Regression Tree and Linear Regression Model

Süleyman Buğra Gülsoy

06 01 2022

Abstract

The project is originally about analyzing the customer personality to have an understanding of a company's target and ideal customers. But for data set and project to be appropriate with regression analyzing I've done small changes. In original data there is no target variable the aim is Classifying the customer profile with data set, determining target customer and modifying company's product based on its target customers from different types of customer segments. Instead to be able to conduct regression analysis I choose the "Amount spent on wine in last 2 years" As my target variable with the aim of determining which customers are more inclined to spent more money To have a clear target customer profile. The models that i will be using are multiple linear regression and random forest The data set that I selected is "marketing_campaign" which is available in kaggle "https://www.kaggle.com/imakash3011/customer-personality-analysis?select=marketing_campaign.csv"

Introduction

Marketing has always been my interest point. Every business model includes some kind of marketing. Moreover, marketing can boost revenue of a product significantly. One might say Marketing is as important as product itself. That is why i wanted to work with a marketing problem. And the data is really good fit for my needs. Yet to be able conduct regression model with my current knowledge some simplifications has been made to original data to have a specific target variable. The objective of the project is predicting the money that a customer spent based on his/her profile Thus, the business can have a better understanding of which customer profile generates more income. So future plans such as advertising to specific customer profile can be made. The data is about the money that an individual spent on a wine. What makes this topic special from others is wine is a product that can have various of prices. Widely spread from very cheap to very expensive.

Methodology

Multiple linear regression and random forest modelling techniques will be used Since data has 29 variables and the target variable is a continuous numeric number Those 2 models are the suitable ones to make a prediction

Data Set

Data set has 2240 observation with 29 variables which are

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

Data is about customer profile from a specific wine business

The data is avaliable in kaggle “https://www.kaggle.com/imakash3011/customer-personality-analysis”

Model Fitting

```
library("rpart")

library("leaps")

library("caret")

## Zorunlu paket yükleniyor: ggplot2

## Zorunlu paket yükleniyor: lattice

my_data = read.csv("marketing_campaign.csv",sep = "\t")

nacount = c()

for(i in 1:ncol(my_data)){
  nacount = c(nacount,sum(is.na(my_data[,i])))
}


nacount

##  [1]  0  0  0  0 24  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0  0
0  0
## [26]  0  0  0  0

my_data = na.omit(my_data)
```

The data is cleared from NA values

```
sum(duplicated(my_data))

## [1] 0
```

Data does not have any duplicated values

```
str(my_data)

## 'data.frame':    2216 obs. of  29 variables:
##  $ ID                 : int  5524 2174 4141 6182 5324 7446 965 6177 4855
5899 ...
##  $ Year_Birth         : int  1957 1954 1965 1984 1981 1967 1971 1985 1974
1950 ...
##  $ Education          : chr  "Graduation" "Graduation" "Graduation"
"Graduation" ...
##  $ Marital_Status     : chr  "Single" "Single" "Together" "Together" ...
##  $ Income             : int  58138 46344 71613 26646 58293 62513 55635
33454 30351 5648 ...
##  $ Kidhome            : int  0 1 0 1 1 0 0 1 1 1 ...
##  $ Teenhome           : int  0 1 0 0 0 1 1 0 0 1 ...
##  $ Dt_Customer        : chr  "04-09-2012" "08-03-2014" "21-08-2013" "10-
02-2014" ...
##  $ Recency            : int  58 38 26 26 94 16 34 32 19 68 ...
##  $ MntWines           : int  635 11 426 11 173 520 235 76 14 28 ...
##  $ MntFruits          : int  88 1 49 4 43 42 65 10 0 0 ...
##  $ MntMeatProducts    : int  546 6 127 20 118 98 164 56 24 6 ...
##  $ MntFishProducts    : int  172 2 111 10 46 0 50 3 3 1 ...
##  $ MntSweetProducts   : int  88 1 21 3 27 42 49 1 3 1 ...
##  $ MntGoldProds       : int  88 6 42 5 15 14 27 23 2 13 ...
##  $ NumDealsPurchases  : int  3 2 1 2 5 2 4 2 1 1 ...
##  $ NumWebPurchases    : int  8 1 8 2 5 6 7 4 3 1 ...
##  $ NumCatalogPurchases: int  10 1 2 0 3 4 3 0 0 0 ...
##  $ NumStorePurchases  : int  4 2 10 4 6 10 7 4 2 0 ...
##  $ NumWebVisitsMonth  : int  7 5 4 6 5 6 6 8 9 20 ...
##  $ AcceptedCmp3       : int  0 0 0 0 0 0 0 0 0 1 ...
##  $ 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 0 0 0 0 0 0 0 0 0 ...
##  $ Complain           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Z_CostContact      : int  3 3 3 3 3 3 3 3 3 3 ...
##  $ Z_Revenue          : int  11 11 11 11 11 11 11 11 11 11 ...
##  $ Response           : int  1 0 0 0 0 0 0 0 1 0 ...
##  - attr(*, "na.action")= 'omit' Named int [1:24] 11 28 44 49 59 72 91 92
93 129 ...
##   ..- attr(*, "names")= chr [1:24] "11" "28" "44" "49" ...
```

Data has character values which needs to be factors

```
my_data$Education = as.factor(my_data$Education)

my_data$Marital_Status = as.factor(my_data$Marital_Status)
```

Irrelevent columns such as id and customer entry date has been removed from data

```
my_data$Dt_Customer = NULL

my_data$ID = NULL

my_data = na.omit(my_data)

str(my_data)

## 'data.frame':    2216 obs. of  27 variables:
##  $ Year_Birth         : int  1957 1954 1965 1984 1981 1967 1971 1985 1974
1950 ...
##  $ Education          : Factor w/ 5 levels "2n Cycle","Basic",..: 3 3 3 3
5 4 3 5 5 5 ...
##  $ Marital_Status     : Factor w/ 8 levels "Absurd","Alone",..: 5 5 6 6 4
6 3 4 6 6 ...
##  $ Income             : int  58138 46344 71613 26646 58293 62513 55635
33454 30351 5648 ...
##  $ Kidhome            : int  0 1 0 1 1 0 0 1 1 1 ...
##  $ Teenhome           : int  0 1 0 0 0 1 1 0 0 1 ...
##  $ Recency            : int  58 38 26 26 94 16 34 32 19 68 ...
##  $ MntWines           : int  635 11 426 11 173 520 235 76 14 28 ...
##  $ MntFruits          : int  88 1 49 4 43 42 65 10 0 0 ...
##  $ MntMeatProducts    : int  546 6 127 20 118 98 164 56 24 6 ...
##  $ MntFishProducts    : int  172 2 111 10 46 0 50 3 3 1 ...
##  $ MntSweetProducts   : int  88 1 21 3 27 42 49 1 3 1 ...
##  $ MntGoldProds       : int  88 6 42 5 15 14 27 23 2 13 ...
##  $ NumDealsPurchases  : int  3 2 1 2 5 2 4 2 1 1 ...
##  $ NumWebPurchases    : int  8 1 8 2 5 6 7 4 3 1 ...
##  $ NumCatalogPurchases: int  10 1 2 0 3 4 3 0 0 0 ...
##  $ NumStorePurchases  : int  4 2 10 4 6 10 7 4 2 0 ...
##  $ NumWebVisitsMonth  : int  7 5 4 6 5 6 6 8 9 20 ...
##  $ AcceptedCmp3       : int  0 0 0 0 0 0 0 0 0 1 ...
##  $ 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 0 0 0 0 0 0 0 0 0 ...
##  $ Complain           : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Z_CostContact      : int  3 3 3 3 3 3 3 3 3 3 ...
##  $ Z_Revenue          : int  11 11 11 11 11 11 11 11 11 11 ...
##  $ Response           : int  1 0 0 0 0 0 0 0 1 0 ...
##  - attr(*, "na.action")= 'omit' Named int [1:24] 11 28 44 49 59 72 91 92
93 129 ...
##   ..- attr(*, "names")= chr [1:24] "11" "28" "44" "49" ...
```

The data has been cleared and ready for modelling

```
set.seed (58270)

split_data = sample(1:nrow(my_data), 0.8*nrow(my_data))

train_set = (my_data)[split_data,]

test_set = (my_data)[-split_data,]


lm_1 = lm(MntWines~., data = train_set)

summary(lm_1)
```

```
## 
## Call:
## lm(formula = MntWines ~ ., data = train_set)
## 
## Residuals:
##     Min      1Q  Median      3Q     Max 
## -920.91  -81.97   -7.82   56.93  757.01 
## 
## Coefficients: (2 not defined because of singularities)
##                           Estimate Std. Error t value Pr(>|t|)    
## (Intercept)             1.479e+03  8.270e+02   1.788 0.073878 .  
## Year_Birth             -1.004e+00  4.122e-01  -2.434 0.015014 *  
## EducationBasic          2.559e+01  3.069e+01   0.834 0.404434    
## EducationGraduation     2.608e+01  1.636e+01   1.594 0.111067    
## EducationMaster         8.091e+01  1.886e+01   4.289 1.89e-05 ***
## EducationPhD            1.197e+02  1.838e+01   6.510 9.82e-11 ***
## Marital_StatusAlone     2.252e+02  1.683e+02   1.338 0.181220    
## Marital_StatusDivorced  2.289e+02  1.317e+02   1.738 0.082403 .  
## Marital_StatusMarried   2.061e+02  1.312e+02   1.571 0.116286    
## Marital_StatusSingle    2.148e+02  1.313e+02   1.636 0.101989    
## Marital_StatusTogether  2.120e+02  1.313e+02   1.614 0.106636    
## Marital_StatusWidow     1.764e+02  1.331e+02   1.326 0.185124    
## Marital_StatusYOLO      1.461e+02  1.844e+02   0.792 0.428361    
## Income                  1.146e-03  2.356e-04   4.867 1.24e-06 ***
## Kidhome                -3.737e+01  1.116e+01  -3.350 0.000825 ***
## Teenhome               -6.073e+00  1.021e+01  -0.595 0.552044
## Recency                 1.614e-01  1.539e-01   1.049 0.294491
## MntFruits              -7.759e-02  1.491e-01  -0.520 0.602788
## MntMeatProducts         1.363e-01  3.273e-02   4.165 3.26e-05 ***
## MntFishProducts        -1.120e-02  1.150e-01  -0.097 0.922388
## MntSweetProducts       -1.729e-01  1.467e-01  -1.179 0.238725
## MntGoldProds            2.385e-01  1.041e-01   2.291 0.022054 *
## NumDealsPurchases      -8.904e+00  2.912e+00  -3.057 0.002268 **
## NumWebPurchases         2.430e+01  2.221e+00  10.941  < 2e-16 ***
## NumCatalogPurchases     2.431e+01  2.474e+00   9.826  < 2e-16 ***
## NumStorePurchases       3.126e+01  1.971e+00  15.857  < 2e-16 ***
## NumWebVisitsMonth       1.897e+01  2.653e+00   7.149 1.29e-12 ***
## AcceptedCmp3            3.372e+01  1.750e+01   1.926 0.054214 .
## AcceptedCmp4            1.843e+02  1.924e+01   9.579  < 2e-16 ***
## AcceptedCmp5            2.654e+02  2.039e+01  13.015  < 2e-16 ***
## AcceptedCmp1            4.077e+01  2.035e+01   2.003 0.045317 *
## AcceptedCmp2            1.574e+02  4.149e+01   3.793 0.000154 ***
## Complain               -3.154e+01  4.120e+01  -0.766 0.443959
## Z_CostContact                  NA         NA      NA       NA
## Z_Revenue                      NA         NA      NA       NA
## Response                4.239e+00  1.452e+01   0.292 0.770433
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 182 on 1738 degrees of freedom
## Multiple R-squared:  0.7248, Adjusted R-squared:  0.7196
## F-statistic: 138.7 on 33 and 1738 DF,  p-value: < 2.2e-16
```

Adj R^2 is 0.7196 for multiple regression with all variables

```
predic_linear = predict(lm_1,test_set)

## Warning in predict.lm(lm_1, test_set): prediction from a rank-deficient
fit may
## be misleading

sqrt(mean((predic_linear-test_set$MntWines)^2))

## [1] 182.7267
```

The mean square error is 182.7267

Improving the model with using optimal variables

```
subset = regsubsets(MntWines~., data = train_set, method = "backward", nvmax
= 25)

## Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax,
force.in =
## force.in, : 2 linear dependencies found
## Reordering variables and trying again:
```

```
which.min(summary(subset)$bic)
```

```
## [1] 13
```

```
subset2 = regsubsets(MntWines~., data = train_set, method = "forward", nvmax
= 25)
```

```
## Warning in leaps.setup(x, y, wt = wt, nbest = nbest, nvmax = nvmax,
force.in =
## force.in, : 2 linear dependencies found
```

```
## Reordering variables and trying again:
```

```
which.min(summary(subset2)$bic)
```

```
## [1] 13
```

```
coef(subset,7)
```

```
##         (Intercept)        EducationPhD              Income
NumWebPurchases
##       -1.621001e+02        8.428681e+01        9.002245e-04
2.934479e+01
## NumCatalogPurchases   NumStorePurchases        AcceptedCmp4
AcceptedCmp5
##        2.872859e+01        2.933250e+01        2.149132e+02
3.024264e+02
```

```
lm_2 = lm(MntWines ~ Education + Income + NumWebPurchases +
NumCatalogPurchases + NumStorePurchases +
            AcceptedCmp4 + AcceptedCmp5, data = train_set)


summary(lm_2)
```

```
##
## Call:
## lm(formula = MntWines ~ Education + Income + NumWebPurchases +
##     NumCatalogPurchases + NumStorePurchases + AcceptedCmp4 +
##     AcceptedCmp5, data = train_set)
##
## Residuals:
##     Min      1Q  Median      3Q     Max
## -901.47  -83.63  -16.24   49.10  816.53
##
## Coefficients:
##                       Estimate Std. Error t value Pr(>|t|)
## (Intercept)         -1.997e+02  1.842e+01 -10.839  < 2e-16 ***
## EducationBasic       3.610e+01  3.138e+01   1.150 0.250195
## EducationGraduation  3.213e+01  1.677e+01   1.916 0.055496 .
## EducationMaster      8.168e+01  1.907e+01   4.283 1.95e-05 ***
## EducationPhD         1.237e+02  1.832e+01   6.757 1.91e-11 ***
## Income               8.634e-04  2.234e-04   3.865 0.000115 ***
## NumWebPurchases      2.946e+01  1.987e+00  14.828  < 2e-16 ***
## NumCatalogPurchases  2.894e+01  1.936e+00  14.946  < 2e-16 ***
## NumStorePurchases    2.925e+01  1.814e+00  16.130  < 2e-16 ***
## AcceptedCmp4         2.110e+02  1.838e+01  11.476  < 2e-16 ***
## AcceptedCmp5         3.013e+02  1.884e+01  15.996  < 2e-16 ***
## ---
## Signif. codes:  0 '***' 0.001 '**' 0.01 '*' 0.05 '.' 0.1 ' ' 1
##
## Residual standard error: 187.9 on 1761 degrees of freedom
## Multiple R-squared:  0.7029, Adjusted R-squared:  0.7013
## F-statistic: 416.7 on 10 and 1761 DF,  p-value: < 2.2e-16
```

Adj R^2 is 0.7013 with best subset of variables

```
predic_linear2 = predict(lm_2,test_set)

sqrt(mean((predic_linear2-test_set$MntWines)^2))

## [1] 186.2892
```

The mean square error is 186.2892

So the model hasn't developed from removing variables

Hence the best model with multiple linear regression is the model that has all the variables

Tree Based Model

Random forest technique

```
tree_model = train(MntWines~., data = train_set, method = "rf",
                   trControl = trainControl("cv", number = 10), ntree = 100)



summary(tree_model)

##                 Length Class      Mode
## call               5   -none-     call
## type               1   -none-     character
## predicted       1772   -none-     numeric
## mse              100   -none-     numeric
## rsq              100   -none-     numeric
## oob.times       1772   -none-     numeric
## importance        35   -none-     numeric
## importanceSD       0   -none-     NULL
## localImportance    0   -none-     NULL
## proximity          0   -none-     NULL
## ntree              1   -none-     numeric
## mtry               1   -none-     numeric
## forest            11   -none-     list
## coefs              0   -none-     NULL
## y               1772   -none-     numeric
## test               0   -none-     NULL
## inbag              0   -none-     NULL
## xNames            35   -none-     character
## problemType        1   -none-     character
## tuneValue          1   data.frame list
## obsLevels          1   -none-     logical
## param              1   -none-     list
```

```
predict_tree = predict(tree_model,test_set)

RMSE(predict_tree,test_set$MntWines)
```

```
## [1] 137.6541
```

The mean square error is 137.6541

Conclusion

The Tree based model which is random forest is better at predicting the amount of money that a customer spent compared to multiple linear regression model. With respect to error "137.6541" to "182.7267" Yet the mean square error is considerably high for each model. Hence, the models need further improvement To have a better predictions. The model can be further used in the wine business to predicting the amount of money that each stereo type of customers spent in the model parameters. To have a better understanding of the business and Targeting the specific stereotype of customers that will bring to most revenue.

References

"https://rpubs.com"

"https://www.r-bloggers.com"

"http://www.sthda.com"

"https://stackoverflow.com"

"https://github.com/bethatkinson/rpart"

"https://github.com/cran/leaps"

"https://github.com/topepo/caret"