Marketing Analysis Final Project

Jinman Rong

12/16/2019

adds new page after title

Contents

adds table of contents

List of Figures

List of Tables

```r
# Load libraries
# install.packages("corrplot")
library(corrplot)                # correlation plot
library(broom)                   # tidy the linear regression results
library(tidyverse)               # data manipulation
library(data.table)              # table manipulation
# install.packages("outliers")
library(outliers)                # check z scores
library(class)                   # kNN
# install.packages("fastDummies")
library(fastDummies)             # transit categorical variables into dummy variables
```

Basic Explanatory Analysis

1. Load the data contained in the file data_telebank.csv and name the variable: dta_bank

```r
set.seed(123456)                                    # Random Number Generation
dta_bank <- read.csv("~/Downloads/Bank Case.csv")    # Load data
summary(dta_bank, maxsum = 50)                      # Check data structure
```

```
##       age                    job            marital
##  Min.   :17.00   admin.       :10422   divorced: 4612
##  1st Qu.:32.00   blue-collar  : 9254   married :24928
##  Median :38.00   entrepreneur : 1456   single  :11568
##  Mean   :40.02   housemaid    : 1060   unknown :   80
##  3rd Qu.:47.00   management   : 2924
##  Max.   :98.00   retired      : 1720
##                  self-employed: 1421
##                  services     : 3969
##                  student      :  875
##                  technician   : 6743
##                  unemployed   : 1014
##                  unknown      :  330
##                education        default         housing
##  basic.4y           : 4176   no     :32588   no     :18622
##  basic.6y           : 2292   unknown: 8597   unknown:  990
##  basic.9y           : 6045   yes    :    3   yes    :21576
##  high.school        : 9515
##  illiterate         :   18
##  professional.course: 5243
##  university.degree  :12168
##  unknown            : 1731
##
##
##
##
##       loan            contact      month       day_of_week
##  no     :33950   cellular :26144   apr: 2632   fri:7827
##  unknown:  990   telephone:15044   aug: 6178   mon:8514
##  yes    : 6248                     dec:  182   thu:8623
##                                    jul: 7174   tue:8090
##                                    jun: 5318   wed:8134
##                                    mar:  546
##                                    may:13769
```

```
##                                     nov: 4101
##                                     oct:  718
##                                     sep:  570
##
##
##     duration        y
##  Min.   :   0.0   no :36548
##  1st Qu.: 102.0   yes: 4640
##  Median : 180.0
##  Mean   : 258.3
##  3rd Qu.: 319.0
##  Max.   :4918.0
##
##
##
##
##
##
```

2. In one sentence, describe variables in each column paying special attention to

a. Type of variable (categorical/numerical) and what are the units (for the numerical only)

b. For the ones that are numerical study whether they have outliers. There is no definition for what an outlier so we can define an outlier as any observation with a value that is more than 4 times its standard deviation.

ANSWER:

age: [numerical] (year)

job: type of job [categorical]

marital: marital status [categorical]

education: [categorical]

default: has credit in default? [categorical]

housing: has housing loan? [categorical]

loan: has personal loan? [categorical]

contact: contact communication type [categorical]

month: last contact month of year [categorical]

day_of_week: last contact day of the week [categorical]

duration: last contact duration, in seconds [numerical] (seconds)

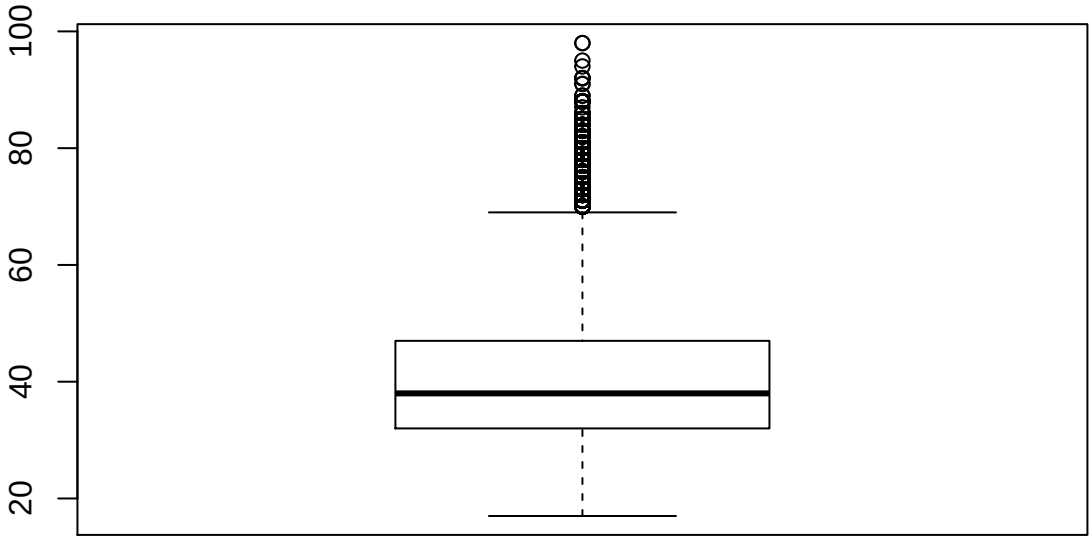
y: has the client subscribed a term deposit? [categorical:binary]

There are outliers in both age and duration column under the assumption that an outlier is any observation with a value that is more than 4 times its standard deviation.

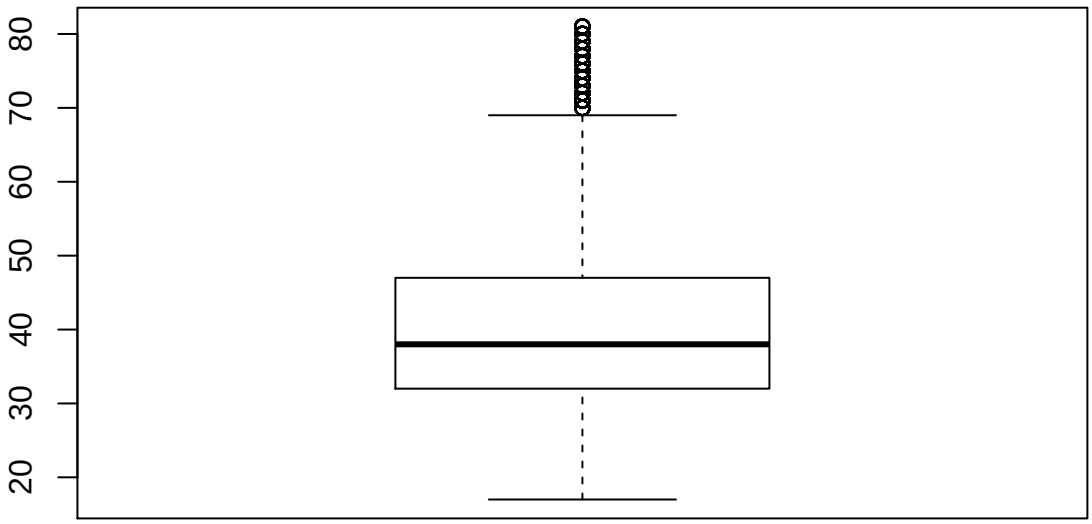
Remove outliers in variables age and duration

```
# Get the z-scores for each value in age and duration
age_outlier_scores      <- scores(dta_bank$age)
duration_outlier_scores <- scores(dta_bank$duration)
```

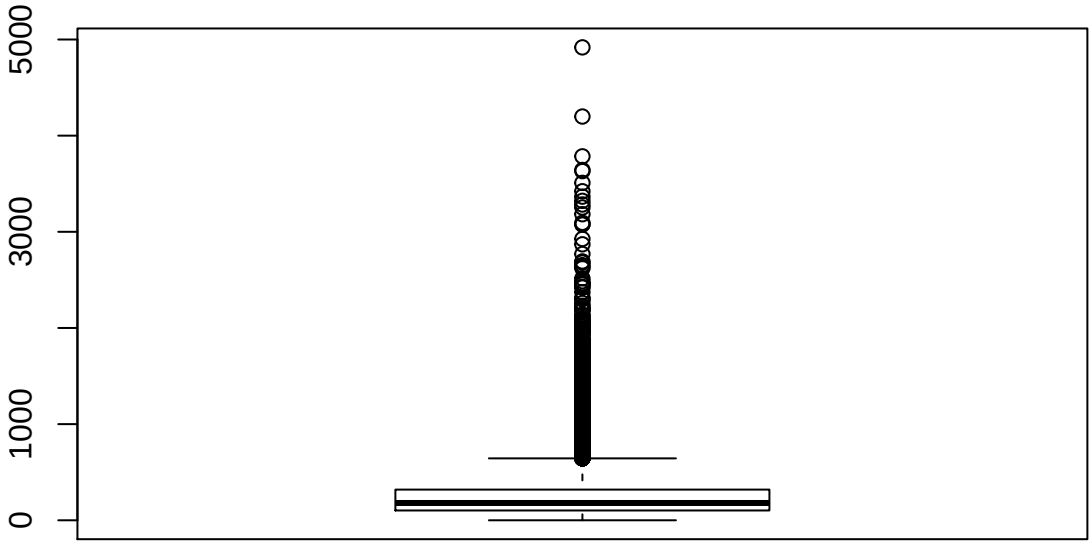
```r
# Create a logical vector the same length as outlier_scores that is "TRUE" if outlier_scores is greater
age_is_outlier          <- scores(dta_bank$age)    > 4 | scores(dta_bank$age)    < -4
duration_is_outlier     <- duration_outlier_scores > 4 | duration_outlier_scores < -4

# Add a column with info whether the age is an outlier
dta_bank$age_is_outlier      <- age_is_outlier
dta_bank$duration_is_outlier <- duration_is_outlier

# Only get rows where the age_is_outlier column we made is equal to "FALSE"
dta_bank_outliers_rm <- dta_bank[dta_bank$age_is_outlier == FALSE &
                                 dta_bank$duration_is_outlier == FALSE, ]

# Draw boxplots to check the final results
boxplot(dta_bank$age)
```



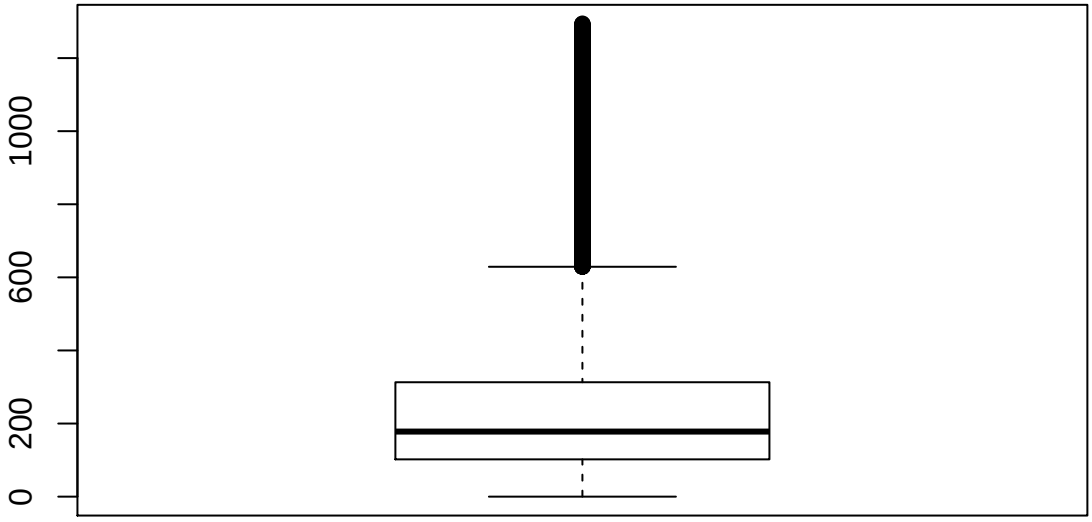

```r
boxplot(dta_bank_outliers_rm$age)
```



```r
boxplot(dta_bank$duration)
```

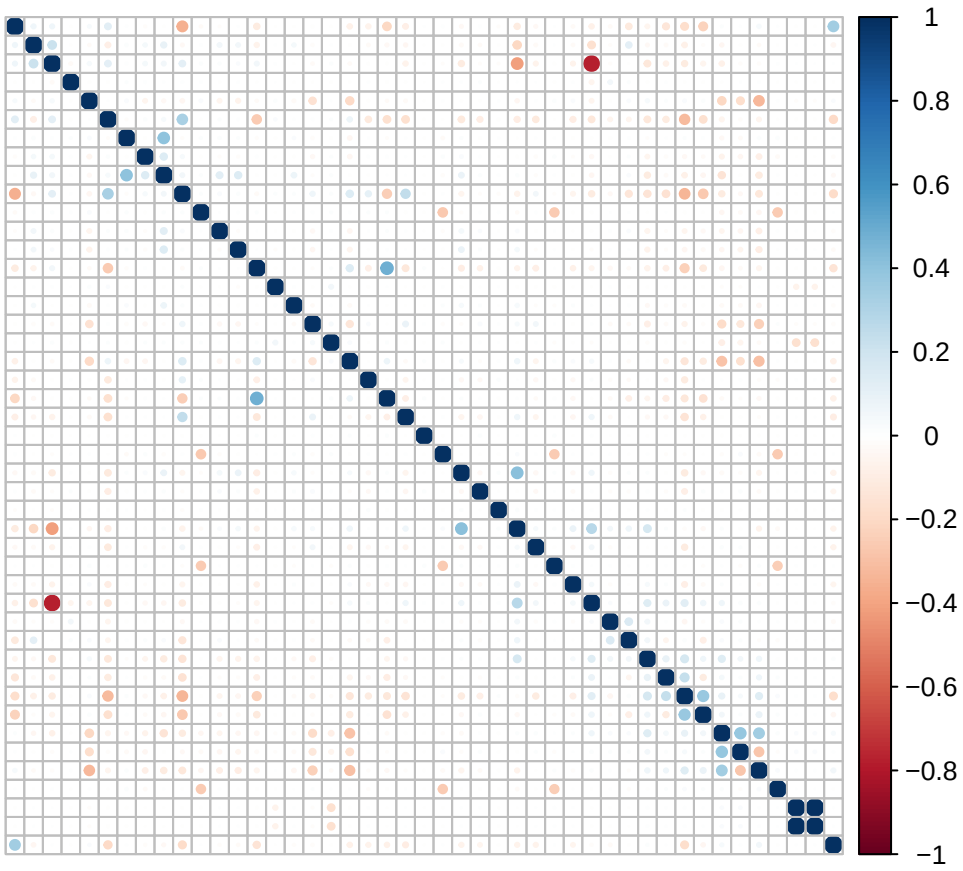



```r
boxplot(dta_bank_outliers_rm$duration)
```



3. Create a corr-plot using the package corrplot. You will have to install it using the command: install.packages()

```r
# Transit dependent variable into binary variable
dta_bank_outliers_rm$y =  ifelse(dta_bank_outliers_rm$y=='yes',1,0)
dta_bank_cleaned       <- dta_bank_outliers_rm %>% select(1:12)
# Draw the corr-plot
M =  cor(model.matrix(~.-1,data=dta_bank_cleaned))
corrplot(M, order = "AOE", tl.pos = "n")
```



4. Run the following command: lm(y~.,data=dta_bank)

a. Write the structural equation that R is estimating?

ANSWER:

y = intercept + a1*age* + *b1*jobblue-collar + ... + b11*jobunknown* + *c1*maritalmarried +... + c3*maritalunknown* + *d1*educationbasic.6y + ... + d7*educationunknown* + *e1*defaultunknown + e2*defaultyes* + *f1*housingunknown + f2*housingyes* +*g1*loanyes + h1*contacttelephone* + *i1*monthaug + ... + i9*monthloan* + *j1*day_of_weekmon + ... + j4*day_of_weekwed* + *k1*duration

b. Comment the results.

i. Best time to perform telemarketing tasks?

ANSWER:

Month: March Day of the week: Wednesday

ii. Best income groups?

ANSWER:

Student

iii. Potential concerns of omitted variable Bias

ANSWER:

client related variable: client's honest history

bank related variable: number of contacts performed during this campaign and for this client, outcome of the previous marketing campaign

social and economic variables: consumer price index

```
reg1                    =  lm(formula = y ~ ., data = dta_bank_cleaned)
result                  <- reg1 %>% tidy() %>% print(n = 1e3)
```

```
## # A tibble: 43 x 5
##    term                           estimate  std.error statistic   p.value
##    <chr>                             <dbl>      <dbl>     <dbl>     <dbl>
##  1 (Intercept)                    0.00835  0.0122         0.684 4.94e-  1
##  2 age                            0.000710 0.000165       4.29  1.77e-  5
##  3 jobblue-collar                -0.0269   0.00491       -5.48  4.23e-  8
##  4 jobentrepreneur               -0.0271   0.00761       -3.56  3.75e-  4
##  5 jobhousemaid                  -0.00781  0.00908       -0.860 3.90e-  1
##  6 jobmanagement                 -0.0199   0.00572       -3.47  5.17e-  4
##  7 jobretired                     0.0624   0.00808        7.73  1.10e- 14
##  8 jobself-employed              -0.0244   0.00764       -3.20  1.38e-  3
##  9 jobservices                   -0.0220   0.00533       -4.14  3.55e-  5
## 10 jobstudent                     0.118    0.00989       11.9   1.66e- 32
## 11 jobtechnician                 -0.0153   0.00472       -3.24  1.21e-  3
## 12 jobunemployed                  0.0193   0.00891        2.16  3.05e-  2
## 13 jobunknown                     0.0102   0.0153         0.664 5.06e-  1
## 14 maritalmarried                 0.00779  0.00435        1.79  7.36e-  2
## 15 maritalsingle                  0.0225   0.00500        4.51  6.63e-  6
## 16 maritalunknown                -0.00881  0.0303        -0.291 7.71e-  1
## 17 educationbasic.6y              0.00600  0.00709        0.846 3.98e-  1
## 18 educationbasic.9y             -0.00315  0.00561       -0.561 5.75e-  1
## 19 educationhigh.school           0.00150  0.00581        0.259 7.96e-  1
## 20 educationilliterate            0.116    0.0629         1.85  6.48e-  2
## 21 educationprofessional.course   0.00822  0.00653        1.26  2.09e-  1
## 22 educationuniversity.degree     0.0176   0.00593        2.97  2.93e-  3
## 23 educationunknown               0.0213   0.00798        2.67  7.68e-  3
## 24 defaultunknown                -0.0396   0.00345      -11.5   2.41e- 30
## 25 defaultyes                    -0.0422   0.154         -0.274 7.84e-  1
## 26 housingunknown                 0.00575  0.00875        0.657 5.11e-  1
## 27 housingyes                     0.00240  0.00269        0.890 3.74e-  1
## 28 loanyes                       -0.00134  0.00369       -0.363 7.16e-  1
## 29 contacttelephone              -0.0778   0.00349      -22.3   1.73e-109
## 30 monthaug                      -0.0680   0.00636      -10.7   1.16e- 26
## 31 monthdec                       0.237    0.0210        11.3   1.58e- 29
## 32 monthjul                      -0.0868   0.00617      -14.1   9.60e- 45
## 33 monthjun                       0.00135  0.00696        0.194 8.46e-  1
## 34 monthmar                       0.312    0.0128        24.4   1.12e-130
## 35 monthmay                      -0.0666   0.00604      -11.0   3.15e- 28
## 36 monthnov                      -0.0711   0.00675      -10.5   6.45e- 26
## 37 monthoct                       0.239    0.0115        20.9   2.80e- 96
## 38 monthsep                       0.228    0.0126        18.1   1.22e- 72
## 39 day_of_weekmon                -0.00961  0.00420       -2.29  2.21e-  2
## 40 day_of_weekthu                 0.00208  0.00419        0.495 6.21e-  1
## 41 day_of_weektue                 0.00369  0.00426        0.866 3.86e-  1
## 42 day_of_weekwed                 0.00546  0.00425        1.28  1.99e-  1
## 43 duration                       0.000564 0.00000618    91.2   0.
```

Predictive Modeling and Tuning

1. Explain (in sentences) why and how we always do that.

ANSWER:

Steps of predictive modeling:

a. Judge whether the variable is useful as predictor
b. Plot correlation plot to think about suitable models to do predictions
c. Set a seed for randomization.
d. Split the data into training, validating and testing sets for the model. The training data is used to train the model and the testing set is used to test it and determine its accuracy.
e. Train the model and test. A good way to split it would be to set aside eighty percent of the data set for training and the remaining for validating and testing.
f. Get out the results from the confusion matrix and seek to improve the performance keys.

Reasons: Firstly, we do the first judgment to save time in the following coding process. Predictive modeling is not like casuality analysis and we should judge before getting down to business. Secondly, we split the data into training, validating and testing data to train the data first, and then test the model. The split process is randomized to ensure a good training.

2. From the point of view of the firm and given that we are running a predictive exercise, is there any variable that should not be included as X? If yes, please drop it.

ANSWER:

From my perspective, I think job, marital, education, default, housing, loan, duration variable should not be included as Xs when doing a predictive analysis. This is because we cannot predict the next customer's job, marital, education, default, housing or loan info. Also, we cannot predict how long will the contact lasts (duration). Therefore, using these variables are not practical in predictive analysis.

3. Explain the problems of overfitting and underfitting.

ANSWER:

Overfitting occurs when a statistical model or machine learning algorithm captures the noise of the data. Intuitively, overfitting occurs when the model or the algorithm fits the data too well. Specifically, overfitting occurs if the model or algorithm shows low bias but high variance.

Underfitting occurs when a statistical model or machine learning algorithm cannot capture the underlying trend of the data. Intuitively, underfitting occurs when the model or the algorithm does not fit the data well enough. Specifically, underfitting occurs if the model or algorithm shows low variance but high bias. Underfitting is often a result of an excessively simple model.

Both overfitting and underfitting lead to poor predictions on new data sets.

4. Explain the meaning of the no free lunch theorem.

ANSWER:

On a particular problem, different search algorithms may obtain different results, but over all problems, they are indistinguishable. It follows that if an algorithm achieves superior results on some problems, it must pay with inferiority on other problems. In this sense there is no free lunch in search.

5. For the following 4 models, write their structural equations and comment:

a. Which one overfits more?

b. Which one underfits more?

c. Is the model that fits the training data the best one that has the best predictive power?

d. Can we use a confusion matrix to analyze the problems of underfitting?

e. Which data set should we use to run these regressions?

ANSWER:

Firstly, After comparing the accuracies of four different models using both training and testing data, I find that they are quite similar. This means that in this case, those models do not make much difference in fitting data. The conclusion makes sense because in all, they are all linear models that have similar model structure. Furthermore, due to the high accuracy of all four models (nearly 90 is very high), we can also guess that the scenario is quite suitable for predicting data using linear model.

Secondly, my opinion about " Is the model that fits the training data the best one that has the best predictive power?" is not exactly. The model that fits the training data the best does not mean to have the best predictive power. When it comes to prediction, we need to consider confusion matrix(accuracy) and overfitting/underfitting problem. Overfits leads to bad prediction because the model fits the training data too well but not the testing data. Sometimes bias is necessary to obtain the best prediction. (Bias-variance tradeoff)

Thirdly, a confusion matrix is a table that is often used to describe the performance of a classification model (or "classifier") on a set of test data for which the true values are known. It allows the visualization of the performance of an algorithm. However, if we want to analyse overfitting/underfitting problem, we need to compare the two confusion matrixs using the training and testing data separately. If the model is overfitting, it shows a higher accuracy in fitting training data. If the model is underfitting, it shows a higher accuracy in fitting testing data.

About the dataset, I think we should use the datset that drops the unrelevant variables.

```
# Dataset manipulation: drop unrelevant variables
dta_bank_drop <- dta_bank_outliers_rm %>% select(age, contact, month, day_of_week, y)
```

split training, validating and testing data

```
# Load train, valid and test datasets: 80%, 10%, 10%
# Training & validation
RAMDOM_SEED = 22
sample_split = function(dta_bank_drop, y, p = 0.8, v = 0.1) {
  len = length(unlist(dta_bank_drop[,1]))

  # Generate index for spliting original dataset
  set.seed(RAMDOM_SEED)
  train_index = sample(1:len, round(len*p, 0))
  val_index = sample((1:len)[-train_index], round(len*v, 0))
  test_index = (1:len)[-c(train_index, val_index)]
  cat(sprintf('[Description]\n  Number of sample: %7s\n  Training set: %11s\n  Validation set: %8s\n  Te
              len, length(train_index), length(val_index), length(test_index)))

  # Generate training set, validation set and testing set
  X = colnames(dta_bank_drop)[colnames(dta_bank_drop) != y]
  train_x = dta_bank_drop[train_index, X]
  train_y = dta_bank_drop[train_index, y]
  val_x = dta_bank_drop[val_index, X]
  val_y = dta_bank_drop[val_index, y]
  test_x = dta_bank_drop[test_index, X]
  test_y = dta_bank_drop[test_index, y]
```

```
  dataset_list = list(train_x=train_x, train_y=train_y, val_x=val_x,
                      val_y=val_y, test_x=test_x, test_y=test_y,
                      train_n=length(train_index), val_n=length(val_index), test_n=length(test_index))
}
```

```
dta_bank_drop = as.data.frame(dta_bank_drop)
sample        = sample_split(dta_bank_drop,'y')
```

```
## [Description]
##   Number of sample:   40703
##   Training set:       32562
##   Validation set:     4070
##   Test set:            4071
```

regression 1:

```
lm1              =  lm     (sample$train_y ~ age + factor(month), data = sample$train_x)
result1          <- lm1    %>% tidy() %>% print(n = 1e3)
```

```
## # A tibble: 11 x 5
##    term               estimate std.error statistic   p.value
##    <chr>                 <dbl>     <dbl>     <dbl>     <dbl>
##  1 (Intercept)       0.200      0.00925     21.6   5.96e-103
##  2 age               0.0000733  0.000163     0.450 6.53e-  1
##  3 factor(month)aug -0.101      0.00777    -13.0   7.98e- 39
##  4 factor(month)dec  0.297      0.0258      11.5   1.68e- 30
##  5 factor(month)jul -0.122      0.00763    -16.0   2.57e- 57
##  6 factor(month)jun -0.104      0.00796    -13.1   6.83e- 39
##  7 factor(month)mar  0.291      0.0160      18.2   1.84e- 73
##  8 factor(month)may -0.142      0.00711    -20.0   4.37e- 88
##  9 factor(month)nov -0.104      0.00834    -12.4   1.85e- 35
## 10 factor(month)oct  0.238      0.0141      16.9   6.71e- 64
## 11 factor(month)sep  0.249      0.0159      15.7   3.25e- 55
```

```
pred_data1       =  predict(lm1, sample$test_x)
pred_y_reg1      =  copy   (pred_data1)
pred_data1_train =  predict(lm1, sample$train_x)
```

```
# Transform continuous data to discrete data: classifier
for (i in 1:length(pred_data1)) {
  if(pred_data1[i] <  0.5){pred_data1[i] = 0}
  if(pred_data1[i] >= 0.5){pred_data1[i] = 1}
}

for (i in 1:length(pred_data1_train)) {
  if(pred_data1_train[i] <  0.5){pred_data1_train[i] = 0}
  if(pred_data1_train[i] >= 0.5){pred_data1_train[i] = 1}
}
```

```
# Confusion matrix and Accuracy
conf_nat1       <- table(pred_data1, sample$test_y)
Accuracy1       <- sum  (diag(conf_nat1))/ sum(conf_nat1)*100
conf_nat1
```

```
##
## pred_data1    0    1
```

```
##          0 3642  425
##          1    2    2
```

```
Accuracy1
```

```
## [1] 89.51118
```

```
conf_nat1_train <- table(pred_data1_train, sample$train_y)
Accuracy1_train <- sum  (diag(conf_nat1_train))/ sum(conf_nat1_train)*100
conf_nat1_train
```

```
##
## pred_data1_train     0     1
##                0 29025  3463
##                1    38    36
```

```
Accuracy1_train
```

```
## [1] 89.2482
```

regression 2:

```
lm2              =  lm     (sample$train_y ~ age + I(age^2) + I(age^3) + factor(month),
                            data = sample$train_x)
result2          <- lm2    %>% tidy() %>% print(n = 1e3)
```

```
## # A tibble: 13 x 5
##    term                 estimate   std.error statistic  p.value
##    <chr>                   <dbl>       <dbl>     <dbl>    <dbl>
##  1 (Intercept)       0.705       0.0624         11.3   1.62e-29
##  2 age              -0.0255      0.00429        -5.95  2.78e- 9
##  3 I(age^2)          0.000312    0.0000951       3.28  1.03e- 3
##  4 I(age^3)         -0.000000225 0.000000676    -0.333 7.39e- 1
##  5 factor(month)aug -0.0953      0.00771       -12.4   4.84e-35
##  6 factor(month)dec  0.266       0.0256         10.4   4.34e-25
##  7 factor(month)jul -0.120       0.00757       -15.8   4.33e-56
##  8 factor(month)jun -0.0964      0.00791       -12.2   4.01e-34
##  9 factor(month)mar  0.273       0.0159         17.1   1.70e-65
## 10 factor(month)may -0.133       0.00707       -18.9   6.94e-79
## 11 factor(month)nov -0.0934      0.00829       -11.3   1.97e-29
## 12 factor(month)oct  0.208       0.0140         14.8   1.64e-49
## 13 factor(month)sep  0.228       0.0158         14.4   6.35e-47
```

```
pred_data2       =  predict(lm2,sample$test_x)
pred_y_reg2      =  copy   (pred_data2)
pred_data2_train =  predict(lm2, sample$train_x)
```

```
# Transform continuous data to discrete data: classifier
for (i in 1:length(pred_data2)) {
  if(pred_data2[i] <  0.5){pred_data2[i] = 0}
  if(pred_data2[i] >= 0.5){pred_data2[i] = 1}
}

for (i in 1:length(pred_data2_train)) {
  if(pred_data2_train[i] <  0.5){pred_data2_train[i] = 0}
  if(pred_data2_train[i] >= 0.5){pred_data2_train[i] = 1}
}
```

```
# Confusion matrix and accuracy
conf_nat2       <- table(pred_data2, sample$test_y)
Accuracy2       <- sum  (diag(conf_nat2))/ sum(conf_nat2)*100
conf_nat2
```

```
##
## pred_data2    0    1
##          0 3617  396
##          1   27   31
```

```
Accuracy2
```

```
## [1] 89.60943
```

```
conf_nat2_train <- table(pred_data2_train, sample$train_y)
Accuracy2_train <- sum  (diag(conf_nat2_train))/ sum(conf_nat2_train)*100
conf_nat2_train
```

```
##
## pred_data2_train     0     1
##                0 28867  3344
##                1   196   155
```

```
Accuracy2_train
```

```
## [1] 89.12843
```

regression 3:

```
lm3             =  lm     (sample$train_y ~ ., data = sample$train_x)
result3         <- lm3    %>% tidy() %>% print(n = 1e3)
```

```
## # A tibble: 16 x 5
##    term              estimate std.error statistic   p.value
##    <chr>                <dbl>     <dbl>     <dbl>     <dbl>
##  1 (Intercept)       0.196     0.00976      20.1  3.93e- 89
##  2 age               0.000328  0.000162      2.03 4.29e-  2
##  3 contacttelephone -0.0982    0.00430     -22.9  8.35e-115
##  4 monthaug         -0.107     0.00773     -13.8  3.80e- 43
##  5 monthdec          0.304     0.0256       11.9  1.57e- 32
##  6 monthjul         -0.116     0.00760     -15.3  1.66e- 52
##  7 monthjun         -0.0289    0.00858      -3.37 7.49e-  4
##  8 monthmar          0.294     0.0159       18.5  4.37e- 76
##  9 monthmay         -0.0917    0.00744     -12.3  9.21e- 35
## 10 monthnov         -0.103     0.00830     -12.4  2.34e- 35
## 11 monthoct          0.249     0.0140       17.8  1.47e- 70
## 12 monthsep          0.254     0.0158       16.1  8.23e- 58
## 13 day_of_weekmon   -0.0135    0.00519      -2.61 9.18e-  3
## 14 day_of_weekthu    0.00846   0.00520       1.63 1.04e-  1
## 15 day_of_weektue    0.00779   0.00528       1.47 1.40e-  1
## 16 day_of_weekwed    0.0106    0.00526       2.01 4.43e-  2
```

```
pred_data3       =  predict(lm3, sample$test_x)
pred_y_reg3      =  copy   (pred_data3)
pred_data3_train =  predict(lm3, sample$train_x)
```

```
# Transform continuous data to discrete data: classifier
for (i in 1:length(pred_data3)) {
```

```
  if(pred_data3[i] < 0.5){pred_data3[i] = 0}
  if(pred_data3[i] >= 0.5){pred_data3[i] = 1}
}

for (i in 1:length(pred_data3_train)) {
  if(pred_data3_train[i] < 0.5){pred_data3_train[i] = 0}
  if(pred_data3_train[i] >= 0.5){pred_data3_train[i] = 1}
}
```

```
conf_nat3       <- table(pred_data3, sample$test_y)
Accuracy3       <- sum  (diag(conf_nat3)) /sum(conf_nat3)*100
conf_nat3
```

```
##
## pred_data3    0    1
##          0 3624  402
##          1   20   25
```

```
Accuracy3
```

```
## [1] 89.634
```

```
conf_nat3_train <- table(pred_data3_train, sample$train_y)
Accuracy3_train <- sum  (diag(conf_nat3_train)) /sum(conf_nat3_train)*100
conf_nat3_train
```

```
##
## pred_data3_train     0     1
##                0 28902  3298
##                1   161   201
```

```
Accuracy3_train
```

```
## [1] 89.37719
```

regression 4:

```
lm4              =  lm     (sample$train_y ~ .^2, data = sample$train_x)
result4          <- lm4    %>% tidy() %>% print(n = 1e3)
```

```
## # A tibble: 79 x 5
##    term                              estimate std.error statistic  p.value
##    <chr>                                <dbl>     <dbl>     <dbl>    <dbl>
##  1 (Intercept)                       0.102       0.0298     3.43    6.11e- 4
##  2 age                              -0.00000665  0.000673  -0.00989 9.92e- 1
##  3 contacttelephone                  0.0738      0.0312     2.37    1.79e- 2
##  4 monthaug                         -0.0250      0.0326    -0.766   4.44e- 1
##  5 monthdec                          0.369       0.114      3.25    1.17e- 3
##  6 monthjul                          0.0149      0.0321     0.466   6.42e- 1
##  7 monthjun                          0.338       0.0373     9.06    1.37e-19
##  8 monthmar                          0.470       0.0590     7.97    1.59e-15
##  9 monthmay                         -0.0146      0.0309    -0.472   6.37e- 1
## 10 monthnov                         -0.0455      0.0374    -1.22    2.24e- 1
## 11 monthoct                          0.538       0.0502    10.7     1.11e-26
## 12 monthsep                          0.513       0.0606     8.47    2.54e-17
## 13 day_of_weekmon                   -0.0557      0.0272    -2.04    4.11e- 2
## 14 day_of_weekthu                    0.183       0.0269     6.80    1.05e-11
## 15 day_of_weektue                    0.177       0.0320     5.53    3.24e- 8
```

```
## 16 day_of_weekwed                    0.164       0.0309     5.30    1.18e- 7
## 17 age:contacttelephone             -0.000893    0.000420  -2.13    3.36e- 2
## 18 age:monthaug                      0.000743    0.000698   1.06    2.88e- 1
## 19 age:monthdec                      0.000418    0.00177    0.236   8.14e- 1
## 20 age:monthjul                     -0.0000596   0.000686  -0.0869  9.31e- 1
## 21 age:monthjun                     -0.000490    0.000790  -0.621   5.35e- 1
## 22 age:monthmar                      0.000169    0.00122    0.139   8.89e- 1
## 23 age:monthmay                      0.000749    0.000683   1.10    2.73e- 1
## 24 age:monthnov                      0.00114     0.000794   1.44    1.51e- 1
## 25 age:monthoct                     -0.00269     0.000979  -2.75    5.95e- 3
## 26 age:monthsep                     -0.00426     0.00115   -3.69    2.23e- 4
## 27 age:day_of_weekmon                0.00127     0.000511   2.48    1.31e- 2
## 28 age:day_of_weekthu                0.000447    0.000512   0.874   3.82e- 1
## 29 age:day_of_weektue                0.000358    0.000516   0.694   4.88e- 1
## 30 age:day_of_weekwed                0.000668    0.000518   1.29    1.97e- 1
## 31 contacttelephone:monthaug        -0.00738     0.0318    -0.232   8.16e- 1
## 32 contacttelephone:monthdec        -0.243       0.0705    -3.45    5.66e- 4
## 33 contacttelephone:monthjul        -0.0854      0.0268    -3.18    1.45e- 3
## 34 contacttelephone:monthjun        -0.419       0.0276   -15.2     6.10e-52
## 35 contacttelephone:monthmar        -0.141       0.0524    -2.68    7.35e- 3
## 36 contacttelephone:monthmay        -0.124       0.0252    -4.94    8.05e- 7
## 37 contacttelephone:monthnov        -0.00741     0.0297    -0.250   8.03e- 1
## 38 contacttelephone:monthoct        -0.0841      0.0387    -2.18    2.96e- 2
## 39 contacttelephone:monthsep        -0.334       0.0473    -7.06    1.76e-12
## 40 contacttelephone:day_of_weekmon -0.00251     0.0136    -0.185   8.53e- 1
## 41 contacttelephone:day_of_weekthu -0.00104     0.0138    -0.0753  9.40e- 1
## 42 contacttelephone:day_of_weektue  0.00742     0.0136     0.544   5.86e- 1
## 43 contacttelephone:day_of_weekwed  0.000262    0.0136     0.0192  9.85e- 1
## 44 monthaug:day_of_weekmon          -0.0158     0.0227    -0.696   4.87e- 1
## 45 monthdec:day_of_weekmon          -0.0321     0.0842    -0.381   7.03e- 1
## 46 monthjul:day_of_weekmon          -0.0325     0.0225    -1.45    1.48e- 1
## 47 monthjun:day_of_weekmon           0.00733    0.0251     0.293   7.70e- 1
## 48 monthmar:day_of_weekmon          -0.208      0.0488    -4.27    1.99e- 5
## 49 monthmay:day_of_weekmon           0.00831    0.0214     0.389   6.97e- 1
## 50 monthnov:day_of_weekmon          -0.0183     0.0246    -0.744   4.57e- 1
## 51 monthoct:day_of_weekmon          -0.199      0.0438    -4.54    5.54e- 6
## 52 monthsep:day_of_weekmon           0.00538    0.0520     0.103   9.18e- 1
## 53 monthaug:day_of_weekthu          -0.220      0.0222    -9.92    3.67e-23
## 54 monthdec:day_of_weekthu          -0.167      0.0868    -1.92    5.45e- 2
## 55 monthjul:day_of_weekthu          -0.229      0.0220   -10.4     2.79e-25
## 56 monthjun:day_of_weekthu          -0.189      0.0254    -7.44    1.02e-13
## 57 monthmar:day_of_weekthu          -0.333      0.0517    -6.45    1.16e-10
## 58 monthmay:day_of_weekthu          -0.206      0.0211    -9.78    1.53e-22
## 59 monthnov:day_of_weekthu          -0.201      0.0241    -8.36    6.80e-17
## 60 monthoct:day_of_weekthu          -0.251      0.0415    -6.06    1.42e- 9
## 61 monthsep:day_of_weekthu          -0.156      0.0471    -3.31    9.46e- 4
## 62 monthaug:day_of_weektue          -0.189      0.0277    -6.83    8.41e-12
## 63 monthdec:day_of_weektue           0.00174    0.102      0.0170  9.86e- 1
## 64 monthjul:day_of_weektue          -0.222      0.0276    -8.03    1.03e-15
## 65 monthjun:day_of_weektue          -0.216      0.0299    -7.23    5.11e-13
## 66 monthmar:day_of_weektue          -0.175      0.0502    -3.49    4.91e- 4
## 67 monthmay:day_of_weektue          -0.207      0.0267    -7.74    1.01e-14
## 68 monthnov:day_of_weektue          -0.204      0.0292    -6.98    2.93e-12
## 69 monthoct:day_of_weektue          -0.258      0.0462    -5.58    2.41e- 8
```

```
## 70 monthsep:day_of_weektue         -0.101       0.0506    -2.00    4.52e- 2
## 71 monthaug:day_of_weekwed         -0.194       0.0266    -7.30    2.91e-13
## 72 monthdec:day_of_weekwed         -0.0766      0.0915    -0.837   4.02e- 1
## 73 monthjul:day_of_weekwed         -0.218       0.0265    -8.22    2.10e-16
## 74 monthjun:day_of_weekwed         -0.179       0.0290    -6.17    7.05e-10
## 75 monthmar:day_of_weekwed         -0.196       0.0591    -3.31    9.31e- 4
## 76 monthmay:day_of_weekwed         -0.194       0.0255    -7.60    3.14e-14
## 77 monthnov:day_of_weekwed         -0.194       0.0281    -6.90    5.14e-12
## 78 monthoct:day_of_weekwed         -0.258       0.0453    -5.70    1.22e- 8
## 79 monthsep:day_of_weekwed         -0.0600      0.0491    -1.22    2.22e- 1
```

```
pred_data4       =  predict(lm4,sample$test_x)
pred_y_reg4      =  copy   (pred_data4)
pred_data4_train =  predict(lm4, sample$train_x)
```

```
# Transform continuous data to discrete data: classifier
for (i in 1:length(pred_data4)) {
  if(pred_data4[i] <  0.5){pred_data4[i] = 0}
  if(pred_data4[i] >= 0.5){pred_data4[i] = 1}
}

for (i in 1:length(pred_data4_train)) {
  if(pred_data4_train[i] <  0.5){pred_data4_train[i] = 0}
  if(pred_data4_train[i] >= 0.5){pred_data4_train[i] = 1}
}
```

```
conf_nat4       <- table(pred_data4, sample$test_y)
Accuracy4       <- sum  (diag(conf_nat4))/ sum(conf_nat4)*100
conf_nat4
```

```
##
## pred_data4    0    1
##          0 3617  394
##          1   27   33
```

```
Accuracy4
```

```
## [1] 89.65856
```

```
conf_nat4_train <- table(pred_data4_train, sample$train_y)
Accuracy4_train <- sum  (diag(conf_nat4_train))/ sum(conf_nat4_train)*100
conf_nat4_train
```

```
##
## pred_data4_train     0     1
##                0 28817  3179
##                1   246   320
```

```
Accuracy4_train
```

```
## [1] 89.4816
```

Improving the predictive power

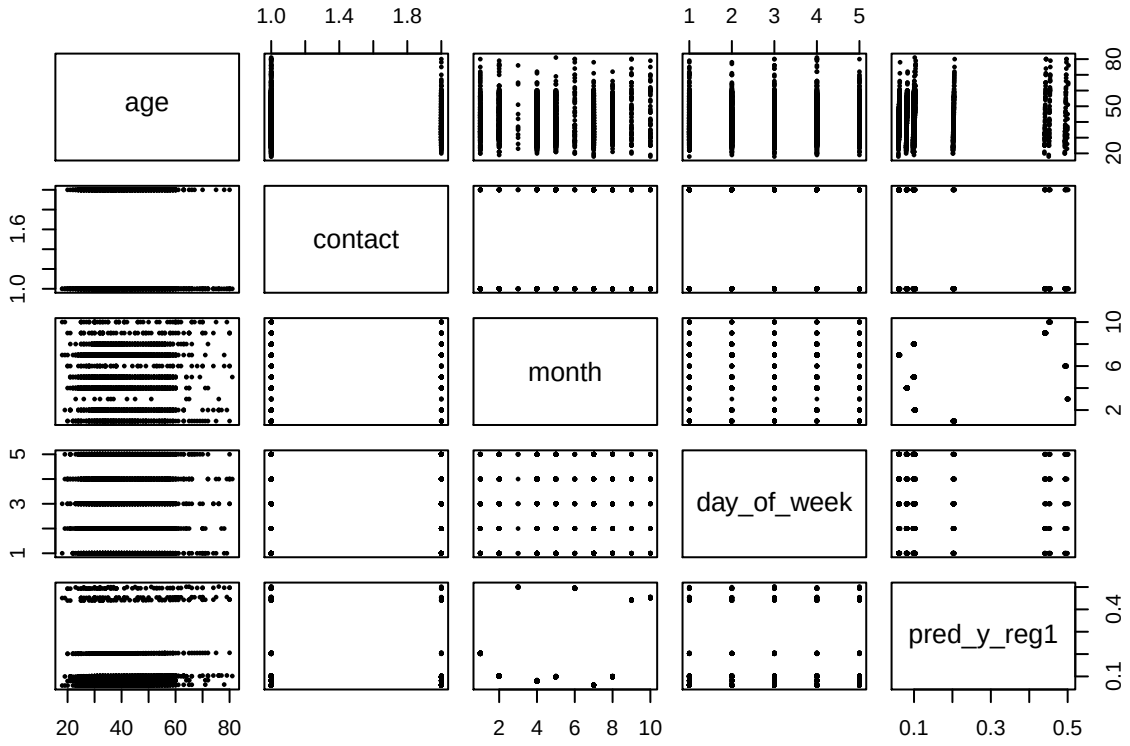
1. Make a visualization to inspect the relationship between the Y and each of the X that you have included in the regressions above.

a. Does it look linear?

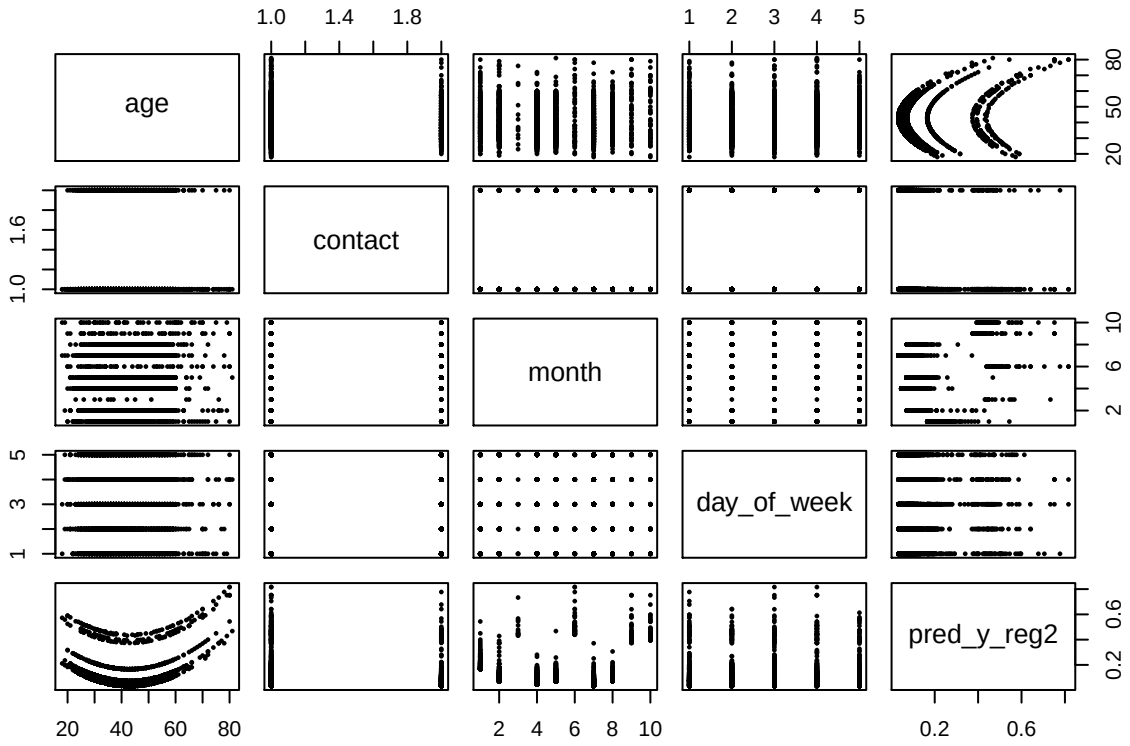
ANSWER:

Regression 1, 3 and 4 look linear while regression 2 looks nonlinear.

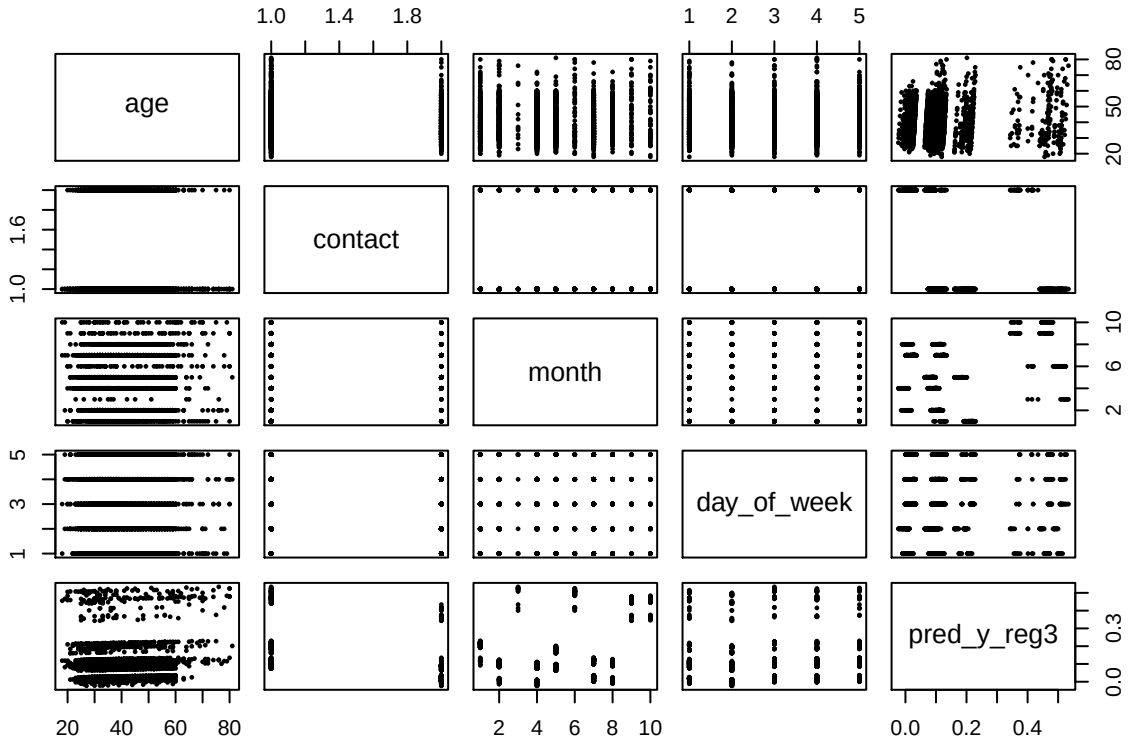
```
pairs(cbind(sample$test_x, pred_y_reg1), pch = 16, cex = .5)
```



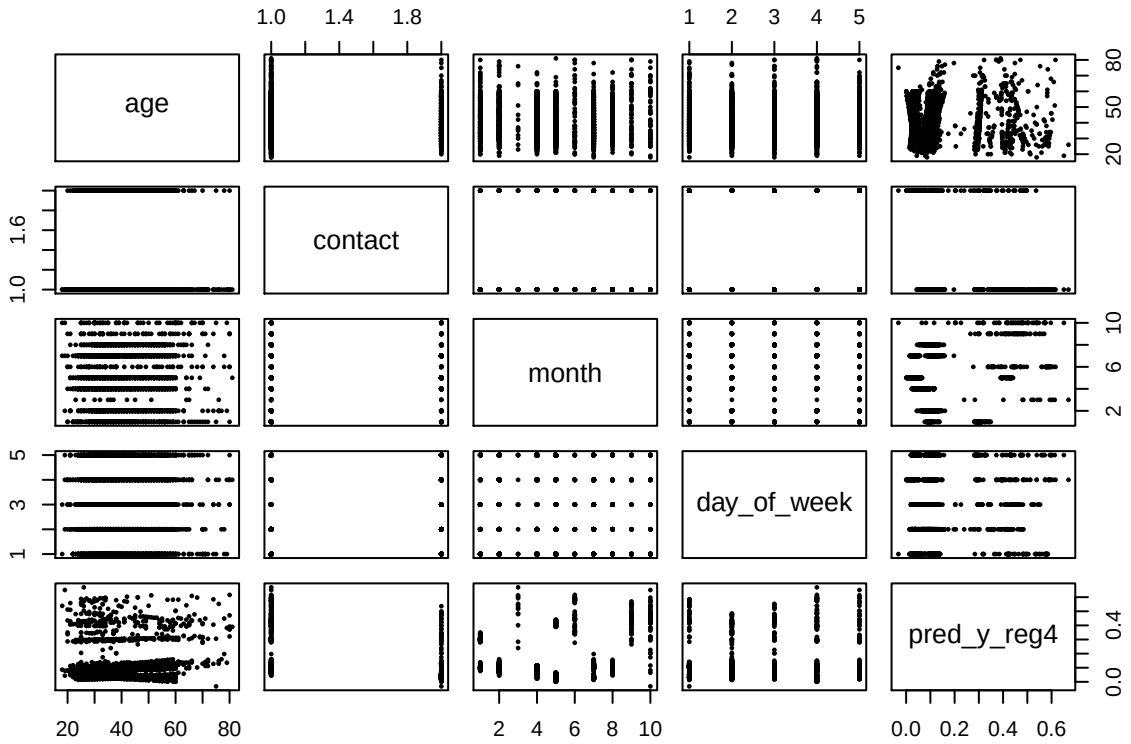

```
pairs(cbind(sample$test_x, pred_y_reg2), pch = 16, cex = .5)
```



```
pairs(cbind(sample$test_x, pred_y_reg3), pch = 16, cex = .5)
```




```
pairs(cbind(sample$test_x, pred_y_reg4), pch = 16, cex = .5)
```



2. Use the other predictive methods seen in class (like NB classifiers or KNN) to check if you can improve the performance.

3. Do they make it better? Worse?

ANSWER:

Both NB classifier and kNN are used to do predictions and their accuracy are quite similar. The accuracy of NB classifier is 89.48661 and the accuracy of kNN 89.43748, is which means NB classifiers is slightly better.

```
# data manipulation for NB classifier
data    = dta_bank_drop %>% mutate(subscribe = factor(y, levels = c(0, 1), labels = c('No', 'Yes'))) %>%
sample2 = sample_split  (data,'subscribe')
```

```
## [Description]
##   Number of sample:   40703
##   Training set:       32562
##   Validation set:     4070
##   Test set:            4071
```

naive bayes

```
# Naive bayes
NBclassifier         =  naivebayes::naive_bayes(formula = sample2$train_y ~ .,
                                                laplace = 1,
                                                data    = sample2$train_x)

# Evaluating model performance on validation set
pred_nb_valid        =  predict(NBclassifier, newdata = sample2$val_x  )
pred_nb_training     =  predict(NBclassifier, newdata = sample2$train_x)
pred_nb_testing      =  predict(NBclassifier, newdata = sample2$test_x )

# Accuracy and cross table of training and testing data
conf_nat_nb_training <- table(pred_nb_training, sample2$train_y)
Accuracy_nb_training <- sum  (diag(conf_nat_nb_training))/ sum(conf_nat_nb_training)*100
conf_nat_nb_training
```

```
##
## pred_nb_training    No   Yes
##              No  28502  2967
##              Yes   561   532
```

```
Accuracy_nb_training
```

```
## [1] 89.16528
```

```
conf_nat_nb_testing  <- table(pred_nb_testing, sample2$test_y)
Accuracy_nb_testing  <- sum  (diag(conf_nat_nb_testing))/ sum(conf_nat_nb_testing)*100
conf_nat_nb_testing
```

```
##
## pred_nb_testing   No  Yes
##             No  3571  355
##             Yes   73   72
```

```
Accuracy_nb_testing
```

```
## [1] 89.48661
```

kNN

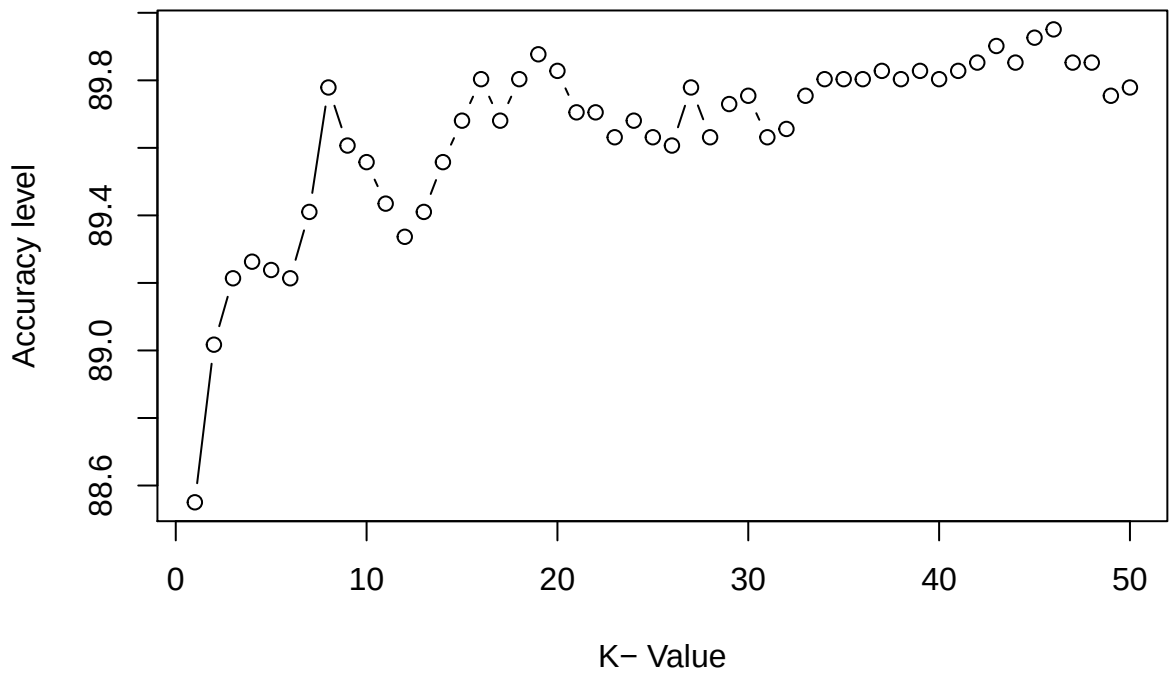
```r
# Transform data into dummy variable
dta_bank_d_final     <- dummy_cols(dta_bank_drop) %>% select(-contact, -month,
                                                             -day_of_week)
# Define a min-max normalize() function
normalize <- function(x) {
return((x - min(x)) / (max(x) - min(x)))
}

# Normalize the age variable
dta_bank_d_final$age =  normalize(dta_bank_d_final$age)
```

```r
#find the optimized k
i      = 1                       # declaration to initiate for loop
k.optm = 1                       # declaration to initiate for loop
for (i in 1:50){
    knn.mod   = class::knn(train = sample3$train_x, cl    = sample3$train_y,
                           test  = sample3$val_x  , k     = i)
    k.optm[i] = 100 * sum(sample3$val_y == knn.mod)/ NROW(sample3$val_y)
    k         = i
    cat(k,'=',k.optm[i],'\n')       # to print % accuracy
}
```

```
## 1 = 88.55037
## 2 = 89.0172
## 3 = 89.21376
## 4 = 89.2629
## 5 = 89.23833
## 6 = 89.21376
## 7 = 89.41032
## 8 = 89.77887
## 9 = 89.60688
## 10 = 89.55774
## 11 = 89.43489
## 12 = 89.33661
## 13 = 89.41032
## 14 = 89.55774
## 15 = 89.68059
## 16 = 89.80344
## 17 = 89.68059
## 18 = 89.80344
## 19 = 89.87715
## 20 = 89.82801
## 21 = 89.70516
## 22 = 89.70516
## 23 = 89.63145
## 24 = 89.68059
## 25 = 89.63145
## 26 = 89.60688
## 27 = 89.77887
## 28 = 89.63145
## 29 = 89.72973
## 30 = 89.7543
## 31 = 89.63145
## 32 = 89.65602
```

```
## 33 = 89.7543
## 34 = 89.80344
## 35 = 89.80344
## 36 = 89.80344
## 37 = 89.82801
## 38 = 89.80344
## 39 = 89.82801
## 40 = 89.80344
## 41 = 89.82801
## 42 = 89.85258
## 43 = 89.90172
## 44 = 89.85258
## 45 = 89.92629
## 46 = 89.95086
## 47 = 89.85258
## 48 = 89.85258
## 49 = 89.7543
## 50 = 89.77887
```

```
# accuracy plot: choose k = 46
plot(k.optm, type = "b", xlab = "K- Value",ylab = "Accuracy level")
```




```
# knn of validating data for k = 46
knn_valid_46 =  knn(train = sample3$train_x, test = sample3$val_x,
                    cl    = sample3$train_y, k    = 46)
ACC_valid_46 <- 100 * sum(valid_actual == knn_valid_46)/ NROW(valid_actual)
ACC_valid_46
```

```
## [1] 89.90172
```

```
# knn of testing data for k = 46
knn_test_46  =  knn(train = sample3$train_x, test = sample3$test_x,
                    cl    = sample3$train_y, k    = 46)
test_actual  =  sample3$test_y
```

```r
ACC_test_46  <- 100 * sum(test_actual == knn_test_46)/ NROW(test_actual)
ACC_test_46
```

```
## [1] 89.53574
```

```r
# knn of training data for k = 46
knn_train_46 =  knn(train = sample3$train_x, test = sample3$train_x,
                    cl    = sample3$train_y, k = 46)
train_actual =  sample3$train_y
ACC_train_46 <- 100 * sum(train_actual == knn_train_46)/ NROW(train_actual)
ACC_train_46
```

```
## [1] 89.39561
```

Causal Questions

1. When we study causality we always focus on the parameters multiplying the X variables instead of the predictive capacity of the model. We then give a causal interpretation to the estimated coefficients.

a. Explain when in marketing is preferable a causal analysis to a predictive analysis.

ANSWER:

In prediction analysis, the goal is to develop a formula for making predictions about the dependent variable, based on the observed values of the independent variables. In causal analysis, the independent variables are regarded as causes of the dependent variable. The aim of the study is to determine whether a particular independent variable really affects the dependent variable, and to estimate the magnitude of that effect, if any.

Marketers care more about the casualities between parameters instead of using independent variables to do predictions about dependent variables. Their goal is to explain the relationships between different variables.

b. In the context of a linear regression, explain the concepts of a biased estimator.

ANSWER:

In statistics, the bias of an estimator is the difference between this estimator's expected value and the true value of the parameter being estimated. An estimator or decision rule with zero bias is called unbiased. In the context of a linear regression , biased estimator means the estimator is different from the true value.

2. Which of the variables could be interesting to analyze from a causal point of view. Give examples.

ANSWER:

Actually, those variables dropped when doing predictive analysis are quite intesting to analyze from a casual point of view. They cannot be predicted as predictors but are effective in explaining the dependent variable.

3. For those variables what would be the potential omitted variables problem?

ANSWER:

The answer is quite similar in the first part: Basic Explanatory Analysis. I come up with several variables that if omitted will cause OVB problem: client related variable: client's honest history income place of birth

bank related variable: number of contacts performed during marketing campaign and for this client, outcome of the previous marketing campaign

social and economic variables: consumer price index