Fall 2021
ISTM 6214 – Foundation of Artificial Intelligence

Customer Personality Analysis

December 12, 2021
Instructor: Professor Wei Chen

Group 5
Haoran Zhang
Getitnet Bogale-alemayehu
Rachel Kim

ORIGINAL WORK STATEMENT

We the undersigned certify that the actual composition of this proposal was done by us and is original work.

	Name	Signature
Contact Author	Getinet Bogale	Getinet Bogale
	Haoran Zhang	Haoran Zhang
	Rachel Kim	Rachel Kim

Table of Contents

Executive Summary

Our research and dataset were obtained from the Customer Personality Analysis which aims to give out a detailed analysis of the "ideal" customer for different companies. The purpose of the research is to help different businesses understand their consumers further and make it a lot more efficient in selling products or services that would most be beneficial for the demographic that they are trying to target. This model helps businesses modify their products to different types of customer segments.

We were able to carry out the project following this general process:

- Cleaning and Processing the Data
- Exploring the data
- Running data models such as Linear Regression, Logistic Regression, KNN, Decision Trees
- Evaluating models
- Drawing comparisons and conclusions

Research Question

This project aims to provide exploratory data analysis through the use of visualization and modeling of the dataset provided named marketing_campaign.csv. The goal of this research project is to answer the following questions:

- What model is most appropriate to predict the customer's response to our product/service?
- What type of customer is favored for our product/service?
- How likely will a customer accept our product or service?

Data Description

The data was sourced from Kaggle (https://www.kaggle.com/imakash3011/customer-personality-analysis) and functioned to help businesses' find efficient ways to modify their products based on finding the "ideal" customer. The owner of this dataset is Akash Patel, who has a background in machine learning and is a current intern at Estylo. This dataset has been updated fairly recently (3 months ago), which ensures that the data is not outdated. The biggest questions that the dataset aims to solve are 1) to better understand customer attitude towards the product and 2) what people are doing as opposed to what they are saying about the product. However, for our general purpose and research, we aim to utilize the information to predict what the most popular customer groups are for target industries.

The number of rows in this dataset is 2240 and has 29 different variables to consider. Each customer has a unique identifier that is in the form of 4 numbers and the “idealness” is based on these variables: birth year, education, marital status, income, number of children in household, number of teenagers in household, date of customer’s enrollment with the company, number of days since the customer last purchased, and whether or not this customer has complained in the last 2 years. As mentioned above, the target industries are wine, fruits, meat, fish, sweets, and gold. These product variables are all the amounts spent in that specific category in the past 2 years. The specific industries that are touched for how much each consumer spends on are wine, fruits, meat, fish, sweet, and gold products. What we aim to achieve out of this customer personality analysis is to see customer attitude to given products and predict how different customer segments will respond to a particular product or service.

There are also 7 variables for the promotion method, ranging from whether the purchase was made with a discount or whether customers accepted an offer in different stages of the marketing campaign. Lastly, 4 variables are indicating the place of purchase such as company website, catalog, in-store, and the number of visits to the company website in the last month. For more contextual definitions of the variables look below:

People

- ID: Customer's unique identifier
- Year_Birth: Customer's birth year
- Education: Customer's education level
- Marital_Status: Customer's marital status
- Income: Customer's yearly household income
- Kidhome: Number of children in customer's household
- Teenhome: Number of teenagers in customer's household
- Dt_Customer: Date of customer's enrollment with the company
- Recency: Number of days since customer's last purchase
- Complain: 1 if the customer complained in the last 2 years, 0 otherwise

Products

- MntWines: Amount spent on wine in last 2 years
- MntFruits: Amount spent on fruits in last 2 years
- MntMeatProducts: Amount spent on meat in last 2 years
- MntFishProducts: Amount spent on fish in last 2 years
- MntSweetProducts: Amount spent on sweets in last 2 years
- MntGoldProds: Amount spent on gold in last 2 years

Promotion

- NumDealsPurchases: Number of purchases made with a discount
- AcceptedCmp1: 1 if customer accepted the offer in the 1st campaign, 0 otherwise
- AcceptedCmp2: 1 if customer accepted the offer in the 2nd campaign, 0 otherwise
- AcceptedCmp3: 1 if customer accepted the offer in the 3rd campaign, 0 otherwise
- AcceptedCmp4: 1 if customer accepted the offer in the 4th campaign, 0 otherwise
- AcceptedCmp5: 1 if customer accepted the offer in the 5th campaign, 0 otherwise
- Response: 1 if customer accepted the offer in the last campaign, 0 otherwise

Place

- NumWebPurchases: Number of purchases made through the company's website
- NumCatalogPurchases: Number of purchases made using a catalog
- NumStorePurchases: Number of purchases made directly in stores
- NumWebVisitsMonth: Number of visits to company's website in the last month

Data Preparation

1. Missing value

While quickly running a summary with the data, there were 24 null values for the variable Income. To overcome this issue, we found the mean of the column and inputted this value into the null cells. This code can be found in our appendix.

2. Data Consistency

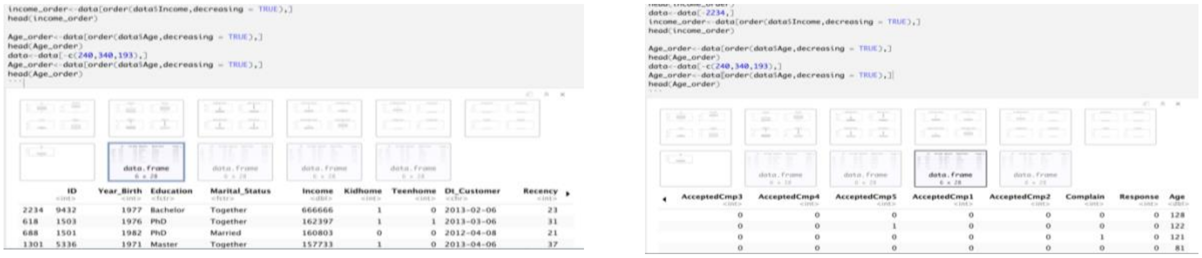
The Dt_Customer variables have two different formats and to have more consistency within our research we changed the date variable to one format. One uses the mm/dd/yyyy format and the other uses the dd-mm-yyyy format. The following codes will make use of the for loop to iterate all date values and transform them into the same format. In each loop, we check if they include “/” or “-” so we can transform accordingly. Library sjmisc is used for identifying and transforming data strings of a particular pattern. Additionally, the variable Year_Birth was not an indicative name, therefore we changed the name to Age.

3. Factorization

Factorized the complaint variable to 1 if the customer has complained within the past 2 years and 0 if the answer is otherwise.

4. Outlier Removal

First, we pinpointed outliers that were in any of the variables. Outliers that we were able to find were the age and income variables, which were removed from our dataset.



```
income_order<-data[order(data$Income,decreasing = TRUE),]
head(income_order)

Age_order<-data[order(data$Age,decreasing = TRUE),]
head(Age_order)
data<-data[-c(240,340,193),]
Age_order<-data[order(data$Age,decreasing = TRUE),]
head(Age_order)
```

```
           ID  Year_Birth  Education  Marital_Status    Income  Kidhome  Teenhome  Dt_Customer   Recency
        <int>       <int>  <fctr>     <fctr>             <dbl>    <int>     <int>  <chr>           <int>
2234    9432         1977  Bachelor   Together          666666        1         0  2013-02-06         23
618     1503         1976  PhD        Together          162397        1         1  2013-03-06         31
688     1501         1982  PhD        Married           160803        0         0  2012-04-08         21
1301    5336         1971  Master     Together          157733        1         0  2013-04-06         37
```

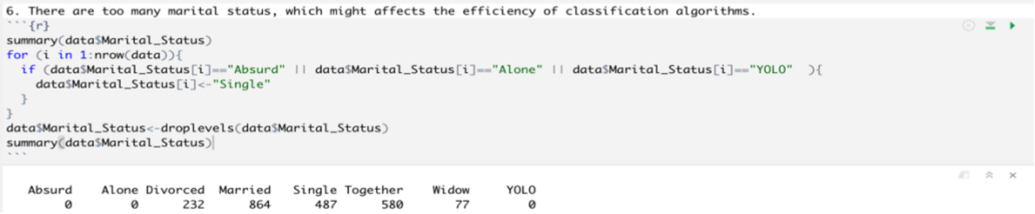
```
data<-data[-2234,]
income_order<-data[order(data$Income,decreasing = TRUE),]
head(income_order)

Age_order<-data[order(data$Age,decreasing = TRUE),]
head(Age_order)
data<-data[-c(240,340,193),]
Age_order<-data[order(data$Age,decreasing = TRUE),]
head(Age_order)
```

```
  AcceptedCmp3  AcceptedCmp4  AcceptedCmp5  AcceptedCmp1  AcceptedCmp2  Complain  Response   Age
         <int>         <int>         <int>         <int>         <int>     <int>     <int> <dbl>
             0             0             0             0             0         0         0   128
             0             0             1             0             0         0         0   122
             0             0             0             0             0         1         0   121
             0             0             0             0             0         0         0    81
```

Figure 1 Income & Age Outlier

5. Marital_Status Variable

There are too many marital status variables that can decrease the efficiency of the classification algorithms. Alone, Absurd, and YOLO were all specific situations of 'Single', so replace all of them with 'Single'.



6. There are too many marital status, which might affects the efficiency of classification algorithms.

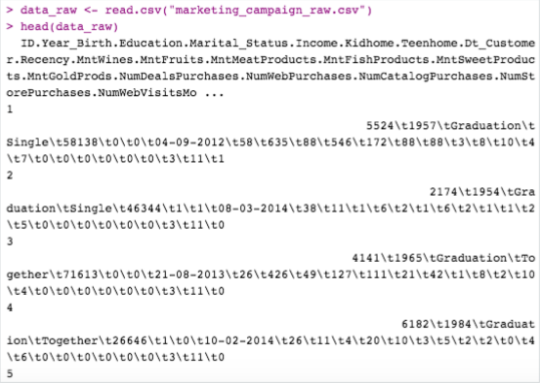
~~~
```{r}
summary(data$Marital_Status)
for (i in 1:nrow(data)){
  if (data$Marital_Status[i]=="Absurd" || data$Marital_Status[i]=="Alone" || data$Marital_Status[i]=="YOLO"  ){
    data$Marital_Status[i]<-"Single"
  }
}
data$Marital_Status<-droplevels(data$Marital_Status)
summary(data$Marital_Status)
```
~~~

```
   Absurd     Alone Divorced  Married   Single Together    Widow     YOLO 
        0         0      232      864      487      580       77        0 
```

Figure 2 Summary of Marital Status

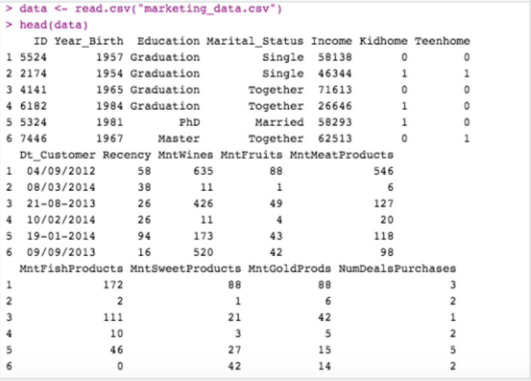
6. Messy Data

When initially downloading the data, all data were mashed into one column. The cause of this could be that the original file was in a different format to a csv. This was a quick but essential fix to ensure that any analysis and regression we ran would function. Below you can see the before and after of cleaning this messy data. Additionally, the Z_Revenue & Z_CostContact have Constant values, which don't provide any information.



```
> data_raw <- read.csv("marketing_campaign_raw.csv")
> head(data_raw)
  ID.Year_Birth.Education.Marital_Status.Income.Kidhome.Teenhome.Dt_Custome
r.Recency.MntWines.MntFruits.MntMeatProducts.MntFishProducts.MntSweetProduc
ts.MntGoldProds.NumDealsPurchases.NumWebPurchases.NumCatalogPurchases.NumSt
orePurchases.NumWebVisitsMo ...
1
                                                    5524\t1957\tGraduation\t
Single\t58138\t0\t0\t04-09-2012\t58\t635\t88\t546\t172\t88\t88\t3\t8\t10\t4
\t7\t0\t0\t0\t0\t0\t0\t3\t11\t1
2
                                                         2174\t1954\tGra
duation\tSingle\t46344\t1\t1\t08-03-2014\t38\t11\t1\t6\t2\t1\t6\t2\t1\t1\t2
\t5\t0\t0\t0\t0\t0\t0\t3\t11\t0
3
                                                  4141\t1965\tGraduation\tTo
gether\t71613\t0\t0\t21-08-2013\t26\t426\t49\t127\t111\t21\t42\t1\t8\t2\t10
\t4\t0\t0\t0\t0\t0\t0\t3\t11\t0
4
                                                       6182\t1984\tGraduat
ion\tTogether\t26646\t1\t0\t10-02-2014\t26\t11\t4\t20\t10\t3\t5\t2\t2\t0\t4
\t6\t0\t0\t0\t0\t0\t0\t3\t11\t0
5
```

Figure 3 Before data clean-up

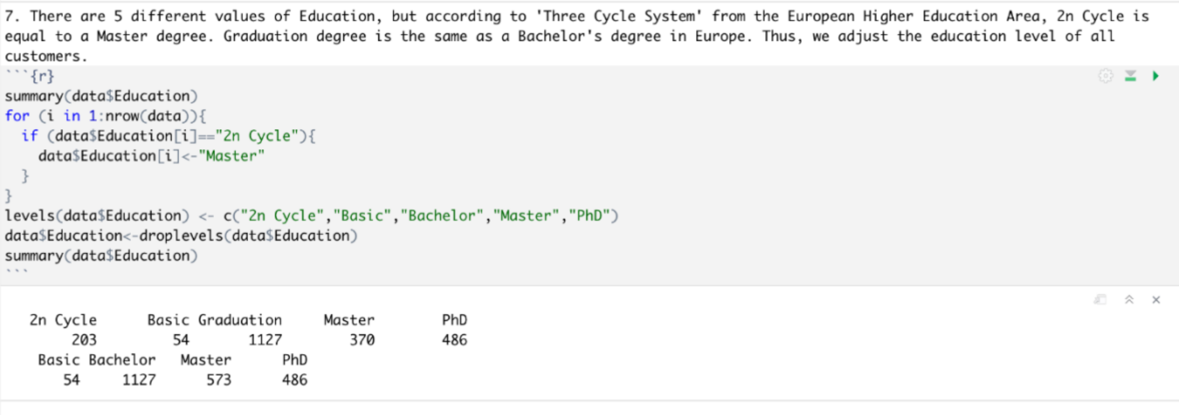


```
> data <- read.csv("marketing_data.csv")
> head(data)
    ID Year_Birth  Education Marital_Status Income Kidhome Teenhome
1 5524       1957 Graduation         Single  58138       0        0
2 2174       1954 Graduation         Single  46344       1        1
3 4141       1965 Graduation       Together  71613       0        0
4 6182       1984 Graduation       Together  26646       1        0
5 5324       1981        PhD        Married  58293       1        0
6 7446       1967     Master       Together  62513       0        1
  Dt_Customer Recency MntWines MntFruits MntMeatProducts
1  04/09/2012      58      635        88             546
2  08/03/2014      38       11         1               6
3  21-08-2013      26      426        49             127
4  10/02/2014      26       11         4              20
5  19-01-2014      94      173        43             118
6  09/09/2013      16      520        42              98
  MntFishProducts MntSweetProducts MntGoldProds NumDealsPurchases
1             172               88           88                 3
2               2                1            6                 2
3             111               21           42                 1
4              10                3            5                 2
5              46               27           15                 5
6               0               42           14                 2
```

Figure 4 After data clean-up

7. Education Variable

There are 5 different values of Education, but according to the 'Three Cycle System' from the European Higher Education Area, 2n Cycle is equal to a Master's degree. A graduation degree is the same as a Bachelor's degree in Europe. Thus, we adjust the education level of all customers.



7. There are 5 different values of Education, but according to 'Three Cycle System' from the European Higher Education Area, 2n Cycle is equal to a Master degree. Graduation degree is the same as a Bachelor's degree in Europe. Thus, we adjust the education level of all customers.

~~~
```{r}
summary(data$Education)
for (i in 1:nrow(data)){
  if (data$Education[i]=="2n Cycle"){
    data$Education[i]<-"Master"
  }
}
levels(data$Education) <- c("2n Cycle","Basic","Bachelor","Master","PhD")
data$Education<-droplevels(data$Education)
summary(data$Education)
```
~~~

```
  2n Cycle      Basic Graduation      Master        PhD 
       203         54       1127         370        486 
   Basic Bachelor    Master       PhD 
      54     1127       573       486 
```

Figure 5 Education level of all Customers

Data Exploration

After preprocessing the data, we ran a simple distribution of our numeric variables utilizing boxplots. Here are some interesting things that we noticed within our research with some of the variables.

- Age: This variable has a pretty normal distribution with the average age being slightly less than 50 years old. There are at least two customers whose ages are over 100, which was able to show us that there are outliers within this data.
- Income: We've seen this variable a lot within the preprocessing part of our workflow. Once we have removed the outlier that is over 600k+. We can see that our preprocessing was successful in removing the outliers, seeing that the average salary now ranges around 50k, which actually makes sense.
- MntMeatProducts: The range is scaled differently as opposed to different products that are listed within this boxplot as well. The mean is around 150 dollars which is a lot higher than the other

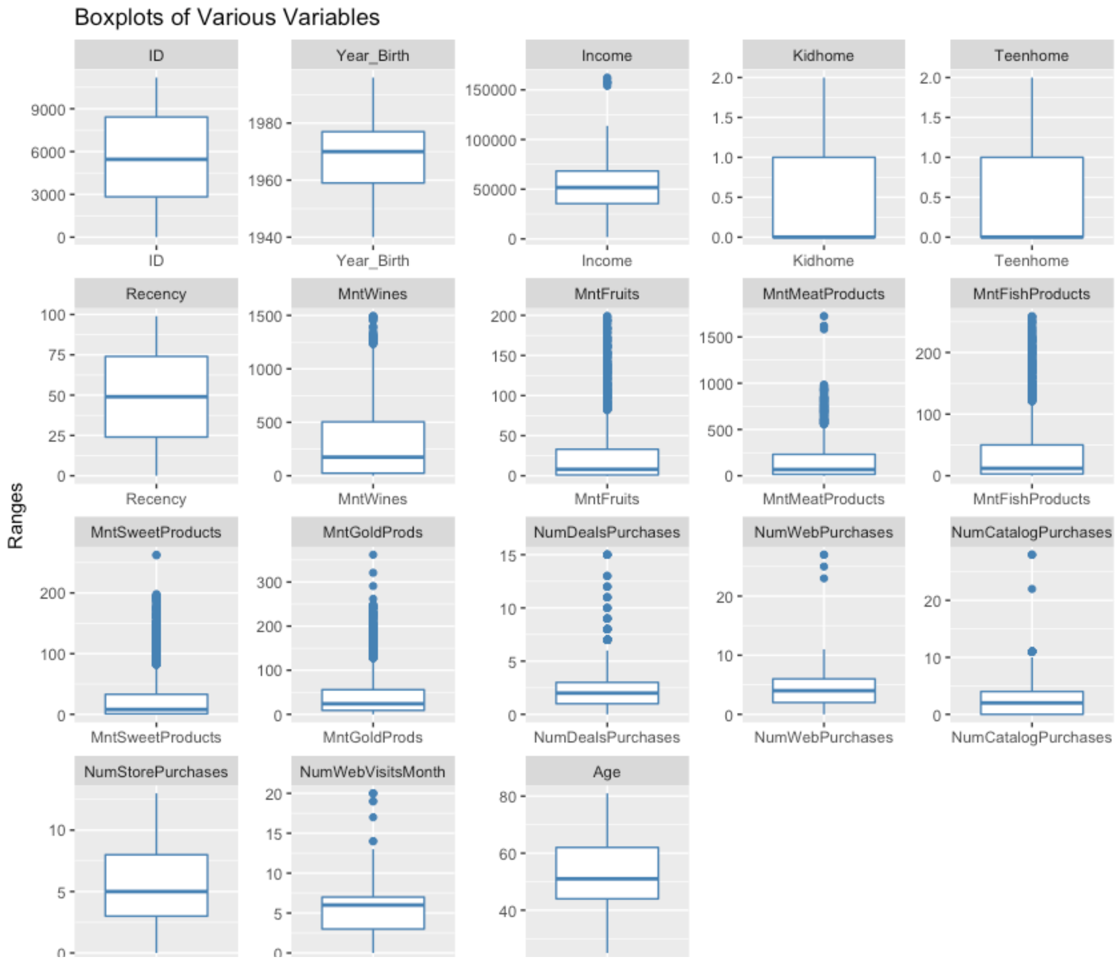


Figure 6 Boxplots of Variables

products. Anything above the price of 250 is very unusual, in which there are instances where there is meat priced as high as in the thousands.

Next, we ran a map of the correlation between the numerical variables. The correlations that are above 0.6 are highlighted in the light blue color, in which the red are ones that are below -0.6. The great things about these variables are that most of them seem to be highly positively correlated. There were many instances of the sum of many which also was the cause of many variables having a positive correlation. In the case of MntSpent and NumPurchases it is obvious that they will be positively correlated with each other and their components. The more money that is spent, it is also just insinuated that more purchases will also be made.

Other variables that are also correlated with good reason are Income to MntSpent, which shows that the higher the income the more money there is spent, the same logic can be applied to Income to NumPurchases. On the other hand, the only negative correlation that we can observe is between income and NumWebVisitsMonth, which indicates that customers with lower incomes will visit the website more frequently but make around the same amount of purchases as people with higher incomes. However, these are just observations that were made within our data exploration part of the workflow and should be taken into consideration, because correlation does not necessarily imply causation.

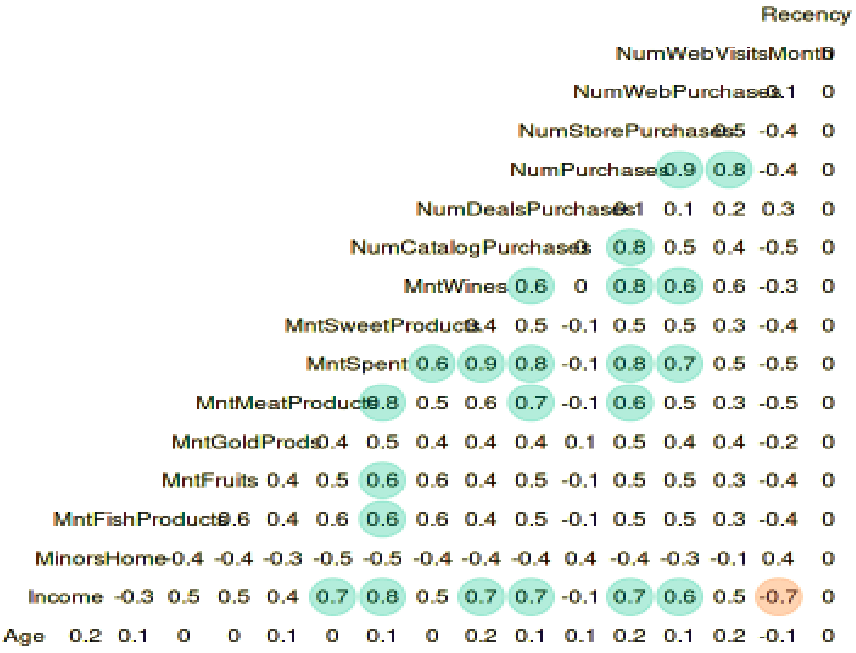


Figure 7 Observation of variables

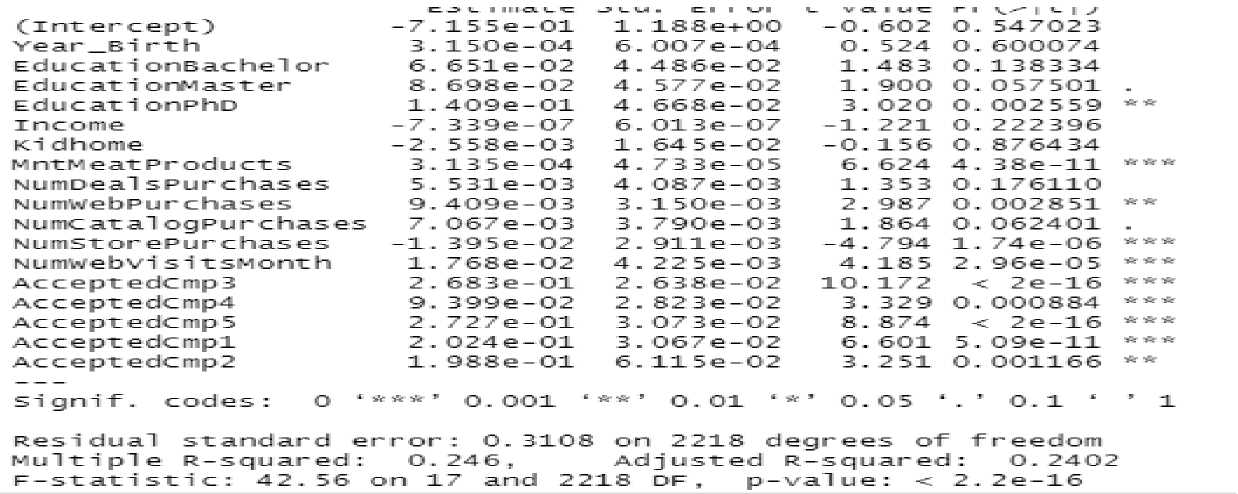
Methodology

To answer our prompted research questions, we ran many different models such as linear regression, logistic regression, KNN, and Decision Trees (more details follow). These models were crucial to our understanding of different consumer bases in different industry sectors. Most importantly we were able to identify key strengths and weaknesses in our models and find the best model for different purposes by taking into account the AE, RMSE, Specificity, Sensitivity, Accuracy, and the Confusion Matrix. Our specific methodology helped us gain a further understanding of our results and answer the research questions.

1. Linear Regression

Linear regression is used to predict the value of a dependent variable based on at least one or multiple independent variables. In our case, we are trying to predict the dependent variable which is the "Response" variable, and it shows that if a customer accepted the offer in the last advertising campaign they are labeled as "1" and if not, they are labeled as "0". We will examine some examples from our model to determine if our linear regression model supports the strength of the relationship between our dependent variable and the independent variable. Before building our model, we defined a few dummy variables, which are: Education, Martial_Status, and Dt_Customer.

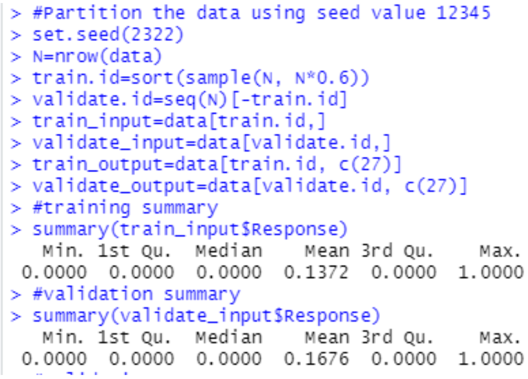
After formalizing the data, our first linear model is trying to determine or figure out our "Response" variable by examining Year_Birth, Education, Income, kidhome, MntMeatProducts, NumDealsPurchases, NumWebPurchases, NumCatalogPurchases, NumStorePurchases, NumWebVisitsMonth, AcceptedCmp3, AcceptedCmp4, AcceptedCmp5, AcceptedCmp1, and AcceptedCmp2.



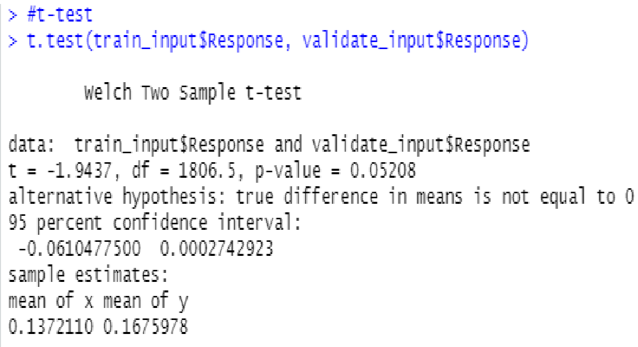
```
                        Estimate Std. Error t value Pr(>|t|)
(Intercept)           -7.155e-01  1.188e+00  -0.602 0.547023
Year_Birth             3.150e-04  6.007e-04   0.524 0.600074
EducationBachelor      6.651e-02  4.486e-02   1.483 0.138334
EducationMaster        8.698e-02  4.577e-02   1.900 0.057501 .
EducationPhD           1.409e-01  4.668e-02   3.020 0.002559 **
Income                -7.339e-07  6.013e-07  -1.221 0.222396
Kidhome               -2.558e-03  1.645e-02  -0.156 0.876434
MntMeatProducts        3.135e-04  4.733e-05   6.624 4.38e-11 ***
NumDealsPurchases      5.531e-03  4.087e-03   1.353 0.176110
NumWebPurchases        9.409e-03  3.150e-03   2.987 0.002851 **
NumCatalogPurchases    7.067e-03  3.790e-03   1.864 0.062401 .
NumStorePurchases     -1.395e-02  2.911e-03  -4.794 1.74e-06 ***
NumWebVisitsMonth      1.768e-02  4.225e-03   4.185 2.96e-05 ***
AcceptedCmp3           2.683e-01  2.638e-02  10.172  < 2e-16 ***
AcceptedCmp4           9.399e-02  2.823e-02   3.329 0.000884 ***
AcceptedCmp5           2.727e-01  3.073e-02   8.874  < 2e-16 ***
AcceptedCmp1           2.024e-01  3.067e-02   6.601 5.09e-11 ***
AcceptedCmp2           1.988e-01  6.115e-02   3.251 0.001166 **
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.3108 on 2218 degrees of freedom
Multiple R-squared:  0.246,	Adjusted R-squared:  0.2402 
F-statistic: 42.56 on 17 and 2218 DF,  p-value: < 2.2e-16
```

Figure 8 1st Linear Regression model

Based on the linear regression model, MntMeatProducts, NumStorePurchases, NumWebVisitsMonth, AcceptedCmp1, Cmp3, Cmp4, and Cmp5 have the strongest statistical significant relationship because their p-value is less than 0.05, which is very closer to 0.001. Followed by Education with PhD, NumWebPurchases, and AcceptedCmp2 with a p-value close to 0.01. The Adjusted R-squared value of this model tells us that this model is not quite reliable to be referred to as a good model due to a low value of 0.24, for it to be a good linear model the adjusted R-squared needs to be 0.54 or higher. After splitting and partitioning the data, we set seed to 2322 and split 60% of the data for the training and 40% for validation and summarizing the train and validated data with the "Response" variable to compose a t-test as shown below.



```
> #Partition the data using seed value 12345
> set.seed(2322)
> N=nrow(data)
> train.id=sort(sample(N, N*0.6))
> validate.id=seq(N)[-train.id]
> train_input=data[train.id,]
> validate_input=data[validate.id,]
> train_output=data[train.id, c(27)]
> validate_output=data[validate.id, c(27)]
> #training summary
> summary(train_input$Response)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 0.0000  0.0000  0.0000  0.1372  0.0000  1.0000 
> #validation summary
> summary(validate_input$Response)
   Min. 1st Qu.  Median    Mean 3rd Qu.    Max. 
 0.0000  0.0000  0.0000  0.1676  0.0000  1.0000 
```

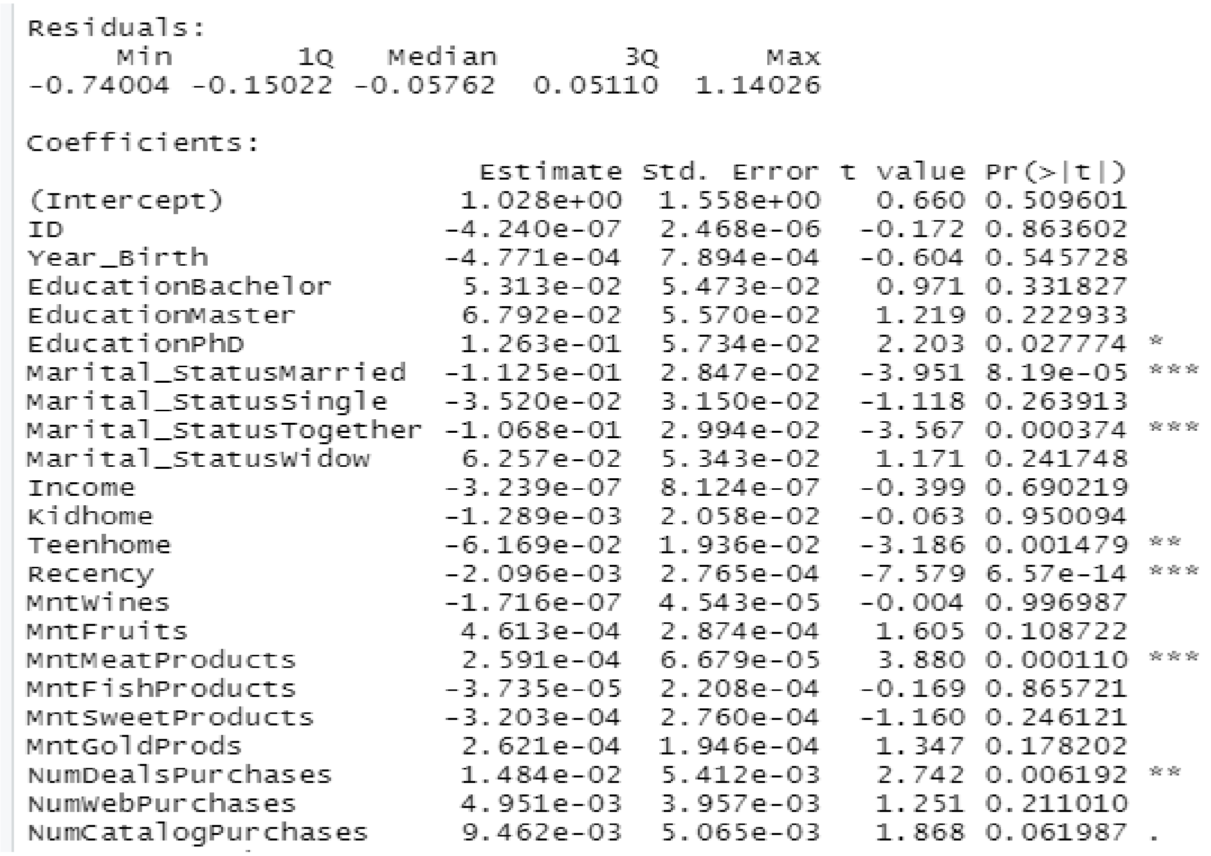


```
> #t-test
> t.test(train_input$Response, validate_input$Response)

	Welch Two Sample t-test

data:  train_input$Response and validate_input$Response
t = -1.9437, df = 1806.5, p-value = 0.05208
alternative hypothesis: true difference in means is not equal to 0
95 percent confidence interval:
 -0.0610477500  0.0002742923
sample estimates:
mean of x mean of y 
0.1372110 0.1675978 
```

Figure 9 Partitioning data sets (60% training, 40% validation)

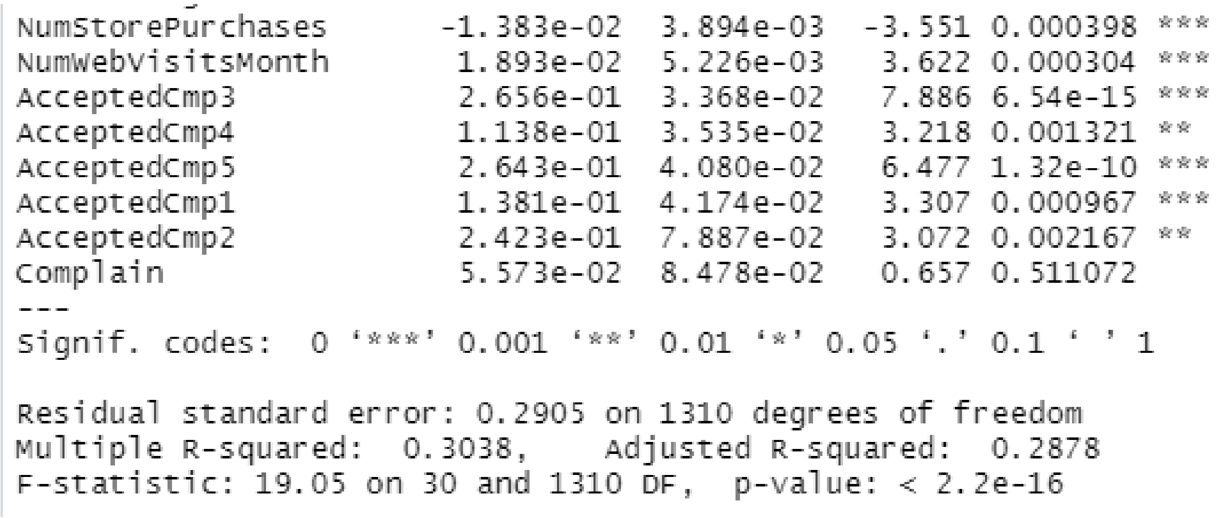
Figure 10 T-test observation

After this, we constructed another linear regression model for the training data which is also the second leaner regression model. As you can see, we omitted columns 8,27,28 and 30. Those columns are: Dt_Customer, Z_CostContact, Z_Revenue, and Age as a dummy variable. The reason that these columns were omitted is because they contain empty values.



```
Residuals:
     Min       1Q   Median       3Q      Max 
-0.74004 -0.15022 -0.05762  0.05110  1.14026 

Coefficients:
                          Estimate Std. Error t value Pr(>|t|)    
(Intercept)              1.028e+00  1.558e+00   0.660 0.509601    
ID                      -4.240e-07  2.468e-06  -0.172 0.863602    
Year_Birth              -4.771e-04  7.894e-04  -0.604 0.545728    
EducationBachelor        5.313e-02  5.473e-02   0.971 0.331827    
EducationMaster          6.792e-02  5.570e-02   1.219 0.222933    
EducationPhD             1.263e-01  5.734e-02   2.203 0.027774 *  
Marital_StatusMarried   -1.125e-01  2.847e-02  -3.951 8.19e-05 ***
Marital_StatusSingle    -3.520e-02  3.150e-02  -1.118 0.263913    
Marital_StatusTogether -1.068e-01  2.994e-02  -3.567 0.000374 ***
Marital_StatusWidow      6.257e-02  5.343e-02   1.171 0.241748    
Income                  -3.239e-07  8.124e-07  -0.399 0.690219    
Kidhome                 -1.289e-03  2.058e-02  -0.063 0.950094    
Teenhome                -6.169e-02  1.936e-02  -3.186 0.001479 ** 
Recency                 -2.096e-03  2.765e-04  -7.579 6.57e-14 ***
MntWines                -1.716e-07  4.543e-05  -0.004 0.996987    
MntFruits                4.613e-04  2.874e-04   1.605 0.108722    
MntMeatProducts          2.591e-04  6.679e-05   3.880 0.000110 ***
MntFishProducts         -3.735e-05  2.208e-04  -0.169 0.865721    
MntSweetProducts        -3.203e-04  2.760e-04  -1.160 0.246121    
MntGoldProds             2.621e-04  1.946e-04   1.347 0.178202    
NumDealsPurchases        1.484e-02  5.412e-03   2.742 0.006192 ** 
NumWebPurchases          4.951e-03  3.957e-03   1.251 0.211010    
NumCatalogPurchases      9.462e-03  5.065e-03   1.868 0.061987 .  
```

Figure 11 Linear Regression model after data partitioning



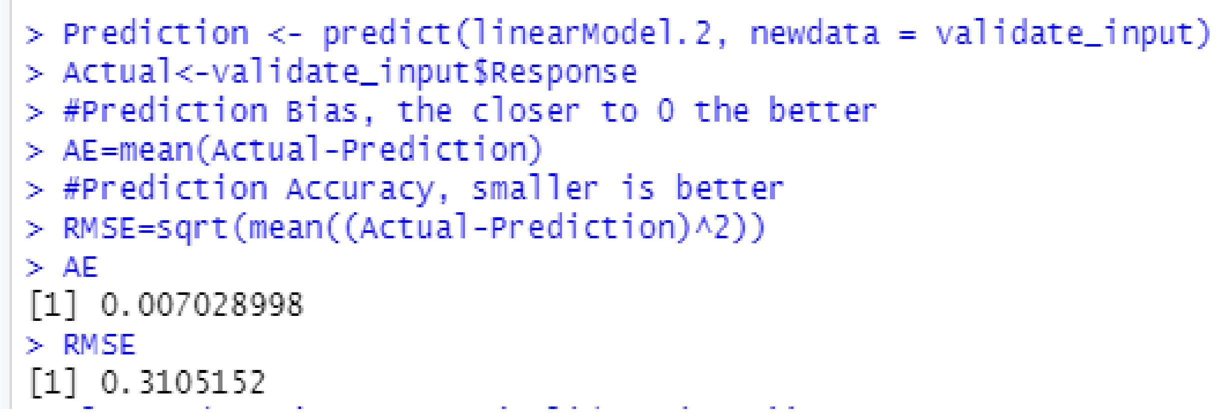
```
NumStorePurchases       -1.383e-02  3.894e-03  -3.551 0.000398 ***
NumWebVisitsMonth        1.893e-02  5.226e-03   3.622 0.000304 ***
AcceptedCmp3             2.656e-01  3.368e-02   7.886 6.54e-15 ***
AcceptedCmp4             1.138e-01  3.535e-02   3.218 0.001321 ** 
AcceptedCmp5             2.643e-01  4.080e-02   6.477 1.32e-10 ***
AcceptedCmp1             1.381e-01  4.174e-02   3.307 0.000967 ***
AcceptedCmp2             2.423e-01  7.887e-02   3.072 0.002167 ** 
Complain                 5.573e-02  8.478e-02   0.657 0.511072    
---
Signif. codes:  0 ‘***’ 0.001 ‘**’ 0.01 ‘*’ 0.05 ‘.’ 0.1 ‘ ’ 1

Residual standard error: 0.2905 on 1310 degrees of freedom
Multiple R-squared:  0.3038,	Adjusted R-squared:  0.2878 
F-statistic: 19.05 on 30 and 1310 DF,  p-value: < 2.2e-16
```

Figure 12 Linear Regression model after data partitioning

The second model includes marital_status, teenhome, and Recency to be of strong statistical significance and shows a better linear model than our first linear regression model, with a slightly higher adjusted R-squared value of 0.29. In the next step, we will examine prediction accuracy, Average Error, and Root-mean-square error of our second linear model.

1. AE and RMSE

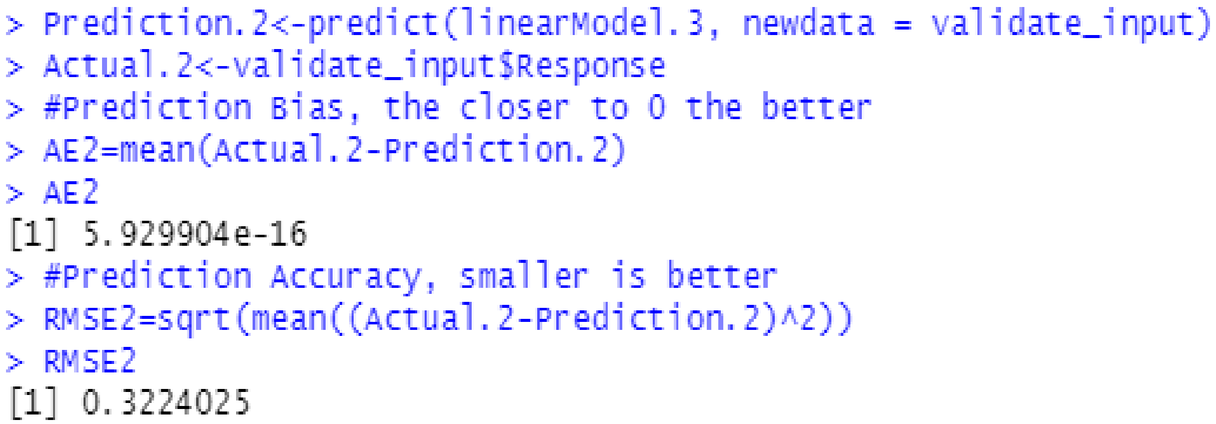
2nd Linear Regression Model



```
> Prediction <- predict(linearModel.2, newdata = validate_input)
> Actual<-validate_input$Response
> #Prediction Bias, the closer to 0 the better
> AE=mean(Actual-Prediction)
> #Prediction Accuracy, smaller is better
> RMSE=sqrt(mean((Actual-Prediction)^2))
> AE
[1] 0.007028998
> RMSE
[1] 0.3105152
```

Figure 13 Average error and root-mean-squared-error prediction

3rd Linear Regression Model



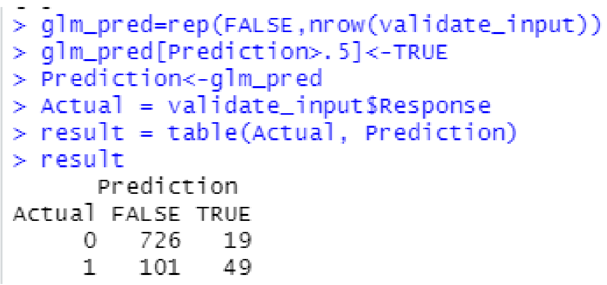
```
> Prediction.2<-predict(linearModel.3, newdata = validate_input)
> Actual.2<-validate_input$Response
> #Prediction Bias, the closer to 0 the better
> AE2=mean(Actual.2-Prediction.2)
> AE2
[1] 5.929904e-16
> #Prediction Accuracy, smaller is better
> RMSE2=sqrt(mean((Actual.2-Prediction.2)^2))
> RMSE2
[1] 0.3224025
```

Figure 14 3rd Linear model average error and root-mean-squared-error prediction

The Average Error value is very close to 0 it is slightly lower than 0 with a value of 0.007 hence creating a prediction unbiased, if the value is closer to 0 the better the Average error value is. On the other hand, the Root-mean-squared error (RMSE) has a value of 0.310, which tells us the prediction accuracy and that the smaller the value the better. In this case, our prediction model of the validation data set is reliable based on the AE and RMSE assessment.

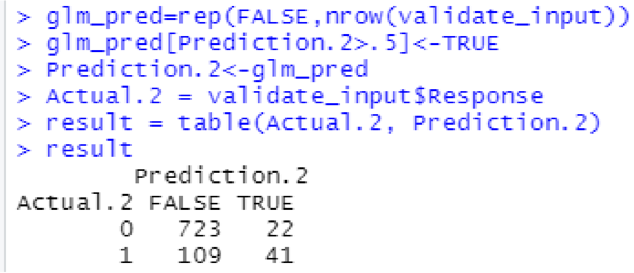
2. Confusion Matrix

2nd Linear Regression Model (left) & 3rd Linear Regression Model (right)



```
> glm_pred=rep(FALSE,nrow(validate_input))
> glm_pred[Prediction>.5]<-TRUE
> Prediction<-glm_pred
> Actual = validate_input$Response
> result = table(Actual, Prediction)
> result
      Prediction
Actual FALSE TRUE
     0   726   19
     1   101   49
```

Figure 15 Confusion Matrix (2nd model)

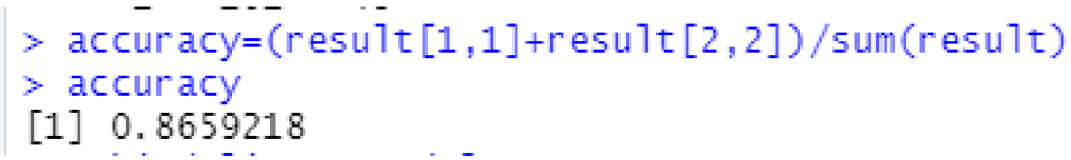


```
> glm_pred=rep(FALSE,nrow(validate_input))
> glm_pred[Prediction.2>.5]<-TRUE
> Prediction.2<-glm_pred
> Actual.2 = validate_input$Response
> result = table(Actual.2, Prediction.2)
> result
        Prediction.2
Actual.2 FALSE TRUE
       0   723   22
       1   109   41
```

Figure 16 Confusion Matrix (3rd model)

3. Prediction Accuracy

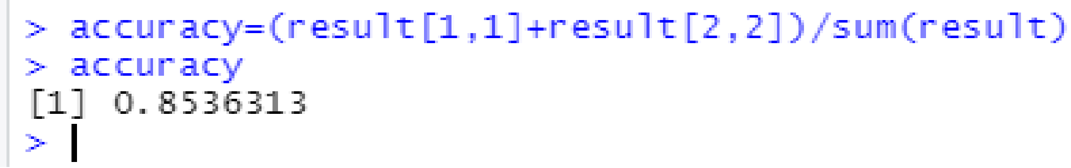
2nd Linear Regression Model



```
> accuracy=(result[1,1]+result[2,2])/sum(result)
> accuracy
[1] 0.8659218
```

Figure 17 Accuracy prediction (2nd model)

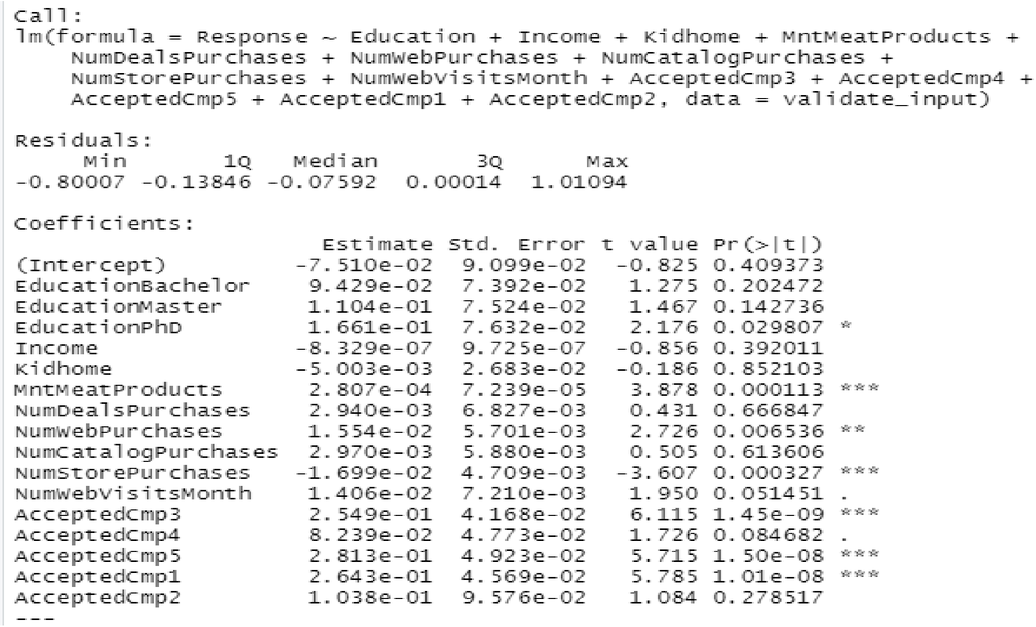
3rd Linear Regression Model



```
> accuracy=(result[1,1]+result[2,2])/sum(result)
> accuracy
[1] 0.8536313
> 
```

Figure 18 Accuracy prediction (3rd model)

Lastly, we will use a third linear regression model to evaluate the validation data sets.



```
Call:
lm(formula = Response ~ Education + Income + Kidhome + MntMeatProducts +
    NumDealsPurchases + NumWebPurchases + NumCatalogPurchases +
    NumStorePurchases + NumWebVisitsMonth + AcceptedCmp3 + AcceptedCmp4 +
    AcceptedCmp5 + AcceptedCmp1 + AcceptedCmp2, data = validate_input)

Residuals:
     Min       1Q   Median       3Q      Max
-0.80007 -0.13846 -0.07592  0.00014  1.01094

Coefficients:
                      Estimate Std. Error t value Pr(>|t|)
(Intercept)         -7.510e-02  9.099e-02  -0.825 0.409373
EducationBachelor    9.429e-02  7.392e-02   1.275 0.202472
EducationMaster      1.104e-01  7.524e-02   1.467 0.142736
EducationPhD         1.661e-01  7.632e-02   2.176 0.029807 *
Income              -8.329e-07  9.725e-07  -0.856 0.392011
Kidhome             -5.003e-03  2.683e-02  -0.186 0.852103
MntMeatProducts      2.807e-04  7.239e-05   3.878 0.000113 ***
NumDealsPurchases    2.940e-03  6.827e-03   0.431 0.666847
NumWebPurchases      1.554e-02  5.701e-03   2.726 0.006536 **
NumCatalogPurchases  2.970e-03  5.880e-03   0.505 0.613606
NumStorePurchases   -1.699e-02  4.709e-03  -3.607 0.000327 ***
NumWebVisitsMonth    1.406e-02  7.210e-03   1.950 0.051451 .
AcceptedCmp3         2.549e-01  4.168e-02   6.115 1.45e-09 ***
AcceptedCmp4         8.239e-02  4.773e-02   1.726 0.084682 .
AcceptedCmp5         2.813e-01  4.923e-02   5.715 1.50e-08 ***
AcceptedCmp1         2.643e-01  4.569e-02   5.785 1.01e-08 ***
AcceptedCmp2         1.038e-01  9.576e-02   1.084 0.278517
---
```

Figure 19 3rd Linear regression model

In this model, we can see that the Adjusted R-squared dropped a little bit than the previous model and the residuals have slight variations compared to our second linear regression model and that can be attributed to excluding some independent variables in this model that we included in our second linear model. The accuracy prediction analysis for our third and last regression model on the basis of the validation data set is as follows; the average error rate is 5.92991e-16 and the RMSE is 0.32, which is good because the smaller the value the better the prediction accuracy. Linear regression is a very useful tool, but not always because our analysis depends on the best fit line and that is okay, it does not mean the data model we are observing is wrong. In our observation, the linear regression model is not a very good model for observing customers' personality analysis.

2. Logistic Regression

Pros and Cons

Logistic regression is a predictive analysis used when the dependent variable is binary. It describes the correlation between one binary variable and other ordinal, nominal, or interval variables.

Advantages:

- Logistic regression is simple and easy to implement. Such is due to its training efficiency, which requires low computation power.
- It provides some training examples in cases of low dimensional datasets hence less prone to over-fitting

- Due to its simplicity, logistic regression can be used as a baseline to assess the performance of other complex algorithms. In other words, it eases the computation of complex models, where their use is paramount.

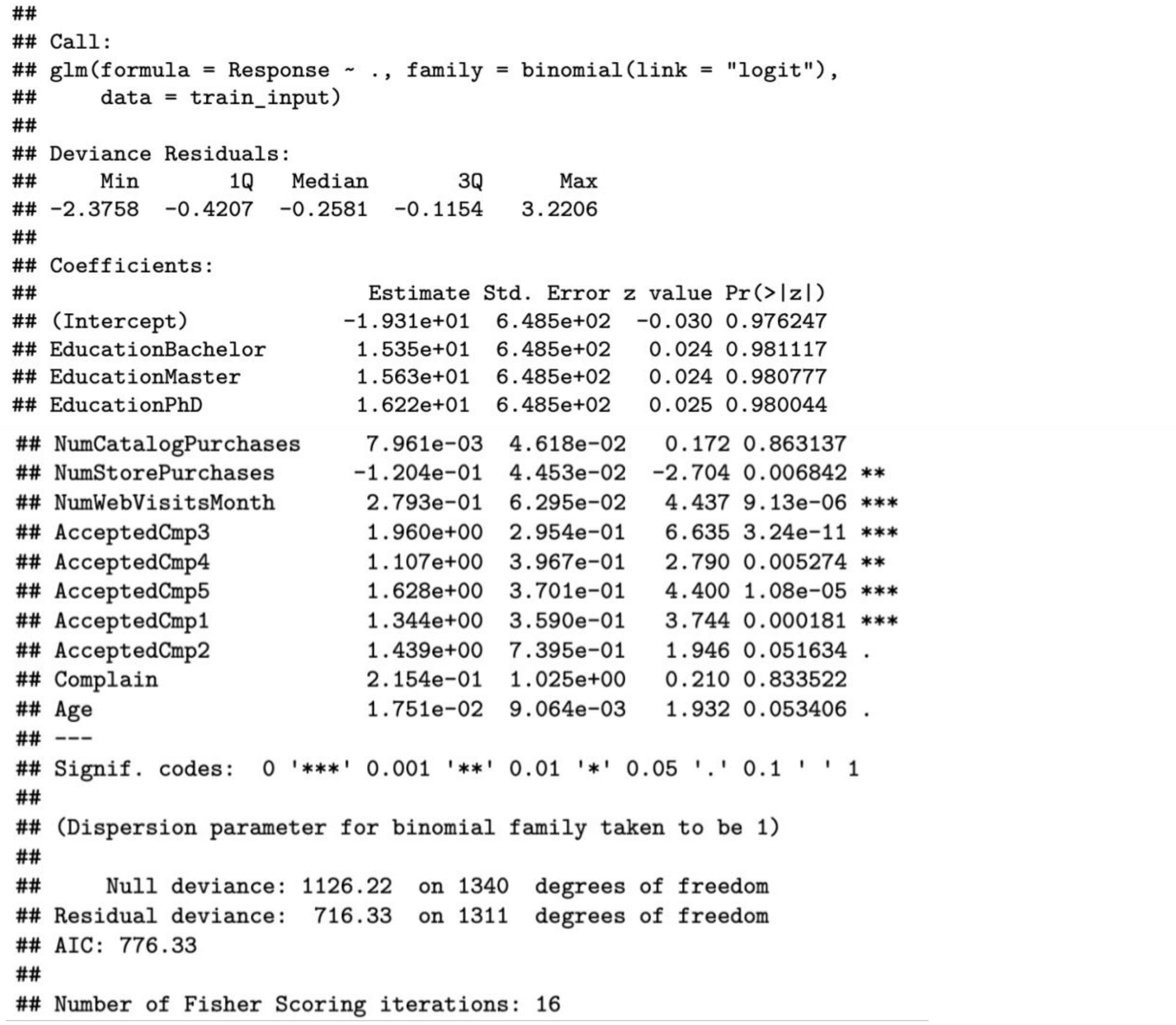
Disadvantages:

- Logistic regression is vulnerable to overfitting due to sampling bias. In other words, the models can depict more predictive power than the actual one hence leading to the inaccuracy of results.
- This model requires the independence of every data point against other points. In this regard, a relationship between observations would make the model overweight their significance. Such is risky because most of the social and scientific research involves multiple observations of similar individuals.
- Logistic regression has limited outcome variables. The model is inapplicable in cases of continuous datasets hence cannot predict progressive outcomes. As such, the researchers using continuous datasets would be forced to categorize data into discrete forms which would sacrifice its precision.

Predict Response

This dataset gives 2240 different customers basic information, their product purchasing preferences as well as their reactions to some marketing campaigns. We will perform logistic prediction tasks on this dataset. As the data description says, the column 'Response' stands for if a certain customer accepted the offer in the last campaign. So the question is whether we can use some customers' responses to this campaign to predict someone else's reactions? If we can achieve this, a business could promote the campaign to customers that are more likely to accept the offer, which could help it make a more efficient marketing plan.

We first run the following codes to generate the logistic regression on the response variable with all variables:



```
##
## Call:
## glm(formula = Response ~ ., family = binomial(link = "logit"),
##     data = train_input)
##
## Deviance Residuals:
##     Min       1Q   Median       3Q      Max
## -2.3758  -0.4207  -0.2581  -0.1154   3.2206
##
## Coefficients:
##                        Estimate Std. Error z value Pr(>|z|)
## (Intercept)          -1.931e+01  6.485e+02  -0.030 0.976247
## EducationBachelor     1.535e+01  6.485e+02   0.024 0.981117
## EducationMaster       1.563e+01  6.485e+02   0.024 0.980777
## EducationPhD          1.622e+01  6.485e+02   0.025 0.980044
## NumCatalogPurchases    7.961e-03  4.618e-02   0.172 0.863137
## NumStorePurchases     -1.204e-01  4.453e-02  -2.704 0.006842 **
## NumWebVisitsMonth      2.793e-01  6.295e-02   4.437 9.13e-06 ***
## AcceptedCmp3           1.960e+00  2.954e-01   6.635 3.24e-11 ***
## AcceptedCmp4           1.107e+00  3.967e-01   2.790 0.005274 **
## AcceptedCmp5           1.628e+00  3.701e-01   4.400 1.08e-05 ***
## AcceptedCmp1           1.344e+00  3.590e-01   3.744 0.000181 ***
## AcceptedCmp2           1.439e+00  7.395e-01   1.946 0.051634 .
## Complain               2.154e-01  1.025e+00   0.210 0.833522
## Age                    1.751e-02  9.064e-03   1.932 0.053406 .
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## (Dispersion parameter for binomial family taken to be 1)
##
##     Null deviance: 1126.22  on 1340  degrees of freedom
## Residual deviance:  716.33  on 1311  degrees of freedom
## AIC: 776.33
##
## Number of Fisher Scoring iterations: 16
```

Figure 20 Logistic regression model

The accuracy of the model is 0.8826816, with a sensitivity of .6630 and specificity of .90784. The Average Error is .0480 and the Root Mean Square of Error (RMSE) is .3425.

3. KNN

Pros and Cons

One of the simplest models to use is the K-nearest neighbor. This is because it uses its proximity to the next nearest data point to classify new information. It does this by counting the majority of votes from its nearest neighbors. For regression, it will average the numbers of its nearest neighbors to make a prediction. How many data points it uses around it is called K.

Advantages:

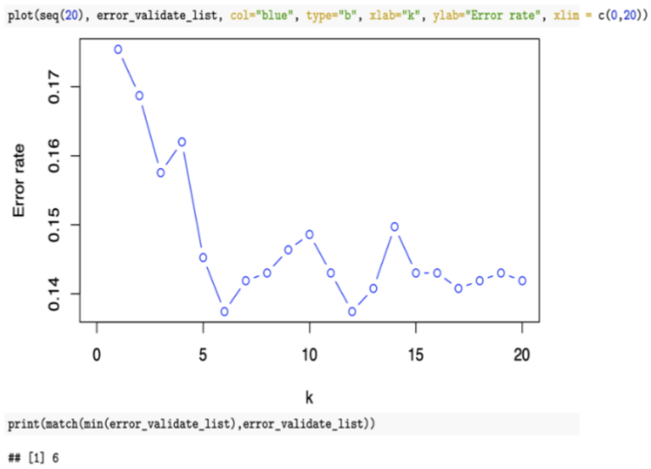
- Its main advantage is its simplicity. It does not require a lot of data if it's only using a few dimensions.
- It can be used for both classification and prediction. It does not assume any underlying parameters.
- It is easy to add new data to make a prediction.

Disadvantages:

The KNN model requires a lot of data to be the most accurate with a large number of dimensions. Because of the large amount of data, the time to process all of the data will also be high.

Prediction

Before applying the KNN to predict customer responses, we need first to understand what's the best number of neighbors to predict responses. We achieve this by running the K range from 1 to 20 and comparing them by plotting the errors. Based on the plots and min value match, the k=6 provides the best prediction.

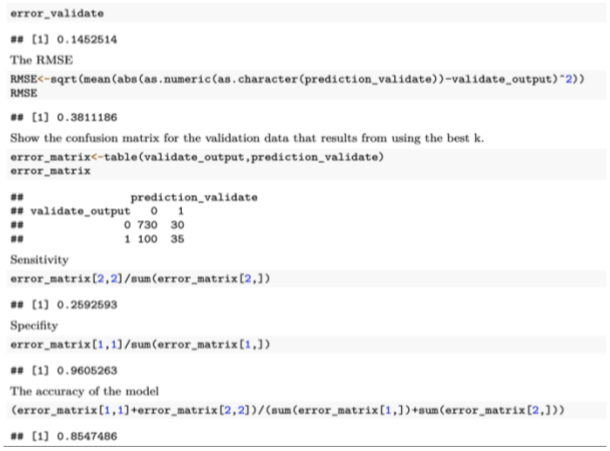


```
plot(seq(20), error_validate_list, col="blue", type="b", xlab="k", ylab="Error rate", xlim = c(0,20))
```

```
print(match(min(error_validate_list),error_validate_list))
```

```
## [1] 6
```

Figure 21 Plot of Error rate



```
error_validate
```

```
## [1] 0.1452514
```

The RMSE

```
RMSE<-sqrt(mean(abs(as.numeric(as.character(prediction_validate))-validate_output)^2))
RMSE
```

```
## [1] 0.3811186
```

Show the confusion matrix for the validation data that results from using the best k.

```
error_matrix<-table(validate_output,prediction_validate)
error_matrix
```

```
##                prediction_validate
## validate_output   0   1
##               0 730  30
##               1 100  35
```

Sensitivity

```
error_matrix[2,2]/sum(error_matrix[2,])
```

```
## [1] 0.2592593
```

Specifity

```
error_matrix[1,1]/sum(error_matrix[1,])
```

```
## [1] 0.9605263
```

The accuracy of the model

```
(error_matrix[1,1]+error_matrix[2,2])/(sum(error_matrix[1,])+sum(error_matrix[2,]))
```

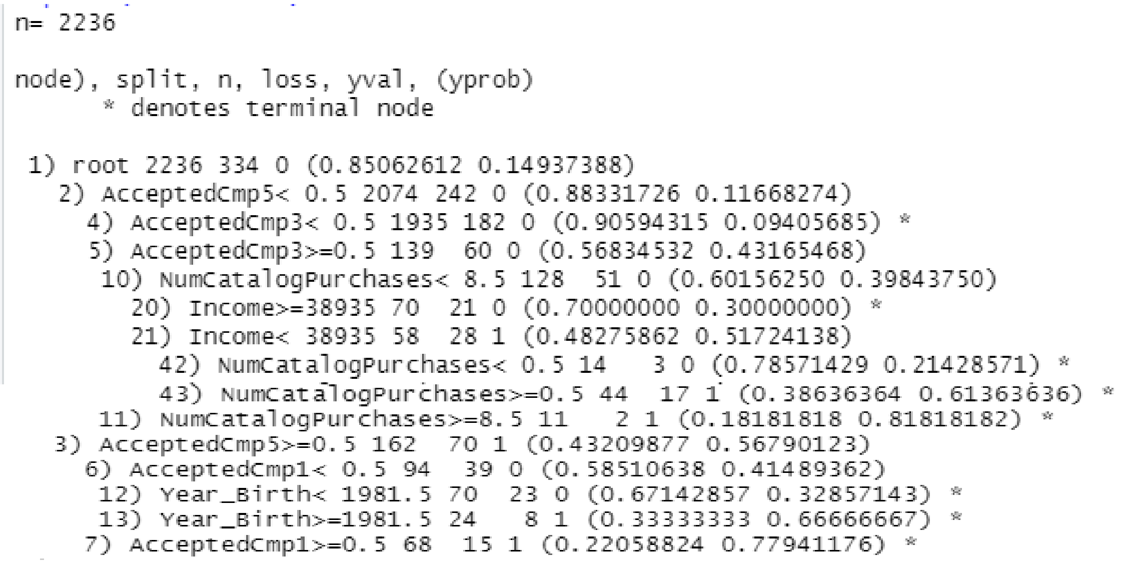
```
## [1] 0.8547486
```

Figure 22 Error rate and Confusion Matrix analysis

The accuracy of the model is 0.8547, with a sensitivity of 0.2593 and specificity of 0.9605. The Average Error is 0.1453 and the Root Mean Square of Error (RMSE) is 0.3811.

4. Decision Tree

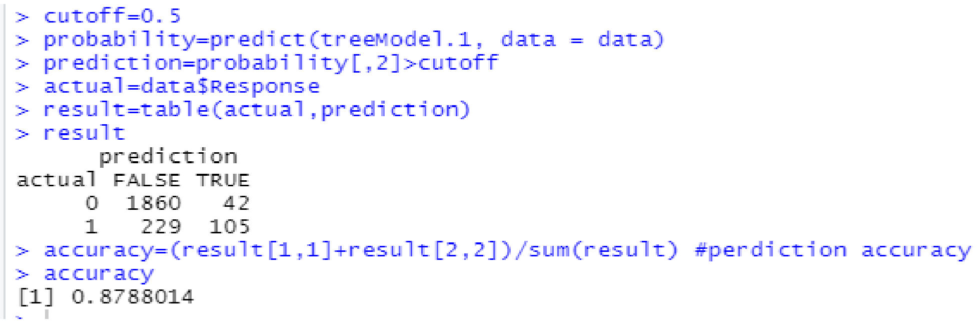
In our decision tree model, the root starts with 2,236 data points, which is the total number of data points in our model, and from that data point, 334 customers accept or say yes to the advertising campaign. The majority of the customers say no to the advertising campaign indicated by the number 0 to the right side of the 334 customers that accept the advertising campaign. The percentage of customers who said no to the campaign is 85% and 15% said yes or accepted the advertising campaign. The second line shows the split between AcceptedCmp5 less than 0.5, and the third node shows AcceptedCmp5 greater than or equal to 0.5. Node 4 shows Cmp3 less than 0.5 doesn't accept the advertising with a majority of 95% saying no to the ad campaign. 70% of customers' income is less than or equal to 38,935.



```
n= 2236

node), split, n, loss, yval, (yprob)
      * denotes terminal node

 1) root 2236 334 0 (0.85062612 0.14937388)
   2) AcceptedCmp5< 0.5 2074 242 0 (0.88331726 0.11668274)
     4) AcceptedCmp3< 0.5 1935 182 0 (0.90594315 0.09405685) *
     5) AcceptedCmp3>=0.5 139  60 0 (0.56834532 0.43165468)
      10) NumCatalogPurchases< 8.5 128  51 0 (0.60156250 0.39843750)
        20) Income>=38935 70  21 0 (0.70000000 0.30000000) *
        21) Income< 38935 58  28 1 (0.48275862 0.51724138)
          42) NumCatalogPurchases< 0.5 14   3 0 (0.78571429 0.21428571) *
          43) NumCatalogPurchases>=0.5 44  17 1 (0.38636364 0.61363636) *
      11) NumCatalogPurchases>=8.5 11   2 1 (0.18181818 0.81818182) *
   3) AcceptedCmp5>=0.5 162  70 1 (0.43209877 0.56790123)
     6) AcceptedCmp1< 0.5 94  39 0 (0.58510638 0.41489362)
      12) Year_Birth< 1981.5 70  23 0 (0.67142857 0.32857143) *
      13) Year_Birth>=1981.5 24   8 1 (0.33333333 0.66666667) *
     7) AcceptedCmp1>=0.5 68  15 1 (0.22058824 0.77941176) *
```

Figure 23 Decision tree model

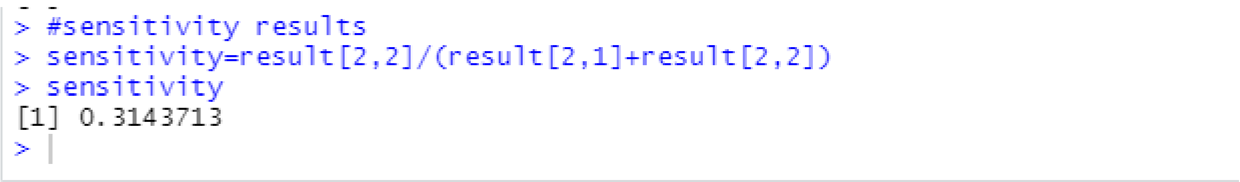
The prediction model for the first decision tree uses a cutoff value of 0.5 and it shows that the value 105 is true and from the prediction 229 were wrong predictions and 1860 of being false predictions. The accuracy prediction shows 88% of the accuracy of the first tree model.



```
> cutoff=0.5
> probability=predict(treeModel.1, data = data)
> prediction=probability[,2]>cutoff
> actual=data$Response
> result=table(actual,prediction)
> result
      prediction
actual FALSE TRUE
     0  1860   42
     1   229  105
> accuracy=(result[1,1]+result[2,2])/sum(result) #perdiction accuracy
> accuracy
[1] 0.8788014
```

Figure 24 Confusion Matrix and prediction accuracy

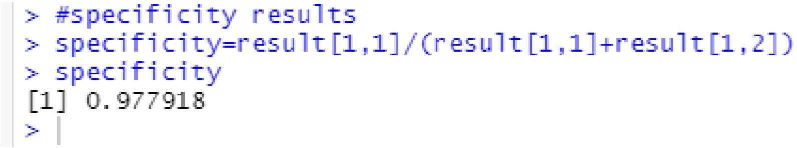
The sensitivity rate shows us how various kinds of independent variables affect the dependent variable. This metric shows the true positives of each category. In this model the true positives correctly identified is 0.31 or 31%.



```
> #sensitivity results
> sensitivity=result[2,2]/(result[2,1]+result[2,2])
> sensitivity
[1] 0.3143713
> 
```

Figure 25 1st model sensitivity result

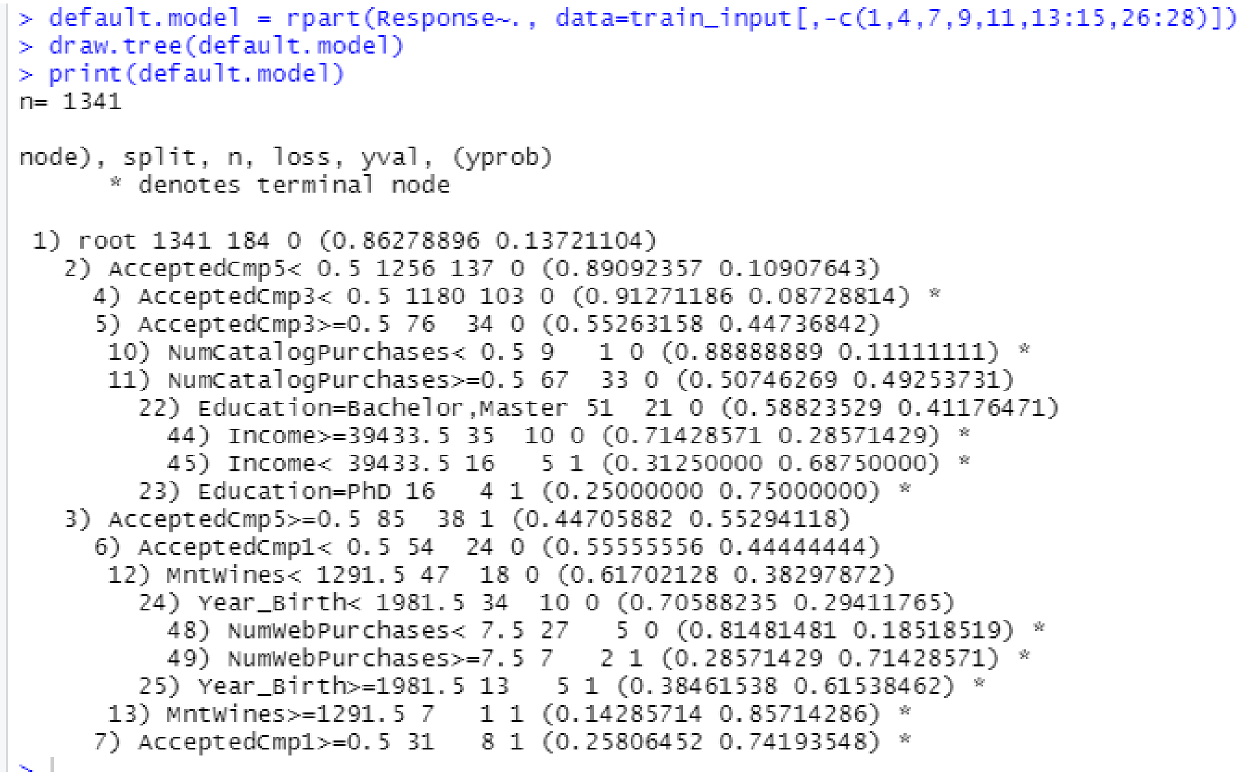
The specificity metric shows the true negatives of each category. In this first decision tree model, there is a 98% rate of true negatives that are identified from this decision model.



```
> #specificity results
> specificity=result[1,1]/(result[1,1]+result[1,2])
> specificity
[1] 0.977918
> 
```

Figure 26 2nd model specificity result

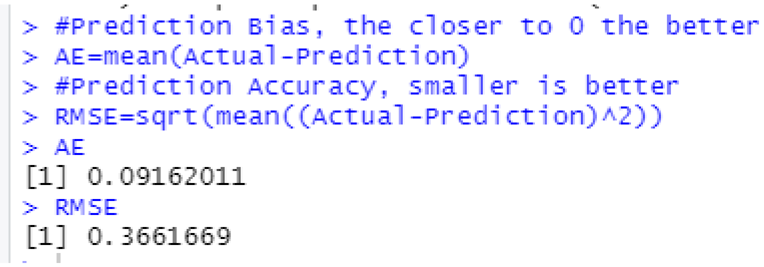
Splitting the data and designating 60% of the data set for training and 40% for validation and creating the second decision tree or the default model for training purposes as shown below. We also excluded certain columns from this training model because it will give us a redundant outcome if we keep in the training data set and build the default model decision tree. The result of the default model is relatively similar to the first decision model we did, but now it includes more variables since it is split for training and validation observations. In the first root line, it shows that out of 1341 customers 86% said no and out of 184 customers 14% said yes to the advertising campaign.



```
> default.model = rpart(Response~., data=train_input[,-c(1,4,7,9,11,13:15,26:28)])
> draw.tree(default.model)
> print(default.model)
n= 1341

node), split, n, loss, yval, (yprob)
      * denotes terminal node

 1) root 1341 184 0 (0.86278896 0.13721104)
   2) AcceptedCmp5< 0.5 1256 137 0 (0.89092357 0.10907643)
     4) AcceptedCmp3< 0.5 1180 103 0 (0.91271186 0.08728814) *
     5) AcceptedCmp3>=0.5 76  34 0 (0.55263158 0.44736842)
      10) NumCatalogPurchases< 0.5 9   1 0 (0.88888889 0.11111111) *
      11) NumCatalogPurchases>=0.5 67  33 0 (0.50746269 0.49253731)
        22) Education=Bachelor,Master 51  21 0 (0.58823529 0.41176471)
          44) Income>=39433.5 35  10 0 (0.71428571 0.28571429) *
          45) Income< 39433.5 16   5 1 (0.31250000 0.68750000) *
        23) Education=PhD 16   4 1 (0.25000000 0.75000000) *
   3) AcceptedCmp5>=0.5 85  38 1 (0.44705882 0.55294118)
     6) AcceptedCmp1< 0.5 54  24 0 (0.55555556 0.44444444)
      12) MntWines< 1291.5 47  18 0 (0.61702128 0.38297872)
        24) Year_Birth< 1981.5 34  10 0 (0.70588235 0.29411765)
          48) NumWebPurchases< 7.5 27   5 0 (0.81481481 0.18518519) *
          49) NumWebPurchases>=7.5 7   2 1 (0.28571429 0.71428571) *
        25) Year_Birth>=1981.5 13   5 1 (0.38461538 0.61538462) *
      13) MntWines>=1291.5 7   1 1 (0.14285714 0.85714286) *
     7) AcceptedCmp1>=0.5 31   8 1 (0.25806452 0.74193548) *
```

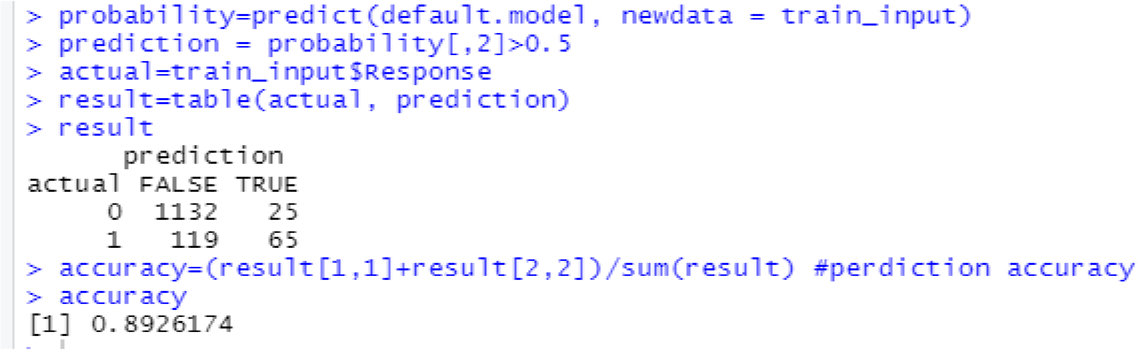
Figure 27 2nd decision tree model

The prediction model for the second decision tree also uses a cutoff value of 0.5 for the new validation data sets and it shows that the value 65 is true and from the prediction 119 were wrong predictions and 1132 of being true positives. The accuracy prediction shows 89% of the accuracy of the default tree model with new data sets for the validation observation.



```
> #Prediction Bias, the closer to 0 the better
> AE=mean(Actual-Prediction)
> #Prediction Accuracy, smaller is better
> RMSE=sqrt(mean((Actual-Prediction)^2))
> AE
[1] 0.09162011
> RMSE
[1] 0.3661669
```

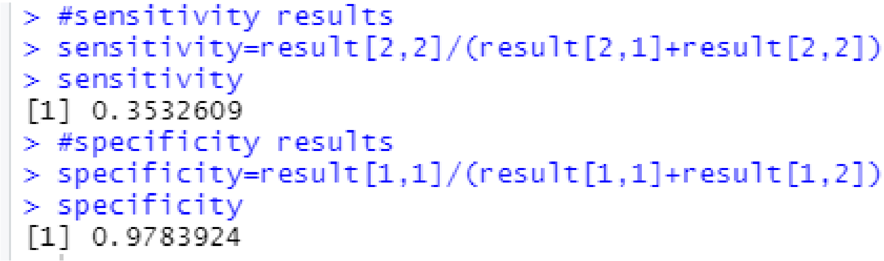
Figure 28 2nd model Average error and root-mean-squared-error



```
> probability=predict(default.model, newdata = train_input)
> prediction = probability[,2]>0.5
> actual=train_input$Response
> result=table(actual, prediction)
> result
      prediction
actual FALSE TRUE
     0  1132   25
     1   119   65
> accuracy=(result[1,1]+result[2,2])/sum(result) #perdiction accuracy
> accuracy
[1] 0.8926174
```

Figure 29 2nd model Confusion matrix and prediction accuracy rate

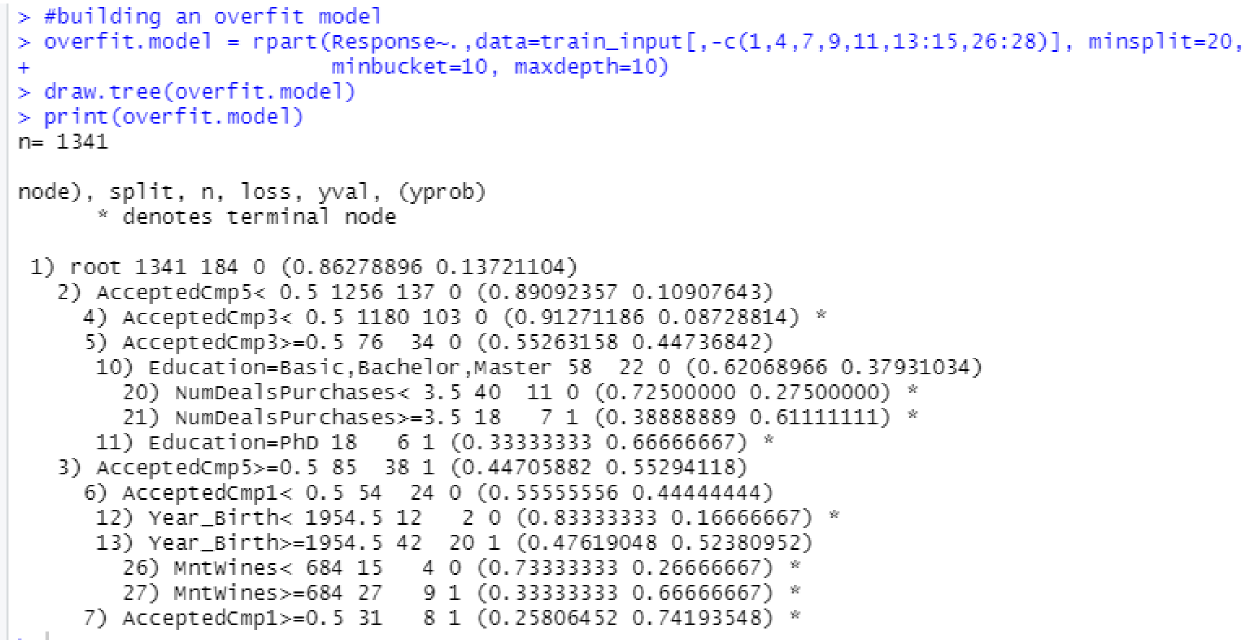
The sensitivity rate shows us how various kinds of independent variables affect the dependent variable. This metric shows the true positives of each category. In this model, the true positives correctly identified is 0.35 or 35%. In addition, the specificity result metric shows the true negatives of each category. In this second decision tree model for the training data sets, there is a 97% rate of true negatives that are identified from this second decision model.



```
> #sensitivity results
> sensitivity=result[2,2]/(result[2,1]+result[2,2])
> sensitivity
[1] 0.3532609
> #specificity results
> specificity=result[1,1]/(result[1,1]+result[1,2])
> specificity
[1] 0.9783924
```

Figure 30 2nd model Sensitivity and Specificity results

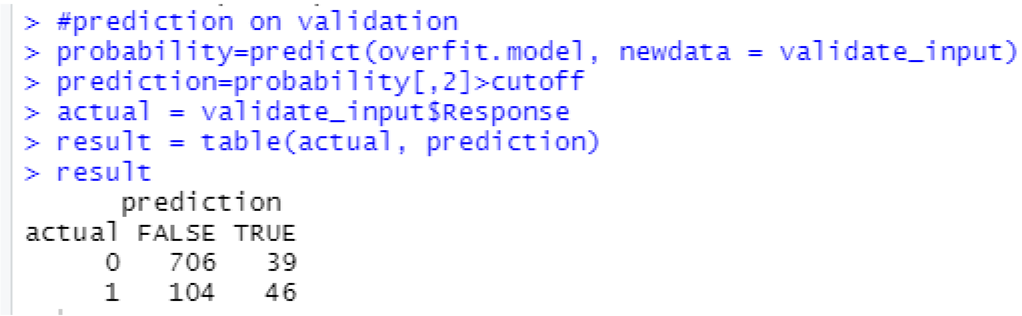
In the overfitted model, we used independent variables with the parameters of minsplit=20, minbucket=10, maxdepth=10. We stayed consistent with the cutoff value prediction of 0.5 throughout all the tree models we used. The tree model has a high significant accuracy of 84%, and low sensitivity of 30% therefore the customers are not easily convinced with targeted aid campaigns or influenced by advertisements to make a purchase of their choice.



```
> #building an overfit model
> overfit.model = rpart(Response~.,data=train_input[,-c(1,4,7,9,11,13:15,26:28)], minsplit=20,
+                       minbucket=10, maxdepth=10)
> draw.tree(overfit.model)
> print(overfit.model)
n= 1341

node), split, n, loss, yval, (yprob)
      * denotes terminal node

 1) root 1341 184 0 (0.86278896 0.13721104)
   2) AcceptedCmp5< 0.5 1256 137 0 (0.89092357 0.10907643)
     4) AcceptedCmp3< 0.5 1180 103 0 (0.91271186 0.08728814) *
     5) AcceptedCmp3>=0.5 76  34 0 (0.55263158 0.44736842)
      10) Education=Basic,Bachelor,Master 58  22 0 (0.62068966 0.37931034)
        20) NumDealsPurchases< 3.5 40  11 0 (0.72500000 0.27500000) *
        21) NumDealsPurchases>=3.5 18   7 1 (0.38888889 0.61111111) *
      11) Education=PhD 18   6 1 (0.33333333 0.66666667) *
   3) AcceptedCmp5>=0.5 85  38 1 (0.44705882 0.55294118)
     6) AcceptedCmp1< 0.5 54  24 0 (0.55555556 0.44444444)
      12) Year_Birth< 1954.5 12   2 0 (0.83333333 0.16666667) *
      13) Year_Birth>=1954.5 42  20 1 (0.47619048 0.52380952)
        26) MntWines< 684 15   4 0 (0.73333333 0.26666667) *
        27) MntWines>=684 27   9 1 (0.33333333 0.66666667) *
     7) AcceptedCmp1>=0.5 31   8 1 (0.25806452 0.74193548) *
```

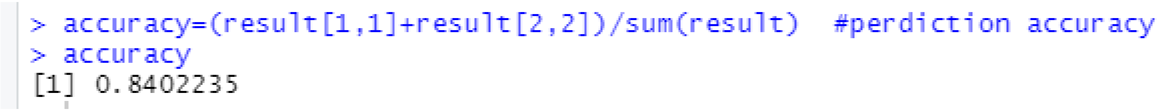
Figure 31 3rd Decision Tree Model (Overfit model)

The prediction model for the overfit decision tree also uses a cutoff value of 0.5 for the new validation data sets and it shows that the value 706 being true and from the prediction 39 were wrong predictions and 104 were falsely predicted by the overfit model. The accuracy prediction shows 84% of accuracy of the overfit decision tree model with new data sets for the validation observation.



```
> #prediction on validation
> probability=predict(overfit.model, newdata = validate_input)
> prediction=probability[,2]>cutoff
> actual = validate_input$Response
> result = table(actual, prediction)
> result
      prediction
actual FALSE TRUE
     0   706   39
     1   104   46
```

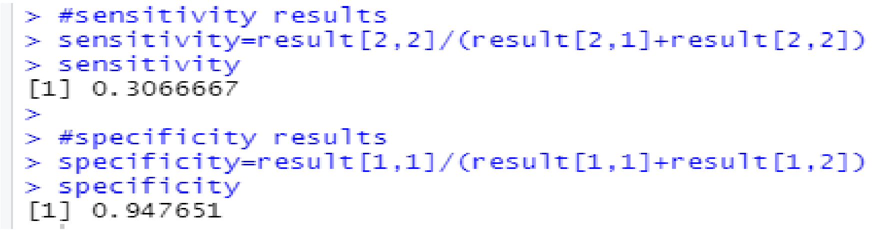
Figure 32 Overfit model's Confusion matrix



```
> accuracy=(result[1,1]+result[2,2])/sum(result)  #perdiction accuracy
> accuracy
[1] 0.8402235
```

Figure 33 Overfit model's prediction accuracy

In the sensitivity analysis the fraction of the true positives found were 30% and for the specificity 94% which is not good because 94% of the specificity predictions are classified as false positives.

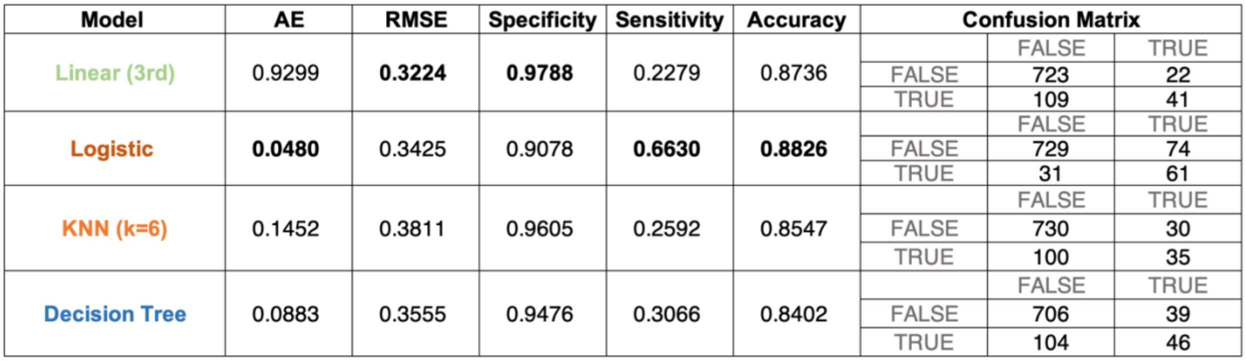


```
> #sensitivity results
> sensitivity=result[2,2]/(result[2,1]+result[2,2])
> sensitivity
[1] 0.3066667
>
> #specificity results
> specificity=result[1,1]/(result[1,1]+result[1,2])
> specificity
[1] 0.947651
```

Figure 34 Overfit model's Sensitivity and Specificity results

Model Comparison

Table 1: Model Comparison



Model	AE	RMSE	Specificity	Sensitivity	Accuracy	Confusion Matrix		
Linear (3rd)	0.9299	**0.3224**	**0.9788**	0.2279	0.8736		FALSE	TRUE
						FALSE	723	22
						TRUE	109	41
Logistic	**0.0480**	0.3425	0.9078	**0.6630**	**0.8826**		FALSE	TRUE
						FALSE	729	74
						TRUE	31	61
KNN (k=6)	0.1452	0.3811	0.9605	0.2592	0.8547		FALSE	TRUE
						FALSE	730	30
						TRUE	100	35
Decision Tree	0.0883	0.3555	0.9476	0.3066	0.8402		FALSE	TRUE
						FALSE	706	39
						TRUE	104	46

We ran four different models and provided comparisons through the Average Error, RMSE, Specificity, Sensitivity, Accuracy, and the Confusion Matrix. To make understanding easier we put our findings in one clean model as you can see above. With our linear regression, our average error came out to be 0.9299 which was one of the highest within our research. KNN is also really high being 0.1452, but not as nearly bad as our linear regression. Our other models such as the Decision Tree and Logistic regression have fairly good values that have an average error that is less than 1.5%. RMSE for all of the models are fairly similar, all of them being within the range of 0.30-0.40. In terms of specificity, all of our models also fit within the range of 0.90-0.98. Tests with high specificity are usually when the result is positive, which is something that can be observed within our confusion matrix. The sensitivity of our models were all relatively low for the linear, KNN, and Decision Tree, however, we notice that the Logistic Model has a noticeable high percentage. Accuracy was also similar for all, but our Logistic Model had the highest accuracy which insinuates better performance.

Recommendation

Regarding our question on which model is best, we concluded that it all depends on what someone is looking for. All of our models have different strengths, and can all be different based on what you want. If you are pursuing low AE, it would be best to go with the logistic regression model. Additionally, if you are pursuing high specificity, the best model to choose would be the linear regression model. As for the sensitivity and accuracy, the logistic regression model is also the best model for these specific choices. To better understand customer buying trends, one can analyze which model fits best for their question and scenario.

Conclusion

After all of our research and recommendations, we bring back all of our research to a full circle with the questions we prompted at the beginning of our project:

- What model is most appropriate to predict the customer's response to our product/service?
- What type of customer is favored for our product/service?
- How likely will a customer accept our product or service?

Within our research, our favorite model is the logistic regression model, given that it has the lowest average error, and the highest specificity, sensitivity, and accuracy. The advantages of using this method of regression are that it is simple and easy to implement and training efficiency, which requires low computation power. Some disadvantages are that it is vulnerable to overfitting because of potential sampling bias. Also, it is inapplicable in the instance that a continuous data set is being used. Therefore, we would recommend the logistic regression model as the most appropriate to predict the customer's response to our product.

When answering the second question, we need to analyze the tree model. Based on the third model of the decision tree, after setting the parameters minsplit=20, minbucket=10, and maxdepth=10 we can see that out of 1341 customers, 184 accepted the cmp5. The second ad campaign which was Cmp3 less than 0.5, out of 1180 customers, 103 said yes to the campaign and Cmp3 greater than or equal to 0.5 had 34 customers that said yes to the ad campaign out of 76 total customers. When answering the question of what type of customer is favored for our product/service, are those customers who have a college degree, with a marital status of married or together, and earn an income greater or equal to $38,000 per year.

Lastly, our third question asks how likely will a customer accept our product or services? This question can be answered by inputting the customer's demographics and ad-campaign response based on each mode that we have constructed. Most customers appear to accept Cmp3 and Cmp5 based on the percentage of the approval rate being between 87 and 89 percent.