

# Regression Project

Vinay Sharma

Sonia Mehra

Rajat Agarwal

Souvik Paul

Ankit Gupta

July 2020

## Abstract

Here , we are working with facebook metric data published in 2014 on a facebook page of a cosmetic brand. Initially we check missing observation in the data(if any found we delete those) then we split data into test and train for further analysis, the EDA( exploratory data analysis) , changing categorical variable into indicators and then move to linear regression assumptions. Next we detect Leverage/influential points with help of Cook's Distance , DFFITS , DFBETAS , COVRATIO. Then move to deal with different problems like normality( checking using plot and Shapiro-Wilk Test if found transform our response), heteroscedasticity(checking using plots of regressor vs residuals and also by Breusch Pagon test if found transformed our regressors) , multicollinearity(check using Condition Number and Variance Inflation Factor if found we deal with different techniques like Principal Component Regression(PCR),variable elimination,Partial Least Square Regression (PLS) and lastly the LASSO using cross validation). LASSO also gives us a way for variable selection as well as a way to remove multicollinearity . Finding MAPE in each of model allows us to compare each models to get the model with best predicting power. Finally we Checking for Significance of first level of each Categorical Variable and correct all with suitable remedies to get a model with minimum MAPE.

**Key words:** MAPE , LASSO , Breusch Pagon test , Cross Validation , VIF , Multicollinearity , Influential , Cook's Distance , EDA , Regression , PLS , PCR.

## **1 Acknowledgement**

We would like to express our heartfelt gratitude to Dr. Minerva Mukhopadhyay for helping us in every difficulty which we face during this project. It has been a great learning experience and has also provided us with a practical insight of the theoretical knowledge gathered during the course Regression Analysis.

We also take this opportunity to thank the authors and publishers of the various books and journals we have consulted. Without those this work would not have been completed.

We would also like to thank our parents for their extensive support throughout the session. Their constant encouragement has enabled us to complete the project within the stipulated time-period.

# Contents

<b>1 Acknowledgement</b>	<b>3</b>
<b>2 Introduction</b>	<b>6</b>
<b>3 Objective</b>	<b>6</b>
<b>4 MAPE</b>	<b>6</b>
<b>5 Data Description</b>	<b>7</b>
<b>6 Data Pre-Processing</b>	<b>9</b>
6.1 Missing Value . . . . .	9
6.2 Categorical Variable . . . . .	10
6.3 Splitting Data into Train and Test . . . . .	10
<b>7 Exploratory Data Analysis</b>	<b>11</b>
7.1 Analysing Categorical Variable using Box-Plot And frequency table . . . . .	12
7.1.1 Type . . . . .	12
7.1.2 Category . . . . .	13
7.1.3 Paid . . . . .	13
7.1.4 Post Month . . . . .	14
7.1.5 Post Hour . . . . .	15
7.1.6 Post Weekday . . . . .	16
7.2 Analysing Numerical Variable using Pair-Plot and correlation matrix . . . . .	17
<b>8 Changing categorical variable into Indicator variable and Scaling Numerical Quantity.</b>	<b>19</b>
<b>9 Primary Model</b>	<b>19</b>
9.1 Assumptions . . . . .	19
<b>10 Diagnostics for Leverage/ Influential Points</b>	<b>20</b>
10.1 Leverage Points . . . . .	20
10.2 Diagnostics for Influential points . . . . .	22
10.2.1 Cook's Distance . . . . .	22
10.2.2 DFFITS . . . . .	24
10.2.3 DFBETAS . . . . .	26
10.2.4 COVRATIO . . . . .	26
<b>11 Normality Checking</b>	<b>28</b>
11.1 Checking Through Plot . . . . .	28
11.2 Checking Through Test . . . . .	28
11.3 Calculations . . . . .	29
11.4 New Model . . . . .	33
<b>12 Heteroscedasticity</b>	<b>33</b>
12.1 Heteroscedasticity Diagnostics . . . . .	33
12.2 Removal of Heteroscedasticity . . . . .	35

<b>13 Multicollinearity</b>	<b>37</b>
13.1 Multicollinearity diagnostics . . . . .	37
13.2 Dealing with multicollinearity . . . . .	37
13.2.1 Principal Component Regression . . . . .	37
13.2.2 Variable Elimination . . . . .	40
13.2.3 Principal Component Regression(PCR) on entire dataset . . . . .	41
13.2.4 Partial Least Square Regression (PLSR)on entire dataset . . . . .	42
13.2.5 The Lasso . . . . .	44
<b>14 Checking for Significance of first level of each Categorical Variable</b>	<b>46</b>
<b>15 Conclusion</b>	<b>52</b>
<b>16 Bibliography</b>	<b>52</b>
<b>17 Appendix</b>	<b>53</b>

## 2 Introduction

Social media marketing is marketing of any product via social media like Facebook , WhatsApp ,Instagram e.t.c. It allows individuals , businesses and other organizations to interact with one another on a single online platform. A major advantage is there direct interactions which helps them to build relationship and communities online .

Although , business retailers have gained over 133% of their revenues from social media marketing . From the recent data there are more than 3 billion active user on social media including Facebook , Instagram , WhatsApp e.t.c. Any company marketing strategies must include one important part of social media marketing . As it is the easiest and fastest way to interact directly with the consumer of a company . It helps company to increase their sales and helps in their overall development.

Here we are working on Facebook metric data collected from a post published on a Facebook page of renowned cosmetic brand . The company want to increase the number of lifetime engaged user with company . While modeling the data with Linear Regression we have encountered the problem of normality, heteroscedasticity ,multicollinearity and fixed all of them with suitable remedies .

## 3 Objective

Our objective is to increase the number of lifetime engaged user . So it will benefit the company in overall development. Keeping focus on features our aim is to maximize the number of users(lifetime engaged).So we need to know,

Which month the brand post more so as to increase lifetime engaged users?

Which type of post engages mostly users?

Which time of the day engaged users?

Do weekday ,paid or unpaid post also play a role in lifetime engaging of users?

What other variables one focus on to increase lifetime engaged users?

## 4 MAPE

We will use Mean Absolute Percentage Error (MAPE) to check how good our model accuracy is and MAPE also will be used for comparing the models. formula for MAPE is as :-

$$MAPE = \frac{1}{n} \sum_{i=1}^n \left| \frac{A_i - F_i}{A_i} \right|$$

Where  $A_i$  is the actual value and  $F_i$  is the forecast value. Multiplying it by 100 gives percentage.

Now we code this function at the beginning of this project -

```

> rm(list=ls())
> getwd()
[1] "/cloud/project"
> setwd("/cloud/project")
> MAPE = function(ycap,y){
+   cat('MAPE',mean(abs((y - ycap)/y)))
+ }

```

Figure 1: Declaring the function

## 5 Data Description

**Title of the Data Set :-** Facebook Metrics Data Set.

**Source :-** This data Set is created by SÃ©rgio Moro, Paulo Rita and Bernardo Vala (ISCTE-IUL) @ 2016.

<http://archive.ics.uci.edu/ml/datasets/Facebook+metrics>

**Data Set Information :-** The data is related to posts published during the year of 2014 on the Facebook's page of a renowned cosmetics brand. This is a multivariate Data Set which contains **500 Instances** and **19 Attributes**

Let's import the data and check the column names -

```

> # Data importing
> X=read.csv("dataset_Facebook.csv",sep=";",header=TRUE,na.string = '?')
> attach(X)
> colnames(X)
[1] "Page.total.likes"
[2] "Type"
[3] "Category"
[4] "Post.Month"
[5] "Post.Weekday"
[6] "Post.Hour"
[7] "Paid"
[8] "Lifetime.Post.Total.Reach"
[9] "Lifetime.Post.Total.Impressions"
[10] "Lifetime.Engaged.Users"
[11] "Lifetime.Post.Consumers"
[12] "Lifetime.Post.Consumptions"
[13] "Lifetime.Post.Impressions.by.people.who.have.liked.your.Page"
[14] "Lifetime.Post.reach.by.people.who.like.your.Page"
[15] "Lifetime.People.who.have.liked.your.Page.and.engaged.with.your.post"
[16] "comment"
[17] "like"
[18] "share"
[19] "Total.Interactions"

```

Figure 2: Data Importing and Colnames

We found that few column names are too large to handle, so, need to shorten them.  
Now see the summary of our data at a glance -

```
> colnames(X)[8:15]=c('Total.Reach','Total.Impressions','Lifetime.Engaged.Users','Consumers','Consumptions','Impressions.liked','Reach.liked','People.engaged.liked')
> str(X)
'data.frame': 500 obs. of 19 variables:
 $ Page.total.likes      : int 139441 139441 139441 139441 139441 139441 139441 139441 139441 ...
1 139441 ...
$ Type                  : Factor w/ 4 levels "Link","Photo",...
$ Category               : int 2 2 3 2 2 2 3 3 2 3 ...
$ Post.Month              : int 12 12 12 12 12 12 12 12 12 12 ...
$ Post.Weekday             : int 4 3 3 2 2 1 1 7 7 6 ...
$ Post.Hour                : int 3 10 3 10 3 9 3 9 3 10 ...
$ Paid                   : int 0 0 0 1 0 0 1 1 0 0 ...
$ Total.Reach              : int 2752 10460 2413 50128 7244 10472 11692 13720 11844 4694 ...
$ Total.Impressions        : int 5091 19057 4373 87991 13594 20849 19479 24137 22538 8668 ...
$ Lifetime.Engaged.Users   : int 178 1457 177 2211 671 1191 481 537 1530 280 ...
$ Consumers                 : int 109 1361 113 790 410 1073 265 232 1407 183 ...
$ Consumptions               : int 159 1674 154 1119 580 1389 364 305 1692 250 ...
$ Impressions.liked          : int 3078 11710 2812 61027 6228 16034 15432 19728 15220 4309 ...
$ Reach.liked                : int 1640 6112 1503 32048 3200 7852 9328 11056 7912 2324 ...
$ People.engaged.liked       : int 119 1108 132 1386 396 1016 379 422 1250 199 ...
$ comment                  : int 4 5 0 58 19 1 3 0 0 3 ...
$ like                      : int 79 130 66 1572 325 152 249 325 161 113 ...
$ share                      : int 17 29 14 147 49 33 27 14 31 26 ...
$ Total.Interactions         : int 100 164 80 1777 393 186 279 339 192 142 ...
```

Figure 3: Summary of data

**Information About Attributes :-** We see that there are 19 Attributes and from the above picture this implies both categorical and numeric variables are present in this data and clearly variable "**Type,Category,Post.Month,Post.Weekday,Post.Hour,Paid**" is categorical and others are "**Numeric Integer**"

- (1) **Page Total Likes** :- Total no of pages likes.
- (2) **Type** :- Factor:{Photo,Status,Video,Link}
- (3) **Category** :- Factor:{action,product,inspiration}
- (4) **Post.month** :- Factor
- (5) **Post.weekday** :- Factor
- (6) **Post.hour** :- Factor
- (7) **Paid** :- Factor{1-yes,0-no}
- (8) **Lifetime Post total Reach**
- (9) **Lifetime post total impressions** :- No of times a post from page is displayed.
- (10) **Lifetime engaged users** :- No of people who clicked anywhere in a post(unique users).
- (11) **Lifetime post consumers** :- The number of people who clicked anywhere in a post.
- (12) **Lifetime post consumption** :- The number of clicks anywhere in a post.
- (13) **Lifetime post impressions by people who have liked your page** :- Total number of impressions just from people who have liked a page
- (14) **Lifetime post reach by people who like your page** :- The number of people who saw a page post because they have liked that page.
- (15) **Lifetime people who have liked your page and engaged with your post** :- The number of people who have liked a Page and clicked anywhere in a post.
- (16) **Comment** :-Number of comments on the publication.
- (17) **Like** :- Number of "Likes" on the publication.
- (18) **share** :- Number of times the publication was shared.
- (19) **total interaction** :-The sum of "likes", "comments", and "shares" of the post.

**"Lifetime engaged users"**, The number of people who clicked anywhere in a post(unique users) is taken as **response variable (Y)**.

## 6 Data Pre-Processing

Let's check out more information and summary statistics of our data including missing values, histograms etc using `skim()` function.

```
> library(ggplot2)
> library(RColorBrewer)
> library(skimr)
> skim(X)
-- Data Summary -----
#> #>   Values
#> Name          X
#> Number of rows 500
#> Number of columns 19
#>
#> Column type frequency:
#>   factor      1
#>   numeric     18
#>
#> Group variables    None
#>
#> -- Variable type: factor -----
#> # A tibble: 1 x 6
#>   skim_variable n_missing complete_rate ordered n_unique top_counts
#>   <chr>           <int>        <dbl> <lgl> <int> <chr>
#> 1 Type              0         1 FALSE      4 Pho: 426, Sta: 45, Lin: 22, Vid: 7
#>
#> -- Variable type: numeric -----
#> # A tibble: 18 x 11
#>   skim_variable n_missing complete_rate      mean       sd     p0     p25     p50     p75     p100 hist
#>   <chr>           <int>        <dbl>      <dbl>     <dbl> <dbl>     <dbl>     <dbl>     <dbl> <chr>
#> 1 Page.total.likes     0         1 123194.  16273.  81370  112676  129600  136393  139441  
#> 2 Category            0         1  1.88     0.853    1     1     2     3     3
#> 3 Post.Month           0         1  7.04     3.31     1     4     7     10    12
#> 4 Post.Weekday          0         1  4.15     2.03     1     2     4     6     7
#> 5 Post.Hour             0         1  7.84     4.37     1     3     9     11    23
#> 6 Paid                 1         0.998  0.279  0.449     0     0     0     1     1
#> 7 Total.Reach          0         1 13903.  22741.   238   3315   5281  13168  180480  
#> 8 Total.Impressions     0         1 29586.  76803.   570   5695.  9051  22086. 1110282  
#> 9 Lifetime.Engaged.Users 0         1  920.    985.     9   394.   626.  1062   11452  
#> 10 Consumers            0         1  799.    883.     9   332.   552.  956.  11328  
#> 11 Consumptions          0         1 1415.   2001.     9   509.   851.  1463  19779  
#> 12 Impressions.liked     0         1 16766.  59791.   567   3970.  6256. 14860. 1107833  
#> 13 Reach.liked           0         1  6585.  7682.   236   2182.  3417  7989  51456  
#> 14 People.engaged.liked  0         1  610.   613.     9   291.   412.  656.  4376
#> 15 comment               0         1  7.48    21.2     0     1     3     7     372
#> 16 like                  1         0.998  178.   323.     0   56.5  101   188.  5172
#> 17 share                  4         0.992  27.3   42.6     0   10   19   32.2  790
#> 18 Total.Interactions     0         1  212.   380.     0    71   124.  228.  6334  
```

Figure 4: More Data Information

### 6.1 Missing Value

We found that there are 6 missing value - 1 in 'Paid', 1 in 'like' and 4 missing values in 'share'. We are not bother with the last 5 missing values as later we will exclude the columns 'like', 'comments', 'share' as these column sums is represented by the column 'Total.interaction'. So, now we only exclude the missing values in 'Paid'.

```

> #see that Paid contains a missing value
> which(is.na(X$Paid)==T) #500
[1] 500
> #deleting last observation due to missing value
> X=X[-500,]
> dim(X) #499 19
[1] 499 19

```

Figure 5: Missing Value

Thus we have no missing value (except in 'like' and 'share' which will be omitted later) now.

## 6.2 Categorical Variable

In Section 3 we found out some categorical variables but few of them were in integer type. Now convert all the categorical variables into factors.

```

> X$Type=as.factor(X$Type)
> X$Category=as.factor(X$Category)
> X$Paid=as.factor(X$Paid)
> X$Post.Month=as.factor(X$Post.Month)
> X$Post.Hour=as.factor(X$Post.Hour)
> X$Post.Weekday=as.factor(X$Post.Weekday)

```

Figure 6: Converting to Factor

## 6.3 Splitting Data into Train and Test

Now we have 499 observations. For checking the model efficiency we need to divide the data into train data and test data such that we check the performance of our model (built using train data) on test data (prediction on test data). We will split the data into 4:1 ratio that means train data consists of 80% observation and test data will contain 20% observations. We will omit the columns 'like', 'comments' and 'share' from both the data and make 'Lifetime Engaged users' as our response.

```

> #splitting into train and test
> set.seed(197)
> ind=sample(499,499*.8)
> X_train=X[ind,]
> X_test=X[-ind,]
> dim(X_train) #399 19
[1] 399 19
> dim(X_test) #100 19
[1] 100 19
>
> #train set
> y=X_train[,which(colnames(X)=='Lifetime.Engaged.Users')]
> X=X_train[,-c(which(colnames(X)=='Lifetime.Engaged.Users'),which(colnames(X)=='like'),which(colnames(X)=='share'),which(colnames(X)=='comment'))]
>
> #test set
> y_test=X_test[,which(colnames(X_test)=='Lifetime.Engaged.Users')]
> X_test=X_test[,-c(which(colnames(X_test)=='Lifetime.Engaged.Users'),which(colnames(X_test)=='like'),which(colnames(X_test)=='share'),which(colnames(X_test)=='comment'))]

```

Figure 7: Train and Test Set

Now we will apply every Regression Technique on Train data and check goodness of the model on the prediction of Test data.

## 7 Exploratory Data Analysis

Now we again see data summary on the train data after pre-processing. We store the index of categorical variables in '**cat\_col**'.

```

> ##Data summary
> skim(X)
-- Data Summary -----
                           Values
Name                      X
Number of rows            399
Number of columns          15

Column type frequency:
 factor                   6
 numeric                  9

Group variables           None

-- Variable type: factor -----
# A tibble: 6 x 6
  skim_variable n_missing complete_rate ordered n_unique top_counts
* <chr>          <int>        <dbl> <lgl>    <int> <chr>
1 Type              0           1 FALSE      4 Pho: 345, Sta: 32, Lin: 19, Vid: 3
2 Category          0           1 FALSE      3 1: 174, 3: 124, 2: 101
3 Post.Month         0           1 FALSE      12 10: 50, 6: 40, 12: 40, 7: 38
4 Post.Weekday       0           1 FALSE      7 6: 68, 7: 63, 1: 58, 4: 56
5 Post.Hour          0           1 FALSE      21 3: 85, 10: 63, 13: 40, 11: 34
6 Paid               0           1 FALSE      2 0: 301, 1: 98

-- Variable type: numeric -----
# A tibble: 9 x 11
  skim_variable n_missing complete_rate   mean     sd    p0    p25    p50    p75    p100 hist
* <chr>          <int>        <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <dbl> <chr>
1 Page.total.likes  0           1 122879. 16597. 81370 110645 129600 136393 139441 ━━█
2 Total.Reach        0           1 13409. 21235. 238  3308  5206  11944  153536 ━━█
3 Total.Impressions 0           1 29652. 80575. 570  5690. 8893  20964. 1110282 ━━█
4 Consumers          0           1 752.   671.   9   330.   555   942   4754 ━━█
5 Consumptions       0           1 1348.   1872.  9   499   847   1440.  19779 ━━█
6 Impressions.liked 0           1 17447. 66396. 567  3982. 6173  13812  1107833 ━━█
7 Reach.liked        0           1 6441.   7730.  236  2179  3370  7620  48368 ━━█
8 People.engaged.liked 0           1 586.   561.   9   289   408   641   4318 ━━█
9 Total.Interactions 0           1 201.   273.   0   71    126   226   2177 ━━█
> cat_col=c(2,3,4,5,6,7)

```

Figure 8: Data summary

We find that there is no missing values now. There 6 categorical variables and rest are numerical. Now we run separate analysis for categorical and numerical variables.

## 7.1 Analysing Categorical Variable using Box-Plot And frequency table

### 7.1.1 Type

```
> data=cbind.data.frame(X,y)
> colnames(data)[16]="Lifetime.Engaged.Users"
> #box plot and frequency tables for categorical variable
> ggplot(data,aes(x=Type,y=Lifetime.Engaged.Users,fill=Type))+ 
+   geom_boxplot()+scale_fill_brewer(palette = "Dark2") + theme_classic()
> table(X[,2])
```

Type	Link	Photo	Status	Video
Count	19	345	32	3

Figure 9: Code for Analysing Variable 'Type'

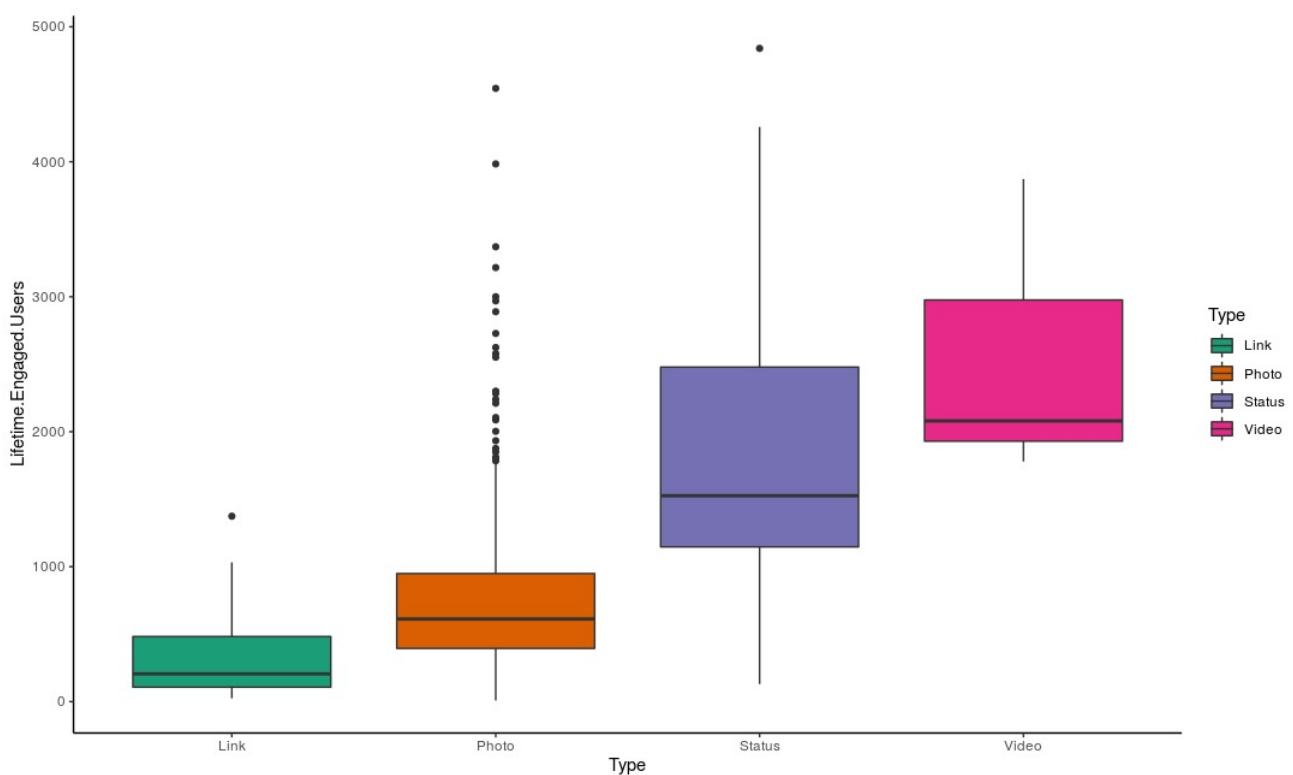


Figure 10: Distribution of Lifetime.Engaged.Users by Type of Post

It has four label namely '**Link**', '**Photo**', '**Status**', '**Video**' Clearly 'Photo' is much more Frequent (345) than other categories. 'Video' has much higher central tendency then comes 'Status' . 'Status' has higher variability and 'Link' has lowest Central tendency and lowest variability with respect to Lifetime.Engaged.Users.All the types here have long tail that implies positive skewness.

### 7.1.2 Category

```
> ggplot(data,aes(x=Category,y=Lifetime.Engaged.Users,fill=Category))+  
+   geom_boxplot()+scale_fill_brewer(palette = "Dark2")+theme_classic()  
> table(X[,3])  
  
1    2    3  
174 101 124
```

Figure 11: Code for analysing Variable 'Category'

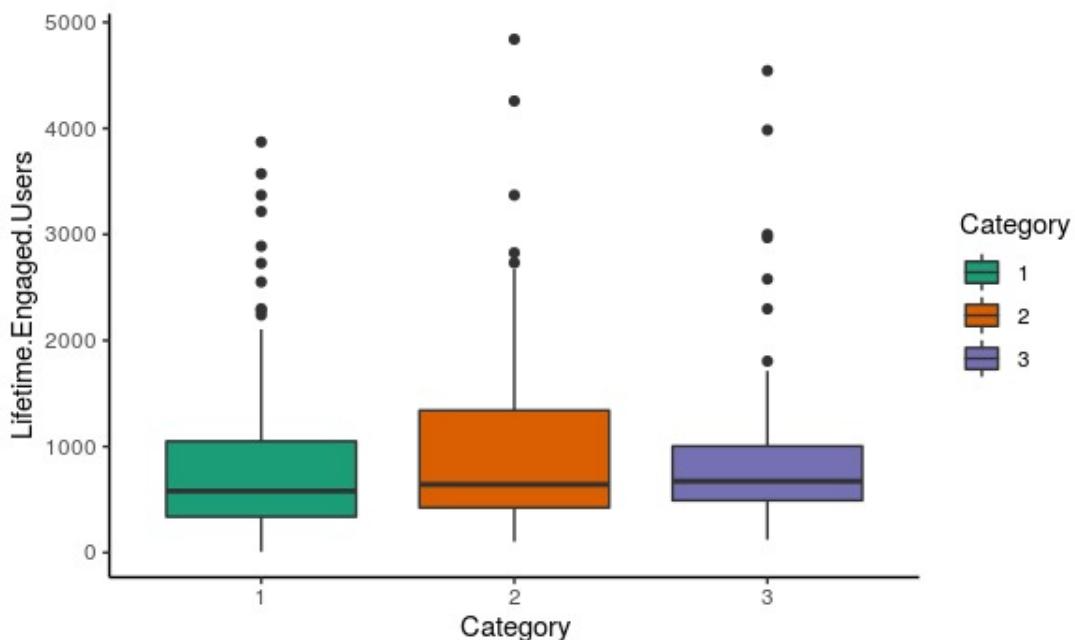


Figure 12: Distribution of Lifetime.Engaged.Users by Category of Post

Variable 'Category' has three label '1','2','3'. Label 1 is most frequent(174). In Fig.12 all three label have almost same central tendency and label 2 has slightly more variability. All three Categories are positively skewed.

### 7.1.3 Paid

```
> ggplot(data,aes(x=Paid,y=Lifetime.Engaged.Users,fill=Paid))+  
+   geom_boxplot()+scale_fill_brewer(palette = "Dark2")+theme_classic()  
> table(X[,7])  
  
0    1  
301  98
```

Figure 13: Code for analysing Variable 'Paid'

Variable 'Paid' has two label '0','1'. Label '0' is most frequent(301). Both labels have almost same central tendency . Both have too much outliers but their variabilities is are not high. Both labels have positive skewness.

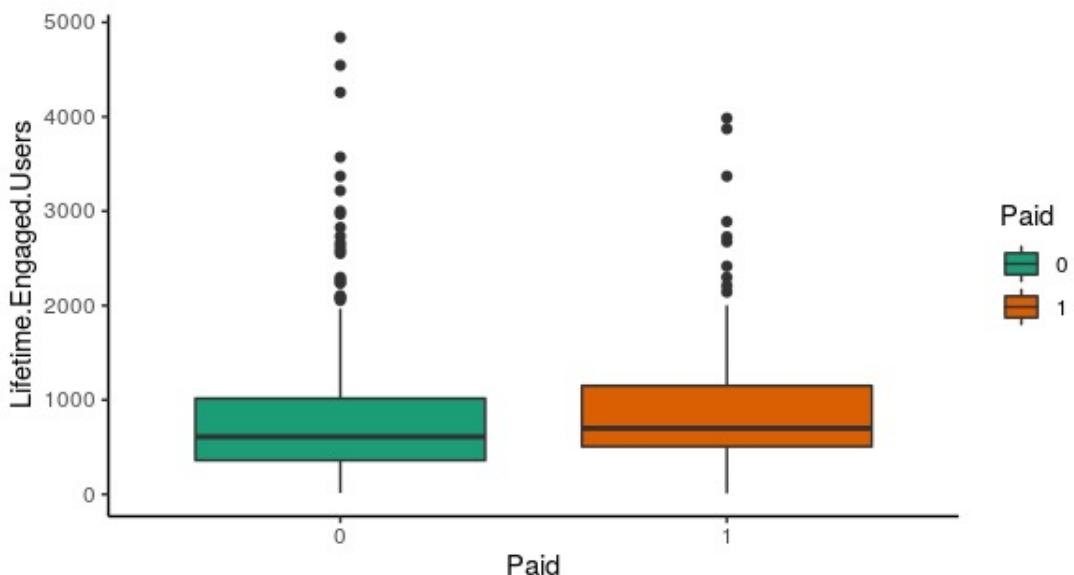


Figure 14: Distribution of Lifetime.Engaged.Users by Paid scenario of Post

#### 7.1.4 Post Month

```
> ggplot(data,aes(x=Post.Month,y=Lifetime.Engaged.Users,fill=Post.Month))+  
+   geom_boxplot()+scale_fill_brewer(palette = "Paired")+theme_classic()  
> table(x[,4])  
  
 1  2  3  4  5  6  7  8  9 10 11 12  
21 24 29 37 29 40 38 26 32 50 33 40
```

Figure 15: Code for analysing Variable 'Post.Month'

Variable Post.Month has 12 label as '1','2'...'12'. Which represent all 12 months of a year. Month October (10th month) is most frequent(50) i.e it has most lifetime engaged users, 'Jun' and 'Dec' are second highest .'Dec' has lowest central tendency where 'Dec' has highest variability and 'May' has lowest variability.

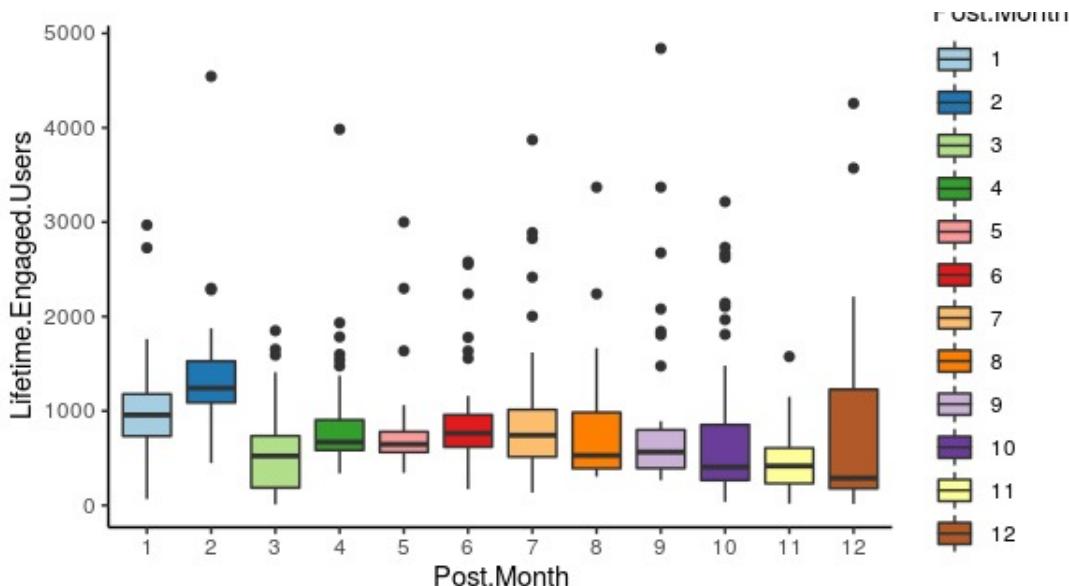


Figure 16: Distribution of Lifetime.Engaged.Users by Post.Month

### 7.1.5 Post Hour

```
> ggplot(data,aes(x=Post.Hour,y=Lifetime.Engaged.Users,fill=Post.Hour))+  
+   geom_boxplot() + theme_classic()  
> table(X[,6])  
  
1  2  3  4  5  6  7  8  9 10 11 12 13 14 15 16 17 18 19 20 22 23  
4 30 85 29 11 14  9  9 25 63 34 21 40 11  4  1  3  3  1  1  1  0
```

Figure 17: Code for analysing Variable Post.Hour

Variable Post.Hour has 23 label as '1','2'...'23'. Which represent hours of a day. Third hour is most frequent(85). Hour 6,10,12,14 have more variability in comparison to others. Hours 3,5,6,7,9,10,13,14 are positively skewed and 1,8,17 are negatively skewed

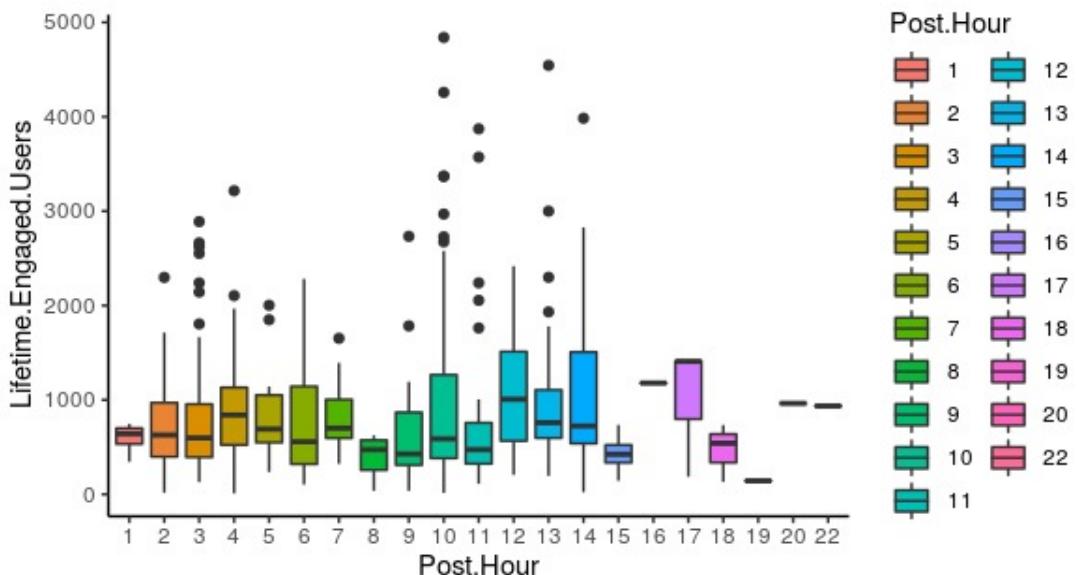


Figure 18: Distribution of Lifetime.Engaged.Users by Post.Hour

### 7.1.6 Post Weekday

```
> ggplot(data,aes(x=Post.Weekday,y=Lifetime.Engaged.Users,fill=Post.Weekday))+  
+   geom_boxplot()+scale_fill_brewer(palette = "Dark2")+theme_classic()  
> table(X[,5])  
  
 1  2  3  4  5  6  7  
58 53 49 56 52 68 63
```

Figure 19: Code for analysing variable Post.Weekday

Variable Post.Weekday has 7 label as '1','2'...'7'. Which represent days of a week. 6th day and 7th day are most frequent(68 and 63 frequency respectively).All days have almost same central tendency .All days have higher number of outliers.

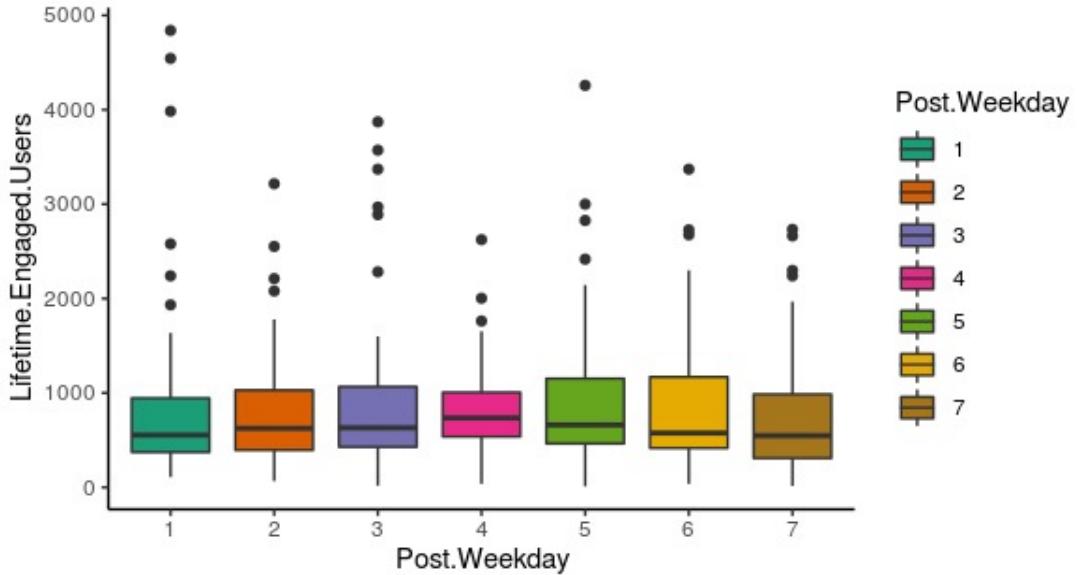


Figure 20: Distribution of Lifetime.Engaged.Users by Post.Weekday

## 7.2 Analysing Numerical Variable using Pair-Plot and correlation matrix

Pairs plot is as follows(using the code `pairs(X[,-cat_col])`):

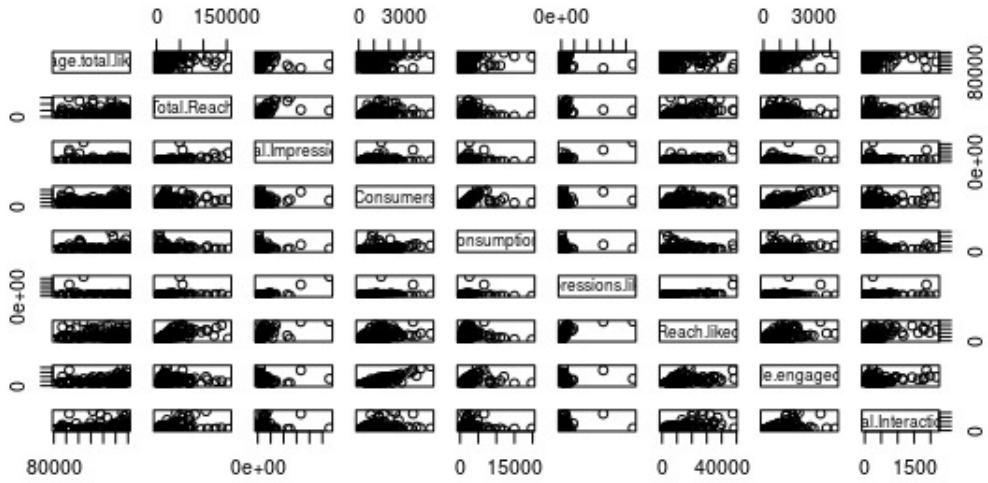


Figure 21: Pair plot among numerical variable

From pairs plot(Fig.21) and correlation matrix(Fig.22) one can infer that there is few strong linear relations among the variables :

1. Lifetime.Post.Total.Reach & Lifetime.Post.Total.Impressions(0.65)
2. Lifetime.Post.Total.Reach & Lifetime.Post.reach.by.people.who.liked.your.page(0.74)
3. Lifetime.Post.Total.Impressions & Lifetime.Post.Impressions.by.people.who.liked.your.page(0.87)
4. Lifetime.Post.reach.by.people.who.liked.your.page & Lifetime.Post.Total.Impressions(0.65),

```

> cor(X[,-cat_col])
      Page.total.likes Total.Reach Total.Impressions Consumers Consumptions Impressions liked
Page.total.likes          1.0000000 -0.07898343 -0.1008282 -0.1810288 -0.1309026 -0.1070115
Total.Reach             -0.07898343  1.00000000  0.6538706  0.4654033  0.2620303  0.3177624
Total.Impressions       -0.10082818  0.65387062  1.0000000  0.3335410  0.2032278  0.8749364
Consumers              -0.18102875  0.46540329  0.3335410  1.0000000  0.5895780  0.2804542
Consumptions            -0.13090257  0.26203031  0.2032278  0.5895780  1.0000000  0.1713110
Impressions.liked      -0.10701146  0.31776237  0.8749364  0.2804542  0.1713110  1.0000000
Reach.liked             -0.08373285  0.73681831  0.6468487  0.5950186  0.3454121  0.5977254
People.engaged.liked   0.01319878  0.38987190  0.3297367  0.9150718  0.5338141  0.3341836
Total.Interactions      0.02616364  0.48429775  0.3276305  0.4182337  0.2289611  0.2914600
                                         Reach.liked People.engaged.liked Total.Interactions
Page.total.likes        -0.08373285          0.01319878        0.02616364
Total.Reach             0.73681831          0.38987190        0.48429775
Total.Impressions       0.64684875          0.32973666        0.32763053
Consumers               0.59501859          0.91507175        0.41823366
Consumptions            0.34541205          0.53381412        0.22896113
Impressions.liked      0.59772542          0.33418363        0.29146004
Reach.liked             1.00000000          0.66565469        0.64449764
People.engaged.liked   0.66565469          1.00000000        0.52950544
Total.Interactions      0.64449764          0.52950544        1.00000000
> cor(X[,-cat_col],y)
      [,1]
Page.total.likes     -0.1404654
Total.Reach          0.5419739
Total.Impressions    0.3746976
Consumers            0.9695748
Consumptions         0.5618013
Impressions.liked   0.3151980
Reach.liked          0.6906750
People.engaged.liked 0.9288043
Total.Interactions   0.6205491

```

Figure 22: Correlation among regressors and with response

5. Lifetime.Post.Consumers &  
Lifetime.People.who.have.liked.your.Page.and.engaged.with.your.post(0.91)
6. Lifetime.Post.reach.by.people.who.liked.your.page &  
Lifetime.People.who.have.liked.your.Page.and.engaged.with.your.post(0.66)

Again From Fig.22 one can infer that response is correlated with:

1. Lifetime.Post.Consumers(0.97)
2. Lifetime.Post.reach.by.people.who.liked.your.page(0.69)
3. Lifetime.People.who.have.liked.your.Page.and.engaged.with.your.post(0.92).

## 8 Changing categorical variable into Indicator variable and Scaling Numerical Quantity.

```
> library(fastDummies)
> length(cat_col)
[1] 6
> X=dummy_cols(X,select_columns = colnames(X)[ cat_col],remove_first_dummy = T,remove_selected_columns = T)
> X_test=dummy_cols(X_test,select_columns = colnames(X_test)[ cat_col],remove_first_dummy = T,remove_selected_columns = T)
> p=(dim(X))[2]
> p      #no. of predictors  53
[1] 53
> n=(dim(X))[1]
> n      #no. of observation  399
[1] 399
```

Figure 23: Code for changing categorical variable into Indicator variable

It is clear from Fig 23 that there are 6 categorical variable (denoted by length of "cat\_col" variable). After changing data into categorical variables we have 53 predictors in which 9 are continuous numerical variables (present in first 9 column of X) and then perform scaling on these variables.

p: no of predictors=53

n: no of observations=399

## 9 Primary Model

Consider our multiple linear regression model as :

$$\mathbf{y} = \mathbf{X}\beta + \epsilon \dots\dots\dots(1)$$

Where  $\mathbf{X}_{n \times p+1} = (\mathbf{1}, x_1, x_2, \dots, x_{53})$  is the non-stochastic matrix of predictors (here  $\mathbf{1}$  is the column whose all elements are 1) and  $\beta = (\beta_0, \beta_1, \dots, \beta_{53})$  is the vector of regression coefficients and  $\epsilon = (\epsilon_1, \epsilon_2, \dots, \epsilon_n)$  is the error vector.

### 9.1 Assumptions

- (i)  $\epsilon_i \stackrel{iid}{\sim} \mathbf{N}(\mathbf{0}, \sigma^2)$   $\forall i$ , where  $\sigma^2$  is an unknown constant.
- (ii)  $\mathbf{X}$  is of full column rank

Let's see our first model and its performance -

```
> my_data=cbind(y,X)
> which(colSums(my_data)==0) #53th column sum is zero
Post.Hour_23
      53
> #exclusion of 53th column
> my_data=my_data[,-53]
> X_test=X_test[,-52]
> model=lm(y~.,data=my_data)
> dim(my_data) #399 53
[1] 399 53
> predictions <- predict(model, X_test)
> MAPE(predictions,y_test) #MAPE 0.04609676
MAPE 0.04609676
```

Figure 24: Model and Prediction

So, performance of this model is quite well as MAPE is only 0.046 or 4.6%

## 10 Diagnostics for Leverage/ Influential Points

### 10.1 Leverage Points

Leverage points are those which have unusual x values but does not affect the model estimates much. Leverage points can be find out with the help of Hat matrix(H), where Hat matrix is given as:

$$\mathbf{H} = \mathbf{X}(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T$$

if  $\mathbf{h}_{ii}$  is ith diagonal element of Hat matrix, then traditionally ith point is considered as leverage if

$$\mathbf{h}_{ii} > \frac{2 * p}{n} \dots\dots\dots (a)$$

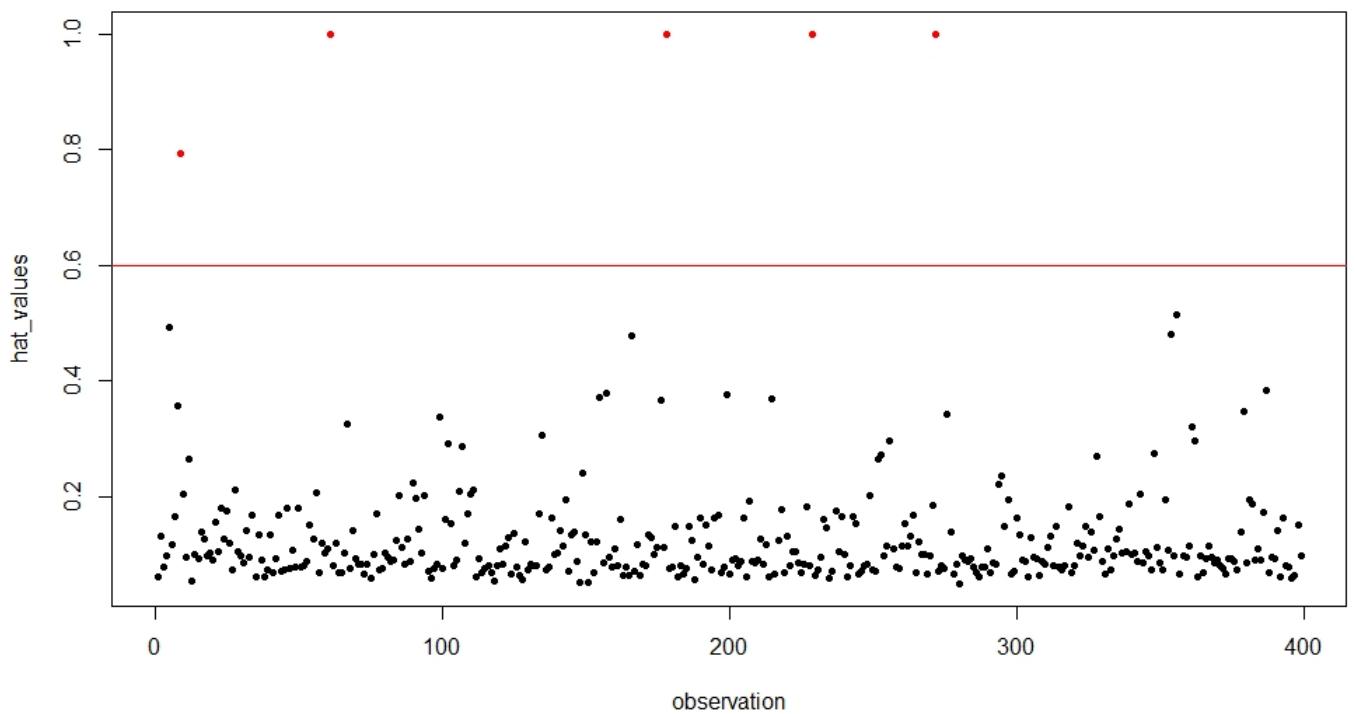


Figure 25: Leverage point detection

If cut-off is considered as given above( which is given in equation (a)) then too many points become leverage,which is clear from fig 22.So to overcome this problem, cut-off is taken to be 0.6, thus yielding 5 leverage points.

```
> #leverage points
> library(olsrr)
> hat_value=hatvalues(model)
> par(mfrow=c(1,1))
> plot(hat_value,xlab ="observation",ylab="hat_values",col=(hat_value>0.6)+1,pch=20)
> abline(h=0.6,col="red")
> lev_point=which(hat_value>0.6) #gives no. of leverage
> length(lev_point) #5
[1] 5
```

Figure 26: Code for leverage point detection

Now we delete the leverage points from the data, thus new data is formed. As there are columns that contain only '0' and '1', so, by deleting rows (or observations) it may happen that few columns is left with '0' only. Then the  $\mathbf{X}$  matrix becomes rank deficient. So, we check those columns and delete them.

Then finally we form a new model without leverage points, then we check its performance.

```

> my_data1=my_data[-lev_point,]
> c=which(colSums(my_data1)==0)
> c
Post.Hour_16 Post.Hour_19 Post.Hour_20 Post.Hour_22
    47        50        51        52
> my_data1=my_data1[,-c]
> X.test=X.test[,-(c-1)]
> row.names(my_data1)=as.character(seq(1,dim(my_data1)[1]))
> model1=lm(y~.,my_data1)
> dim(my_data1)
[1] 394 49
> predictions <- predict(model1, X_test)
> MAPE(predictions,y_test) ##MAPE 0.0477752
MAPE 0.0477752

```

Figure 27: Model without Leverage points

New model performs well with MAPE 4.7% but slightly worse than previous model.

## 10.2 Diagnostics for Influential points

Influential points are those which have both X ad Y unusual. These points have considerable influence on estimates of coefficients and these pull the direction of regression line towards itself.

Our aim is to compute *Cook's distance* ,*DFFITS<sub>i</sub>*, *DFBETAs<sub>ji</sub>*, *CovRatio* and identify the influential point .

### 10.2.1 Cook's Distance

Cook's distance is one of the measure used for identification of Influential Points. Cook's distance is calculated as :

$$\mathbf{C}_i = \frac{(\hat{\beta} - \hat{\beta}_{(i)})^T (\mathbf{X}^T \mathbf{X}) (\hat{\beta} - \hat{\beta}_{(i)})}{pMSres}$$

$$\mathbf{C}_i = \frac{r_i^2 * h_{ii}}{1 - h_{ii}}$$

where, p is number of regressors, n is total no. of observations .  $r_i = \frac{e_i}{\sqrt{MSres(1-h_{ii})}}$  is the internal studentized residual and  $e_i$  is the ith residuals.

It is clear from formula that it contain both part  $r_i$ (studentized residual) and  $h_{ii}$ (ith diagonal element of hat matrix) which measure how far observation is from data and how well the model fits the ith observation  $y_i$

```

> # cooks distance
> cd=cooks.distance(model1)
> plot(cd,xlab="observation",ylab="cooks distance",pch=20)
> cd=cooks.distance(model1)
> plot(cd,xlab="observation",ylab="cooks distance",pch=20,col=(cd>0.1)+1)
> abline(h=0.1,col="red")
> inf_cd=which(cd>0.1)
> length(inf_cd) #5
[1] 5
> ols_plot_cooksd_bar(model1)

```

Figure 28: Code for detection of influential points using cook's distance

Traditionally, the points for which  $C_i$  is greater than 1 is consider influential. But,in this case 0.1 considered as cut-off point and it is clear from Fig.29,30 that 5 points are influential points

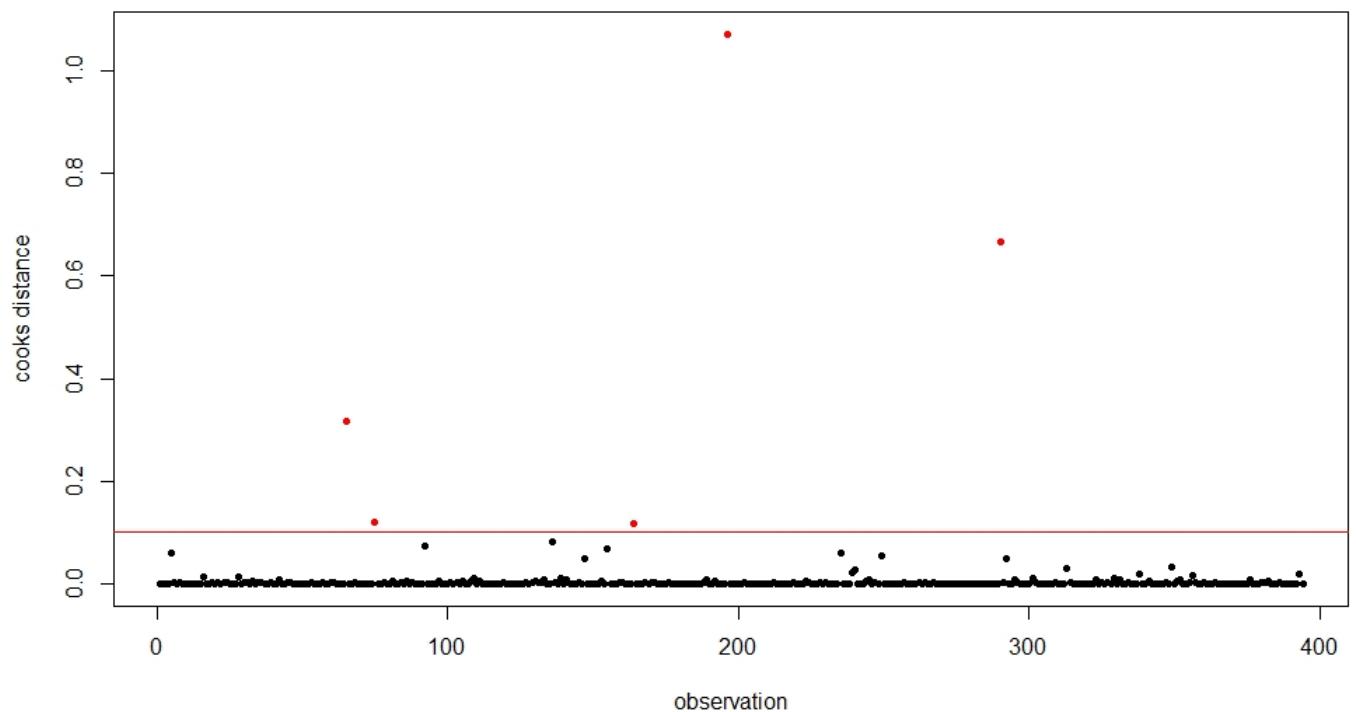


Figure 29: Cook's distance representation when threshold=0.1

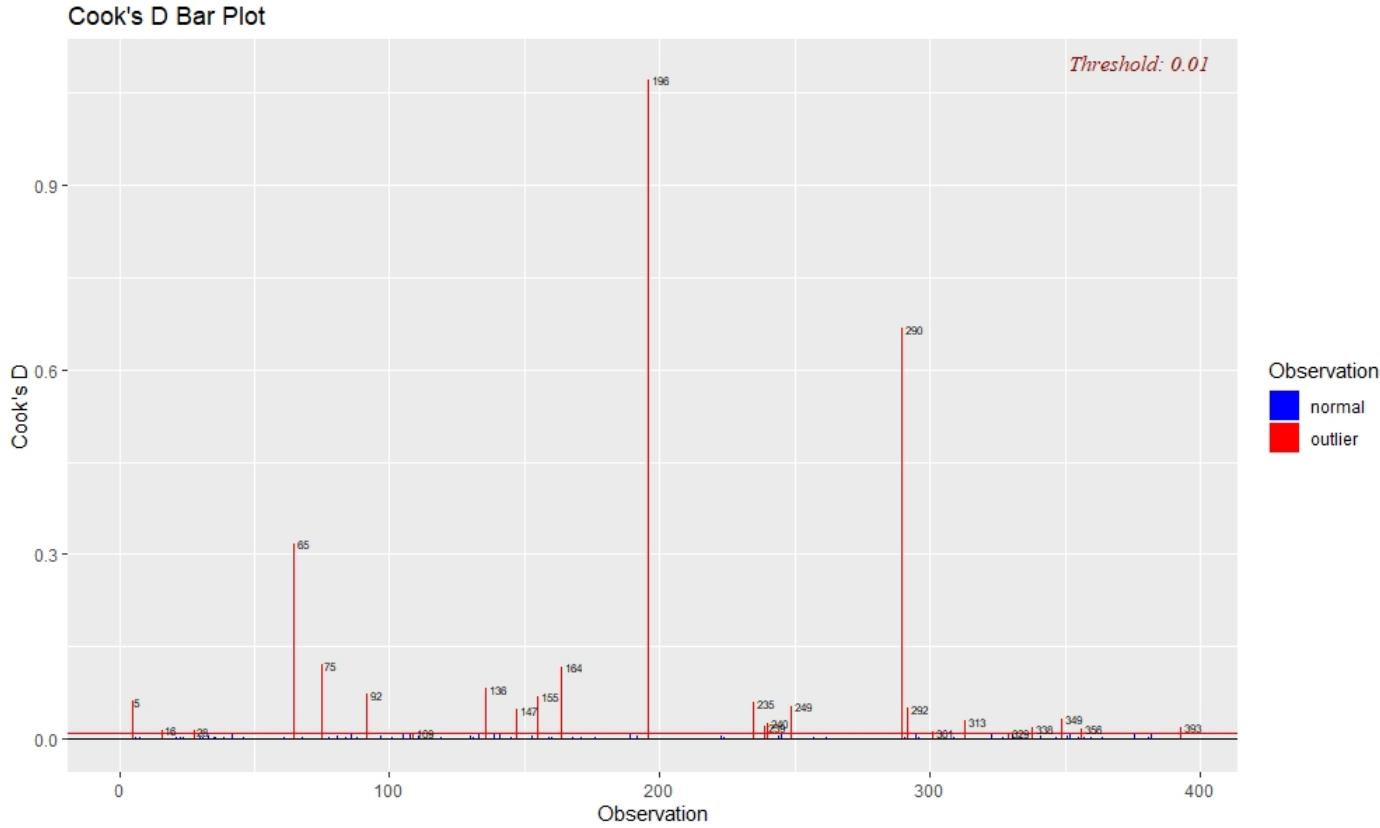


Figure 30: Cook's distance Bar plot when threshold=0.01

### 10.2.2 DFFITS

DFFITS is another method for influential point detection and it also contain both part ( $r_i^{*2}$  and  $h_{ii}$ ) which measure how far observation is from data and how well the model fits the ith observation  $y_i$ .  $DFFITs_i$  is calculated as :

$$DFFITs_i = \sqrt{\frac{r_i^{*2} * h_{ii}}{1 - h_{ii}}}$$

$r_i^* = \frac{e_i}{\sqrt{MSres_i(1-h_{ii})}}$  is the external studentized residual.  $e_i$  is the ith residuals.  $MSres_i =$

$$\frac{\sum_{j=1}^n e_j^2}{(n - 1 - p)}$$

$h_{ii}$  is the ith diagonal element of matrix  $H = \mathbf{X}(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T$

```
> df_fit=dffits(model1)
> plot(1:length(df_fit),df_fit,xlab="observation",ylab="DFFIT",pch=20,col=(abs(df_fit)>2.5)+1)
> abline(h=c(2.5,-2.5),col="red")
> ols_plot_dffits(model)
> influence_DFFIT=which(abs(df_fit)>2.5)
> length(influence_DFFIT) #4
[1] 4
```

Figure 31: Code for calculation of influential points using DFFITS

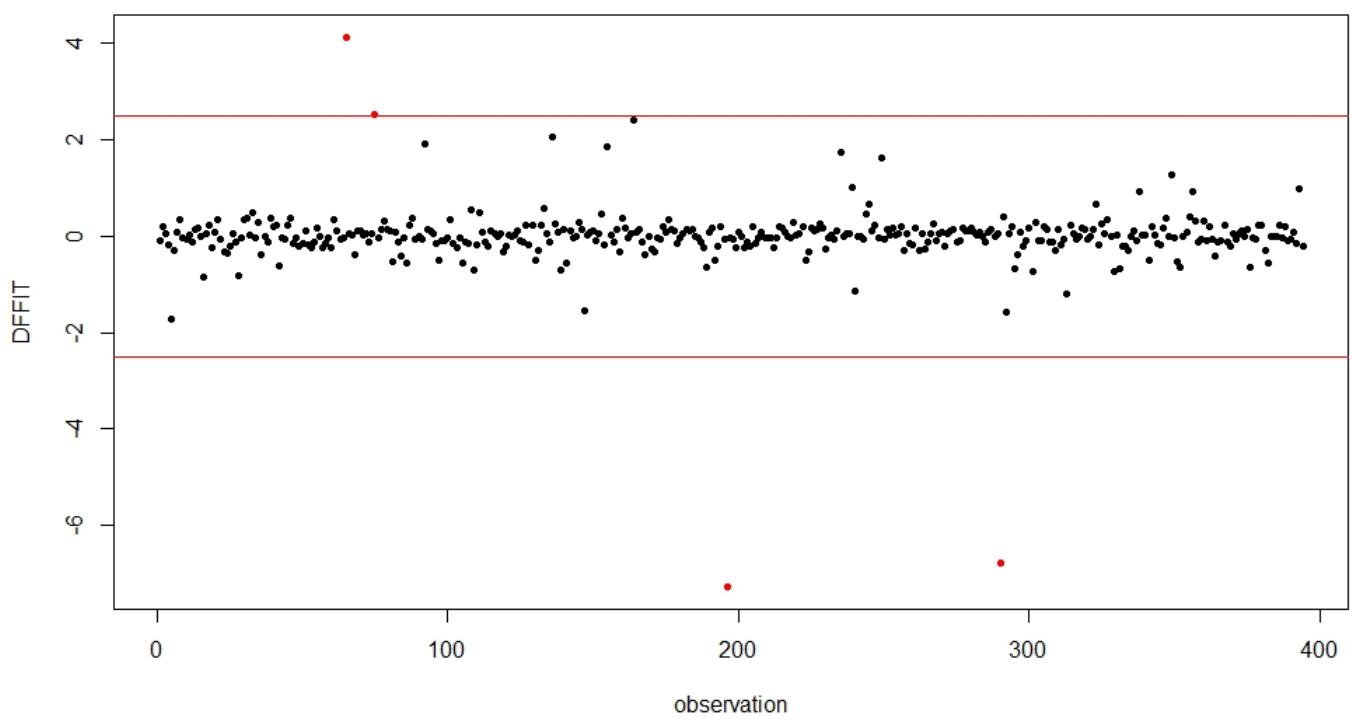


Figure 32: DFFITS representation when threshold=2.5

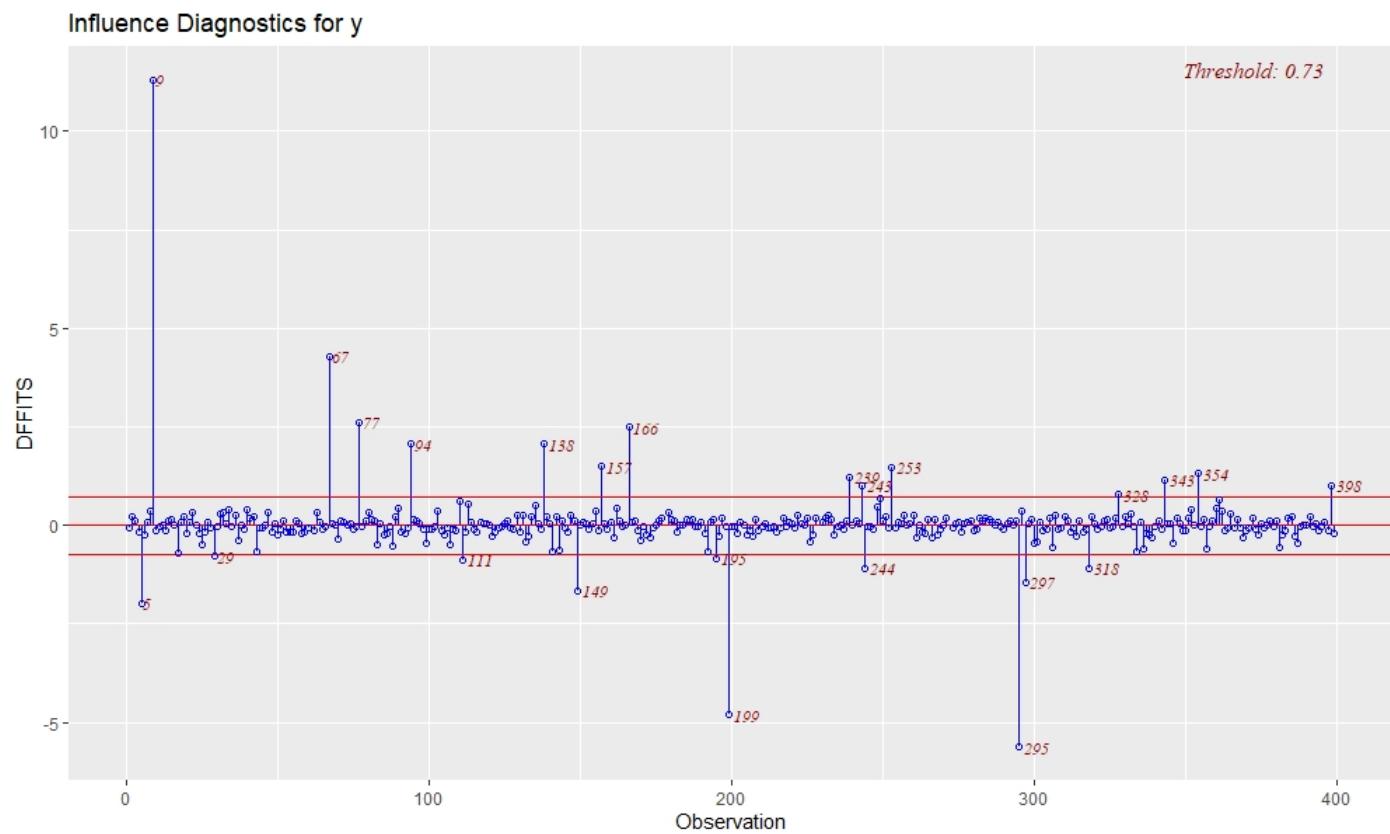


Figure 33: DFFITS representation when threshold=0.73

Traditionally cut-off is given by :

$$|DFITs|_i > \frac{2 * \sqrt{p}}{\sqrt{n}}$$

and if  $DFITs_i$  is greater than this cut-off that indicates that ith sample point is influential point. In this case to avoid too much points to be influential, 2.5 is consider as threshold.

### 10.2.3 DFBETAS

It is also used for influential point detection and it also contain both part ( $r_i^{*2}$  and  $h_{ii}$ ) which measure how far observation is from data and how well the model fits the ith observation  $y_i$ , which is clear from formula of  $DFBETAs_{ji}$   $DFBETAs_{ji}$  is calculated as:

$$DFBETAs_{ji} = \frac{r_i^* * C_j}{\sqrt{h_{jj}(1 - h_{ii})}}$$

$r_i^*$  is the external studentized residual.  $\mathbf{h}_{ii}$  ,  $\mathbf{h}_{jj}$  is the ith,jth diagonal element of matrix  $\mathbf{H}=\mathbf{X}(\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$  respectively and  $C_{ji}$  is the is  $j^{th}$  row of the matrix  $\mathbf{C}=((\mathbf{X}^T\mathbf{X})^{-1}\mathbf{X}^T$

```
> #####DFBETA#####
> DFBETA_ij=dfbetas(model1)
> dim(my_data1) #394 49
[1] 394 49
> influence_DFBETA=c()
> for(j in 2:10)
+   influence_DFBETA=c(influence_DFBETA ,which(abs(DFBETA_ij[,j])>2/sqrt(394)) )
> influence_DFBETA = as.numeric(names(which(table(influence_DFBETA)>=7) ) )
> length(influence_DFBETA) #13
[1] 13
```

Figure 34: Code for calculation of influential points using  $DFBETAs_{ji}$

Generally threshold is given as for this case

$$DFBETAs_{ji} > \frac{2}{\sqrt{n}}$$

And if  $DFBETAs_{ji}$  exceeds this cut off than we say that ith observation is influential. From Fig.34 it is clear that only those points are considered to be influential points which are considered to be influential by 7 or more regressors out of 9 numerical regressors which are tested for DFBETA

### 10.2.4 COVRATIO

The  $Cook's distance$ ,  $DFITs_i$ ,  $DFBETAs_{j,i}$  provides insight about the effect of observation on estimates of coefficients and fitted value  $y$  but they do not provide precision of estimation. COVRATIO also used for this purpose. We express the role of ith observation on precision of estimates in terms of  $COVRATIO_i$ .

We define  $COVRATIO_i$  as

$$\text{covratio}_i = \frac{(MSres_i)^p}{(MSres)^p * (1 - h_{ii})}$$

$MSres_i$  is the estimate of the variance of response variable when  $i$ th data point is removed from the data.  $MSres$  is the estimate of the variance of response variable using all the data.  $\mathbf{h}_{ii}$  is the  $i$ th diagonal element of matrix  $\mathbf{H} = \mathbf{X}(\mathbf{X}^T \mathbf{X})^{-1} \mathbf{X}^T$ .

```
> COVRATIO=covratio(model1)
> par(mfrow=c(1,1))
> plot(COVRATIO,main="Fig 3:COVRATIO for training data Red:influential points and Black :Non
n",
+       col=(abs(COVRATIO-1)>1)+1,cex.main=0.85, pch=20)
> abline(h=c( 2, 0) , col="red", lwd=2, lty="dotted")#check threshold
> influence_COVRATIO=which(abs(COVRATIO-1)>1)
> length(influence_COVRATIO) # 2
[1] 2
```

Figure 35: Code for computation of influential points using COVRATIOS

If  $COVRATIO_i > 1$  then  $i$ th observation improves the precision of estimates and if  $COVRATIO_i < 1$  then  $i$ th observation degrades precision. Traditionally, Cut-off for COVRATIOS is defined as

$$|covratio_i - 1| > \frac{3 * p}{n}$$

There is one influential point using this approach

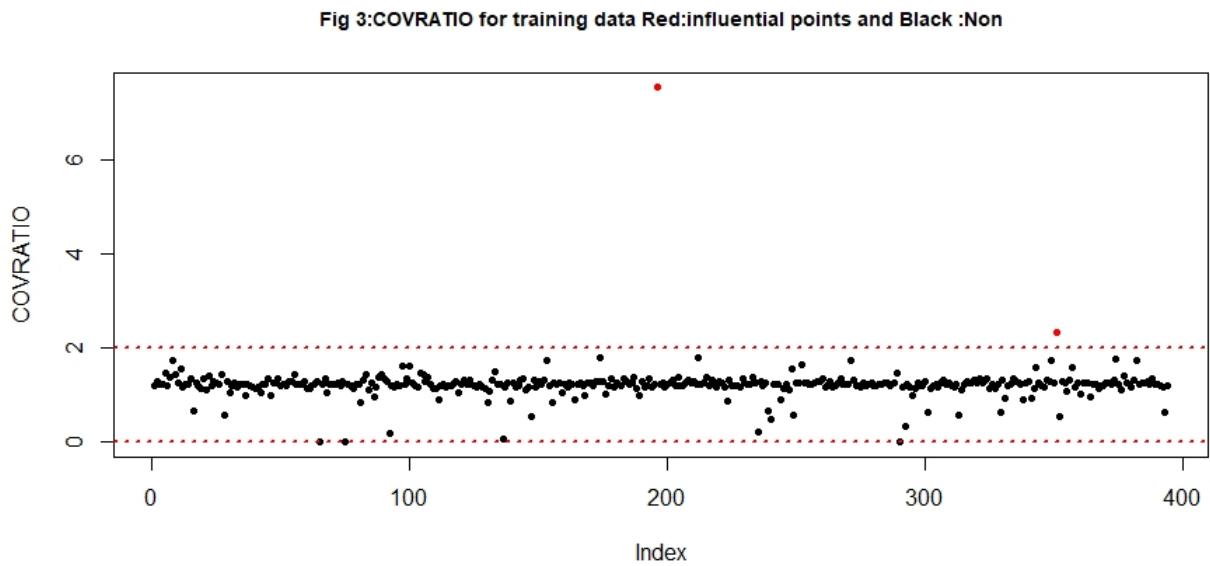


Figure 36: COVRATIO representation

Overall there are 5 influential points (selected by at least 2 methods of finding influentials) and after removing influential points from data we check if there is any columns whose sum is zero and found none. After deleting leverage and influential points we have now 389 observations are left.

```

> #####Final Influential#####
> influential = sort(c(inf_cd,influence_DFBETA,influence_DFFIT,influence_COVRATIO ) )
> final=as.numeric(names(which(table(influential)>=2)))
> length(final) #5
[1] 5
> my_data2=my_data1[-final,]
> which(colSums(my_data2)==0) #0
named integer(0)
> row.names(my_data2)=as.character(seq(1,dim(my_data2)[1]))
> dim(my_data2) #389 49
[1] 389 49

```

Figure 37: Removal of influential points

Now with new data we build a new model and check its performance.

```

> model2=lm(y~.,data=my_data2)
> predictions <- predict(model2, x_test)
> MAPE(predictions,y_test) # MAPE 0.05115962
MAPE 0.05115962
> my_data22=my_data2

```

Figure 38: Model without Outliers

Its MAPE is 5.1% which implies it is very good model but worse than previous ones. That means There is no effect of Outliers in the model w.r.t. to this train and test data.

## 11 Normality Checking

In section 7.1 we assumed  $\epsilon_i \stackrel{iid}{\sim} \mathbf{N}(\mathbf{0}, \sigma^2)$  which is equivalent to assume  $\mathbf{y} \sim \mathbf{N}_n(X\beta, \sigma^2 I_n)$ . Now as  $\mathbf{y}$  is observed quantity, we will check only its Normality.

### 11.1 Checking Through Plot

We will check Normality by Density curve of  $\mathbf{y}$  and by Normal Q-Q Plot. In Density Curve if the curve looks like bell shaped like normal curve , then normal assumption is appropriate. On the other hand in Normal Q-Q Plot order statistics of vector of interest are drawn in the y-axis corresponding to the theoretical order statistics from  $\mathbf{N}(0, 1)$  in the x-axis. Normality assumption can be regarded as true if the points almost in a straight line.

### 11.2 Checking Through Test

Shapiro-Wilk Test is used to test the Normality. Suppose we want to check the normality of the vector  $\mathbf{y} = (y_1, y_2, \dots, y_n)$ . Then the corresponding test statistic is

$$W = \frac{(\sum_{i=1}^n a_i y_{(i)})^2}{\sum_{i=1}^n (y_i - \bar{y})^2}$$

Where  $(a_1, a_2, \dots, a_n) = \frac{m^T V^{-1}}{\|V^{-1}m\|}$ ,  $V$  is the covariance matrix of the order statistic  $y_{(i)}$  and  $m$  is the mean vector of  $y_{(i)}$ .

We check run test using p-value.

### 11.3 Calculations

At first we check that distribution of  $y$  is not normal by following graphs and test:

```
> ##normality checking
> y=my_data[,1]
> library(ggpubr)
> ggdensity(y,fill='blue')+ggtitle('Density Curve')+labs(x='y')+theme_dark()
> ggqqplot(y)+ggtitle('Normal Q-Q Plot')+theme_light()
> shapiro.test(y) #p-value < 2.2e-16 so, it is non normal

Shapiro-Wilk normality test

data: y
W = 0.78424, p-value < 2.2e-16
```

Figure 39: Checking Normality of  $y$

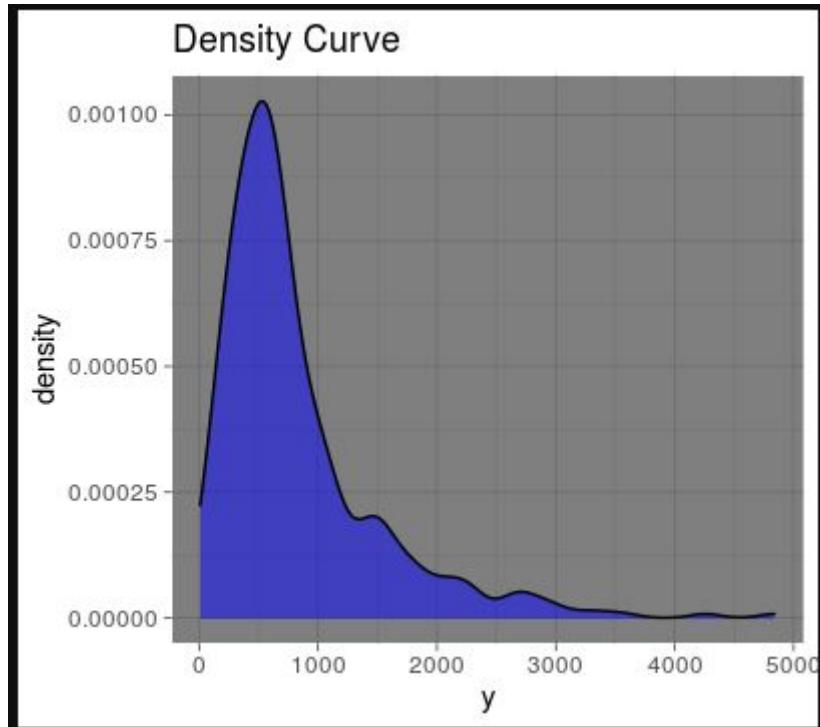


Figure 40: Density Plot

From the test p-value came as less than  $2.2 \times 10^{-16}$  (very very low). So, We need to do transformation of  $y$ . We see that density plot is positively skewed and the Q-Q plot has some kind of convex shape. So, we can try using log transformation.

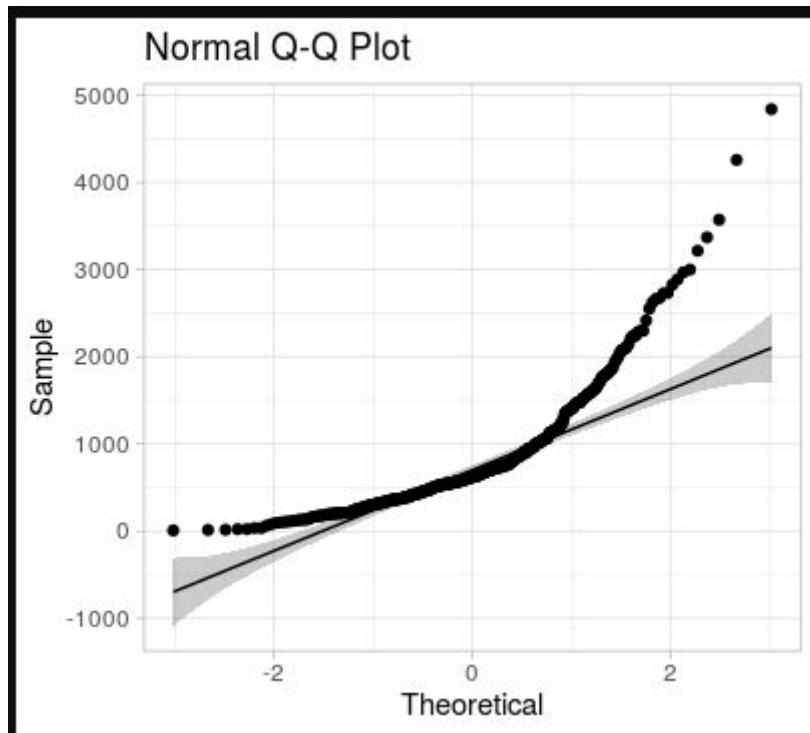


Figure 41: Q-Q Plot

```

> #transformation
> z=log(y)
> ggqqplot(z)+ggtitle('Normal Q-Q Plot')+theme_light()
> shapiro.test(z)  #p-value = 1.28e-10 so, it is non normal

Shapiro-Wilk normality test

data: z
W = 0.94671, p-value = 1.28e-10

```

Figure 42: Transformation From  $y$  to  $\log(y)$

Here we see that the Q-Q plot is almost appropriate except for the first few points. But Shapiro test tells that still it is non normal as the  $p\text{-value} = 1.28 \times 10^{-16}$  is very small. Now we will check excluding some lower order statistic in the begining.

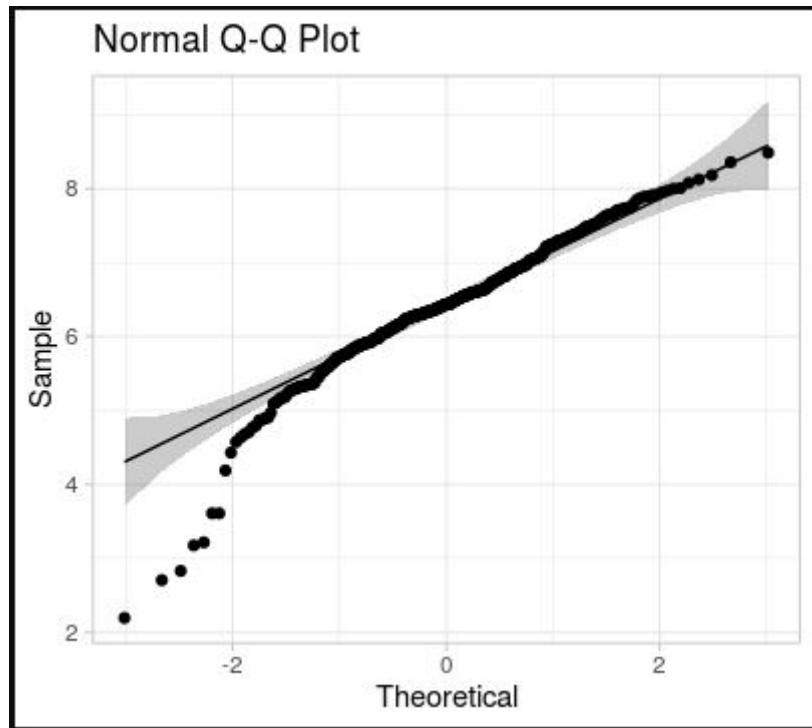


Figure 43: Q-Q Plot for  $\log(y)$

```

> #excluding first 6 points
> z1=sort(z)[7:389]
> ggqqplot(z1)+ggtitle('Normal Q-Q Plot')+theme_light()
> ggdensity(z1,fill='blue')+ggtitle('Density Curve')+theme_dark()
> shapiro.test(z1) #p-value = 0.1075 so, it is normal

Shapiro-Wilk normality test

data: z1
W = 0.99363, p-value = 0.1075

> del_point=which(z<=sort(z)[6])
\ |

```

Figure 44: Excluding first 6 points

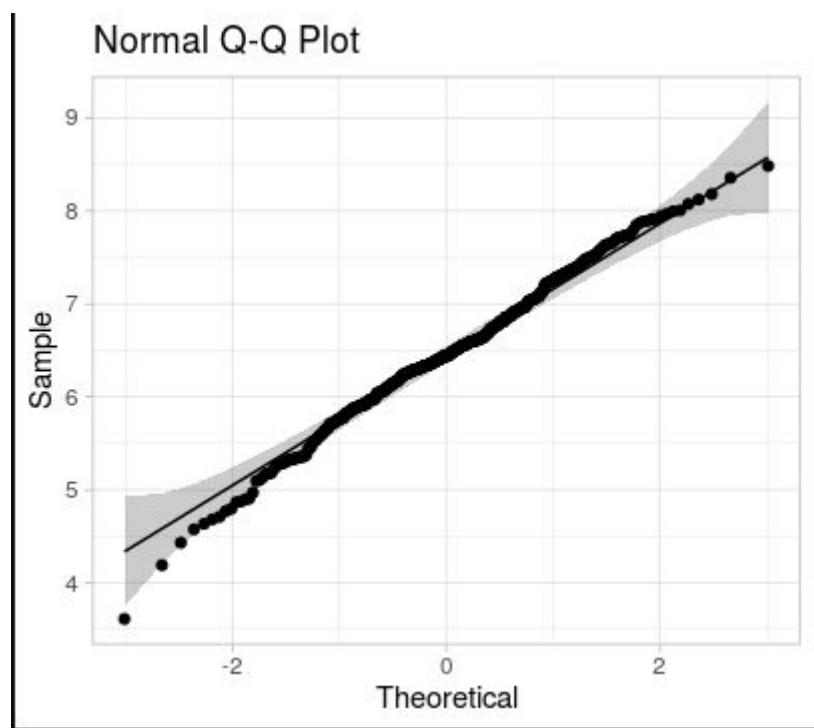


Figure 45: Q-Q Plot for  $\log(y)$

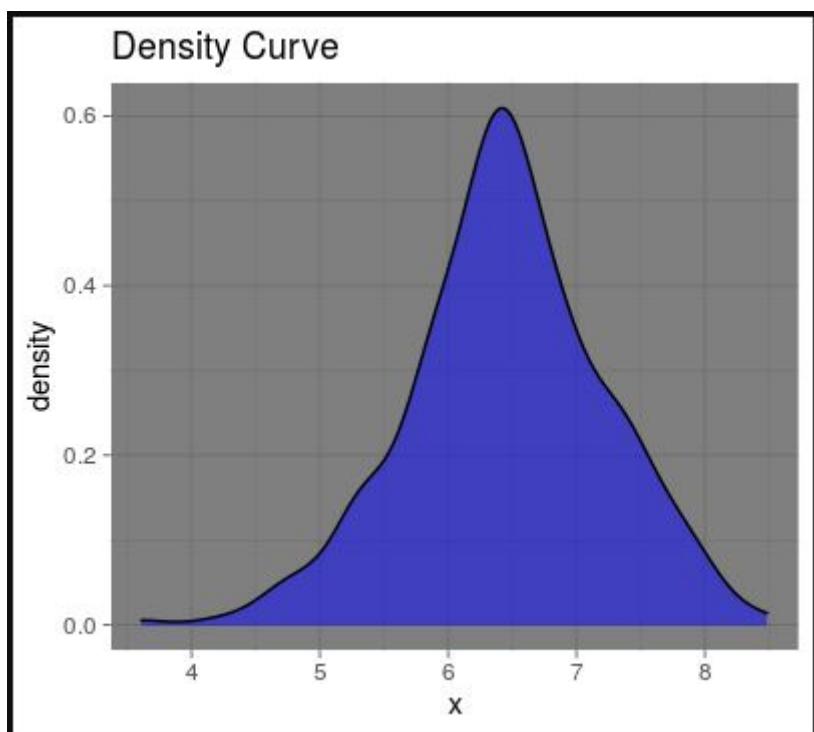


Figure 46: Density Plot for  $\log(y)$

Here we see an impressive result through the graphs and Shapiro-Wilk Test. P-value is  $0.1075 > 0.05$ . So, we can assume normality of  $\log(y)$  at 5% level of significance. Thus our new response variable becomes  $z = \log(y)$  and number of observations reduce to  $(389-6) = 383$ . But instead of 383 we have 382 observation as we have removed 7 observation because of a common observation i.e. 5 distinct and 2 observation with same value that is  $z[217] = 3.610318 = z[368]$ .

```
> #new data with transformed response
> my_data2[,1]=z
> my_data2=my_data2[-del_point,]
> colnames(my_data2)[1]='z'
> dim(my_data2)  #382  49
[1] 382 49
> model3=lm(z~.,data=my_data2) #Our new or transformed model
> predictions <- predict(model3, X_test)
> MAPE(exp(predictions),y_test) # MAPE 8.761486
MAPE 8.761486
```

Figure 47: New Response

Now with new data (transformed response), new model is built and 876% MAPE is found which implies this model's prediction power is very bad. We should look for betterment of it.

## 11.4 New Model

Now we assume our new model as:

$$\mathbf{z} = \mathbf{X}\beta + \epsilon \dots \dots \dots \quad (2)$$

Where  $\mathbf{X}, \beta, \epsilon$  are same as in (1). And assumptions are also same as in 7.1.

## 12 Heteroscedasticity

One of the assumption of our regression model is that our data should be homoscedastic. Consider the regression equation  $y_i = \beta x_i + \epsilon_i$ .  $i=1(1)n$   
In this model, if  $\epsilon_i$  has constant variance say  $\sigma^2$  than data is said to be homoscedastic and if variance of  $\epsilon_i$  depends on  $i$  then data is said to be heteroscedastic and it is said that heteroscedasticity is present in data.

### 12.1 Heteroscedasticity Diagnostics

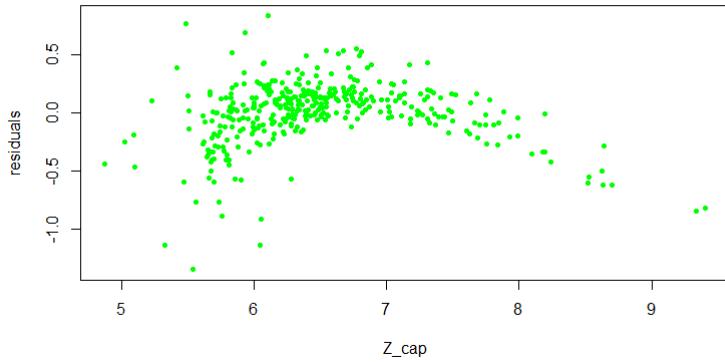


Figure 48: Graph between predicted value and residuals

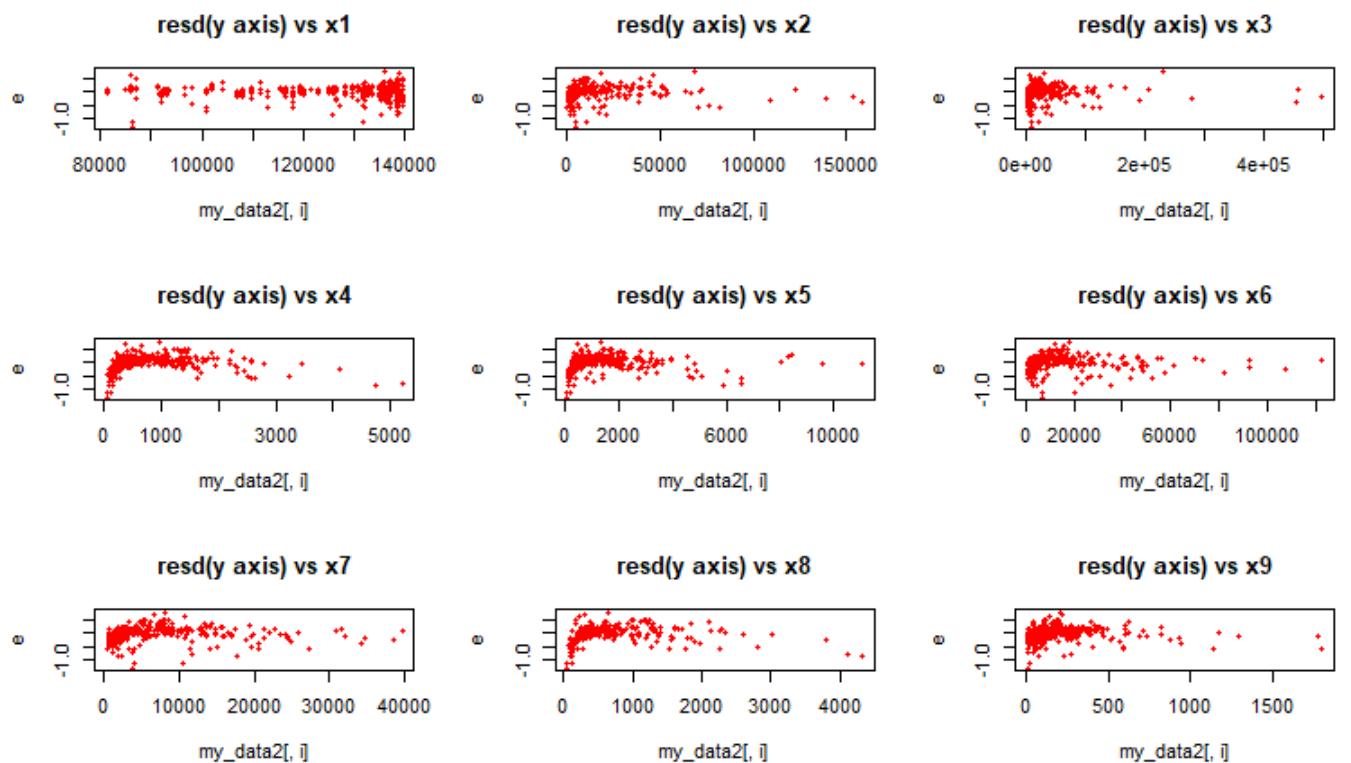


Figure 49: Graph between regressors and residuals

Graph between regressors and residuals have pattern and the graph between residuals and fitted values has also pattern (concave shape) which is indication of heteroscedasticity.

For confirming Heteroscedasticity, we also perform Breusch Pagon test (BP test). This is the test which is used for testing heteroscedasticity. After performing this test, **p value comes out 3.147e-05 which is very low**. This confirm that heteroscedasticity is present in model.

```

> #heteroscedasticity
> library(lmtest)
> bptest(model3)  #p-value =3.147e-05 implies hetero
      studentized Breusch-Pagan test

data: model3
BP = 97.545, df = 48, p-value = 3.147e-05

> z_cap=model3$fitted.values
> e=model3$residuals
> #we check through graph
> plot(z_cap,e,pch=20,col='green')

```

Figure 50: BP TEST

## 12.2 Removal of Heteroscedasticity

```

> par(mfrow=c(1,1))
> #we need to transform x
> my_data4=my_data2
> my_data4[,2:10]=log(my_data4[,2:10])
> model4=lm(z~.,data=my_data4)
> z_fit=model4$fitted.values
> resid=model4$residuals
> plot(z_fit,resid,pch=20,col='green')
> par(mfrow=c(3,3))
> for(i in 2:10)
+   plot(my_data4[,i],resid,pch=20,col='red',main = sub("z","resid(y axis) vs x",i-1))

```

Figure 51: transformation of data

For removal of Heteroscedasticity, We transformed the data. For doing so, all numerical variable of X transformed by log.(response has already transformed into log).

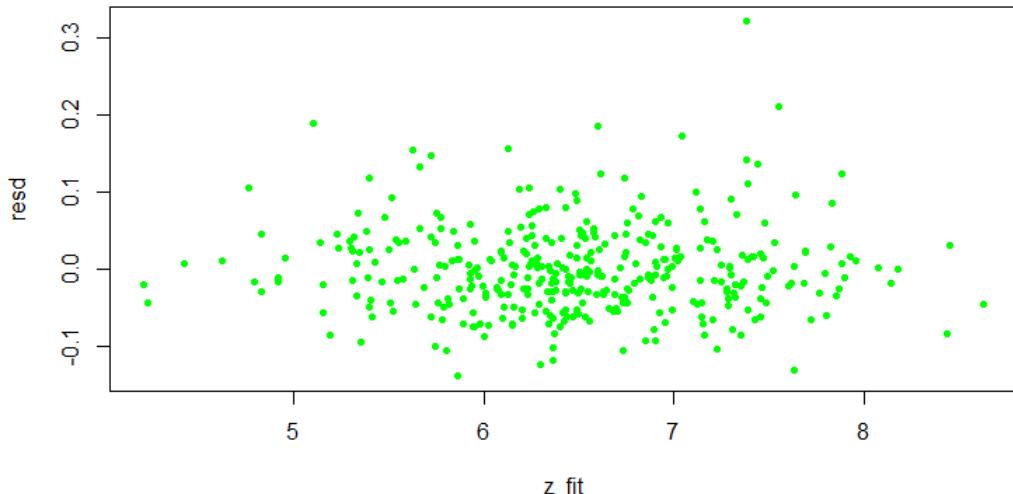


Figure 52: graph between predicted value and residuals

From these graphs it can be seen that there is no pattern in graph between predicted value and residuals and graph between regressors and predictors. This indicate that heteroscedasticity is not significant now.

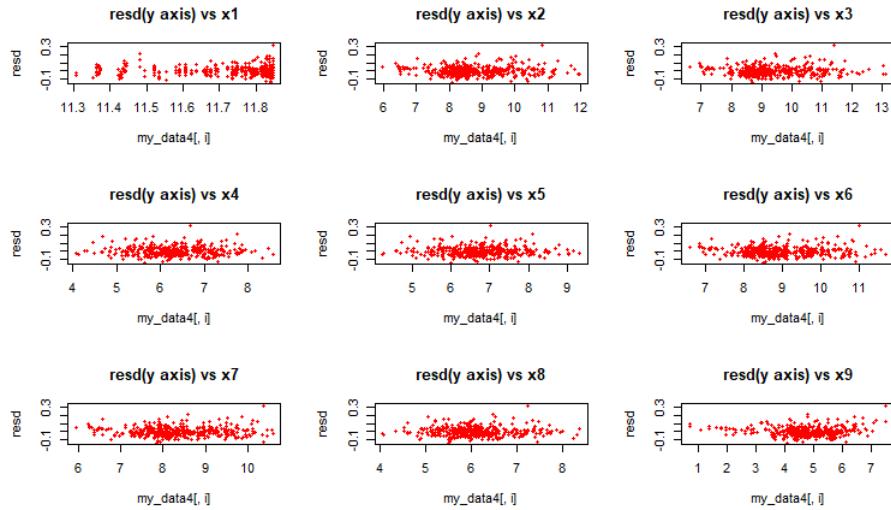


Figure 53: graph between regressors and residuals

Now we have performed Breusch Pagan test

```
> bptest(model4) #p_value=0.02235
studentized Breusch-Pagan test
data: model4
BP = 69.615, df = 48, p-value = 0.02235
```

Figure 54: Breusch Pagan test for transformed data

On applying Breusch Pagan test it can be seen that p value is 0.02235 which is more than 0.02. So heteroscedasticity is not significant at 0.02 level of significant.

So now it is clear that after removing heteroscedasticity, MAPE(mean absolute per-

```
> X_test[,1:9]=log(X_test[,1:9])
> X.test[,1:9]=log(X.test[,1:9])
> predictions <- predict(model4, X_test)
> MAPE(exp(predictions),y_test) #
MAPE 0.05810667> summary(model4)
```

Figure 55: MAPE on transformed data after removing heteroscedasticity

centage errore)= 0.05810667 or 5.8% which is quite better than previous model.

## 13 Multicollinearity

One of the assumptions of linear model is that there should be no multicollinearity in the model i.e there should be no high correlation between two or more predictor variable.

### 13.1 Multicollinearity diagnostics

It is detected with the help of

#### 1. Condition Number

$$CN = \frac{\lambda_{max}}{\lambda_{min}}$$

where  $\lambda_{max}$  and  $\lambda_{min}$  are maximum and minimum values of  $X^T X$ . Here, X is matrix containing only numerical values of X(numerical data)

Convention: Condition number  $> 100$  indicates moderate multicollinearity and  $> 1000$  indicates severe multicollinearity.

In **my\_data4**( dataset obtained after removing influential points,normality,heteroscedasticity) Condition number is coming out to be **972.91** which is near to 1000 and it is strong evidence of Multicollinearity.

Note: Only numerical variables are considered in calculation of condition number since dummy variables don't contribute much to multicollinearity

```
> #Computation of X'X
> X_num1=as.numeric(unlist(X_num1))
> X_num1=matrix(X_num1,ncol=9,byrow = FALSE)
> C=t(X_num1)%*%(X_num1) # it is [9,9] matrix
> eigen_C=eigen(C)$values
> k=max(eigen_C)/min(eigen_C)
> |
```

Figure 56: Calculation of Condition Number

#### 2. Variance Inflation Factor

$$VIF_i = \frac{1}{1 - R_i^2}$$

where  $R_i^2$  is  $R^2$  obtained after regressing  $i^{th}$  regressors on remaining other regressors.

Convention: VIF  $> 10$  indicates severe multicollinearity

**my\_data4** VIFs are shown in Fig.57 . It is once again evident from Fig.57 that VIFs of all the numerical variables except Total.Interactions is greater than 10 indicating multicollinearity

### 13.2 Dealing with multicollinearity

This work will consider various approaches for removing multicollinearity as discussed below:

#### 13.2.1 Principal Component Regression

In this instead of regressing on X's, regression is done on Principal Components explaining majority of variability. PCR is done for Total.Reach,Total.Impressions,Impressions.liked, Reach.liked and Consumers,Consumptions and People.engaged.liked separately. The choice of these combination is made from Fig 58.

```

> car::vif(model4)
    Page.total.likes      Total.Reach      Total.Impressions      Consumers
    172.021036          47.219909        36.125087          37.910952
    Consumptions         Impressions.liked   Reach.liked          People.engaged.liked
    11.043999          78.765205        108.921135          37.617345
    Total.Interactions   Type_Photo       Type_Status           Type_Video
    4.556315            4.509644        6.043793          1.297171
    Category_2           Category_3       Post.Month_2          Post.Month_3
    2.625969            2.622575        5.665529          14.956115
    Post.Month_4          Post.Month_5      Post.Month_6          Post.Month_7
    45.778048           59.784956        108.391189         127.630195
    Post.Month_8          Post.Month_9      Post.Month_10         Post.Month_11
    101.668431          129.826652        199.126246         145.417176
    Post.Month_12         Post.Weekday_2    Post.Weekday_3        Post.Weekday_4
    178.919745          1.934550        1.844524          1.933761
    Post.Weekday_5        Post.Weekday_6    Post.Weekday_7        Post.Hour_2
    1.880285             2.071466        1.957206          8.511637
    Post.Hour_3            Post.Hour_4       Post.Hour_5          Post.Hour_6
    18.843143            8.066417        3.863535          4.721545
    Post.Hour_7            Post.Hour_8       Post.Hour_9          Post.Hour_10
    3.080242             3.111540        7.123220          14.499524
    Post.Hour_11           Post.Hour_12      Post.Hour_13         Post.Hour_14
    9.165697              6.427875        10.494133         3.474812
    Post.Hour_15           Post.Hour_17      Post.Hour_18         Paid_1
    2.599175              2.184343        2.341917          1.150080

```

Figure 57: Calculation of VIF

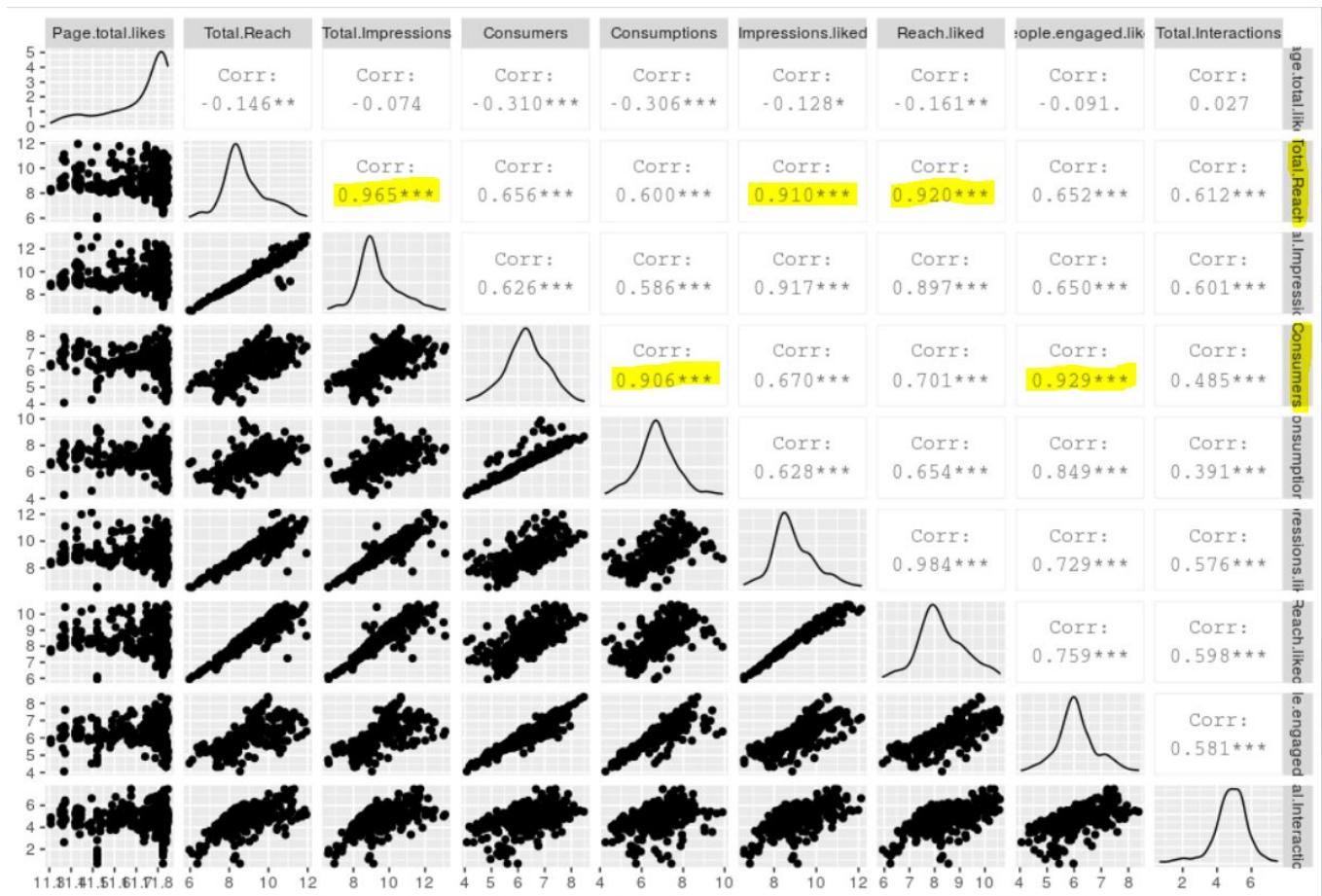


Figure 58: Calculation of VIF

```

> x_new1=my_data4[,c(3,4,7,8)]
> pcr1=prcomp(x_new1,center=T,scale=T)
> plot(pcr1,type='l')
> summary(pcr1)
Importance of components:
              PC1       PC2       PC3       PC4
Standard deviation   1.9484  0.39079  0.20665  0.09092
Proportion of Variance 0.9491  0.03818  0.01068  0.00207
Cumulative Proportion  0.9491  0.98726  0.99793  1.00000
> PC11=pcr1$x[,1]
> PC12=pcr1$x[,2]

```

Figure 59: Calculation of PCR1

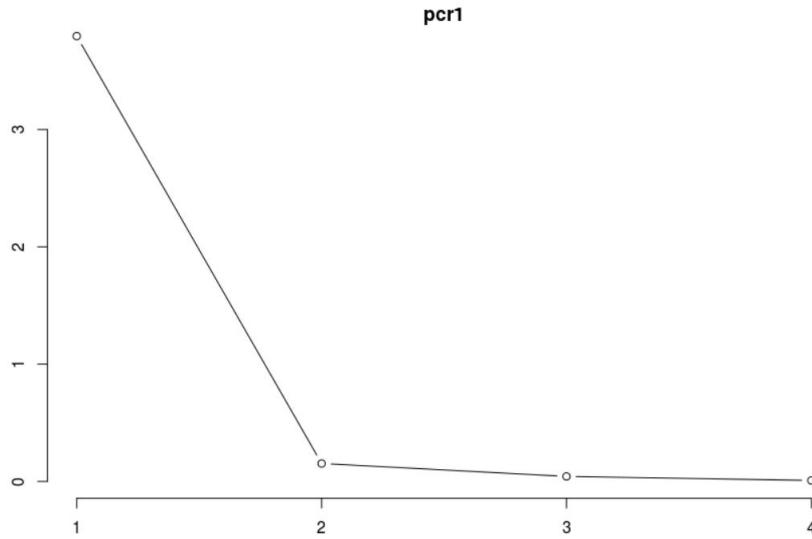


Figure 60: Scree Plot for PCR1

**PCR1**-Principal Component Regression for Total.Reach,Total.Impressions,Impressions.liked, Reach.liked is shown in Fig 59 and 60 indicating majority of variability(98.7%) is explained by first two principal components which will be further used for regression rather than all four Principal Components.

```

> x_new2=my_data4[,c(5,6,9)]
> pcr2=prcomp(x_new2,center=T,scale=T)
> plot(pcr2,type='l')
> summary(pcr2)
Importance of components:
              PC1       PC2       PC3
Standard deviation   1.6705  0.39043  0.23894
Proportion of Variance 0.9302  0.05081  0.01903
Cumulative Proportion  0.9302  0.98097  1.00000
> PC21=pcr2$x[,1]
> PC22=pcr2$x[,2]

```

Figure 61: Calculation of PCR2

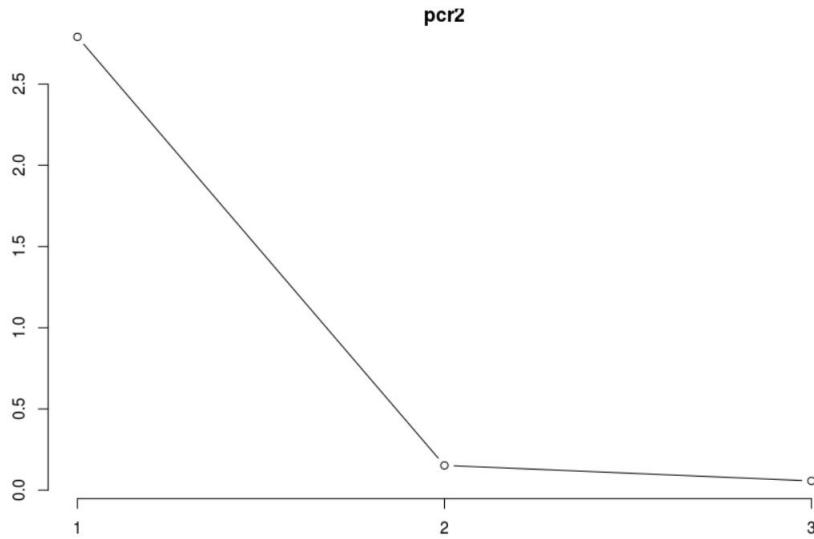


Figure 62: Scree Plot for PCR2

**PCR2**-Principal Component Regression for Consumers,Consumptions and People.engaged.liked is shown in Fig 61 and 62 indicating majority of variability(98.09%) is explained by first two principal components which will be further used for regression rather than all four Principal Components.

### 13.2.2 Variable Elimination

Page.Total.Likes is removed because of low correlation with response(-0.24688) and Large VIF(172.02) (See Fig.57)

Finally model 5 is fitted with following regressors shown in Fig.63 and VIFs are again calculated which are all satisfactorily less than 10 except for 3 dummy variables

```
> cor(my_data4$z,my_data4$Page.total.likes)
[1] -0.2468888
> my_data5=cbind(z=my_data4[,1],PC11,PC12,PC21,PC22,my_data4[,10:49])
> # first variable is excluded due to large vif and low correlation with response
>
> model5=lm(z~.,data=my_data5)
> car::vif(model5)
      PC11          PC12          PC21          PC22  Total.Interactions
4.587446    1.400175   5.277552   2.657922        4.142753
Type_Photo    Type_Status    Type_Video Category_2       Category_3
4.122261    5.859499   1.274570   2.597659        2.582639
Post.Month_2   Post.Month_3  Post.Month_4 Post.Month_5     Post.Month_6
2.376707    3.097954   3.170204   2.570805        3.064728
Post.Month_7   Post.Month_8  Post.Month_9 Post.Month_10    Post.Month_11
3.229145    2.718401   3.144987   4.012563        3.836043
Post.Month_12  Post.Weekday_2 Post.Weekday_3 Post.Weekday_4 Post.Weekday_5
3.855943    1.930078   1.795166   1.906234        1.856982
Post.Weekday_6 Post.Weekday_7  Post.Hour_2  Post.Hour_3     Post.Hour_4
2.036323    1.942236   8.353164   18.671737       8.007098
Post.Hour_5    Post.Hour_6  Post.Hour_7  Post.Hour_8     Post.Hour_9
3.833700    4.692999   3.055828   3.081153       7.043367
Post.Hour_10   Post.Hour_11 Post.Hour_12 Post.Hour_13    Post.Hour_14
14.434657   9.096026   6.363029  10.435312       3.404022
Post.Hour_15   Post.Hour_17 Post.Hour_18  Paid_1
2.419778    2.145530   2.247199  1.145449
```

Figure 63: VIFs for fitted model with Principal components and after Variable elimination

When model 5 is applied on test data it shows a MAPE of 103.9799 which clearly

show that Principal Component of numerical variables is not a fruitful approach always.Sometimes using it may create a mess as in Figure 64.

```
> #loadings used for prediction of test data
> loading1=pcr1$rotation
> loading2=pcr2$rotation
> X_test_pc1=as.matrix(X_test[,c(2,3,6,7)])%*%loading1
> X_test_pc2=as.matrix(X_test[,c(4,5,8)])%*%loading2
> X_test_pc=(as.data.frame(cbind(X_test_pc1[,1:2],X_test_pc2[,1:2],X_test[,9:53])))
> colnames(X_test_pc)[1:4]=c("PC11","PC12","PC21","PC22")
> predictions <- predict(model5, X_test_pc)
> MAPE(exp(predictions),y_test) #
MAPE 103.9799
```

Figure 64: MAPE calculation on test data for fitted model with Principal components and after Variable elimination

Still an attempt is made to improve performance of model 5 by stepwise regression of variables which is a combination of forward selection and backward elimination starting with Backward elimination in this case since full model 5 is being considered.This approach gives model shown in fig 65 with a MAPE of 96.976.So, this model won't be used for analysis since results are not as expected.

```
> model6=lm(formula = z ~ PC12 + PC21 + PC22 + Total.Interactions + Type_Status +
+           Post.Month_2 + Post.Month_3 + Post.Month_4 + Post.Month_5 +
+           Post.Month_6 + Post.Month_7 + Post.Month_8 + Post.Month_9 +
+           Post.Month_10 + Post.Month_11 + Post.Month_12 + Post.Weekday_5 +
+           Post.Hour_2 + Post.Hour_5 + Post.Hour_6 + Post.Hour_8 + Post.Hour_11 +
+           Post.Hour_14 + Post.Hour_15 + Post.Hour_12, data = my_data5)
> predictions <- predict(model6, X_test_pc)
> MAPE(exp(predictions),y_test) #
MAPE 96.97676
```

Figure 65: MAPE calculation on test data for fitted model with Principal components and after Variable elimination and after Stepwise regression

### 13.2.3 Principal Component Regression(PCR) on entire dataset

The principal component regression approach involves constructing the first M principal components  $Z_1, Z_2, \dots, Z_M$  and then using these components as the predictors in a linear regression model that is fit using least squares.The key idea is that often a small number of principal components suffice to explain most of the variability in the data as well as the relationship with the response.In other words, *we assume that the direction in which  $X_1, X_2, \dots, X_p$  show the most variation are the directions that are associated with Y*

PCR is performed on entire dataset and validation="CV" causes pcr() to compute the ten-fold cross-validation error for each possible value of M.Fig 66 shows validation plot for MSE vs no.of components.By looking at Fig.66, 38 principal components are opted which are explaining 98.65% variability of X and 97.78% variability of Z(Use summary(pcr.fit) to get these numbers)

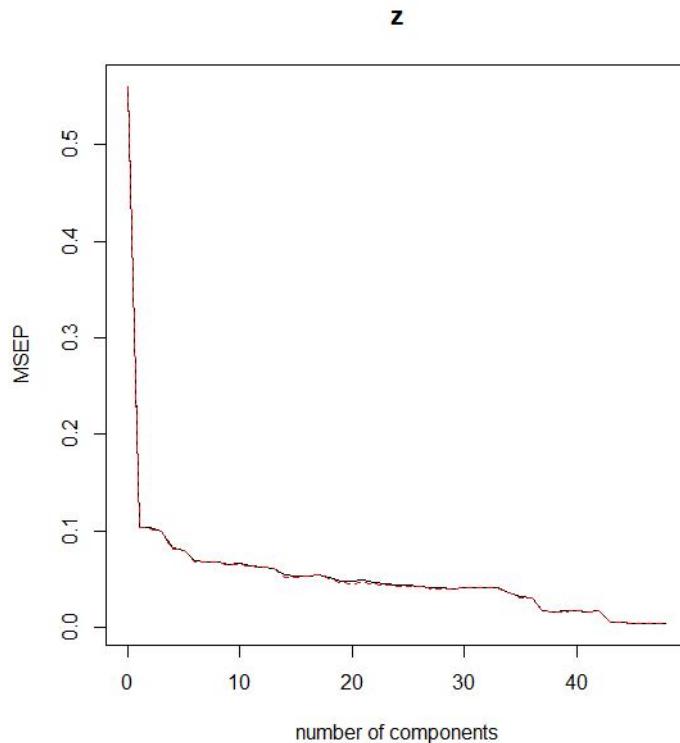


Figure 66: Cross validation MSE v/s number of components using PCR approach

```
> #pcr
> library(pcr)
> pcr.fit=pcr(z~,data=my_data4,scale=T,validation='CV')
> validationplot(pcr.fit,val.type = 'MSEP')
> pcr.pred=predict(pcr.fit,X.test,ncomp = 38 )   # change ncomp accordingly
> MAPE(exp(pcr.pred),y_test)
MAPE 0.1178884
```

Figure 67: MAPE on test data when using PCR

When PCR is tested on test data it provides a MAPE of 0.1178 which is pretty good and this model can be used for analysis. However, as a result of the way PCR is implemented ,the final model is more difficult to interpret because it does not perform any kind of variable selection or even directly produce coefficient estimates.

#### 13.2.4 Partial Least Square Regression (PLSR)on entire dataset

PCR suffers from a drawback: there is no guarantee that the direction that best explains the predictors will also be the best directions to use for predicting the response.PLS identifies the new feature in a supervised way-that is it makes use of response Y in order to identify new features that not only approximate old features well,but also that are related to response.PLS places the highest weight on the variable that are most strongly related to the response

PLS is performed on entire dataset and validation="CV" causes pls() to compute the ten-fold cross-validation error for each possible value of M.Fig.68 shows validation plot for MSE vs no.of components.By looking at Fig.68,10 components are opted which are explaining 37.68% variability of X and 99.37% variability of Z(Use summary(pls.fit) to get these numbers)

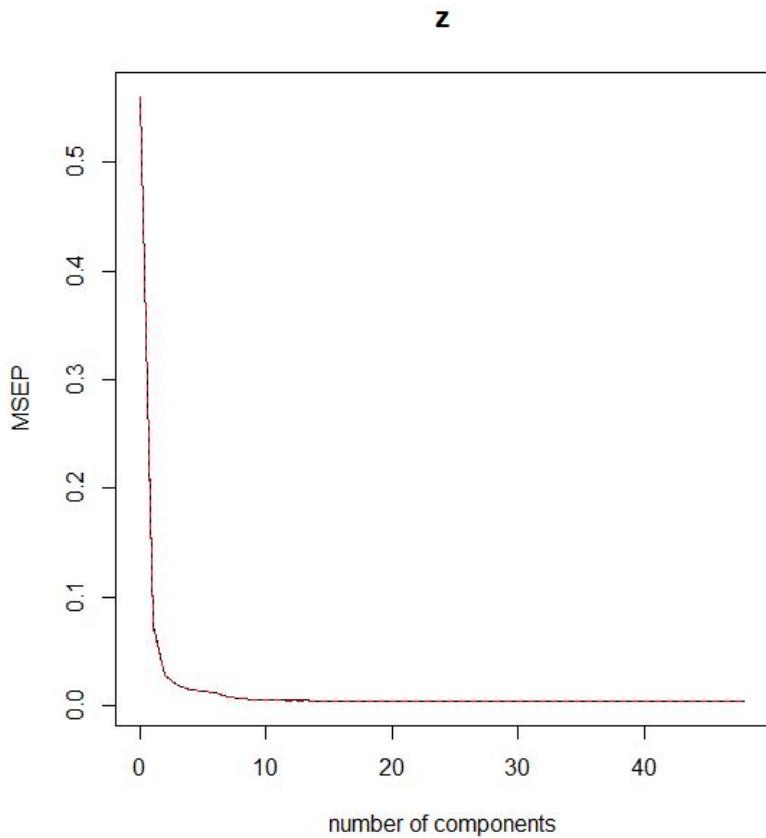


Figure 68: Cross validation MSE v/s number of components using PLS approach

```
> library(pls)
> # pls
> pls.fit=pls(z~.,data=my_data4,scale=T,validation='CV')
> validationplot(pls.fit,val.type = 'MSEP')
> pls.pred=predict(pls.fit,x.test,ncomp = 10 ) # change ncomp accordingly
> MAPE(exp(pls.pred),y_test)
MAPE 0.06176425
```

Figure 69: MAPE on test data when using PLS

When PLS is tested on test data it provides a MAPE of 0.0617 which is lower than MAPE of PCR so this model can also be used for analysis. However, as a result of the way PLS is implemented ,the final model is more difficult to interpret because it does not perform any kind of variable selection or even directly produce coefficient estimates.

### 13.2.5 The Lasso

Lasso is a techniques in which we fit a model containing all p predictors and use the techniques regularise the some of coefficients of estimates or shrinks the estimates towards zero. Shrinking of the coefficient estimates reduces their variance.

**Consider the equation**

$$L = \sum_{i=1}^n (y_i - \beta_0 - \sum_{j=1}^p \beta_j x_{ij})^2 + \lambda \sum_{j=1}^p |\beta_j|$$

The coefficients obtained by minimising this equation is called lasso coefficients is denoted by  $\hat{\beta}_\lambda^L$ . In this equation first is RSS and second term is known as **LASSO PENALTY** and  $\lambda$  is tuning parameter.

As  $\lambda$  increases coefficients of estimates shrinks towards zero and as  $\lambda$  become sufficient large some of the coefficients become zero. So we can say Lasso also performs variable selection that is model contain a subset of the variable.

```
> #Lasso to remove multicollinearity + variable selection
> set.seed(197)
> library(glmnet)
> xtrain=as.matrix(my_data4[,-1])
> ytrain=as.vector(my_data4[,1])
> cv.out=cv.glmnet(xtrain,ytrain,type.measure = "mse",
+                     alpha=1, family="gaussian")
> a=coef(cv.out)[,1]
> length(a[a!=0]) #17 non zero coefficients
[1] 17
> selected_variable=names(a[a!=0])
> selected_variable
[1] "(Intercept)"      "Total.Reach"        "Consumers"          "People.engaged.liked"
[5] "Total.Interactions" "Type_Photo"         "Category_3"         "Post.Month_2"
[9] "Post.Month_4"       "Post.Month_5"        "Post.Month_7"        "Post.Month_11"
[13] "Post.Month_12"      "Post.Weekday_5"     "Post.Hour_2"         "Post.Hour_9"
[17] "Post.Hour_14"
> |
```

Figure 70: LASSO using CROSS VALIDATION

Here LASSO is performed on training data set using CROSS VALIDATION and we have the model corresponding to minimum MSE ( `cv.glmnet()` and `alpha=1` indicate we have performed lasso using cross validation). In this case tuning parameter  $\lambda$  is 10 (by default). It is known that as  $\lambda$  increases coefficient estimates shrinks towards zero. From the fig70. It is clear that 17 coefficients estimates of predictors are non zero and rest 32 are shrinks to zero. So these predictor having non zero coefficients estimates are significant and rest are insignificant ( when we use  $\lambda=10$ ). Name of significant predictors is given in figure70

```
> plot(cv.out,xlab="log(lambda)",ylab="mean squared error")
> bestlam=cv.out$lambda.min
> bestlam
[1] 0.000429268
> lasso.pred=predict(cv.out,s=bestlam,newx = as.matrix(X.test))
> MAPE(exp(lasso.pred),y_test) #0.0571
MAPE 0.05717803
```

Figure 71: MAPE by LASSO using CROSS VALIDATION

Fig 72: is the graph between  $\log(\lambda)$  and Mean-Squared Error where x axis represent  $\log(\lambda)$  and y axis represent Mean-Squared Error. From this figure 72 and Fig 71: it is clear that when  $\lambda = 0.000429268$  or  $\log(\lambda) = -7.753429$  (as  $\log(0.000429268) = -7.753429$ ) than Mean-squared Error is minimum as "`cv.out$lambda.min`" provide the tuning Parameter  $\lambda$  corresponding the model having minimum MSE Using Cross Validation. ("`cv.out`" is defined in fig 70:).

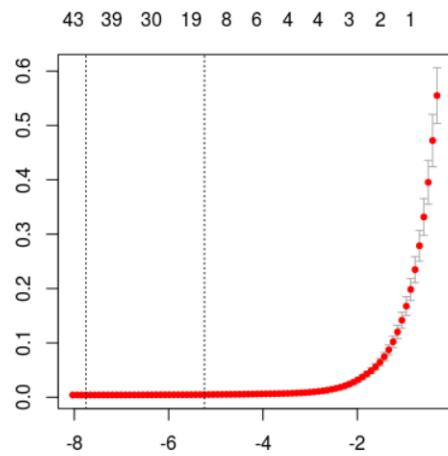


Figure 72: Graph Between  $\log(\lambda)$  and Mean Squared Error

When the model obtained by LASSO using Cross-Validation is tested on test data it provides a MAPE of 0.0571 which is lower than MAPE of PCR and MAPE os PCS so this model is well enough than other model obtained by PCS and PCR.

## 14 Checking for Significance of first level of each Categorical Variable

In our model, in train data, ‘Post.Hour\_16’, ‘Post.Hour\_19’, ‘Post.Hour\_20’ and ‘Post.Hour\_22’ contain only one observation as ‘1’, but due to exclusion of leverage points these observations are also removed. As a result now these columns contain only elements as ‘0’. But this will reduce rank of full rank matrix  $\mathbf{X}$  which will create a problem as full rank was our basic assumption. So, to overcome this difficulty we have removed those four columns (this will put no effect on our analysis as we found that in the whole data set there were only one observation of these ‘Post.Hour’s).

In our initial model we found that there are 52 variables (including dummies) but only few of them have serious effect on the model and others are insignificant in prediction. Checking the p-value and setting our level of significance at 5% we get a sets of significant and insignificant variables which shown in the table below have bold effects on the model. This model consists of 399 observations and 52 variables.

**Highly Significant Variables :** ‘Total.Reach’, ‘Total.Impression’, ‘Consumer’, ‘Impression.liked’, ‘Reach.liked’, ‘People.engaged.liked’ and ‘Total.interaction’

**Moderately Significant Variables :** ‘Post.week\_4’, ‘Consumption’, ‘Type\_photo’, ‘Type\_video’, ‘Paid\_1’

**Insignificant Variables :** Intercept, ‘Page.total.like’, ‘Type\_status’, All ‘Category’ (except Category\_1), All ‘Post.Month’ (except Month\_1), ‘Post.week\_5,6,7’ , All ‘Post.Hour’ (except Hour\_1) and ‘Post.week\_2,3’

It is notable that ‘Intercept’ has no significant effect. From separate Analysis first level of each categorical variable we found that

```

> #first Category
> first=function(dat){
+   Type_Link=rep(0,nrow(dat))
+   Category_1=rep(0,nrow(dat))
+   Post.Month_1=rep(0,nrow(dat))
+   Post.Weekday_1=rep(0,nrow(dat))
+   Post.Hour_1=rep(0,nrow(dat))
+   s=rep(0,nrow(dat))
+   a=which(colnames(dat)=='Type_Photo')
+   for (i in a:a+2)
+     s=s+dat[,i]
+   for(i in 1:length(s)){
+     if (s[i]==0)
+       Type_Link[i]=1
+   }
+   s=rep(0,nrow(dat))
+   a=which(colnames(dat)=='Category_2')
+   for (i in a:a+1)
+     s=s+dat[,i]
+   for(i in 1:length(s)){
+     if (s[i]==0)
+       Category_1[i]=1
+   }
+   s=rep(0,nrow(dat))
+   a=which(colnames(dat)=='Post.Month_2')
+   for (i in a:a+10)
+     s=s+dat[,i]
+   for(i in 1:length(s)){
+     if (s[i]==0)
+       Post.Month_1[i]=1
+   }
+   s=rep(0,nrow(dat))
+   a=which(colnames(dat)=='Post.Weekday_2')
+   for (i in a:a+6)
+     s=s+dat[,i]
+   for(i in 1:length(s)){
+     if (s[i]==0)
+       Post.Weekday_1[i]=1
+   }
+   s=rep(0,nrow(dat))
+   a=which(colnames(dat)=='Post.Hour_2')
+   b=which(colnames(dat)=='Paid_1')
+   for (i in a:b-1)
+     s=s+dat[,i]
+   for(i in 1:length(s)){
+     if (s[i]==0)
+       Post.Hour_1[i]=1
+   }
+   a=which(colnames(dat)=='Paid_1')
+   Paid_0=abs(dat[,a]-1)
+   one=cbind(Type_Link,Category_1,Post.Month_1,Post.Weekday_1,Post.Hour_1,Paid_0)
+   assign('one',one,envir = .GlobalEnv)
+ }
```

Figure 73: Function for creating first level of indicator variable

‘Type\_link’ is significant, ‘Category\_1’ is insignificant, ‘Post.Month\_1’ is significant, ‘Post.week\_1’ is insignificant, ‘Post.Hour\_1’ is insignificant and ‘Paid\_0’ is insignificant . This implies that categorical variables ‘Type’, ‘Post.Month’, ‘Paid’ are significant and ‘Category’, ‘Weekday’, ‘Hour’ are insignificant in predicting.

```

> #model
> one=0
> first(my_data)
> data=cbind(my_data[,-c(11:53)],one) #399 16
> mod=lm(y~,data=data)
> summary(mod)$coefficients[,4]
            (Intercept) Page.total.likes Total.Reach
    7.826506e-03   8.620527e-01  5.882447e-11
  Total.Impressions Consumers Consumptions
  3.943871e-07   2.790514e-241  4.599473e-02
 Impressions.liked Reach.liked People.engaged.liked
  7.978126e-05   1.157271e-01  4.440711e-05
 Total.Interactions Type_Link Category_1
  2.870272e-197  8.607828e-06  9.878424e-01
 Post.Month_1     Post.Weekday_1 Post.Hour_1
  3.408234e-03   5.163657e-01  7.124724e-01
 Paid_0           6.321172e-02

```

Figure 74: Calculation for initial model

We found F-statistic=2790 with df 52 and 346 And  $F_{0.05;52,346} = 1.38$ , So, with a high confidence we can say some of the variables have very significant effects.

After excluding leverage points we built another model and find the following table significant and insignificant varibales. This model is built with 394 observations and 48 variables.

**Highly Significant Variables :** ‘Total.Reach’, ‘Total.Impression’, ‘Consumer’, ‘Impression.liked’, ‘People.engaged.liked’ and ‘Total.interaction’

**Moderately Significant Variables :** ‘Post.week\_2,4’, ‘Consumption’, ‘Type\_photo’, ‘Type\_video’, ‘Paid\_1

**Insignificant Variables :** Intercept, ‘Page.total.like’, ‘Reach.liked’, ‘Type\_status’, All ‘Category’ (except Category\_1), All ‘Post.Month’ (except Month\_1), All ‘Post.Hour’ (except Hour\_1) and ‘Post.week\_3,5,6,7’

It is notable that ‘Intercept’ has no significant effect. From separate Analysis first level of each categorical variable we found that

```

> #model1
> first(my_data1)
> data=cbind(my_data1[,-c(11:49)],one) #394 16
> mod=lm(y~,data=data)
> summary(mod)$coefficients[,4]
            (Intercept) Page.total.likes Total.Reach
    5.767697e-03   9.450720e-01  2.266481e-09
  Total.Impressions Consumers Consumptions
  1.035552e-05   7.920785e-243  5.106258e-02
 Impressions.liked Reach.liked People.engaged.liked
  3.327689e-01   7.255697e-01  5.223577e-05
 Total.Interactions Type_Link Category_1
  1.404010e-199  5.000214e-06  8.956308e-01
 Post.Month_1     Post.Weekday_1 Post.Hour_1
  2.178325e-03   4.357047e-01  7.018525e-01
 Paid_0           9.608225e-02

```

Figure 75: Calculation for model after excluding leverages

‘Type\_link’ is insignificant, ‘Category\_1’ is insignificant, ‘Post.Month\_1’ is significant, ‘Post.week\_1’ is insignificant, ‘Post.Hour\_1’ is insignificant and ‘Paid\_0’ is insignificant . This implies that categorical variables ‘Type’, ‘Category’, ‘Post.Month’, ‘Post.weekday’, ‘Paid’ are significant and ‘Post.Hour’ is insignificant in prediction.

We found F-statistic=3286 with df 48 and 345 And  $F_{0.05;48,345} = 1.395$ , So, with a high

confidence we can say some of the variables have very significant effects.

After deleting influential points we built another model with 389 observations and 48 variables. Let's see its significant and insignificant variables.

**Highly Significant Variables** : 'Total.Reach', 'Total.Impression', 'People.engaged.liked', 'Consumer', 'Total.interaction' and 'Type\_video'

**Moderately Significant Variables** : 'Consumption', 'Impression.liked', 'Reach.like', 'Type\_photo', 'Post.week\_4' and 'Category\_2,3'

**Insignificant Variables** : Intercept, 'Page.total.like', 'Type\_status', All 'Post.Month' (except Month\_1), 'Post.week\_2,3,5,6,7', All 'Post.Hour' (except Hour\_1) and 'Paid\_1'

It is notable that 'Intercept' has no significant effect. From separate Analysis first level of each categorical variable we found that

```
> #model2
> first(my_data22)
> data=cbind(my_data22[,-c(11:49)],one) #389 16
> mod=lm(y~.,data=data)
> summary(mod)$coefficients[,4]
            (Intercept) Page.total.likes Total.Reach
8.271948e-06    3.540761e-01 2.800821e-09
Total.Impressions Consumers Consumptions
6.049717e-06    8.500825e-272 1.305147e-02
Impressions.Liked Reach.Liked People.engaged.liked
2.902905e-02    9.459164e-02 2.214187e-04
Total.Interactions Type_Link Category_1
5.729192e-210   2.863170e-08 9.658753e-02
Post.Month_1      Post.Weekday_1 Post.Hour_1
2.261048e-04    5.053850e-01 4.355406e-01
Paid_0           ...
2.827748e-01
```

Figure 76: Calculation for model after excluding influentials

'Type\_link' is insignificant, 'Category\_1' is insignificant, 'Post.Month\_1' is significant, 'Post.week\_1' is insignificant, 'Post.Hour\_1' is insignificant and 'Paid\_0' is insignificant. This implies that categorical variables 'Type', 'Category', 'Post.Month', 'Post.Weekday' are significant and 'Post.Hour', 'Paid' are insignificant in prediction.

We found F-statistic=4531 with df 48 and 340 And  $F_{0.05;48,340} = 1.396$ , So, with a high confidence we can say some of the variables have very significant effects.

After Normality correction we made a logarithmic transformation of our original response and built a new model. Let's check its significant and insignificant variables after transformation.

**Highly Significant Variables** : 'Intercept', 'Consumer', 'Total.interaction' and 'Type\_photo,status'

**Moderately Significant Variables** : 'Consumption'

**Insignificant Variables** : Intercept, 'Page.total.like', 'People.engaged.liked', 'Type\_video', 'Total.Reach', 'Total.Impression', 'Impression.liked', 'Total.Reach.like', All 'Category' (except Category\_1), All 'Post.Month' (except Month\_1), All 'Post.week' (except week\_1), All 'Post.Hour' (except Hour\_1) and 'Paid\_1'

Here 'Intercept' has high significance (this is due to transformation in response). From separate Analysis first level of each categorical variable we found that

'Type\_link' is insignificant, 'Category\_1' is significant, 'Post.Month\_1' is significant, 'Post.week\_1' is insignificant, 'Post.Hour\_1' is insignificant and 'Paid\_0' is insignificant. This implies that categorical variables 'Type', 'Post.Month', 'Post.weekday' are significant and 'Category', 'Post.Hour', 'Paid' are insignificant.

```

> #model3
> first(my_data2)
> data=cbind(my_data2[,-c(11:49)],one) #382 16
> mod=lm(z~.,data=data)
> summary(mod)$coefficients[,4]
            (Intercept) Page.total.likes Total.Reach
            5.148351e-52   1.058469e-01  6.091946e-02
Total.Impressions Consumers Consumptions
3.095117e-02      7.236998e-25  5.099426e-02
Impressions.liked Reach.liked People.engaged.liked
7.463595e-02      2.799289e-02  3.180192e-02
Total.Interactions Type_Link Category_1
1.258817e-15     6.949517e-01  1.193722e-02
Post.Month_1       Post.Weekday_1 Post.Hour_1
9.866792e-09      4.297200e-01  4.504939e-01
Paid_0             9.378542e-02

```

Figure 77: Calculation for model after transformation in response

We found F-statistic=40 (as response is reduced in measure F-statistics fell by a significant amount) with df 48 and 333 And  $F_{0.05;48,333} = 1.396$ , So, still with a high confidence we can say some of the variables have very significant effects.

To remove heteroscedasticity we made a logarithmic transformation of our numeric continuous variables this time. With transformed response and regressors a new model was built with 382 observations and 48 variables. Now we check its significant and insignificant variables.

**Highly Significant Variables :** ‘Total.Reach’, ‘Total.Impressions’, ‘Consumer’, ‘Consumption’, ‘People.engaged.liked’, ‘Total.interaction’, ‘Type\_photo’

**Moderately Significant Variables :** ‘Category\_3’ and ‘Post.week\_5’

**Insignificant Variables :** Intercept, ‘Page.total.like’, ‘Type\_video’, ‘Impression.like’, ‘Total.Reach.like’, ‘Category\_2’, All ‘Post.Month’ (except Month\_1), ‘Post.week\_2,3,4,6,7’ , All ‘Post.Hour’ (except Hour\_1) and ‘Paid\_1’

Here again ‘Intercept’ has no significant effect. From separate Analysis first level of each categorical variable we found that

```

> #model4
> first(my_data4)
> data=cbind(my_data4[,-c(11:49)],one) #382 16
> mod=lm(z~.,data=data)
> summary(mod)$coefficients[,4]
            (Intercept) Page.total.likes Total.Reach
            1.187149e-01  3.047873e-01  3.440757e-10
Total.Impressions Consumers Consumptions
2.072637e-03      2.003520e-106  5.391620e-01
Impressions.liked Reach.liked People.engaged.liked
1.013098e-02      1.540605e-05   8.772038e-19
Total.Interactions Type_Link Category_1
1.407662e-44     1.833956e-01   7.343936e-02
Post.Month_1       Post.Weekday_1 Post.Hour_1
1.973357e-02      9.146612e-02   7.658354e-01
Paid_0             6.556931e-01

```

Figure 78: Calculation for model after transformation in numeric variables

‘Type\_link’ is insignificant, ‘Category\_1’ is insignificant, ‘Post.Month\_1’ is significant, ‘Post.week\_1’ is insignificant, ‘Post.Hour\_1’ is insignificant and ‘Paid\_0’ is insignificant . This implies that categorical variables ‘Type’, ‘Category’, ‘Post.Month’, ‘Post.week’ are significant and ‘Post.Hour’, ‘Paid’ are insignificant.

We found F-statistic=1273 with df 48 and 333 And  $F_{0.05;48,333} = 1.396$ , So, with a high

confidence we can say some of the variables have very significant effects.

In order to remove multicollinearity from the model two PCA are done for ‘Total.Reach’, ‘Total.impression’, ‘Impression.liked’ , ‘Reach.liked’ and ‘Consumer’, ‘Consumption’ and ‘People.engaged.like’, then instead of all pcs only first two pcs are taken from both. Variability explained by first two principal components in both cases are 98%. ‘Page.total.like’ is removed due to low correlation with response and high VIF. Finally the model was built with 382 observations and 44 variables whose significant and insignificant variables are listed below.

**Highly Significant Variables :** Intercept, PC12, PC21, PC22, ‘Total.interaction’ and ‘Post.Month\_3,4,...

**Moderately Significant Variables :** ‘Type\_status’, ‘Post.Month\_2’ and ‘Post.Hour\_2’

**Insignificant Variables :** PC11, ‘Type\_photo,video’, All ‘Category’ (except Category\_1), All ‘Post.week’ (except week\_1), ‘Post.Hour\_3,4,...,22’ and ‘Paid\_1’

Here ‘Intercept’ has high significance (this is due to transformation in response). From separate Analysis first level of each categorical variable we found that

```
> first(my_data5)
> data=cbind(my_data5[,-c(7:45)],one) #382 12
> mod=lm(z~.,data=data)
> summary(mod)$coefficients[,4]
            PC11          PC12
(Intercept) 5.083124e-02 2.873850e-16
PC21        9.376486e-26 4.552722e-12
4.471583e-228 Category_1 Post.Month_1
Type_Link    6.219339e-04 5.479736e-01
6.974055e-01 Post.Hour_1 Paid_0
Post.Weekday_1 8.164352e-01 9.726894e-01
1.575759e-01
```

Figure 79: Calculation for model after partial pca

‘Type\_link’ is insignificant, ‘Category\_1’ is insignificant, ‘Post.Month\_1’ is significant, ‘Post.week\_1’ is insignificant, ‘Post.Hour\_1’ is insignificant and ‘Paid\_0’ is insignificant . This implies that categorical variables ‘Type’, ‘Post.Month’, ‘Post.Hour’ are significant and ‘Category’, ‘Post.week’, ‘Paid’ are insignificant.

We found F-statistic=672 (as response is reduced in measure F-statistics fell by a significant amount) with df 44 and 337 and  $F_{0.05;44,337} = 1.41$ , So, still with a high confidence we can say some of the variables have very significant effects.

On the above model we applied Stepwise variable selection method, thus only 25 important variables (these variables resulted minimum AIC) are selected from 44 variables and others totally ignored. So, our new model built with 25 variables and 382 observations. Let’s check significant and insignificant variables.

**Highly Significant Variables :** Intercept, PC12, PC21, PC22, ‘Total.interaction’ , ‘Post.Month\_3,4,...,12’ ‘Type\_status’ and ‘Post.Hour\_2’

**Moderately Significant Variables :** ‘Post.Month\_2’ and ‘Post.Hour\_5,6’

**Insignificant Variables :** ‘Post.week\_5’ and ‘Post.Hour\_8,9,...,12’

Clearly due searching best model using variable selection method there are very few insignificant variables and relatively larger number of significant variables. From separate Analysis first level of each categorical variable we found that

‘Type\_link’ is insignificant, ‘Category\_1’ is significant, ‘Post.Month\_1’ is significant, ‘Post.week\_1’ is significant, ‘Post.Hour\_1’ is insignificant and ‘Paid\_0’ is insignificant . This implies among the categorical variables only ‘Paid’ is insignificant.

```

> first(my_data5)
> data=cbind(my_data5[,1:6],one,my_data5[,-c(1:7,10,13,23,29,45)])
> mod=lm(z~.,data=data)
> mod1=lm(z~1,data=data)
> step1=step(mod1,direction="both",scope=formula(mod))
> summary(step1)$call
lm(formula = z ~ PC21 + Total.Interactions + PC22 + PC12 + Type_Status +
tus +
Post.Month_10 + Post.Month_4 + Post.Month_9 + Post.Month_8 +
Post.Month_5 + Post.Month_7 + Post.Month_1 + Post.Month_11 +
Post.Month_6 + Post.Weekday_1 + Post.Hour_15 + Post.Hour_18 +
Post.Hour_17 + Post.Hour_14 + Post.Hour_6 + Post.Hour_5 +
Post.Weekday_5 + Post.Hour_12 + Category_1, data = data)

```

Figure 80: Checking if first levels are selected by Stepwise method

We found F-statistic=1213 (as response is reduced in measure F-statistics fell by a significant amount) with df 25 and 356

To remove multicollinearity in another way, again we applied same procedure (PCA) but in different approach. This time PCA technique is applied over all the 48 variables and obviously got a better result than before using first 38 principal components (which explain 97.78% variability).

Another method (PLS) was used to remove multicollinearity and this time using only first 10 component (which explain 99.37% variability) we got very good result in prediction. Here a good number of variables are reduced.

Lastly Lasso was used to remove multicollinearity and to select variables simultaneously (on 48 variables). Lasso signified only 17 main variables into account.

**Significant Variables (taken into the model)** : Intercept, ‘Total.Reach’, ‘Consumer’, ‘People.engaged.like’, ‘Total.interaction’, ‘Type\_photo’, ‘Category\_3’, ‘Post.Month\_2,4,5,7,11,12’, ‘Post.week\_5’ and ‘Post.Hour\_2,9,14’.

**Insignificant Variables (not taken into the model)** : ‘Page.Total.like’, ‘Total.impression’, ‘Impression.like’, ‘Reach.like’, ‘Consumption’, ‘Type\_status,video’, ‘Category\_2’, ‘Post.Month\_3,6,8,9’, ‘Post.week\_2,3,4,6,7’ and ‘Post.Hour\_3,4,...,8,10,11,12,15,17,18’

From separate Analysis first level of each categorical variable we found that

```

> #lasso
> set.seed(197)
> first(my_data4)
> xtrain1=as.matrix(cbind(my_data4[,c(2:10)],one,my_data4[,-c(1:11,14,16,27,33,49)]))
> ytrain1=as.vector(my_data4[,1])
> cv.out1=cv.glmnet(xtrain1,ytrain1,type.measure = "mse",
+ alpha=1, family="gaussian")
> a1=coef(cv.out1)[,1]
> names(a1[a1!=0])
 [1] "(Intercept)"           "Total.Reach"
 [3] "Consumers"              "People.engaged.liked"
 [5] "Total.Interactions"     "Type_Link"
 [7] "Category_1"              "Post.Month_1"
 [9] "Post.Weekday_1"          "Type_Status"
 [11] "Type_Video"              "Post.Month_4"
 [13] "Post.Month_5"              "Post.Month_7"
 [15] "Post.Month_11"             "Post.Month_12"
 [17] "Post.Weekday_5"            "Post.Hour_9"
 [19] "Post.Hour_12"              "Post.Hour_14"

```

Figure 81: Checking if first levels are selected by Lasso

‘Type\_link’ is insignificant, ‘Category\_1’ is significant, ‘Post.Month\_1’ is significant, ‘Post.week\_1’ is significant, ‘Post.Hour\_1’ is insignificant and ‘Paid\_0’ is insignificant. This implies among the categorical variables only ‘Paid’ is insignificant and other categorical variables have significant effect in prediction.

## 15 Conclusion

Finally we got three models **(i)** firstly using principal component of all variables with MAPE 11.8%

**(ii)** secondly using partial least square with MAPE 6.17%

**(iii)** thirdly using Lasso with MAPE 5.71%.

Though the first model’s efficiency is not bad, it took all the regressors (48 regressors are hard to interpret at a time) into account and then used first 38 components (which explain 97.78% of total variability). This will take too much time and cost for larger number of observations and from this model we can’t specify the regressors which have severe effects on the response (Lifetime.engaged.users). Second model, with better efficiency than first model, also took all the regressors into account (which will again alert a statistician regarding time and cost) but then using only first 10 components (which explain 99.37% of total variability) it predicted very well. Also, here is no scope for choosing best regressors. Thirdly with Lasso we overcame all the difficulties. It identified 17 regressors and with the help of these regressors it predicted very much well than first two models. Clearly, here we can check the effects of these selected 17 regressors and time, cost will be much less here. So, regarding the response Lifetime.engaged.users few factors are there to be noticed. Those are -

**(i)** Posts in April, May, July induced lesser number of users where Feb, Nov, Dec ensured higher number of users. So, the company should post in year-end or in the beginning of the year.

**(ii)** More users are engaged with post type ‘Photo’

**(iii)** Inspirational posts gathered more users

**(iv)** More users can be engaged with the posts in the night and noon

**(v)** Weekdays have not such heavy effects on the users

**(vi)** No effects of paid or unpaid posts

**(vi)** As ‘Total.Reach’, ‘like’, ‘comment’, ‘share’, ‘Consumer’, ‘People.engaged.like’ are directly related to clicking or seeing any post by the users, more amount of these variables certainly will imply more the engaged users.

## 16 Bibliography

1. An Introduction to Statistical Learning,with Applications in R by Gareth James, Daniela Witten, Trevor Hastie and Robert Tibshirani
2. Introduction to Linear Regression Analysis by Douglas C Montgomery, Elizabeth A Peck, G. Geoffrey Vining
3. The Elements of Statistical Learning:Data Mining, Inference, and Prediction by Trevor Hastie,Robert Tibshirani,Jerome Friedman
4. <http://dx.doi.org/10.1016/j.jbusres.2016.02.010> ((Moro et al., 2016) Moro, S., Rita, P., Vala, B. (2016))
5. <https://stackoverflow.com>

6. www.analyticsvidhya.com

## 17 Appendix

The files used in this project can be found at the following link. This include data file and R code.

[https://drive.google.com/drive/folders/1\\_Tw7cTaX78s8HzM6nmt-PVEgtDvDLcTN?usp=sharing](https://drive.google.com/drive/folders/1_Tw7cTaX78s8HzM6nmt-PVEgtDvDLcTN?usp=sharing)