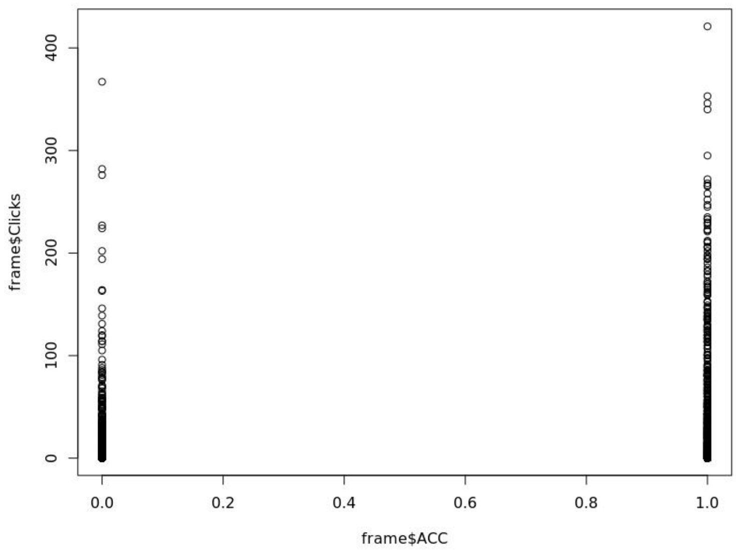

Figure: Sensitivity Analysis Plot for ACC and Clicks

The plot below demonstrates the relationship between gender of the user and the final purchase movement, the result from this plot expresses that gender has no significant correlation with the final decision.

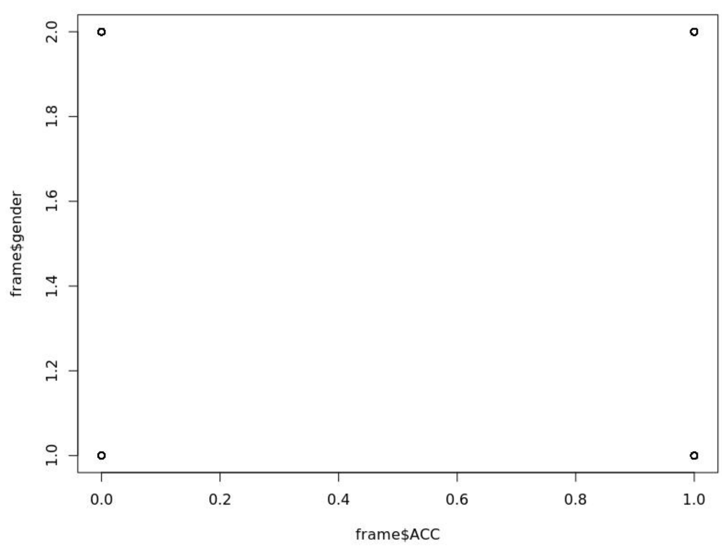

Figure: Sensitivity Analysis Plot for ACC and Genders

When discussing the relationship between impression and ACC, the answer is intuitive, based on common sense and the plot. The higher the impression, the higher chance for the user to become a customer.

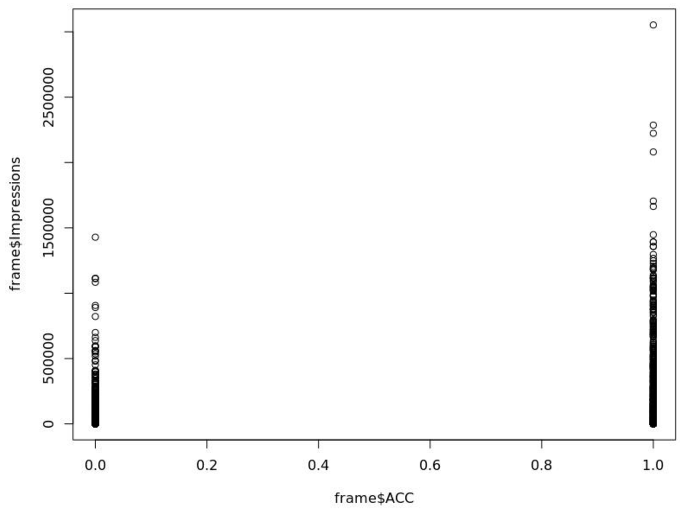

Figure: Sensitivity Analysis Plot for ACC and Impression

The last independent variable we have to discuss is spent. Spent refers to the amount of money paid by the company to Facebook to start the campaign. The plot result shows that the higher amount paid by the company, the higher chance users would purchase the advertised product, potential reason behind this could be that facebook makes the advertisement significantly more visually appealing for the higher payers, which attracts more users to check out the product.

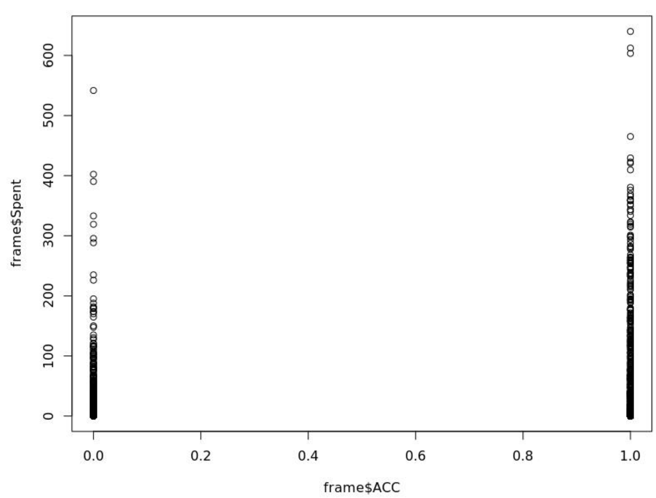

Figure: Sensitivity Analysis Plot for ACC and Spent

Linear Discriminant Analysis

The linear discriminant analysis, one of the most basic classification techniques, is used to build a model for predicting new customer's involvement rate and test for the accurary in the report.

The process of linear discriminant analysis for sales conversion optimization is shown in the following figures.

```
> library(MASS)
> frame<-read.csv("frame2.csv")
> head(frame)
   ad_id xyz_campaign_id fb_campaign_id   age gender interest Impressions Clicks Spent Total_Conversion Approved_Conversion
1 708746             916         103916 30-34      M       15        7350      1  1.43                2                   1
2 708749             916         103917 30-34      M       16       17861      2  1.82                2                   0
3 708771             916         103920 30-34      M       20         693      0  0.00                1                   0
4 708815             916         103928 30-34      M       28        4259      1  1.25                1                   0
5 708818             916         103928 30-34      M       28        4133      1  1.29                1                   1
6 708820             916         103929 30-34      M       29        1915      0  0.00                1                   1
```

Figure: Read CSV

The above-figure shows the steps of loading the package and it also prints the first few observations of that datasets to the console.

```
> #standardize
> frame$stinterest = scale(frame$interest)
> frame$stImpressions = scale(frame$Impressions)
> frame$stSpent = scale(frame$Spent)
> frame$stClicks = scale(frame$Clicks)
```

Figure: Standardized Independent Variables

There are four independent variables including interest, impression,spent, and clicks. These variables are standardized before using linear discriminant analysis. The above-figure shows the process of standardization variables in R.

```
> #Use 75% as train_data
> split.at = round(0.75*length(frame$ad_id))
> print(split.at)
[1] 857
> ind<-sample(1:nrow(frame), split.at)
> print(ind)
  [1]  696  578  350  175   15  778  371  535   59  595  604  395  340 1102  682  449  923
 [18]  402  608  222  499 1026  662  104  420   83  855 1103  345  221  206  887  991    8
 [35]  392  966 1106  841  441 1107   38   45   48 1001  357  283  404  512  173 1067 1133
 [52]   99  103  913  151  930  847   87  687  140  265  346  671  129  524  293  212 1038
 [69] 1112  838   58  188  561  439  105   84  820  193  651 1051  832 1105  728  864  338
 [86] 1136  385    6  892  381    4  127  312  941  810  256  940   11  710  917  880  858
[103]  928  936  147  369  766  117 1109  853  289  434  834   74  777  170  625  191  903
```

Figure: Setup Training Data

There are 1143 observations in 11 variables in the database, 75% of which are equivalent to 857 observations used for training data. The remaining 286 observations are used for the test dataset. The above-figure shows the steps of setting training data.

```
> frame$ACC <-ifelse(frame$Approved_Conversion == 0,0,1)
> frame$TCC <-ifelse(frame$Total Conversion == 0 | frame$Total Conversion == 1,0,1)
```

Figure: Define ACC and TCC

In the ACC model, the group 0 includes people who do not purchase the item after watching the advertisement. On the contrary, group 1 includes people who purchase the item after watching the advertisement. In the TCC model, group 0 includes those who have asked about the product content 0 to 1 times, and group 1 includes those who have asked about the product content more than once. The above-figure shows how to define ACC and TCC in the R.

```
Call:
lda(ACC ~ Spent + Impressions + interest + Clicks, data = train_data)

Prior probabilities of groups:
        0         1 
0.4749125 0.5250875 

Group means:
     Spent Impressions interest   Clicks
0 28.72501    97278.04 31.24324 19.16216
1 72.62869   272482.38 34.59778 47.04222

Coefficients of linear discriminants:
                      LD1
Spent       -1.295078e-02
Impressions  7.052835e-06
interest     1.886227e-03
Clicks      -1.849925e-03
```

Figure: Model for ACC

In the ACC model, the prior probabilities of group 0 is about 47.5%. It indicates that 47.5% data in the training dataset is in group 0. In other words, 47.5% of people would not purchase the product after watching the advertisement. On the contrary, the prior probabilities of group 1 is about 52.5%. It represents that 52.5% data in the training dataset is in group 1. It indicates that 52.5% of people would purchase the product after watching the advertisement. For group 0, it has relatively lower spending, impressions, interest and clicks. For group 1, it has relatively higher spent, impressions, interest and clicks.It indicates that the amount paid by company xyz to Facebook is lower to people who do not purchase the product compared with people who buy the product. Also, for people who buy the product, they watch longer advertisements and click advertisements more times compared with people who do not buy the product. Besides, people who buy the product have more interests compared with people who do not buy products. The coefficient of linear discriminant for spent is about -1.295078e-02.The coefficient of linear discriminant for impressions is about 7.052835e-06 The coefficient of linear discriminant for interest is about 1.886227e-03 The coefficient of linear discriminant for clicks is about -1.849925e-03. The details of the model for ACC are shown in above-figure.

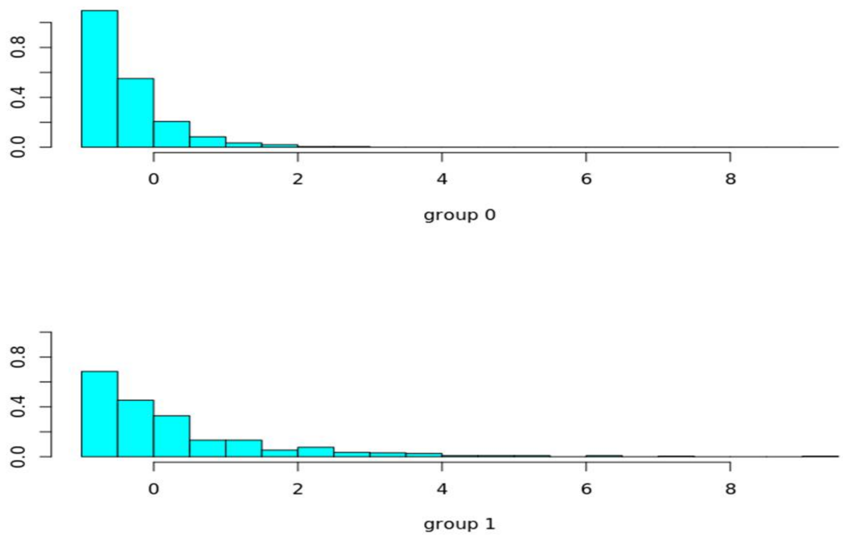

Figure: ACC_Model_Fit

The above-figure shows the density for the linear discriminant score that comes out for group 0 including people who do not purchase the products and group 1 including people who purchase the products.

```
> #Predicted posterior probability
> pred_posteriors=pred$posterior
> print(pred_posteriors)
              0         1
1    0.5660279548 0.4339720
9    0.5680890639 0.4319109
10   0.5639503047 0.4360497
15   0.5758778885 0.4241221
17   0.5654398400 0.4345602
20   0.5581684663 0.4418315
37   0.5480469167 0.4519531
43   0.5683189856 0.4316810
46   0.5575660112 0.4424340
47   0.5679467244 0.4320533
49   0.5698398323 0.4301602
52   0.5693647461 0.4306353
57   0.5645632391 0.4354368
```

Figure: ACC Prediction Posterior Probability

The above-figure shows the ACC prediction posterior probability. These predictions show that the observations have higher probability in group 0 including people who do not purchase the products rather than group 1 including people who buy the products.

```
> print(pred$class)
  [1] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0
 [44] 0 0 0 0 0 0 0 1 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
 [87] 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0 0
[130] 0 0 0 0 0 0 1 1 1 0 1 1 0 0 1 0 0 1 0 1 1 1 1 0 0 0 0 0 1 1 1 1 1 1 1 0 1 0 0 0 1 0 0
[173] 1 1 1 0 1 1 0 0 0 1 1 1 1 0 1 0 0 1 0 0 0 1 1 1 1 1 1 0 0 0 1 1 1 1 1 1 1 1 0 1 0 0 1
[216] 1 1 0 1 1 1 1 1 1 0 1 0 0 1 0 0 0 1 1 0 0 0 0 1 1 1 1 1 1 0 0 0 1 1 1 0 0 0 0 0 0 1 1
[259] 0 0 0 1 1 0 1 1 0 1 1 1 1 0 1 1 1 1 1 1 1 1 1 1 0 0 1 1
Levels: 0 1
```

Figure: ACC Prediction Class

The above-figure shows each individual which group they are assigned. There is a higher probability in group 0 instead of group 1 by looking at the ACC prediction class.

```
> #Confusion Matrix
> CT<-table(test_data$ACC, pred$class)
> print(CT)

      0   1
  0 132  20
  1  62  72
> #Accurary Rate
> mean(test_data$ACC == pred$class)
[1] 0.7132867
>
```

Figure: ACC Confusion Matrix and Accuracy

The above-figure shows the accuracy rate for the ACC is 0.71. The rate of accuracy for the ACC is good to make predictions to verify if people purchase products after watching the advertisement in the future.

```
> CT<-table(test_data$TCC, frame3$Pred_class)
> print(CT)

      0   1
  0 158  12
  1  42  74
>
> #report the accuracy rate
> mean(test_data$TCC == frame3$Pred_class)
[1] 0.8111888
>
```

Figure: TCC Confusion Matrix and Accuracy

The above-figure shows the rate of accuracy for the TCC is 0.81. The rate of accuracy for TCC is good to make predictions to verify whether people asked about the product 0 to 1 time or more than once

Ordered Logit

```
Logistic Regression Model

 lrm(formula = TCC ~ stSpent + stImpressions + stinterest + stClicks,
     data = train_data)

                        Model Likelihood     Discrimination    Rank Discrim.
                           Ratio Test           Indexes           Indexes
 Obs            857    LR chi2     485.06    R2       0.582    C       0.906
  0             504    d.f.             4    g        3.696    Dxy     0.812
  1             353    Pr(> chi2) <0.0001    gr      40.293    gamma   0.812
 max |deriv| 5e-10                           gp       0.362    tau-a   0.394
                                             Brier    0.120

                   Coef    S.E.   Wald Z Pr(>|Z|)
 Intercept          0.5329 0.1450  3.67  0.0002
 stSpent[1]        -5.4646 2.8381 -1.93  0.0542
 stImpressions[1]   8.8864 1.2191  7.29  <0.0001
 stinterest[1]     -0.0192 0.1071 -0.18  0.8576
 stClicks[1]        0.7696 2.2604  0.34  0.7335
```

Figure: Logit Model for TCC

There are four coefficients. The coefficient of stSpent is -5.4646. The coefficient of stImpression is 8.8864. The coefficient of stInterest is -0.0192. The coefficient of stClick is 0.7696. The above-figure shows the logit model for TCC.

```
Logistic Regression Model

 lrm(formula = TCC ~ stSpent + stImpressions, data = train_data)

                        Model Likelihood     Discrimination    Rank Discrim.
                           Ratio Test           Indexes           Indexes
 Obs            857    LR chi2     484.94    R2       0.582    C       0.905
  0             504    d.f.             2    g        3.685    Dxy     0.810
  1             353    Pr(> chi2) <0.0001    gr      39.835    gamma   0.811
 max |deriv| 2e-10                           gp       0.362    tau-a   0.393
                                             Brier    0.120

                   Coef    S.E.   Wald Z Pr(>|Z|)
 Intercept          0.5272 0.1422  3.71  0.0002
 stSpent[1]        -4.5218 0.8854 -5.11  <0.0001
 stImpressions[1]   8.6905 1.0635  8.17  <0.0001
```

Figure: Logit Model without Insignificant Variables

Also, the Pr(>|z|) represents the p-value associated with the value in the z value column. It indicates that it has a statistically significant relationship with the response variable in the model. These two variables, stSpent and stImpression, plays an important role in the logit model because of their smaller values of Pr(>|z|).

```
> #predict on testdata
> fitteddata = predict(ologit,test_data, type = "fitted.ind")
> frame2 = data.frame(fitteddata)
> #write the result to csv
> #write.csv(fitteddate, file="fitteddata.csv)
> #read from csv
> frame3 = data.frame(fitteddata)
> #frame2<-read.csv("fitteddate.csv)
> #add column to store the prediction class
> frame3$Pred_class = ifelse(frame2$fitteddata>0.5,1,0)
```

Figure: Logit_Predict Test Data and Write

The above-figure shows the processes about measuring the probability of the observations in each one of the groups.

Classification Tree

After running our prediction models on the dataset and knowing the accuracy of predicting upcoming campaign data, we decided to use classification trees to determine binarily the determining factors of a customer, potentially, when they are making a decision on becoming a paid customer or not. The first step we do is to load the data and to get the current structure of our dataset components.

```
> # Naive Bayes
> library(e1071)
> frame<-read.csv("frame2.csv")
> head(frame)
  ad_id xyz_campaign_id fb_campaign_id   age gender interest Impressions Clicks Spent Total_Conversion Approved_Conversion
1 708746             916         103916 30-34      M       15        7350      1  1.43                2                   1
2 708749             916         103917 30-34      M       16       17861      2  1.82                2                   0
3 708771             916         103920 30-34      M       20         693      0  0.00                1                   0
4 708815             916         103928 30-34      M       28        4259      1  1.25                1                   0
5 708818             916         103928 30-34      M       28        4133      1  1.29                1                   1
6 708820             916         103929 30-34      M       29        1915      0  0.00                1                   1
>
> str(train_data)
'data.frame':   857 obs. of  17 variables:
 $ ad_id              : int  1121429 1121203 779944 747795 709059 1121619 780799 1121113 734266 1121244 ...
 $ xyz_campaign_id    : int  1178 1178 936 936 916 1178 936 1178 936 1178 ...
 $ fb_campaign_id     : int  144595 144554 116115 110933 103968 144627 116273 144534 108664 144562 ...
 $ age                : Factor w/ 4 levels "30-34","35-39",..: 2 1 2 1 1 4 1 1 1 1 ...
 $ gender             : Factor w/ 2 levels "F","M": 2 2 2 1 2 2 1 2 2 2 ...
 $ interest           : int  7 29 10 15 20 20 22 18 25 36 ...
 $ Impressions        : int  152454 1048861 2549 8410 14669 127125 44699 894911 605 181683 ...
 $ Clicks             : int  22 128 0 2 7 20 13 120 0 20 ...
 $ Spent              : num  37.85 219.77 0 2.36 10.28 ...
 $ Total_Conversion   : int  1 22 1 1 1 2 2 7 1 2 ...
 $ Approved_Conversion: int  1 8 0 1 1 0 0 4 0 1 ...
 $ ACC                : num  0 1 1 0 0 1 1 1 1 0 ...
 $ TCC                : num  0 1 0 0 0 1 1 1 0 1 ...
 $ stinterest         : num [1:857, 1] -0.956 -0.14 -0.845 -0.659 -0.474 ...
 $ stImpressions      : num [1:857, 1] -0.11 2.756 -0.589 -0.57 -0.55 ...
 $ stSpent            : num [1:857, 1] -0.155 1.938 -0.591 -0.564 -0.473 ...
 $ stClicks           : num [1:857, 1] -0.2 1.663 -0.587 -0.552 -0.464 ...
>
> frame$ACC <-ifelse(frame$Approved_Conversion == 0,0,1)
> frame$TCC <-ifelse(frame$Total_Conversion == 0 | frame$Total_Conversion == 1,0,1)
```

Figure: Classification tree 1

The second step we took was to split up the dataset into training dataset and testing dataset, and, for the sake of efficiency and to escape potential errors in the process of data analysis, we decided to turn the datasets' newly introduced features: ACC and TCC, into 'factor' type. And we are ready to do the analysis.

```
> frame$ACC <-ifelse(frame$Approved_Conversion == 0,0,1)
> frame$TCC <-ifelse(frame$Total_Conversion == 0 | frame$Total_Conversion == 1,0,1)
> train_data$ACC<-as.factor(train_data$ACC)
> train_data$TCC<-as.factor(train_data$TCC)
>
> str(train_data)
'data.frame':   857 obs. of  17 variables:
 $ ad_id              : int  1121429 1121203 779944 747795 709059 1121619 780799 1121113 734266 1121244 ...
 $ xyz_campaign_id    : int  1178 1178 936 936 916 1178 936 1178 936 1178 ...
 $ fb_campaign_id     : int  144595 144554 116115 110933 103968 144627 116273 144534 108664 144562 ...
 $ age                : Factor w/ 4 levels "30-34","35-39",..: 2 1 2 1 1 4 1 1 1 1 ...
 $ gender             : Factor w/ 2 levels "F","M": 2 2 2 1 2 2 1 2 2 2 ...
 $ interest           : int  7 29 10 15 20 20 22 18 25 36 ...
 $ Impressions        : int  152454 1048861 2549 8410 14669 127125 44699 894911 605 181683 ...
 $ Clicks             : int  22 128 0 2 7 20 13 120 0 20 ...
 $ Spent              : num  37.85 219.77 0 2.36 10.28 ...
 $ Total_Conversion   : int  1 22 1 1 1 2 2 7 1 2 ...
 $ Approved_Conversion: int  1 8 0 1 1 0 0 4 0 1 ...
 $ ACC                : Factor w/ 2 levels "0","1": 1 2 2 1 1 2 2 2 2 1 ...
 $ TCC                : Factor w/ 2 levels "0","1": 1 2 1 1 1 2 2 2 1 2 ...
 $ stinterest         : num [1:857, 1] -0.956 -0.14 -0.845 -0.659 -0.474 ...
 $ stImpressions      : num [1:857, 1] -0.11 2.756 -0.589 -0.57 -0.55 ...
 $ stSpent            : num [1:857, 1] -0.155 1.938 -0.591 -0.564 -0.473 ...
 $ stClicks           : num [1:857, 1] -0.2 1.663 -0.587 -0.552 -0.464 ...
```

Figure: turning data types to factor

Now we would like to draw our classification trees with respect to ACC and TCC, our made up new factors, the code below refers to the dependent and independent variables in our classification tree for both the ACC and two of our TCC trees.

```
library(partykit)
#ctout <- ctree(ACC ~ Clicks + Spent + Impressions + interest,data=train_data)
ctout <- ctree(TCC ~ Clicks + Spent + interest,data=train_data)
ctpred <- predict(ctout,test_data) #This predicts the categories the borrowers will fall into. Note that
# for demonstration purposes here we're making predictions with the same set of data we used # to make the classificati
print(ctpred)
```

Figure: Classification tree codes

The following plot refers to our ACC classification tree we got. The first binary node is determined by impression; if the customer has a somewhat bad impression of the company or the item, with impression <= 157534, the individual would be classified into node 2, with a population of 587. Our leaf nodes are determined by a decision node where clicks on the advertisement become the key, if the clicks are more than 46, the individual would be classified to node 6, if below or equal to 46, the node would be classified to node 5.

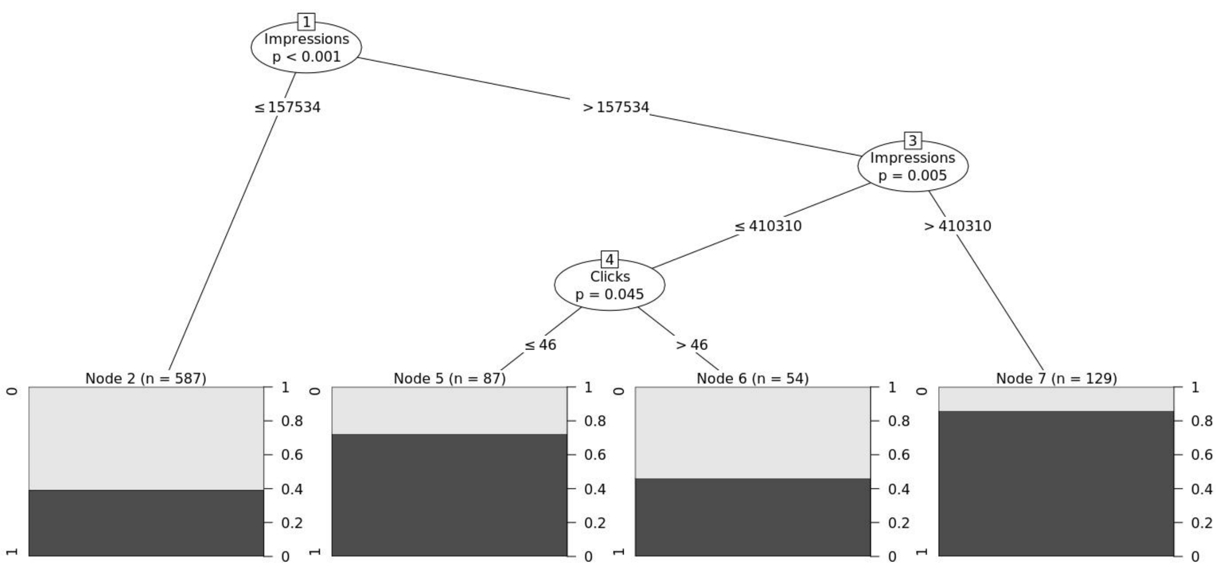

Figure: ACC classification tree

Moving on to the classification tree of TCC, here we did the classification tree twice. In the first attempt we used all the relevant independent variables determined from past steps to express our dependent variable, and try to see what would be the defining factors for the classification tree. The result is that all the decision nodes are determined by impression, which made sense, as impression of the brand or item certain customer holds before seeing the advertisement would be a defining factor for one's future move, just by common sense, but we were afraid that the impression would be a biased data which has limited correlation of the real effect of the campaign, as such impression, by definition, is carried by the user before seeing the ads. With that to bare in mind, we did another classification tree for TCC, this time with only three independent variables, the result showed totally different decision nodes which, based on our understanding, would be more valuable for the topic of our project.

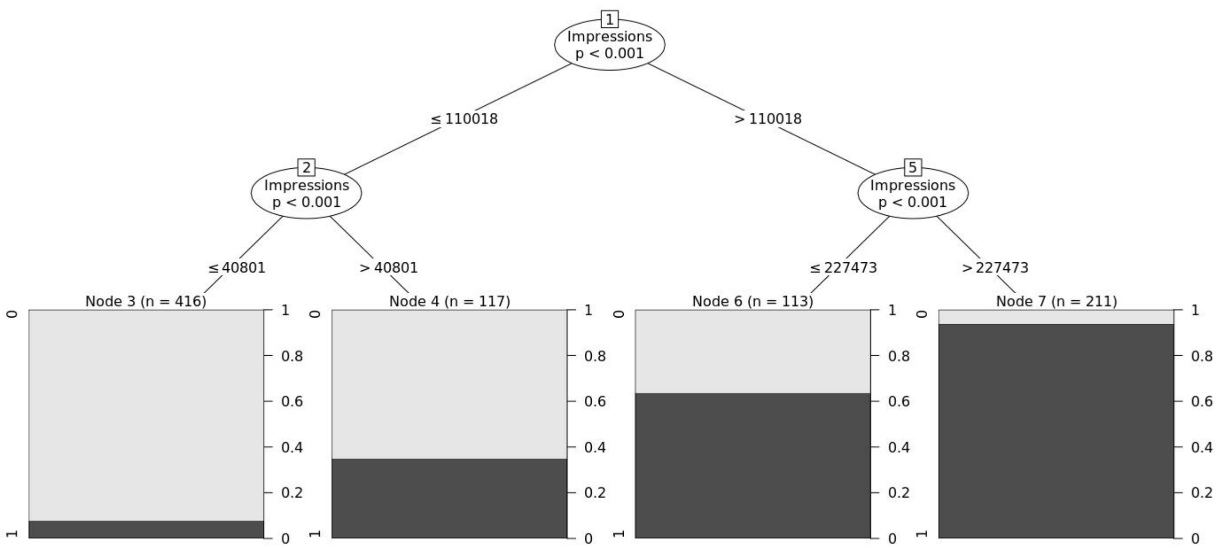

Figure: TCC Classification Tree with Impression

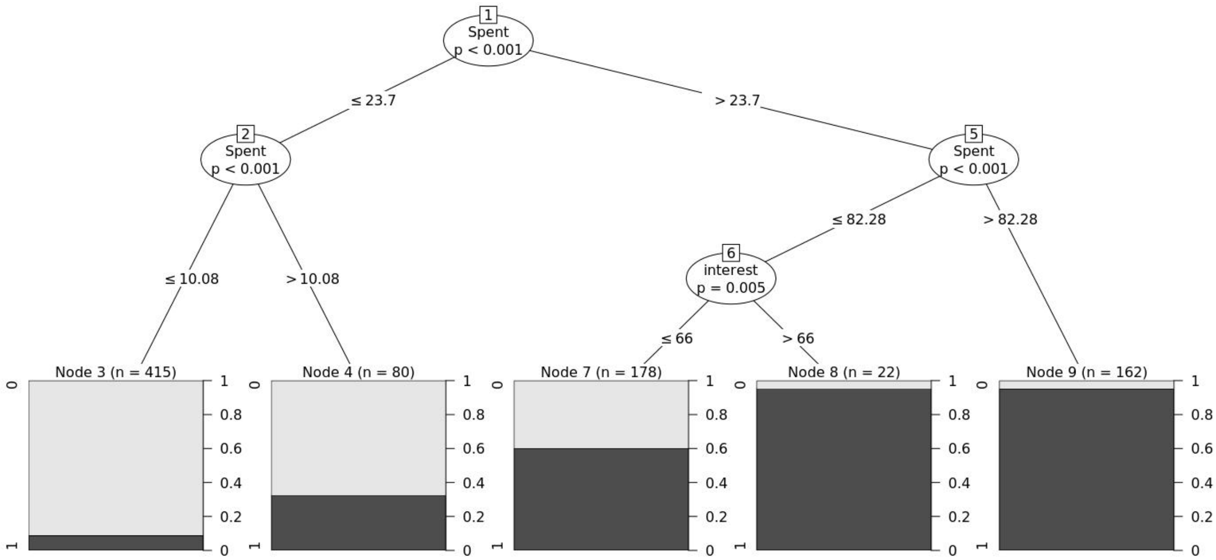

Figure: TCC Classification Tree without Impression

After we have the classification trees and we studied their hidden meanings, it's time to put our models under test, the result was not great but acceptable by our standard. The classification accuracy for our model without impression as independent variable reported a correct classifying rate of 0.787, which means that our model successfully classified the user's move on the advertised merchandise 78.7% correctly, which, based on the tangled nature of the dataset, is acceptable for us. It is worth mentioning that the resulting classification accuracy for our tree with impression actually returned a better accuracy, which reads 82.16%, but our team believes that if future data were collected and the user refuse to give feedback on the impression of the item or company, this result would lose its meaning, so for reality reason we prefer the model with no impression over the one with impression, but which to use would be conditional based on the situation companies face in the future.

```
ctpred   0   1
     0 140  20
     1  41  85
> mean(ctpred == test_data$TCC) #Check the percentage
fies a data point
[1] 0.7867133
> plot(ctout) #plot your classification tree
>
```

Figure: Rate of Accuracy for No Impression Model

```
ctpred   0   1
     0 151  21
     1  30  84
> mean(ctpred == test_data$TCC) :
fies a data point
[1] 0.8216783
```

Figure: Rate of Accuracy for the Impression Model

Conclusion

The objective of this study was to analyze the relationship between advertising campaign and customer conversion rate, and to investigate the factors that can affect a customer's buying decision. The sensitivity analysis was to identify the features that have positive-like correlation with the independent variables. Several variables such as age, and gender were eliminated after performing the sensitivity analysis. For the linear discriminant analysis, the group got 0.71 accuracy rate for ACC and 0.81 accuracy rate for TCC. These rates are useable for this model to predict whether customers are interested in the product and willing to buy the product after watching the advertisement. The group also performed ordered logit to measure the probability of the observations in each one of the groups. Besides, by conducting classification tree model, the considering factors of a customer when deciding to buy product or not were determined. The result indicated that impression, clicks on the ads, spent, interest are important factors which will affect customers decision-making when shopping. The accuracy rate of TCC classification tree with impression and without impression is 78.7% and 82.16% which are acceptable, indicating that the group's model successfully predicted the customers' interests and move of buying advertised product.

Reference

1.Gokagglers, "Sales conversion optimization," *Kaggle*, 26-Sep-2017. [Online]. Available: https://www.kaggle.com/loveall/clicks-conversion-tracking.