SALES MODEL IN R

Vaibhav Khurana

2023-03-06

```
rm(list=ls())

# Loading the packages
library(Hmisc)
library(dplyr)
library(haven)
library(tidyr)
library(ggplot2)
library(fastDummies)
library(MASS)

# Load dataset
data.raw = read.csv("C:/Users/Vaibhav Khurana/Downloads/S10(1).csv")
View(data.raw)
str(data.raw)
```

```
## 'data.frame':    5392 obs. of  28 variables:
##  $ SeqNum: int  1 2 3 5 6 7 8 9 10 11 ...
##  $ MOSTYP: int  3 7 7 10 34 9 3 3 22 21 ...
##  $ MOSHOO: int  9 9 9 1 6 10 9 9 4 4 ...
##  $ MOPLHO: int  1 1 2 1 0 0 0 3 0 1 ...
##  $ MSKB1 : int  0 2 2 1 6 2 2 0 0 1 ...
##  $ MHHUUR: int  2 4 4 4 5 3 5 3 2 3 ...
##  $ MSKB2 : int  6 5 2 4 2 6 4 3 6 6 ...
##  $ MAUT2 : int  1 0 0 5 0 0 0 0 0 4 ...
##  $ MGEMOM: int  2 5 5 4 5 4 3 1 4 3 ...
##  $ MINKGE: int  0 0 0 5 0 0 0 1 0 0 ...
##  $ MAANTH: int  1 0 0 4 0 0 0 0 0 0 ...
##  $ MSKC  : int  6 5 4 0 4 5 5 8 4 0 ...
##  $ MGODPR: int  8 7 2 5 0 3 9 0 9 9 ...
##  $ MFWEKI: int  0 1 0 2 3 0 4 2 1 2 ...
##  $ MGODRK: int  1 2 2 1 3 1 2 3 2 1 ...
##  $ MSKA  : int  8 6 9 5 9 9 6 7 6 5 ...
##  $ MAUT1 : int  1 3 0 4 0 0 3 2 3 4 ...
##  $ MFALLE: int  0 2 4 0 5 4 2 7 2 0 ...
##  $ MRELGE: int  4 0 5 0 2 3 5 2 3 3 ...
##  $ MGEMLE: int  0 0 0 0 0 0 0 0 0 2 ...
##  $ MAUT0 : int  3 4 4 3 3 5 3 3 7 7 ...
##  $ PPERSA: int  6 0 6 0 6 6 0 5 0 6 ...
##  $ PWAPAR: int  0 0 0 0 0 0 3 0 0 0 ...
##  $ PAANHA: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ AMOTSC: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ APERSA: int  0 0 0 0 0 0 1 0 0 0 ...
##  $ AWAPAR: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Resp  : int  0 0 0 0 0 0 0 0 0 0 ...
```

```
#arranging the variables and forming anew data set
data <- data.raw %>% dplyr::select(SeqNum, MOSTYP, MOSHOO, MGEMOM, MAANTH, MGEMLE, everything())
View(data)
str(data)
```

```
## 'data.frame':    5392 obs. of  28 variables:
##  $ SeqNum: int  1 2 3 5 6 7 8 9 10 11 ...
##  $ MOSTYP: int  3 7 7 10 34 9 3 3 22 21 ...
##  $ MOSHOO: int  9 9 9 1 6 10 9 9 4 4 ...
##  $ MGEMOM: int  2 5 5 4 5 4 3 1 4 3 ...
##  $ MAANTH: int  1 0 0 4 0 0 0 0 0 0 ...
##  $ MGEMLE: int  0 0 0 0 0 0 0 0 0 2 ...
##  $ MOPLHO: int  1 1 2 1 0 0 0 3 0 1 ...
##  $ MSKB1 : int  0 2 2 1 6 2 2 0 0 1 ...
##  $ MHHUUR: int  2 4 4 4 5 3 5 3 2 3 ...
##  $ MSKB2 : int  6 5 2 4 2 6 4 3 6 6 ...
##  $ MAUT2 : int  1 0 0 5 0 0 0 0 0 4 ...
##  $ MINKGE: int  0 0 0 5 0 0 0 1 0 0 ...
##  $ MSKC  : int  6 5 4 0 4 5 5 8 4 0 ...
##  $ MGODPR: int  8 7 2 5 0 3 9 0 9 9 ...
##  $ MFWEKI: int  0 1 0 2 3 0 4 2 1 2 ...
##  $ MGODRK: int  1 2 2 1 3 1 2 3 2 1 ...
##  $ MSKA  : int  8 6 9 5 9 9 6 7 6 5 ...
##  $ MAUT1 : int  1 3 0 4 0 0 3 2 3 4 ...
##  $ MFALLE: int  0 2 4 0 5 4 2 7 2 0 ...
##  $ MRELGE: int  4 0 5 0 2 3 5 2 3 3 ...
##  $ MAUT0 : int  3 4 4 3 3 5 3 3 7 7 ...
##  $ PPERSA: int  6 0 6 0 6 6 0 5 0 6 ...
##  $ PWAPAR: int  0 0 0 0 0 0 3 0 0 0 ...
##  $ PAANHA: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ AMOTSC: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ APERSA: int  0 0 0 0 0 0 1 0 0 0 ...
##  $ AWAPAR: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Resp  : int  0 0 0 0 0 0 0 0 0 0 ...
```

```
#mutating data before prepping for logistic regression analysis
#FOR L1
L1 <- data %>% mutate(across(MOSHOO, ~ case_when(
  (.x == 0 ~ 25),
  (.x == 1 ~ 35),
  (.x == 2 ~ 45),
  (.x == 3 ~ 55),
  (.x == 5 ~ 65),
  (.x == 6 ~ 75),
  (TRUE ~ -99)
)))

View(L1)
str(L1)
```

```
## 'data.frame':    5392 obs. of  28 variables:
##  $ SeqNum: int  1 2 3 5 6 7 8 9 10 11 ...
##  $ MOSTYP: int  3 7 7 10 34 9 3 3 22 21 ...
##  $ MOSHOO: num  -99 -99 -99 35 75 -99 -99 -99 -99 -99 ...
##  $ MGEMOM: int  2 5 5 4 5 4 3 1 4 3 ...
##  $ MAANTH: int  1 0 0 4 0 0 0 0 0 0 ...
##  $ MGEMLE: int  0 0 0 0 0 0 0 0 0 2 ...
##  $ MOPLHO: int  1 1 2 1 0 0 0 3 0 1 ...
##  $ MSKB1 : int  0 2 2 1 6 2 2 0 0 1 ...
##  $ MHHUUR: int  2 4 4 4 5 3 5 3 2 3 ...
##  $ MSKB2 : int  6 5 2 4 2 6 4 3 6 6 ...
##  $ MAUT2 : int  1 0 0 5 0 0 0 0 0 4 ...
##  $ MINKGE: int  0 0 0 5 0 0 0 1 0 0 ...
##  $ MSKC  : int  6 5 4 0 4 5 5 8 4 0 ...
##  $ MGODPR: int  8 7 2 5 0 3 9 0 9 9 ...
##  $ MFWEKI: int  0 1 0 2 3 0 4 2 1 2 ...
##  $ MGODRK: int  1 2 2 1 3 1 2 3 2 1 ...
##  $ MSKA  : int  8 6 9 5 9 9 6 7 6 5 ...
##  $ MAUT1 : int  1 3 0 4 0 0 3 2 3 4 ...
##  $ MFALLE: int  0 2 4 0 5 4 2 7 2 0 ...
##  $ MRELGE: int  4 0 5 0 2 3 5 2 3 3 ...
##  $ MAUT0 : int  3 4 4 3 3 5 3 3 7 7 ...
##  $ PPERSA: int  6 0 6 0 6 6 0 5 0 6 ...
##  $ PWAPAR: int  0 0 0 0 0 0 3 0 0 0 ...
##  $ PAANHA: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ AMOTSC: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ APERSA: int  0 0 0 0 0 0 1 0 0 0 ...
##  $ AWAPAR: int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Resp  : int  0 0 0 0 0 0 0 0 0 0 ...
```

```
#for L0&2
L2 <- dummy_cols(L1, select_columns = c('MOSTYP', 'MOSHOO'),remove_selected_columns = TRUE)
View(L2)
str(L2)
```

```
## 'data.frame':    5392 obs. of  72 variables:
##  $ SeqNum    : int  1 2 3 5 6 7 8 9 10 11 ...
##  $ MGEMOM    : int  2 5 5 4 5 4 3 1 4 3 ...
##  $ MAANTH    : int  1 0 0 4 0 0 0 0 0 0 ...
##  $ MGEMLE    : int  0 0 0 0 0 0 0 0 0 2 ...
##  $ MOPLHO    : int  1 1 2 1 0 0 0 3 0 1 ...
##  $ MSKB1     : int  0 2 2 1 6 2 2 0 0 1 ...
##  $ MHHUUR    : int  2 4 4 4 5 3 5 3 2 3 ...
##  $ MSKB2     : int  6 5 2 4 2 6 4 3 6 6 ...
##  $ MAUT2     : int  1 0 0 5 0 0 0 0 0 4 ...
##  $ MINKGE    : int  0 0 0 5 0 0 0 1 0 0 ...
##  $ MSKC      : int  6 5 4 0 4 5 5 8 4 0 ...
##  $ MGODPR    : int  8 7 2 5 0 3 9 0 9 9 ...
##  $ MFWEKI    : int  0 1 0 2 3 0 4 2 1 2 ...
##  $ MGODRK    : int  1 2 2 1 3 1 2 3 2 1 ...
##  $ MSKA      : int  8 6 9 5 9 9 6 7 6 5 ...
##  $ MAUT1     : int  1 3 0 4 0 0 3 2 3 4 ...
##  $ MFALLE    : int  0 2 4 0 5 4 2 7 2 0 ...
##  $ MRELGE    : int  4 0 5 0 2 3 5 2 3 3 ...
##  $ MAUT0     : int  3 4 4 3 3 5 3 3 7 7 ...
##  $ PPERSA    : int  6 0 6 0 6 6 0 5 0 6 ...
##  $ PWAPAR    : int  0 0 0 0 0 0 3 0 0 0 ...
##  $ PAANHA    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ AMOTSC    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ APERSA    : int  0 0 0 0 0 0 1 0 0 0 ...
##  $ AWAPAR    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Resp      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_1  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_2  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_3  : int  1 0 0 0 0 0 1 1 0 0 ...
##  $ MOSTYP_4  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_5  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_6  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_7  : int  0 1 1 0 0 0 0 0 0 0 ...
##  $ MOSTYP_8  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_9  : int  0 0 0 0 0 1 0 0 0 0 ...
##  $ MOSTYP_10 : int  0 0 0 1 0 0 0 0 0 0 ...
##  $ MOSTYP_11 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_12 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_13 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_14 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_15 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_16 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_17 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_18 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_19 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_20 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_21 : int  0 0 0 0 0 0 0 0 0 1 ...
##  $ MOSTYP_22 : int  0 0 0 0 0 0 0 0 1 0 ...
##  $ MOSTYP_23 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_24 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_26 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_27 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_28 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_29 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_30 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_31 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_32 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_33 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_34 : int  0 0 0 0 1 0 0 0 0 0 ...
##  $ MOSTYP_35 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_36 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_37 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_38 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_39 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_40 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_41 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSHOO_-99: int  1 1 1 0 0 1 1 1 1 1 ...
##  $ MOSHOO_35 : int  0 0 0 1 0 0 0 0 0 0 ...
##  $ MOSHOO_45 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSHOO_55 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSHOO_65 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSHOO_75 : int  0 0 0 0 1 0 0 0 0 0 ...
```

```
#mutating data on the basis of L3
#Alternate code
L3 <- L2 %>% mutate(across(MOPLHO:MAUT0, ~ case_when(
  (.x == 0 ~ 0),
  (.x == 1 ~ 5.5),
  (.x == 2 ~ 17),
  (.x == 3 ~ 30),
  (.x == 4 ~ 43),
  (.x == 5 ~ 56),
  (.x == 6 ~ 69),
  (.x == 7 ~ 82),
  (.x == 8 ~ 94),
  (.x == 9 ~ 100),
  (TRUE ~ -99)
)))

View(L3)
str(L3)
```

```
## 'data.frame':    5392 obs. of  72 variables:
##  $ SeqNum    : int  1 2 3 5 6 7 8 9 10 11 ...
##  $ MGEMOM    : int  2 5 5 4 5 4 3 1 4 3 ...
##  $ MAANTH    : int  1 0 0 4 0 0 0 0 0 0 ...
##  $ MGEMLE    : int  0 0 0 0 0 0 0 0 0 2 ...
##  $ MOPLHO    : num  5.5 5.5 17 5.5 0 0 0 30 0 5.5 ...
##  $ MSKB1     : num  0 17 17 5.5 69 17 17 0 0 5.5 ...
##  $ MHHUUR    : num  17 43 43 43 56 30 56 30 17 30 ...
##  $ MSKB2     : num  69 56 17 43 17 69 43 30 69 69 ...
##  $ MAUT2     : num  5.5 0 0 56 0 0 0 0 0 43 ...
##  $ MINKGE    : num  0 0 0 56 0 0 0 5.5 0 0 ...
##  $ MSKC      : num  69 56 43 0 43 56 56 94 43 0 ...
##  $ MGODPR    : num  94 82 17 56 0 30 100 0 100 100 ...
##  $ MFWEKI    : num  0 5.5 0 17 30 0 43 17 5.5 17 ...
##  $ MGODRK    : num  5.5 17 17 5.5 30 5.5 17 30 17 5.5 ...
##  $ MSKA      : num  94 69 100 56 100 100 69 82 69 56 ...
##  $ MAUT1     : num  5.5 30 0 43 0 0 30 17 30 43 ...
##  $ MFALLE    : num  0 17 43 0 56 43 17 82 17 0 ...
##  $ MRELGE    : num  43 0 56 0 17 30 56 17 30 30 ...
##  $ MAUT0     : num  30 43 43 30 30 56 30 30 82 82 ...
##  $ PPERSA    : int  6 0 6 0 6 6 0 5 0 6 ...
##  $ PWAPAR    : int  0 0 0 0 0 0 3 0 0 0 ...
##  $ PAANHA    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ AMOTSC    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ APERSA    : int  0 0 0 0 0 0 1 0 0 0 ...
##  $ AWAPAR    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Resp      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_1  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_2  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_3  : int  1 0 0 0 0 0 1 1 0 0 ...
##  $ MOSTYP_4  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_5  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_6  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_7  : int  0 1 1 0 0 0 0 0 0 0 ...
##  $ MOSTYP_8  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_9  : int  0 0 0 0 0 1 0 0 0 0 ...
##  $ MOSTYP_10 : int  0 0 0 1 0 0 0 0 0 0 ...
##  $ MOSTYP_11 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_12 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_13 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_14 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_15 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_16 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_17 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_18 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_19 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_20 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_21 : int  0 0 0 0 0 0 0 0 0 1 ...
##  $ MOSTYP_22 : int  0 0 0 0 0 0 0 0 1 0 ...
##  $ MOSTYP_23 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_24 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_26 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_27 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_28 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_29 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_30 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_31 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_32 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_33 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_34 : int  0 0 0 0 1 0 0 0 0 0 ...
##  $ MOSTYP_35 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_36 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_37 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_38 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_39 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_40 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_41 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSHOO_-99: int  1 1 1 0 0 1 1 1 1 1 ...
##  $ MOSHOO_35 : int  0 0 0 1 0 0 0 0 0 0 ...
##  $ MOSHOO_45 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSHOO_55 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSHOO_65 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSHOO_75 : int  0 0 0 0 1 0 0 0 0 0 ...
```

```
#For L4, naming it as before split

beforesplit <- L3 %>% mutate(across(starts_with("P"), ~ case_when(
  (.x == 0 ~ 0),
  (.x == 1 ~ 25),
  (.x == 2 ~ 75),
  (.x == 3 ~ 150),
  (.x == 5 ~ 350),
  (.x == 5 ~ 750),
  (.x == 6 ~ 3000),
  (.x == 7 ~ 7500),
  (.x == 8 ~ 15000),
  (.x == 9 ~ 30000),
  (TRUE ~ -99)
)))

View(beforesplit)
str(beforesplit)
```

```
## 'data.frame':    5392 obs. of  72 variables:
##  $ SeqNum    : int  1 2 3 5 6 7 8 9 10 11 ...
##  $ MGEMOM    : int  2 5 5 4 5 4 3 1 4 3 ...
##  $ MAANTH    : int  1 0 0 4 0 0 0 0 0 0 ...
##  $ MGEMLE    : int  0 0 0 0 0 0 0 0 0 2 ...
##  $ MOPLHO    : num  5.5 5.5 17 5.5 0 0 0 30 0 5.5 ...
##  $ MSKB1     : num  0 17 17 5.5 69 17 17 0 0 5.5 ...
##  $ MHHUUR    : num  17 43 43 43 56 30 56 30 17 30 ...
##  $ MSKB2     : num  69 56 17 43 17 69 43 30 69 69 ...
##  $ MAUT2     : num  5.5 0 0 56 0 0 0 0 0 43 ...
##  $ MINKGE    : num  0 0 0 56 0 0 0 5.5 0 0 ...
##  $ MSKC      : num  69 56 43 0 43 56 56 94 43 0 ...
##  $ MGODPR    : num  94 82 17 56 0 30 100 0 100 100 ...
##  $ MFWEKI    : num  0 5.5 0 17 30 0 43 17 5.5 17 ...
##  $ MGODRK    : num  5.5 17 17 5.5 30 5.5 17 30 17 5.5 ...
##  $ MSKA      : num  94 69 100 56 100 100 69 82 69 56 ...
##  $ MAUT1     : num  5.5 30 0 43 0 0 30 17 30 43 ...
##  $ MFALLE    : num  0 17 43 0 56 43 17 82 17 0 ...
##  $ MRELGE    : num  43 0 56 0 17 30 56 17 30 30 ...
##  $ MAUT0     : num  30 43 43 30 30 56 30 30 82 82 ...
##  $ PPERSA    : num  3000 0 3000 0 3000 3000 0 350 0 3000 ...
##  $ PWAPAR    : num  0 0 0 0 0 0 150 0 0 0 ...
##  $ PAANHA    : num  0 0 0 0 0 0 0 0 0 0 ...
##  $ AMOTSC    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ APERSA    : int  0 0 0 0 0 0 1 0 0 0 ...
##  $ AWAPAR    : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ Resp      : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_1  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_2  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_3  : int  1 0 0 0 0 0 1 1 0 0 ...
##  $ MOSTYP_4  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_5  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_6  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_7  : int  0 1 1 0 0 0 0 0 0 0 ...
##  $ MOSTYP_8  : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_9  : int  0 0 0 0 0 1 0 0 0 0 ...
##  $ MOSTYP_10 : int  0 0 0 1 0 0 0 0 0 0 ...
##  $ MOSTYP_11 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_12 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_13 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_14 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_15 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_16 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_17 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_18 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_19 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_20 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_21 : int  0 0 0 0 0 0 0 0 0 1 ...
##  $ MOSTYP_22 : int  0 0 0 0 0 0 0 0 1 0 ...
##  $ MOSTYP_23 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_24 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_26 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_27 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_28 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_29 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_30 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_31 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_32 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_33 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_34 : int  0 0 0 0 1 0 0 0 0 0 ...
##  $ MOSTYP_35 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_36 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_37 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_38 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_39 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_40 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSTYP_41 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSHOO_-99: int  1 1 1 0 0 1 1 1 1 1 ...
##  $ MOSHOO_35 : int  0 0 0 1 0 0 0 0 0 0 ...
##  $ MOSHOO_45 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSHOO_55 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSHOO_65 : int  0 0 0 0 0 0 0 0 0 0 ...
##  $ MOSHOO_75 : int  0 0 0 0 1 0 0 0 0 0 ...
```

```
#Splitting into training and testing data
set.seed(1)
train <- beforesplit %>% dplyr::sample_frac(0.70)
test  <- dplyr::anti_join(beforesplit, train, by = "SeqNum" )

#remove seq num and train logistic model

train<-subset(train,select=-c(SeqNum))
test <-subset(test,select=-c(SeqNum))

fullModel = glm(Resp ~ ., family = 'binomial', data = train) # model with all variables
nullModel = glm(Resp ~ 1, family = 'binomial', data = train) # model with intercept only

interim<-summary(stepAIC(nullModel, # start with a model containing no variables
                         direction = 'forward', # run forward selection
                         scope = list(upper = fullModel, # the maximum to consider is a model wi
th all variables
                                      lower = nullModel), # the minimum to consider is a model w
ith no variables
                         trace = 0)) # do not show the step-by-step process of model selection

coef<-data.frame(interim[['coefficients']])
final<-coef[coef$Pr...z..<0.05,]
print(final)
```

```
##                   Estimate   Std..Error    z.value     Pr...z..
## (Intercept) -4.8584510919 0.4613254738 -10.531504 6.183843e-26
## PPERSA       0.0003337191 0.0000420016   7.945390 1.935807e-15
## MAANTH      -0.3081431015 0.1184850988  -2.600691 9.303628e-03
## MGODPR       0.0061649133 0.0023784334   2.592006 9.541814e-03
## MGEMOM       0.2248161206 0.0521018752   4.314933 1.596510e-05
## MOSTYP_21   -1.0661028647 0.4476368274  -2.381625 1.723646e-02
## MGEMLE      -0.3339378977 0.1458562723  -2.289500 2.205033e-02
## MOSTYP_34   -1.3274436010 0.6439455008  -2.061422 3.926277e-02
## MAUT2        0.0147114272 0.0053270395   2.761652 5.750981e-03
## MSKC         0.0084881182 0.0043250662   1.962541 4.969958e-02
```

```
#Retyping list of variables into final model build
varnames<-rownames(final)
varnames<-varnames[2:length(varnames)]
finalmodel<-glm(Resp ~ PPERSA+MSKC+MAANTH+MGODPR+MGEMOM+MGEMLE+MOSTYP_21+MOSTYP_34+MAUT2, family
= 'binomial', data = train)


#Evaluating performance on test data
test$pred<-predict(finalmodel,newdata=test,type="response")
test<-test[order(-test$pred),]
test$one<-1
test$cumprospects<-cumsum(test$one)
test$cumresp    <-cumsum(test$Resp)

Perf<-subset(test,select=c(pred,cumprospects,cumresp))
Perf$PctProspect<-Perf$cumprospects/nrow(Perf)
Perf$PctResp    <-Perf$cumresp/max(Perf$cumresp)

cutpoint<-subset(Perf,PctProspect>0.745 & PctProspect<0.755)
cutpoint
```

```
##            pred cumprospects cumresp PctProspect   PctResp
## 808  0.02219505         1206      80   0.7453646 0.9195402
## 867  0.02219505         1207      80   0.7459827 0.9195402
## 1118 0.02219505         1208      80   0.7466007 0.9195402
## 1543 0.02219505         1209      80   0.7472188 0.9195402
## 842  0.02218221         1210      80   0.7478368 0.9195402
## 755  0.02214679         1211      80   0.7484549 0.9195402
## 171  0.02174806         1212      80   0.7490729 0.9195402
## 118  0.02167437         1213      80   0.7496910 0.9195402
## 123  0.02167437         1214      80   0.7503090 0.9195402
## 265  0.02167437         1215      80   0.7509271 0.9195402
## 419  0.02156289         1216      80   0.7515451 0.9195402
## 989  0.02156289         1217      80   0.7521632 0.9195402
## 954  0.02153066         1218      80   0.7527812 0.9195402
## 1592 0.02153066         1219      80   0.7533993 0.9195402
## 505  0.02149320         1220      80   0.7540173 0.9195402
## 580  0.02144553         1221      80   0.7546354 0.9195402
```

```
View(cutpoint)
```