CAPSTONE PROJECT

FACEBOOK COMMENT VOLUME PREDICTION

MANU VATS
PGPBAIO

1 CONTENTS

2 INTRODUCTION

2.1 PROBLEM STATEMENT

We are given the dataset consisting of details of Facebook pages and their posts. The goal is to predict how many comments a user generated posts is expected to receive in the given set of hours on Facebook. We need to model the user comments pattern over a set of variables which are provided and get to the right number of comments for each post with minimum error possible.

2.2 NEED OF THE STUDY

Brands are constantly trying to drive engagements with the customers and trying to get their feedback and response. For this, social media is a widespread tool which is being used by almost every major brand. But to measure the customers' response and feedback on social media and generate content on the basis of those feedback is a challenge for the brands. For this, brands need to be able to understand what kind of a post drives higher or lower engagement on Facebook.

2.3 BUSINESS OPPORTUNITY

For both small businesses and large corporations, social media is playing a key role in brand building and customer communication. Facebook is the social networking site relevant for firms to make themselves real for customers. Just to put things in context, the advertising revenue of Facebook in the United States in 2018 stands up to 14.89 billion US dollars. The advertising revenue outside the United States comes down to 18.95 billion US dollars. Latest research reports have indicated that user generated content on Facebook drives higher engagement than ads. The amount of data that gets added to the network increases day by day and it is a gold mine of researchers who want to understand the intricacies of user behavior and user engagement. In this Project, we discuss one such problem where we take a step towards understanding the highly dynamic behavior of users towards Facebook posts.

3 DATA REPORT

Data Dictionary

Variable name	***Description***	***Feature type***
Page Popularity/likes	Defines the popularity or support for the source	Page feature
Page Checkins	Describes how many individuals so far visited this place. This feature is only associated with the places eg:some institution, place, theater etc.	Page feature
Page talking about	Defines the daily interest of individuals towards source. The people who actually come back to the page, after liking the page. This include activities such as comments, likes to a post, shares, etc by visitors to the page.	Page feature
Page Category	Defines the category of the source eg: place, institution, brand etc	Page feature

Feature 5 – Feature 29	These features are aggregated by page, by calculating min, max, average, median and standard deviation of essential features.	Derived features
CC1	The total number of comments before selected base date/time	Essential feature
CC2	The number of comments in last 24 hours, relative to base date/time.	Essential feature
CC3	The number of comments in last 48 to last 24 hours relative to base date/time.	Essential feature
CC4	The number of comments in the first 24 hours after the publication of post but before base date/time	Essential feature
CC5	The difference between CC2 and CC3	Essential feature
Base time	Selected time in order to simulate the scenario	Other feature
Post length	Character count in the post	Other feature
Post Share Count	This features counts the no of shares of the post, that how many peoples had shared this post on to their timeline.	Other feature
Post Promotion Status	To reach more people with posts in News Feed, individual promote their post and this features tells that whether the post is promoted(1) or not(0).	Other feature
H Local	This describes the H hrs., for which we have the target variable/ comments received.	Other feature
Post published weekday	This represents the day (Sunday...Saturday) on which the post was published.	Day of the week (Categorical)
Base Date Time weekday	This represents the day(Sunday...Saturday) on selected base Date/Time.	Day of the week (Categorical)
Comments	The no of comments in next H hrs.(H represents H Local).	Target Variable

Table 1 - Data Dictionary

Let us examine the data. The dimensions of the dataset are as follows-:

No. of Rows	32759
No. of Columns	43

Table 2 - Dimensions of Data

We draw following inferences from a visual inspection and also from viewing the structure and summary of the data-:

- We can group all independent variables of the dataset into following types-:
    - **Page features** – Features which give information about the Facebook Page (Eg. ***Page likes, Page Category***, etc.)

- **Post features** – Features which give information about the post shared by the Facebook page (Eg. ***Post Length, Post Share Count,*** etc.).
- **Post Comments features** – Features which give information about the number of comments on that post (***CC1*** to ***CC5***). They are supposed to be essential features.
- **Unknown features** – Features which are unknown to us but will help us in model building and prediction (***Feature 5*** to ***Feature 29***). They are also the derived features.

- The column ***ID*** is basically the serial number of the Facebook post. So, all values in ***ID*** are different. We shall not use this variable in any analysis.
- The target variable is ***Comments***. It is basically the number of comments in the next specific number of hours.
- The variables seem to have a varied distribution with a lot of them having '0' values (including the Target variable). There are few variables with a very few unique values too like ***H local***.
- Some variables have floating point values and two variables namely ***Feature 27*** and ***CC5*** have negative values too. We have to see it later how they will affect the model.
- ***Post Promotion Status*** seems to be a constant variable. It need not be included in the model building.
- All the variables are of the data-type numeric except two namely ***Post published weekday*** and ***Base DateTime weekday*** which are character variables. They will need to be converted to factor type for analysis
- ***Page Category*** should be a categorical variable but here it is in the numeric form. It needs to be transformed into a categorical/factor variable.
- Following variables have missing values-:

	missing	%
Page.Checkins	3255	10
Page.talking.about	3255	10
Page.likes	3208	10
CC5	3200	10
CC1	3199	10
CC4	3198	10
Page.Category	3024	9
Feature.15	1692	5
Feature.7	1679	5
Feature.13	1643	5
Feature.10	1632	5
Feature.18	1605	5
Feature.22	1601	5
Feature.20	1600	5
Feature.25	1600	5
Feature.29	1600	5
Feature.27	1598	5

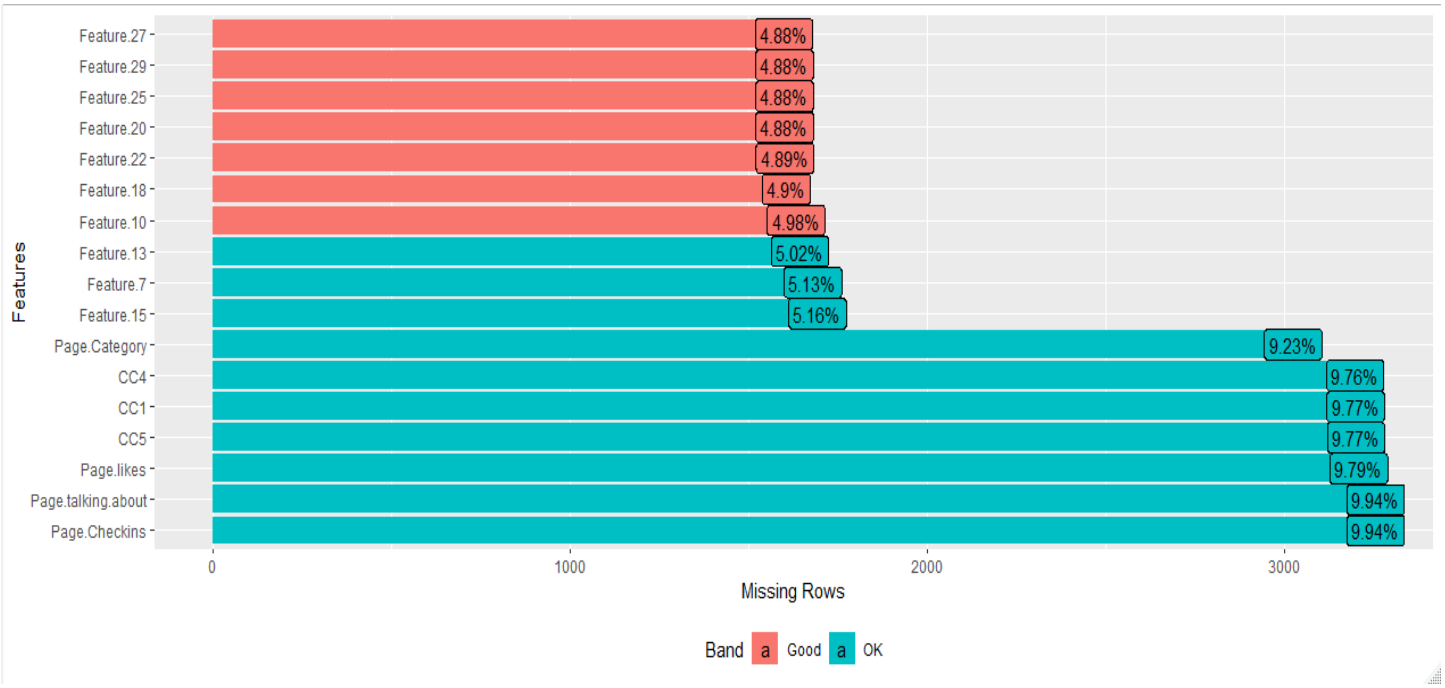

Figure 1 - Missing Values

- 17 variables out of 43 have missing values. 7 of them have about 10% of the values as missing or NA. The rest have 5% missing data. In all, there are 10,531 rows, i.e. around 32% of total number of rows, with missing or NA data. We have to figure out a way to impute the missing data as it may affect our regression model.

4 Univariate Analysis

4.1 Continuous Variables

For univariate analysis of continuous variables, we will study the distribution and spread of each variable. For that we will analyze the following-:

- Density Plots
- Boxplots

Let us first examine the density plots of all continuous variables-:

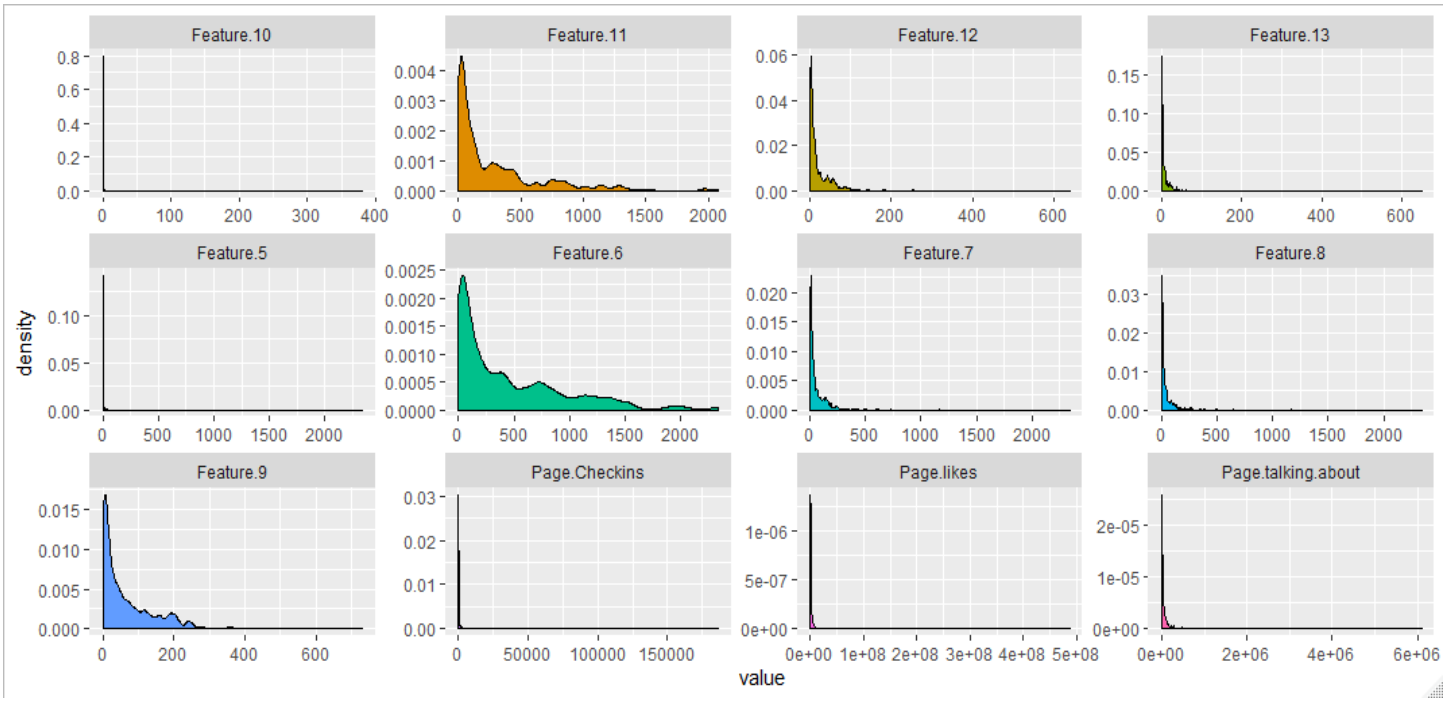

Figure 2 - Density plots of Continuous Variables (1-12)

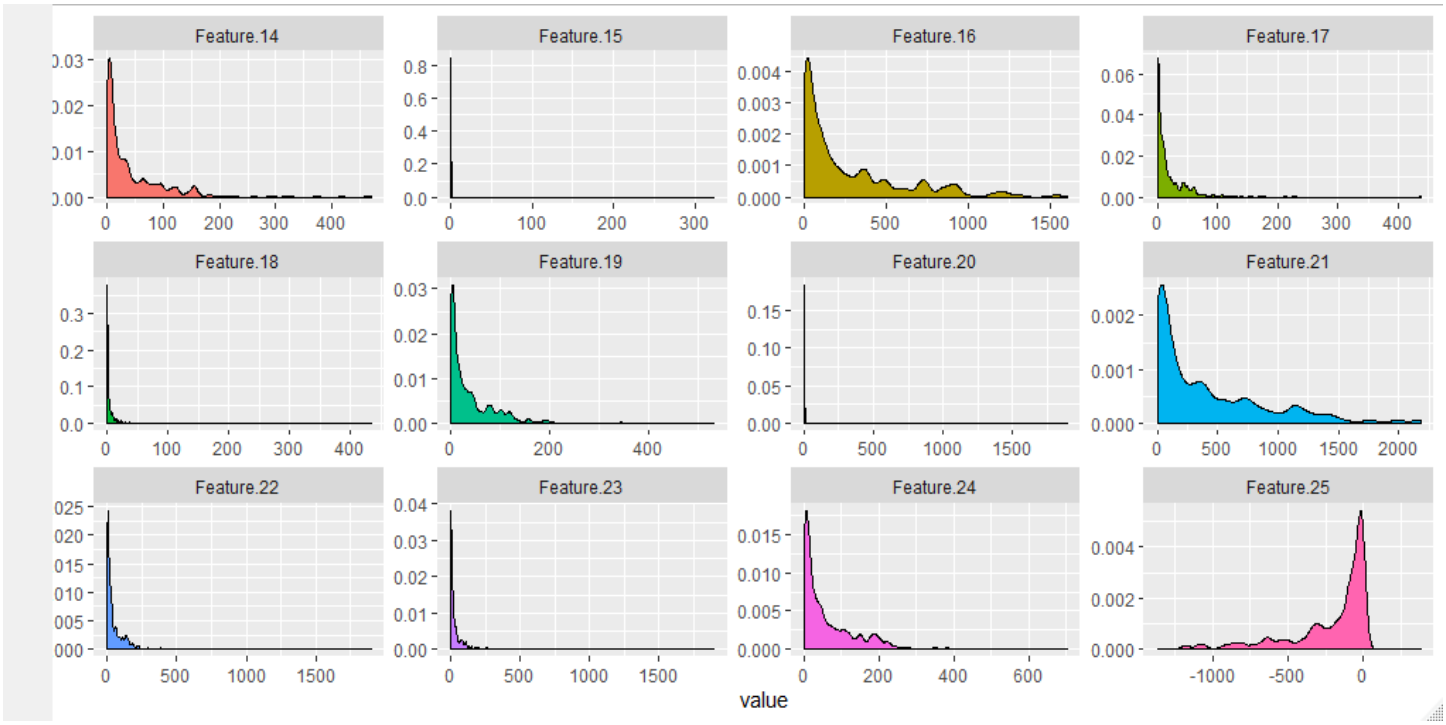

Figure 3 - Density plots of Continuous Variables (13-24)

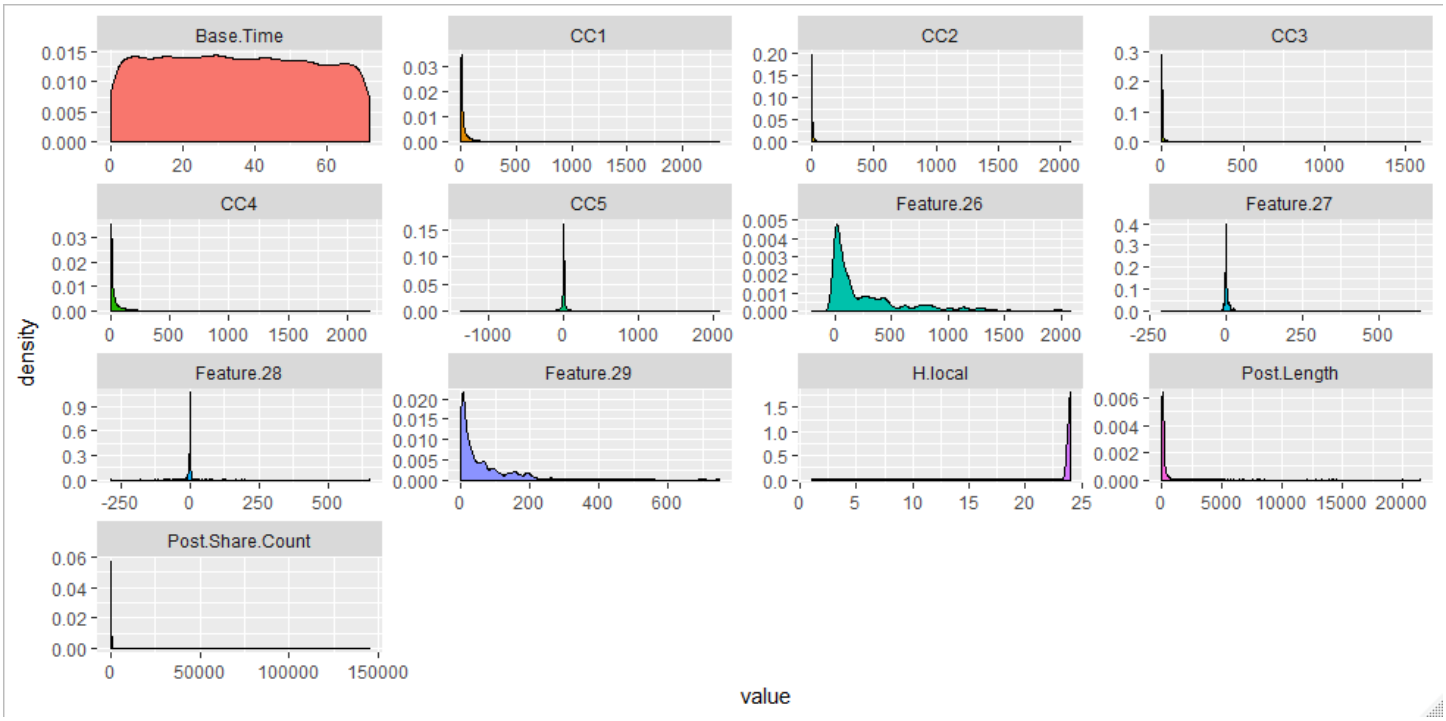

Figure 4 - Density plots of Continuous Variables (25-37)

All the variables are highly skewed except ***Base.Time***. Most of them are right skewed with a few of them left skewed. This kind of highly skewed data is not favorable for a good regression model. Hence, we need to transform the variables.

It is obvious that only after transforming variables, it makes sense to find and treat outliers and impute missing values. Nevertheless, let us take a look at the outliers graphically through boxplots-:

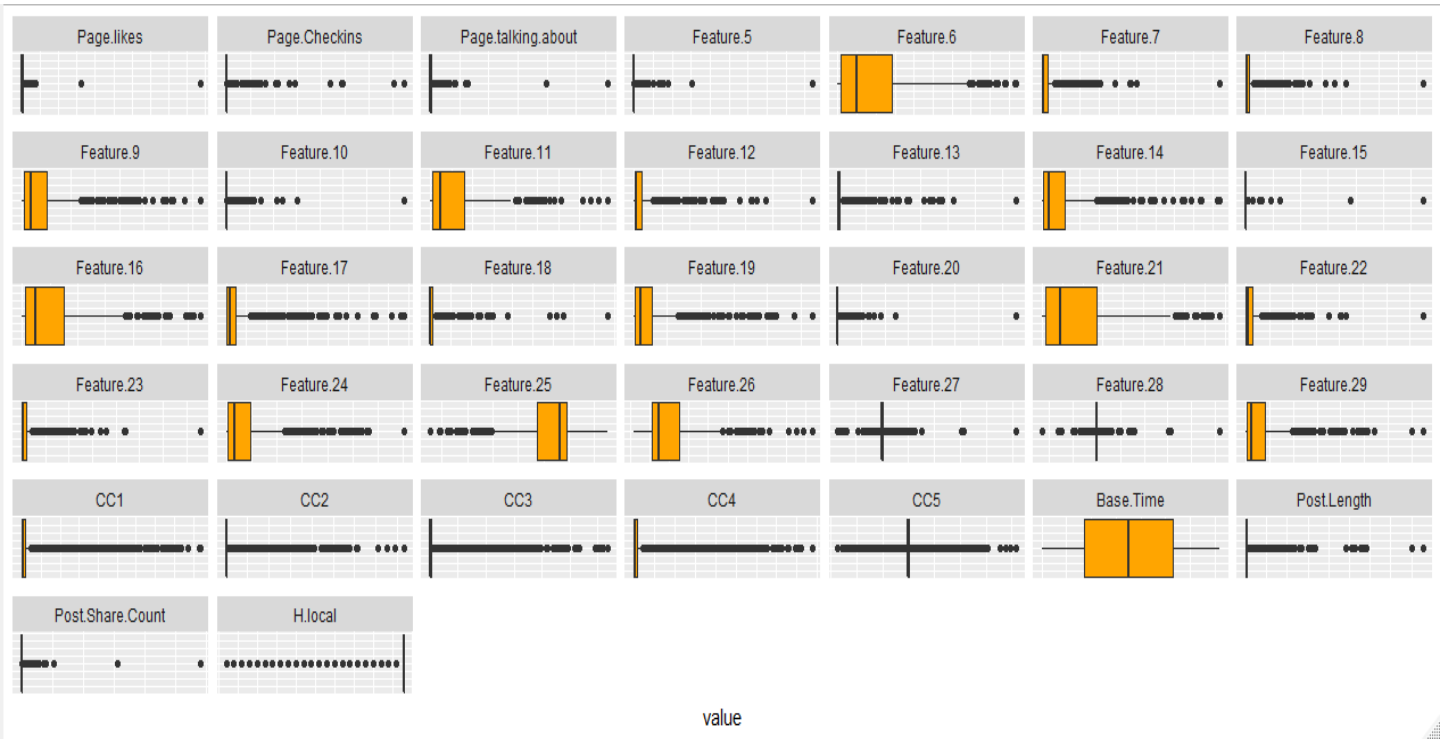

Figure 5 - Boxplots of Continuous Variables

As expected, the boxplot graph shows presence of too many outliers. The variables first need transformation before the treatment of outliers.

Let us summarize basic statistics of each continuous variable in the form of a table-:

	Range	Stdev	Skewness	Kurtosis
Feature.15	324.00	3.53	80.26	6816.70
Feature.5	2341.00	25.72	68.64	5962.02
Page.likes	486972261.00	5433876.34	64.37	5559.22
Feature.20	1897.00	21.52	63.02	5256.06
Feature.10	151.00	3.34	26.11	832.70
Feature.28	937.00	12.60	25.95	1368.98
Post.Share.Count	18691.00	500.87	16.30	415.30
Post.Length	14185.00	344.50	15.96	478.93
Target.Variable	1139.00	36.46	14.16	289.18
Feature.18	433.00	13.54	13.34	281.22
Feature.13	649.00	20.69	13.28	272.77
Feature.27	849.50	16.40	12.46	441.95
Page.talking.about	3959779.00	99974.85	9.71	239.50
CC2	1975.00	76.73	9.13	130.88
Feature.8	2341.00	71.81	8.91	160.86
CC3	1592.00	70.06	8.39	99.07
Feature.23	1897.00	67.49	7.92	117.96
Page.Checkins	186370.00	20406.60	6.20	43.11
CC1	2341.00	135.13	5.94	51.49
CC4	1975.00	126.38	5.71	46.45
Feature.7	2341.00	87.82	5.55	71.34
Feature.22	1897.00	81.54	4.90	51.58
Feature.12	639.00	35.80	4.89	43.76
Feature.17	437.68	30.63	4.77	41.53
CC5	3175.00	94.27	2.97	73.52
Feature.19	533.64	49.80	2.63	11.50
Feature.14	469.54	53.92	2.54	9.90
Feature.29	749.71	72.33	2.48	9.30
Feature.26	2283.00	371.47	2.04	4.49
Feature.9	665.61	81.36	1.99	5.78
Feature.11	2079.00	372.83	1.97	4.21
Feature.24	563.54	76.18	1.90	4.83
Feature.16	1605.00	326.01	1.58	2.04
Feature.21	2184.00	473.31	1.42	1.56
Feature.6	2341.00	498.10	1.37	1.42
Base.Time	72.00	20.95	0.04	-1.18
Feature.25	1425.00	279.50	-1.73	2.47
H.local	23.00	1.87	-9.14	87.66

Table 3 - Basic Statistic Summary of Continuous Variables

- **Range** and **Standard Deviation** – Variables like ***Page.likes***, ***Page.talking.about*** and ***Page.Checkins*** have a very large spread which is evident from their range and standard deviation. Whereas, variables like ***Base.Time***, ***Feature.10*** and ***Feature.18*** have low range and standard deviation which shows less spread. The dataset is mixed up of variables ranging from low to high range and variances

- **Skewness** -- As we can see in the above table, most of the variables have a very high value of skewness. All skewness values except for few are more than 1. Also, all values except two are positive values. This indicates that most of the variables are very **heavily right skewed** and need transformation for the model preparation. ***H.local*** is **heavily left skewed.**
- **Kurtosis** – The Kurtosis values are also very large most of them being positive. This also indicates that the variables are very thick tailed and quite far from exhibiting normal distribution.

*To transform the variables, we shall use **box-cox method** to get lambda values. We shall then raise the variables to the power of those lambda values and hopefully achieve a normal distribution which will be beneficial for our regression model.*

4.2 Categorical Variables

Let us see distribution of level in the categorical variables.

First we shall check is ***Base.DateTime.weekday*** -:

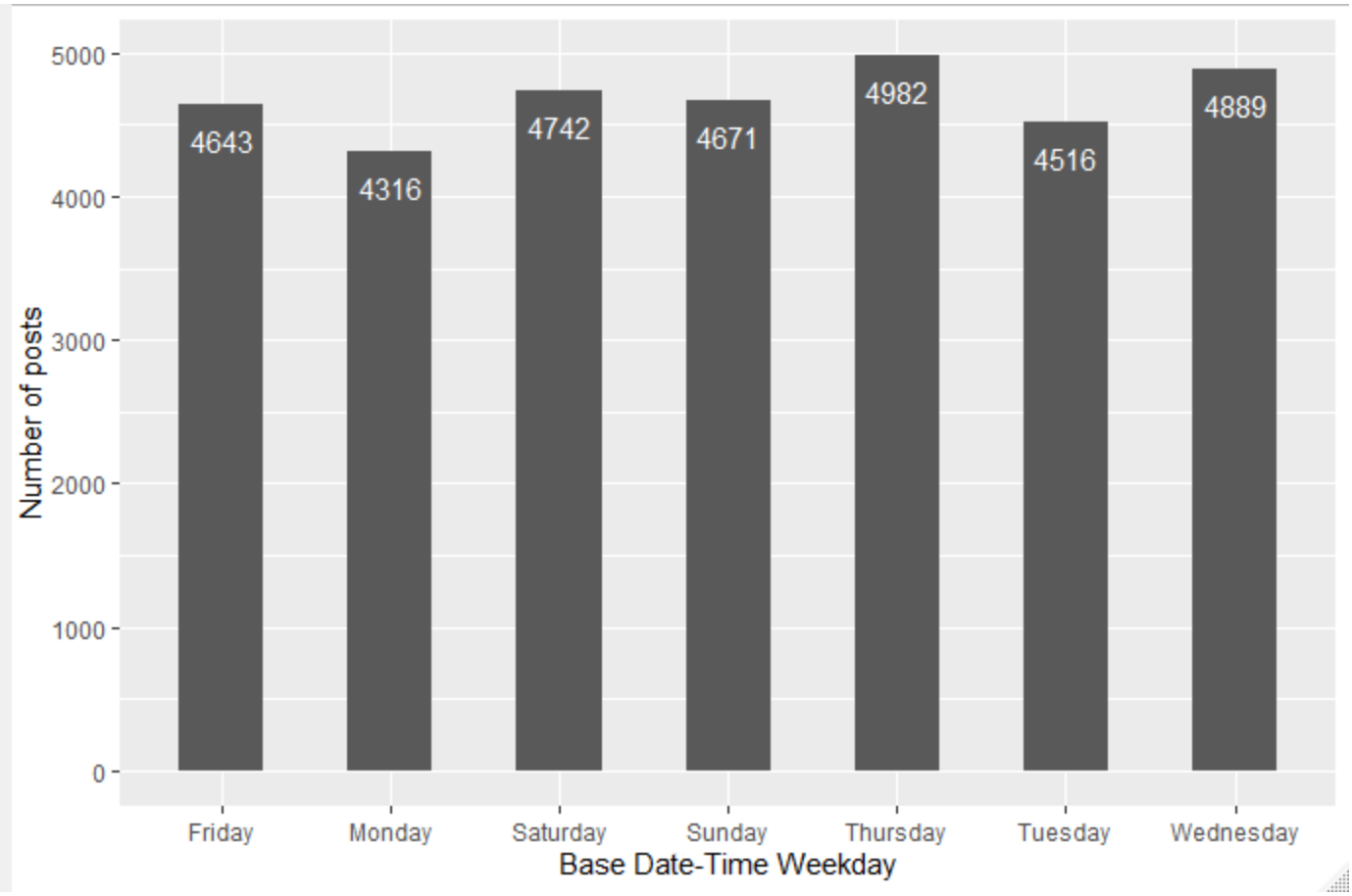

Figure 6 - Base Date-Time vs Number of Posts Barchart

The base day of Thursday had the greatest number of posts followed by Wednesday. The base day of Monday had the least number of posts.

Let us produce a similar barplot for ***Post.published.weekday*** -:

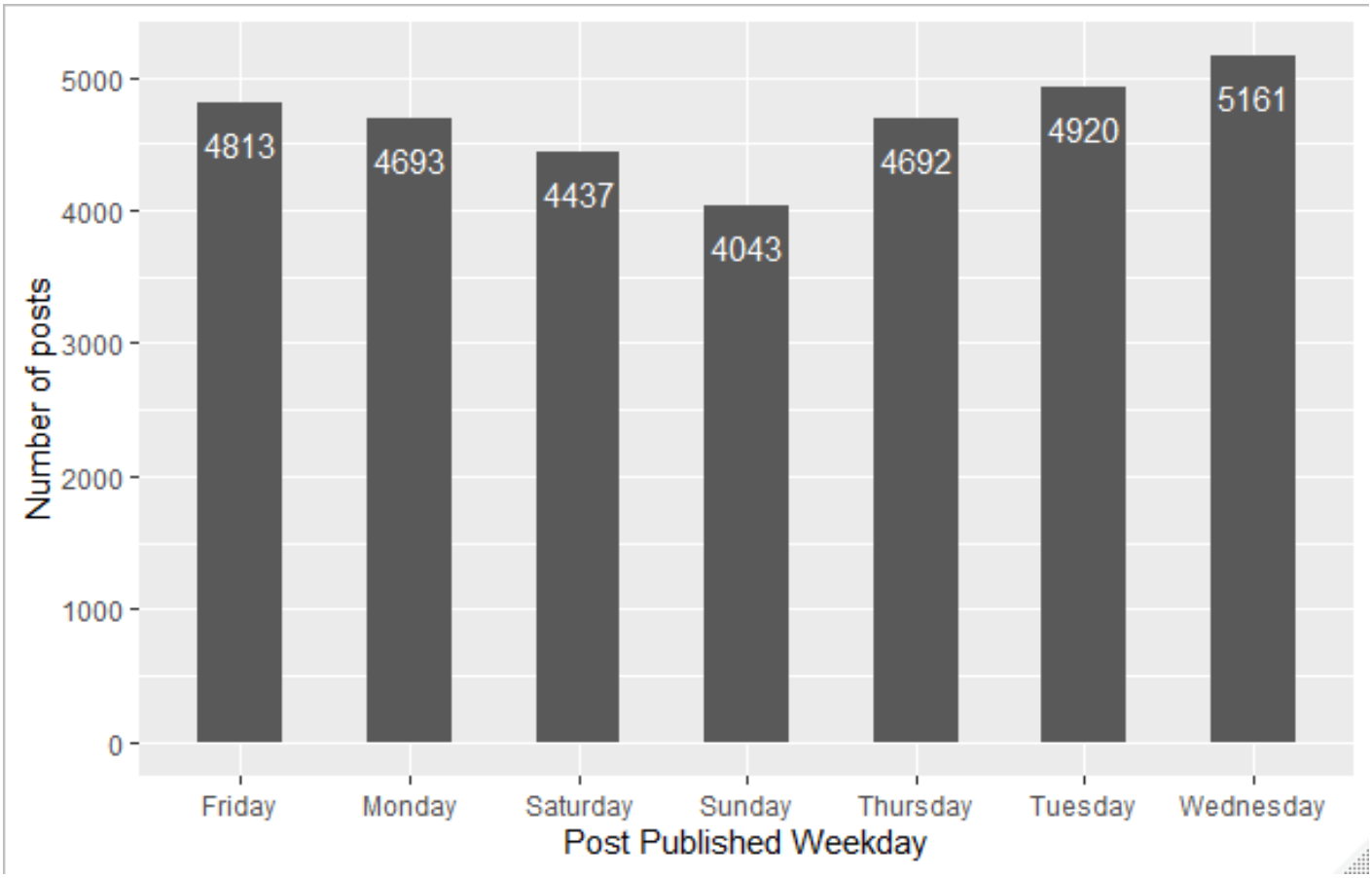

Figure 7 - Post Published Weekday vs Number of Posts Barchart

We clearly see that the greatest number of Facebook posts were published on Wednesday, followed by Tuesday. This is against the popular assumption that people engage in social media more on weekends than on weekdays.

5 Bivariate Analysis

5.1 Continuous and Continuous

Let us see a correlation plot for the continuous variables of the dataset-:

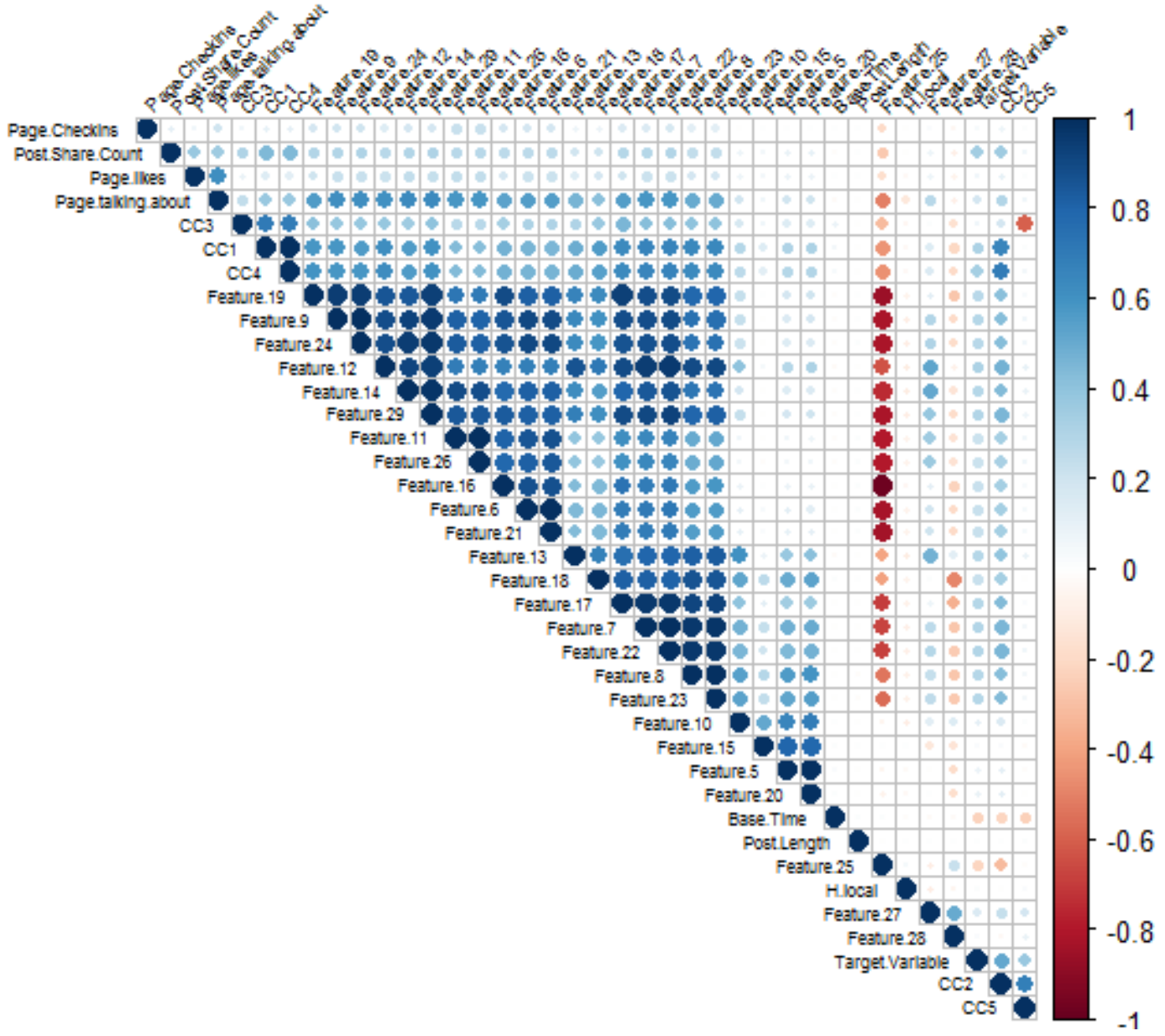

Figure 8 - Correlation Plot

We can clearly see that a lot of variables have a very high correlation between them, some of them having more than 90%. There seems a clear presence of multicollinearity, which has to be treated before formation of the model.

The high correlation exists most among the **Feature** variables. ***Feature.25*** has high negative correlation with other **Feature** variables**.** As these are derived variables, there is a possibility that they may be highly correlated among themselves. It may require to remove few of these variables to remove multicollinearity from the dataset so that we have a better regression model.

5.1.1 Correlation with Target Variable

The Target variable seems to not have significant correlations with individual variables, but highest with ***CC2*** (about 60%)***.*** It seems that ***CC2*** influences the ***Target.Variable*** the most. Also it seems to be most negatively correlated with ***Base.Time.*** Let us check scatterplots for each.

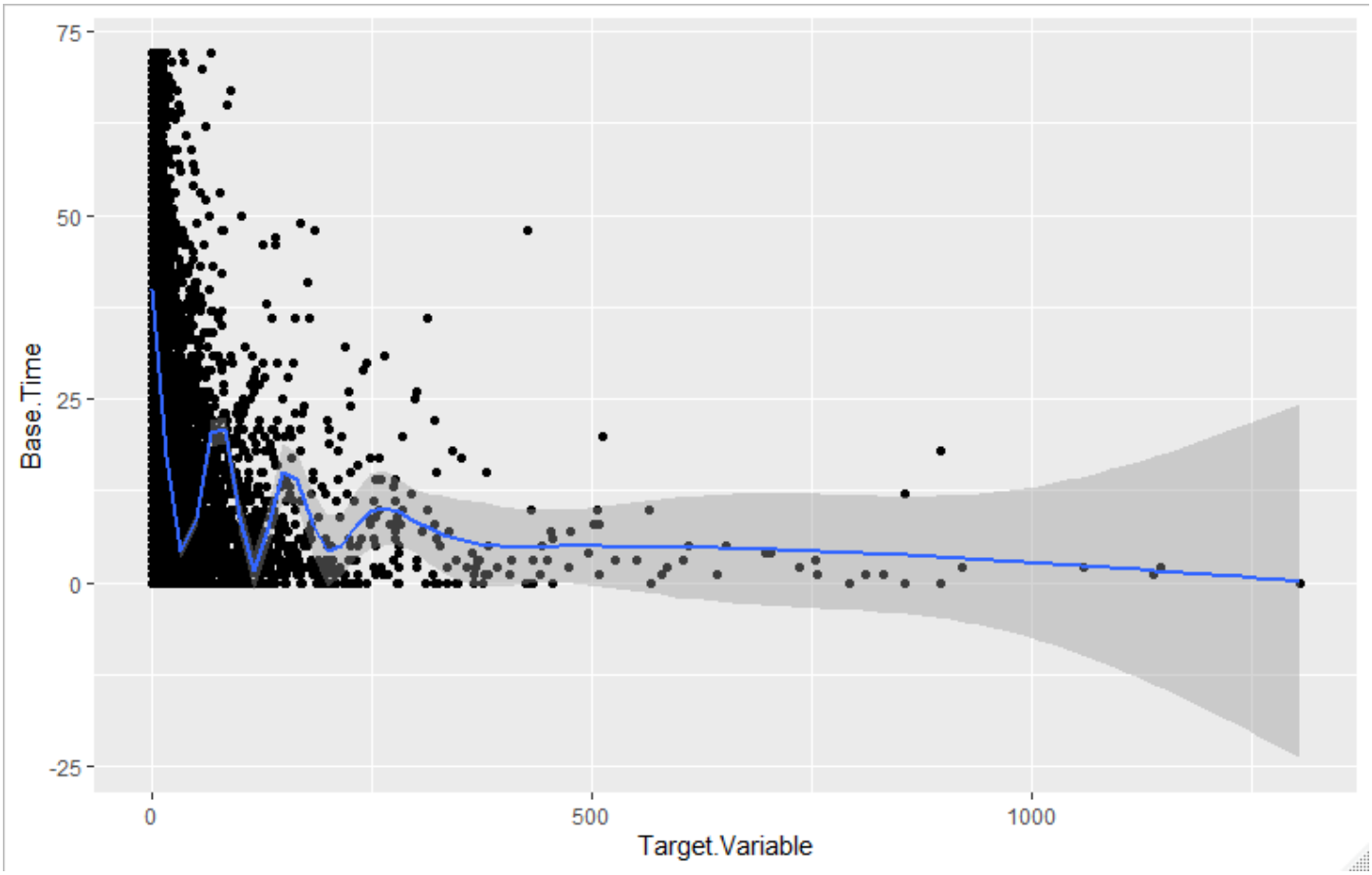

Figure 9 - Target Variable vs Base Time Scatterplot

There doesn't seem a significant correlation between the ***Target.Variable*** and ***Base.Time***, though howsoever little it seems negative.

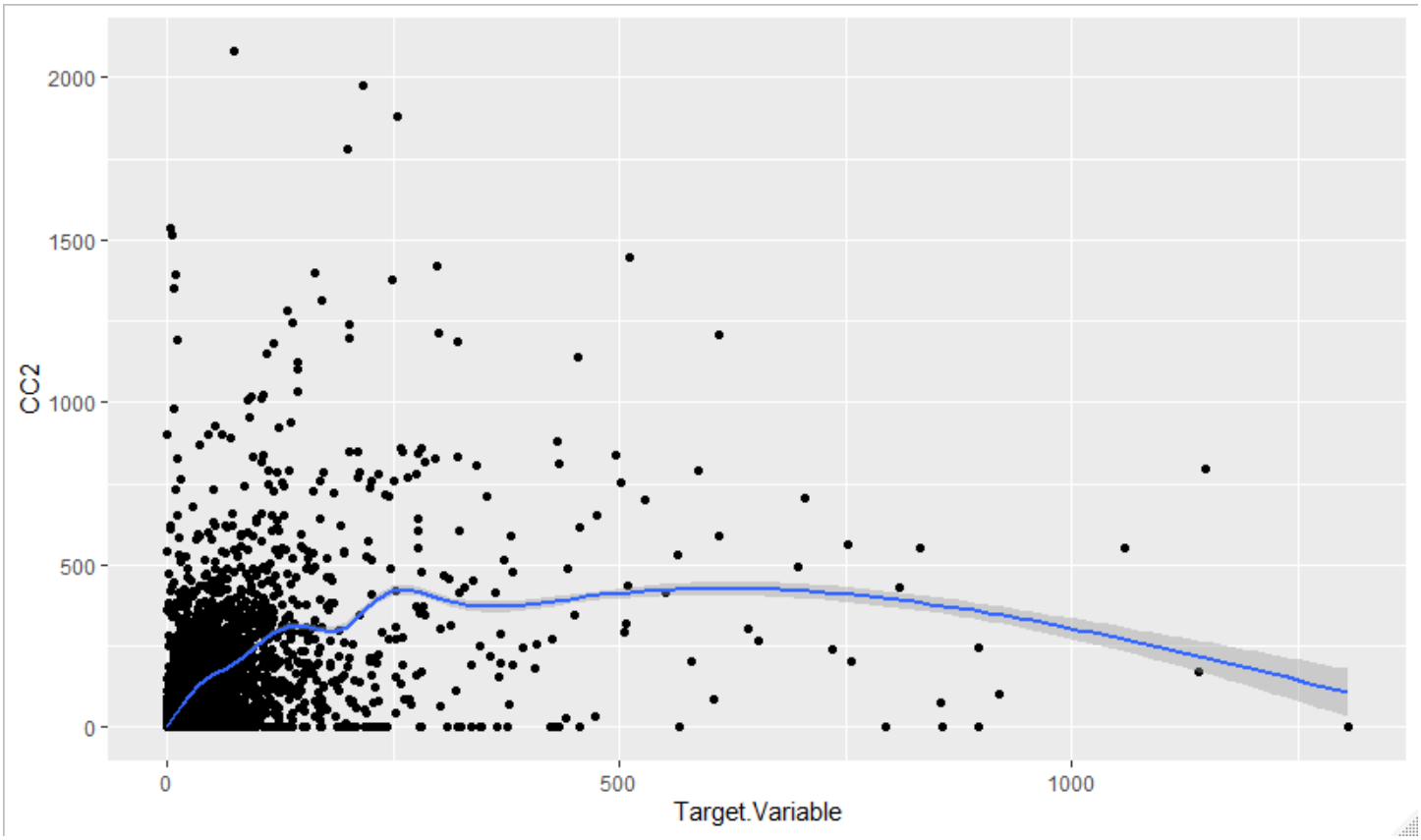

Figure 10 - Target Variable vs CC2 Scatterplot

Here too, the correlation is not significant.

Hence the ***Target.Variable*** doesn't seem to have any significant correlation with any independent variable.

5.2 Categorical and Categorical

Following are the two categorical variables in our dataset-:

- ***Base.DateTime.weekday***
- ***Post.published.weekday***

Let us construct a contingency table for them-:

Base.DateTime.weekday	Post.Published.weekday						
	Friday	Monday	Saturday	Sunday	Thursday	Tuesday	Wednesday
Friday	647	0	0	0	1515	902	1579
Monday	929	623	1452	1312	0	0	0
Saturday	1599	0	649	0	1583	0	911
Sunday	1638	0	1493	592	948	0	0
Thursday	0	903	0	0	646	1587	1846
Tuesday	0	1593	843	1352	0	728	0
Wednesday	0	1574	0	787	0	1703	825

Also performing a chi square test, we get-:

```
Pearson's Chi-squared test

data:  dataset$Base.DateTime.weekday and dataset$Post.published.weekday
X-squared = 31196, df = 36, p-value < 2.2e-16
```

We have got a p-value much less than the significance level of 0.05, we reject the null hypothesis and conclude that the t wo variables are in fact dependent.
Hence both the categorical variables of the dataset are dependent or correlated to each other.

6 IMPORTANT VARIABLES

We find out the important variables using caret package.

	Importance
Page.likes	6.23225111
Page.Checkins	2.93601645
Page.talking.about	3.58850773
Page.Category	0.89920270
Feature.5	**8.32721636**
Feature.6	2.91096900
Feature.7	6.10237088
Feature.8	0.46649210
Feature.9	3.14257141
Feature.10	**8.09762745**
Feature.11	0.85914180
Feature.12	2.54473769
Feature.13	0.65219586
Feature.14	0.77368580
Feature.15	3.26173687
Feature.16	1.23249722
Feature.17	2.00378427
Feature.18	1.64860458
Feature.19	1.38304244
Feature.20	**8.23558310**
Feature.21	2.89123301
Feature.22	6.44276690
Feature.23	0.57802152
Feature.24	3.22691251
Feature.25	0.18980996
Feature.26	1.35860569
Feature.28	7.01685818
Feature.29	0.54940612
CC1	0.92327256
CC2	**26.33749738**
CC3	4.40923710
CC4	1.79784129
Base.Time	**13.42034221**
Post.Length	0.46868541
Post.Share.Count	**26.88490067**
H.local	2.75344394
Post.published.weekdayMonday	0.04558898
Post.published.weekdaySaturday	0.77908198
Post.published.weekdaySunday	0.56820613
Post.published.weekdayThursday	0.56357408
Post.published.weekdayTuesday	0.06437791
Post.published.weekdayWednesday	0.48232013
Base.DateTime.weekdayMonday	0.10946409
Base.DateTime.weekdaySaturday	1.35561801
Base.DateTime.weekdaySunday	0.34350292
Base.DateTime.weekdayThursday	0.56112139
Base.DateTime.weekdayTuesday	0.33333577
Base.DateTime.weekdayWednesday	0.24040437

We find the above highlighted variables as most important.

- It seems that ***Post.Share.Count*** is the most important variable. It is obvious in the real world that the greater are the number of shares, the greater are the number of comments on it.

- ***CC2*** is the 2nd most important variable. Therefore, it is clear that the number of comments in the last 24 hours is a major determination factor.
- ***Base.Time*** also is an important variable. The time from which the shares and the comments are being counted, thus, holds importance.
- ***Feature.5*** and ***Feature.10*** seem to be more important than other feature variables. The reason for that is unknown.

7 Some Valuable Insights

Let us examine the categorical variables of the dataset which are ***Page.Category***, ***Post.published.weekday*** and ***Base.DateTime.weekday.***

Post.published.weekday

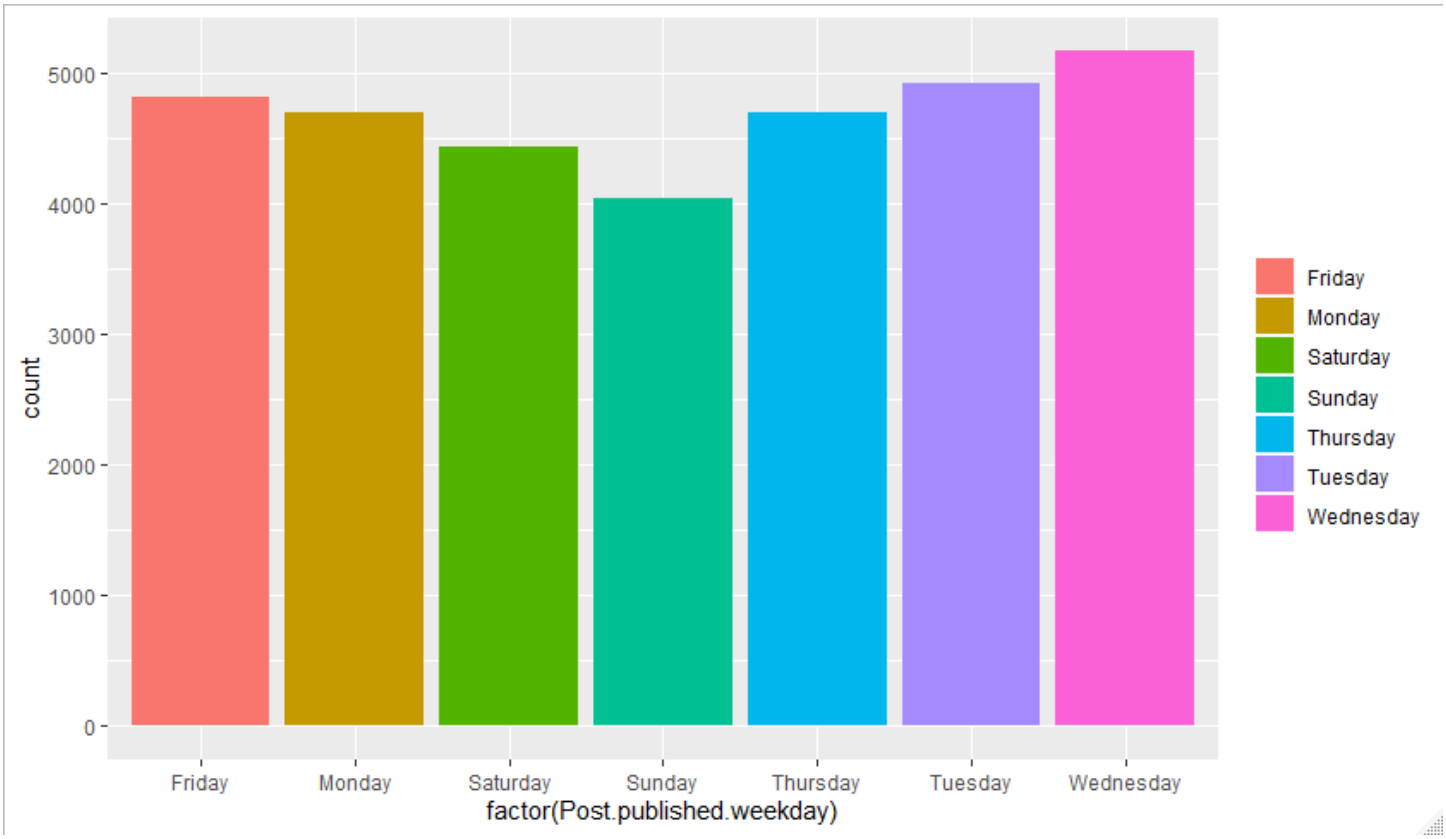

Figure 11 – Count of Post.Published.weekday Barchart

The greatest number of posts were published on Wednesday, i.e in the middle of the week. Hence most people are busy on social media on Wednesday.

Page.Category

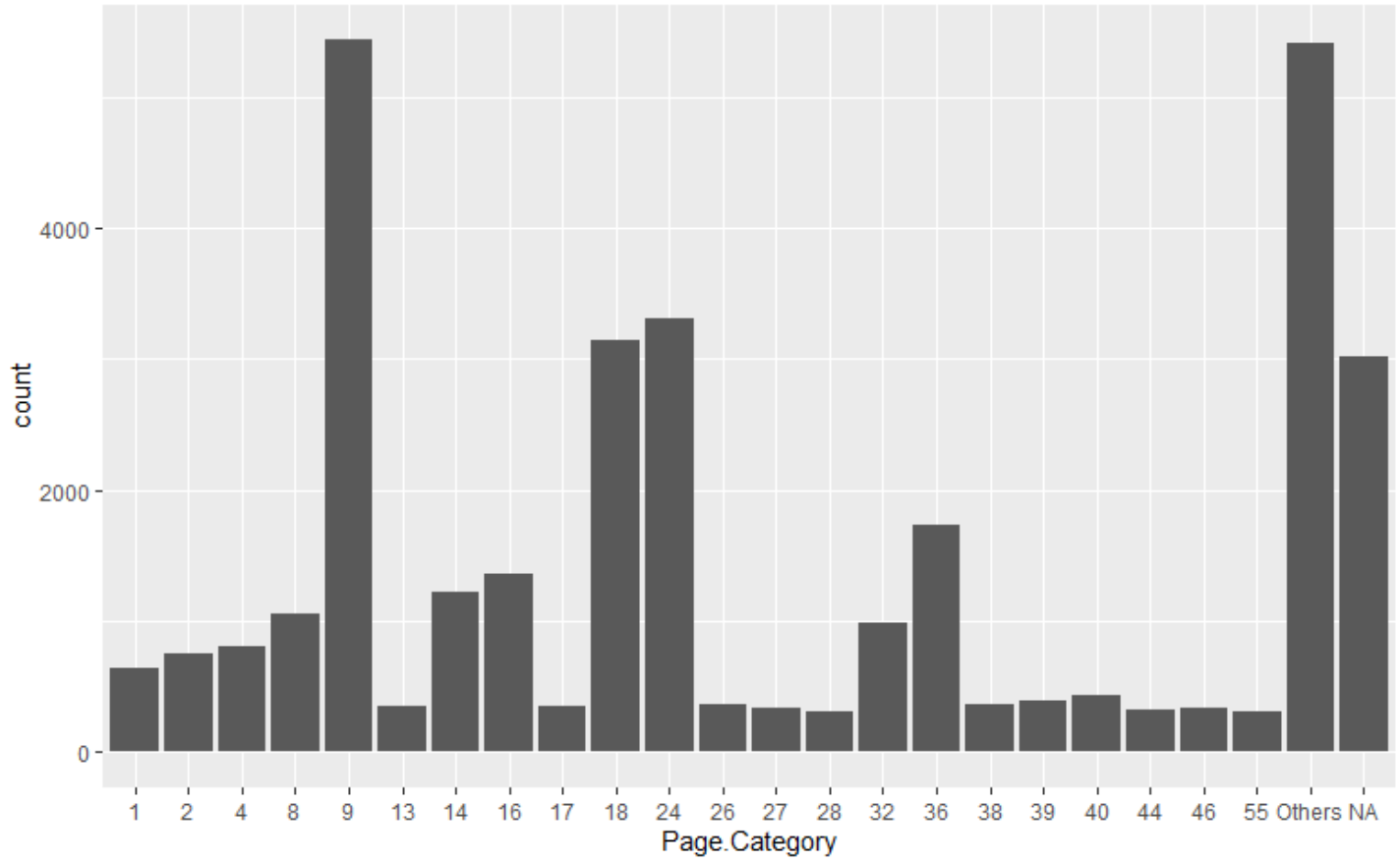

Figure 12 – Count of Page.Category barchart

We see the greatest number of the Facebook pages are of the **Category 9. Category 24** and **18** are also prominent categories among the Facebook pages.

Base.DateTime.weekday

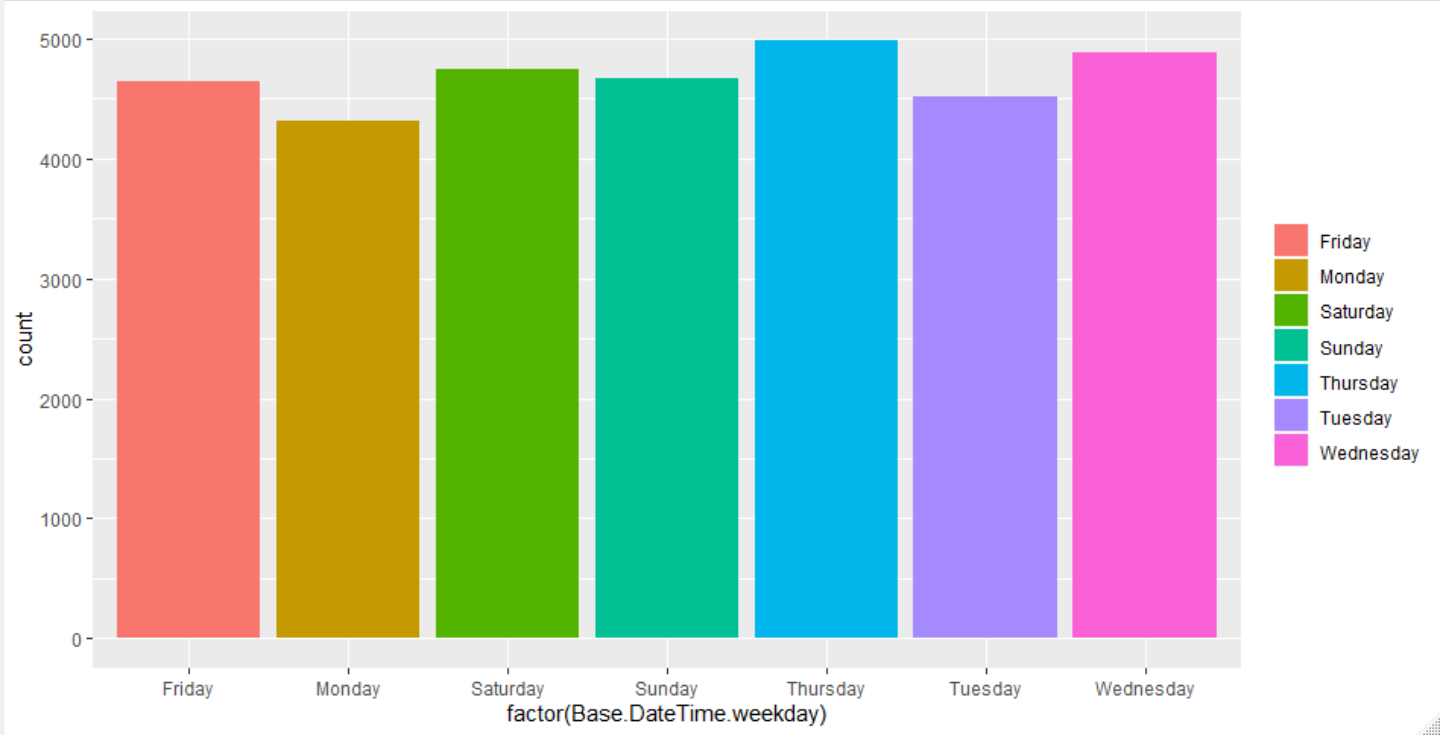

Figure 13 - Count of Base.Datetime.weekday barchart

Thursday is the base weekday for the greatest number of posts.

Let us also examine some important numeric variables-:

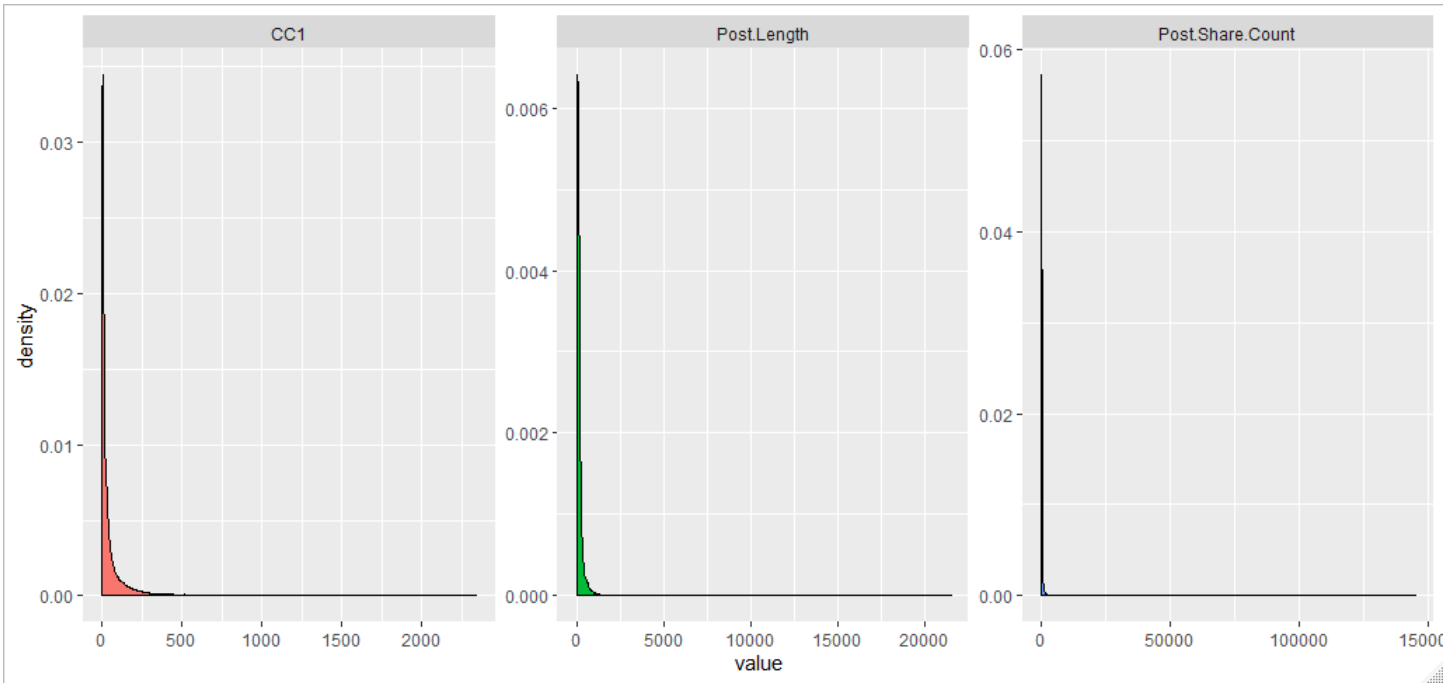

Figure 14 - Density plot of important numeric variables

- ***CC1,*** which is the total number of comments before selected base time is less than 150 for most of the posts. Very few posts have total number of comments as greater than 150.
- Most number of posts have ***Post.Length*** as less than 1000.
- Very few posts have a significant ***Post.Share.Count.*** A lot of posts have no shares at all.

8 DATA PRE-PROCESSING

8.1 REMOVAL OF UNWANTED VARIABLES

First of all, let us remove the variable ***Post.Promotion.Status*** as it has only one value. Therefore, it is a constant variable and hence we don't need it in our model.

Also we shall remove the ***ID*** variable as it has no in the analysis.

8.2 VARIABLE TRANSFORMATION

8.2.1 Page.Category

The variable is ***Page.Category*** is given as numeric variable in the original dataset. But actually, it should be a category variable as it gives the information about the category of the Facebook page. Therefore, we shall convert it to a category variable.

But the variable has 81 levels. These are too many levels for a categorical variable in a dataset. To avoid overfitting, we need to reduce the number of levels to around 20 to 25.

The logic followed to reduce the number of levels in the above-mentioned variable would be to group levels having frequency less than 10% (levels occurring less than 10% in the dataset) into one level by the name "Other". After doing so we get 24 levels in the ***Page.Category*** variable.

8.2.2 Normalizing Variable

Making the relationship between the independent and dependent variables linear can help us create efficient Machine Learning models.

If we are using models like Random Forest and Gradient Boosting, we shall normalize only the dependent variable as these models are quite robust to outliers and skewed distributions in the independent variables.

But if we are using models like Linear Regression, CART and Neural Networks, normalizing the independent variables will required to build a better model.

We use **yeo-johnson** method for transforming variables instead of **box-cox** method, because some variables have negative values and box-cox doesn't work on negative values.

Before transformation-:

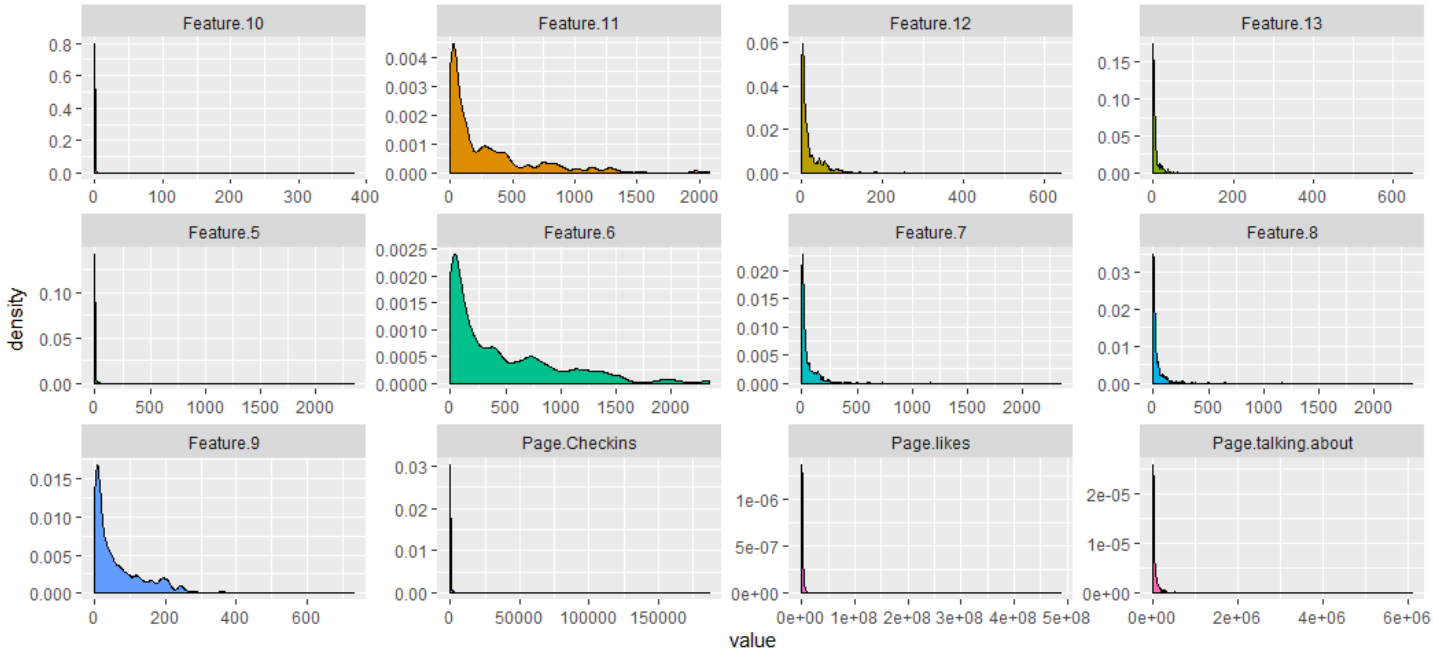

Figure 15 - Density plots before transformation (1-12)

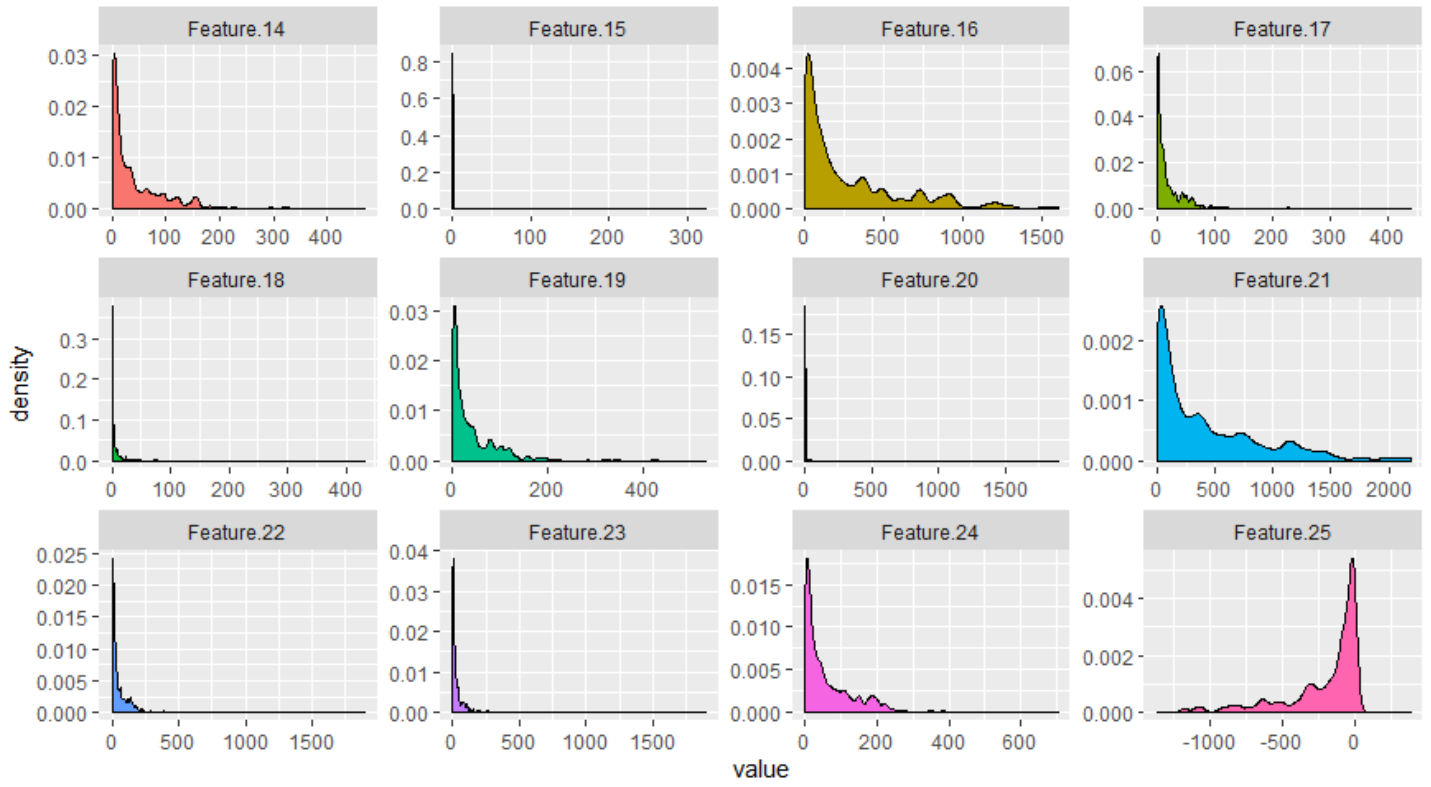

Figure 16 - Density plots before transformation (13-24)

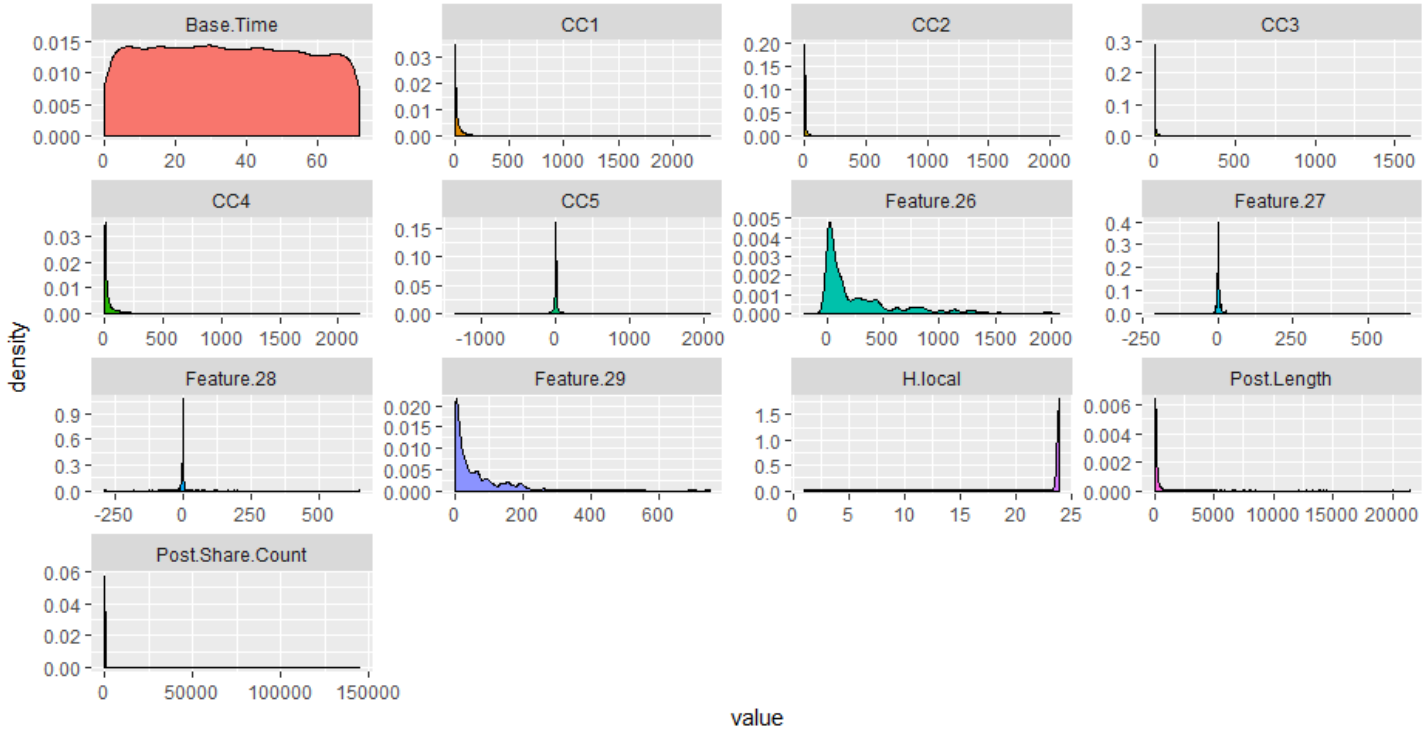

Figure 17 - Density plots before transformation (25-37)

After transformation-:

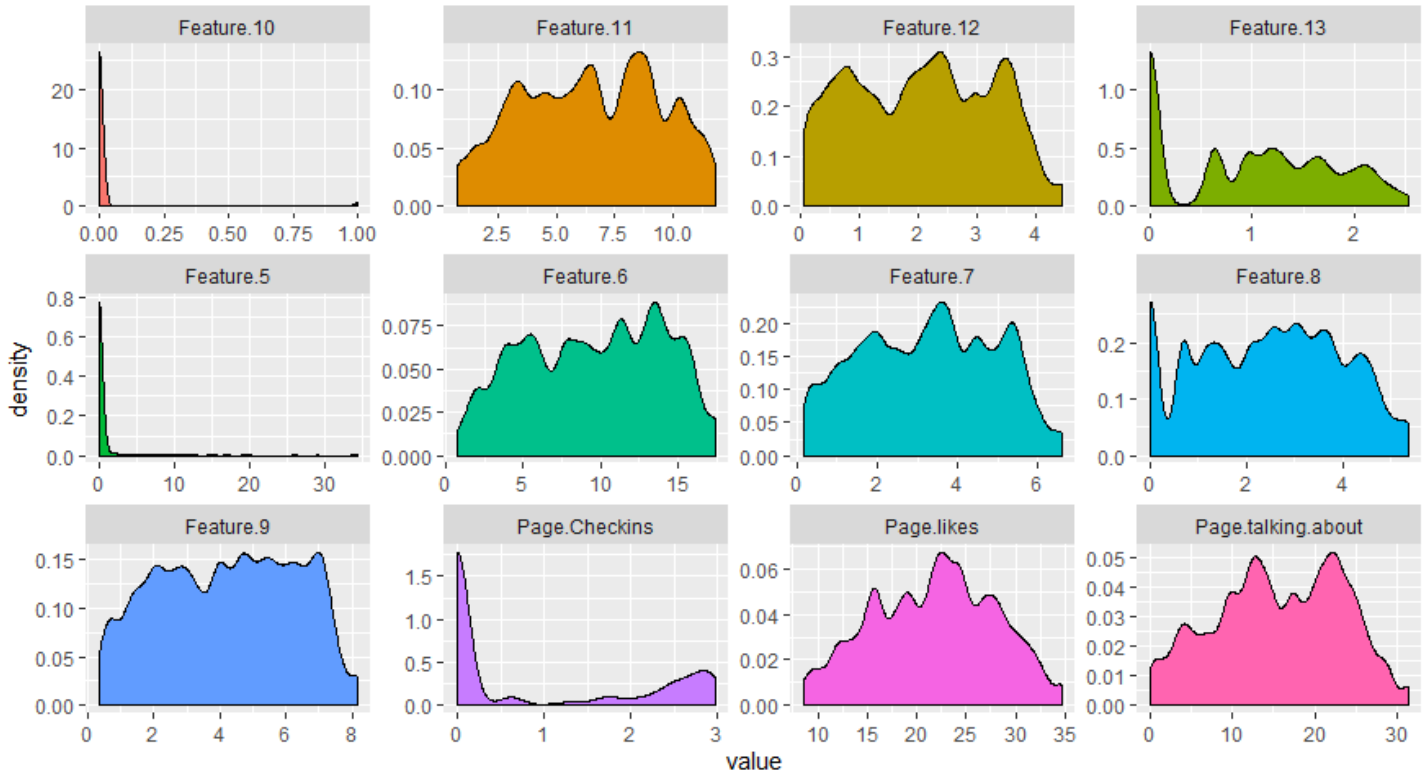

Figure 18 - Density plots after transformation (1-12)

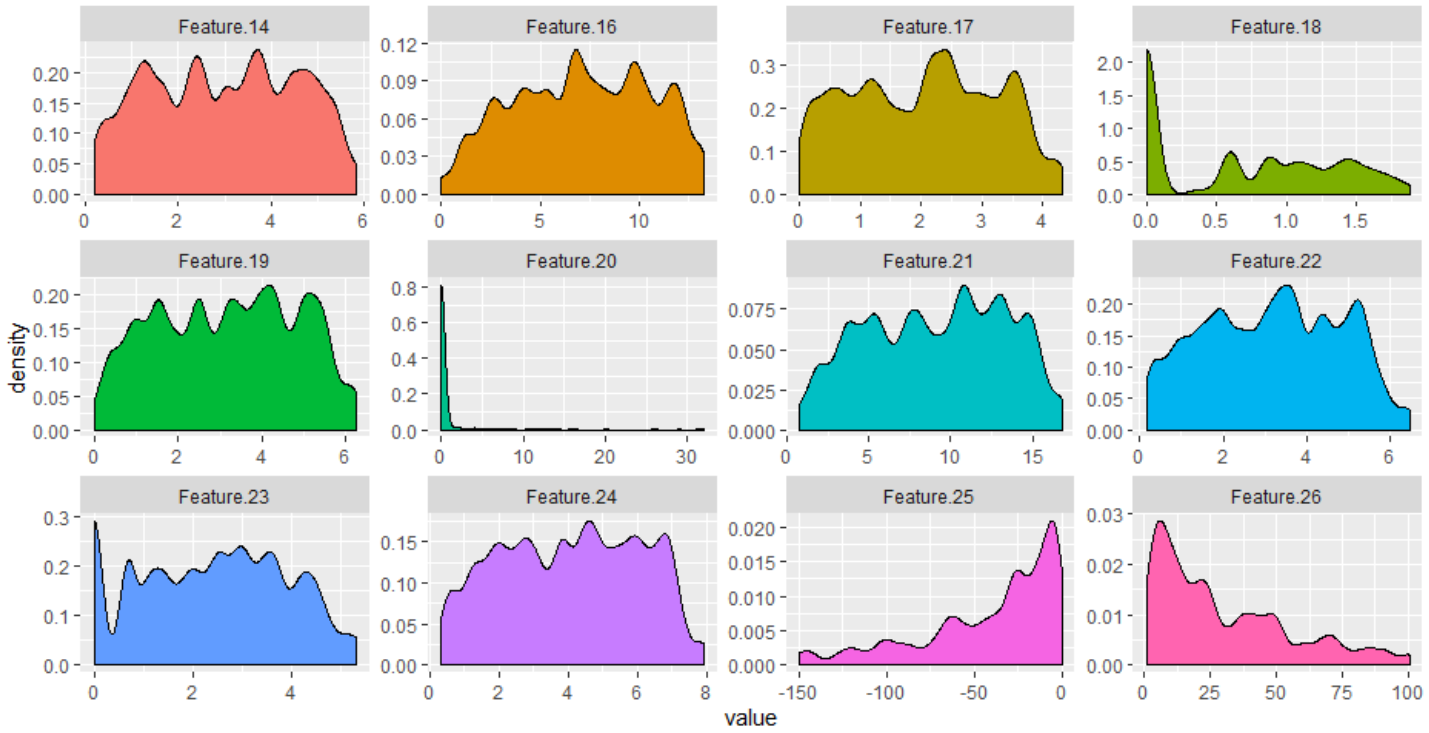

Figure 19 - Density plots after transformation (13-24)

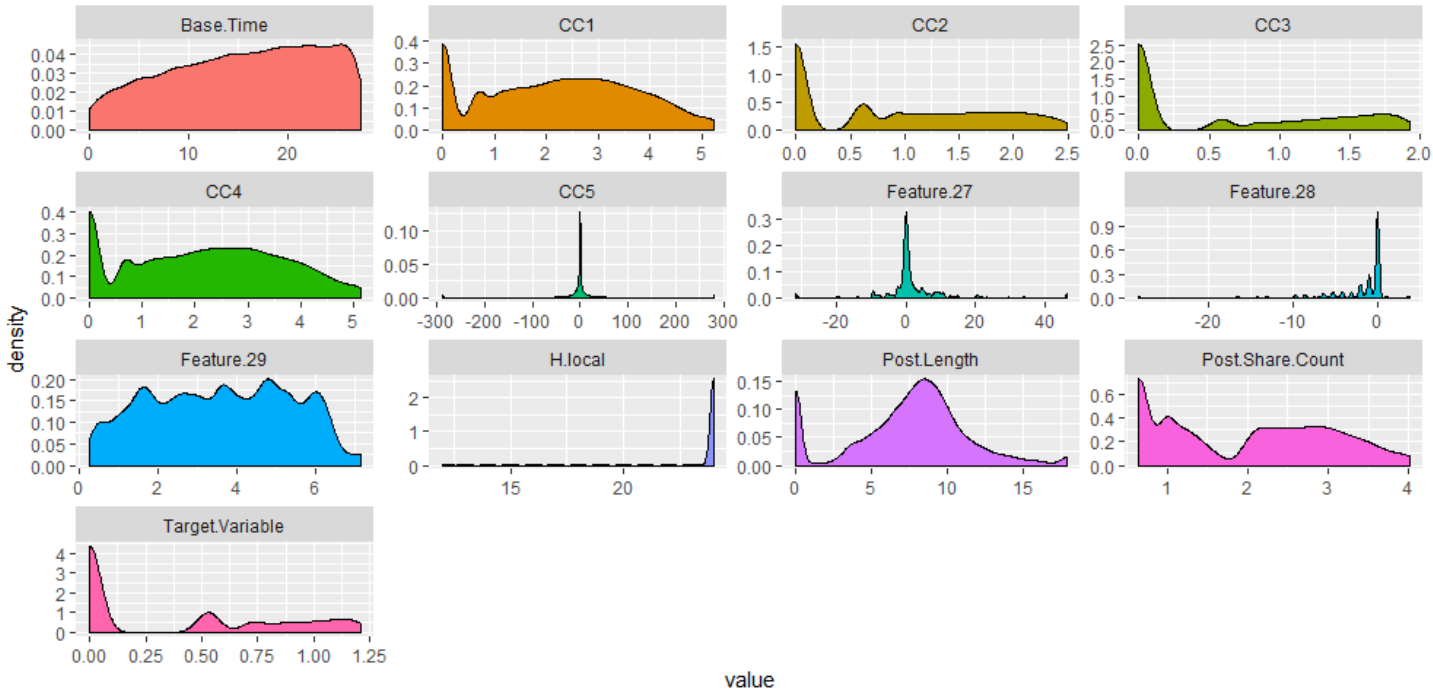

Figure 20 - Density plots after transformation (25-37)

We see that we could achieve normalization of some variables to a certain extent, whereas we could see no effect on certain variables. Thus, we are going to use the above normalized variables for model building.

8.3 Outlier Treatment

Let us look at the boxplots of the variables to find outliers-:

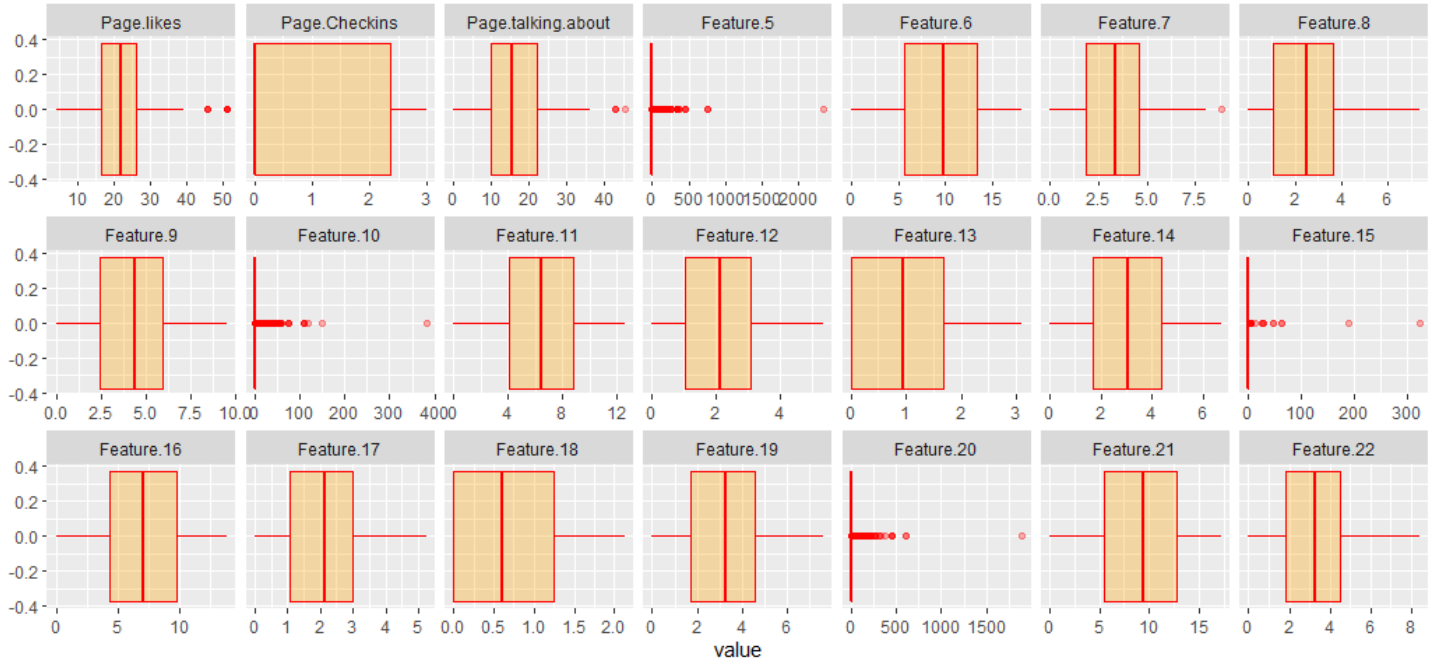

Figure 21 - Boxplots before outlier treatment (1-21)

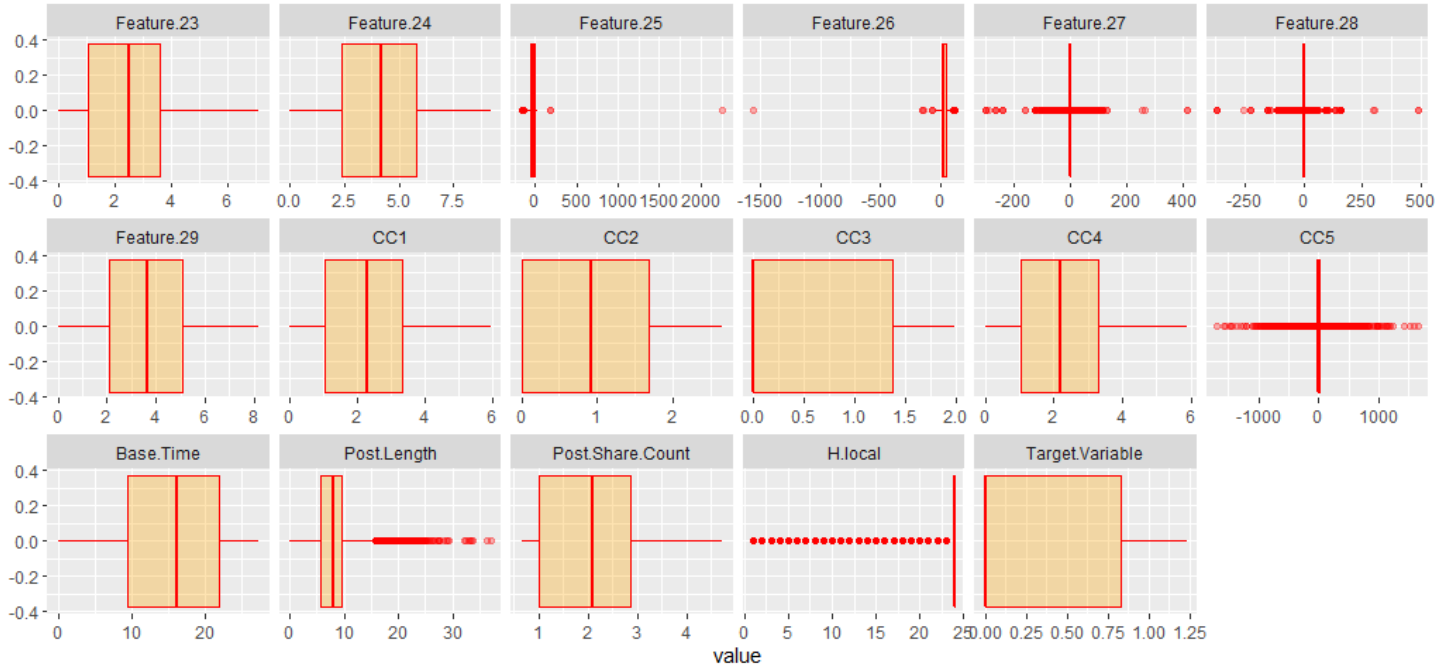

Figure 22 - Boxplots before outlier treatment (22-38)

For this, we shall remove all values which are less than 1 percentile and more than 99 percentiles of the total values for each column.

After completing the above-mentioned operation, we shall again see the boxplots and the density plots of the variable

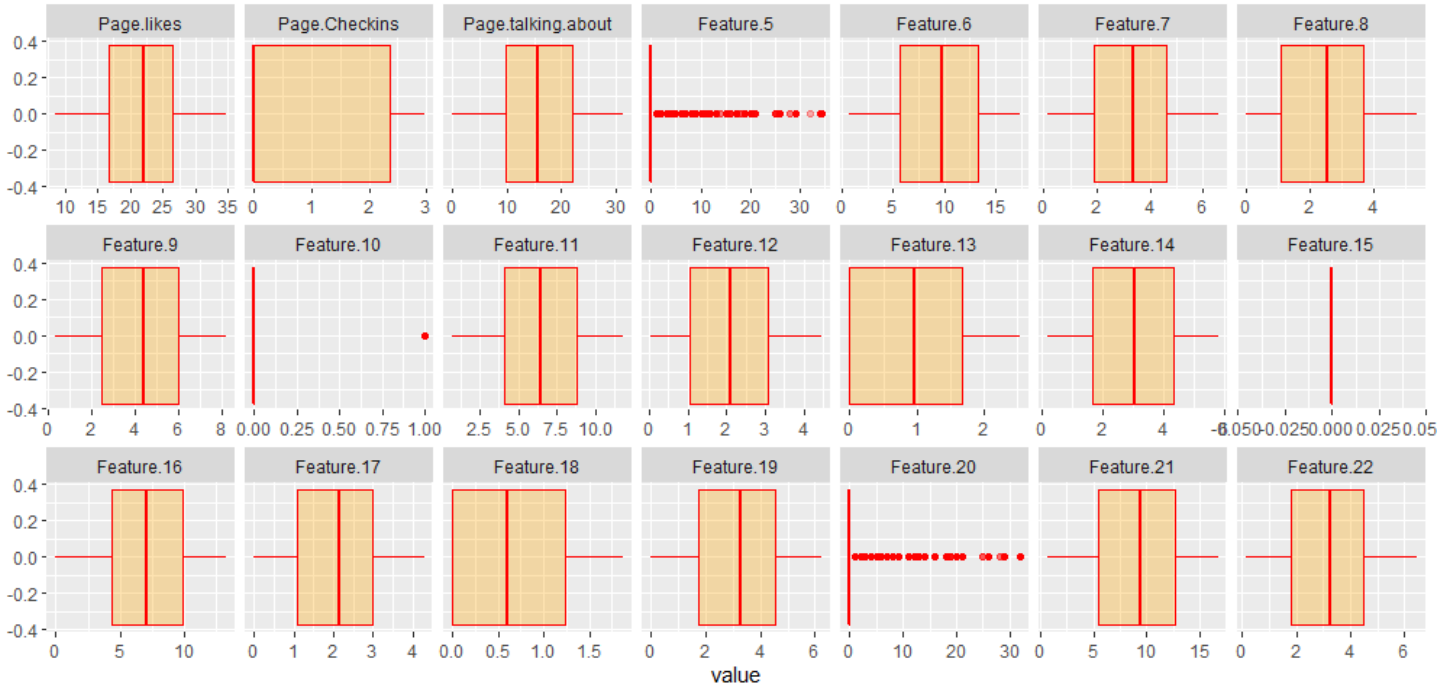

Figure 23 - Boxplots after outlier treatment (1-21)

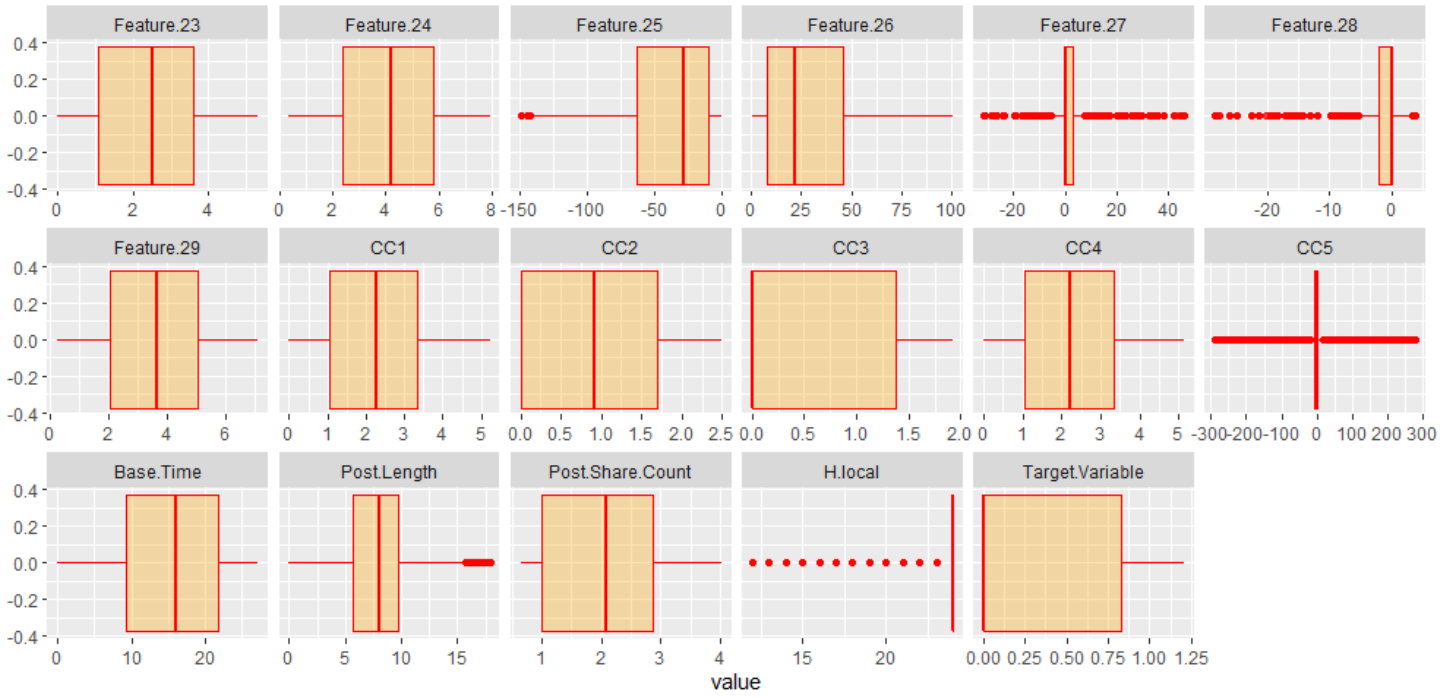

Figure 24 - Boxplots after outlier treatment (22-38)

Here we can see a significant improvement in the behavior of the variables. The isolated values have disappeared and the variables now behave a little more normally. We also see that ***Feature.15*** has become a constant variable. We shall remove it.

8.4 Missing Values Treatment

Let us see how many values are missing in the dataset-:

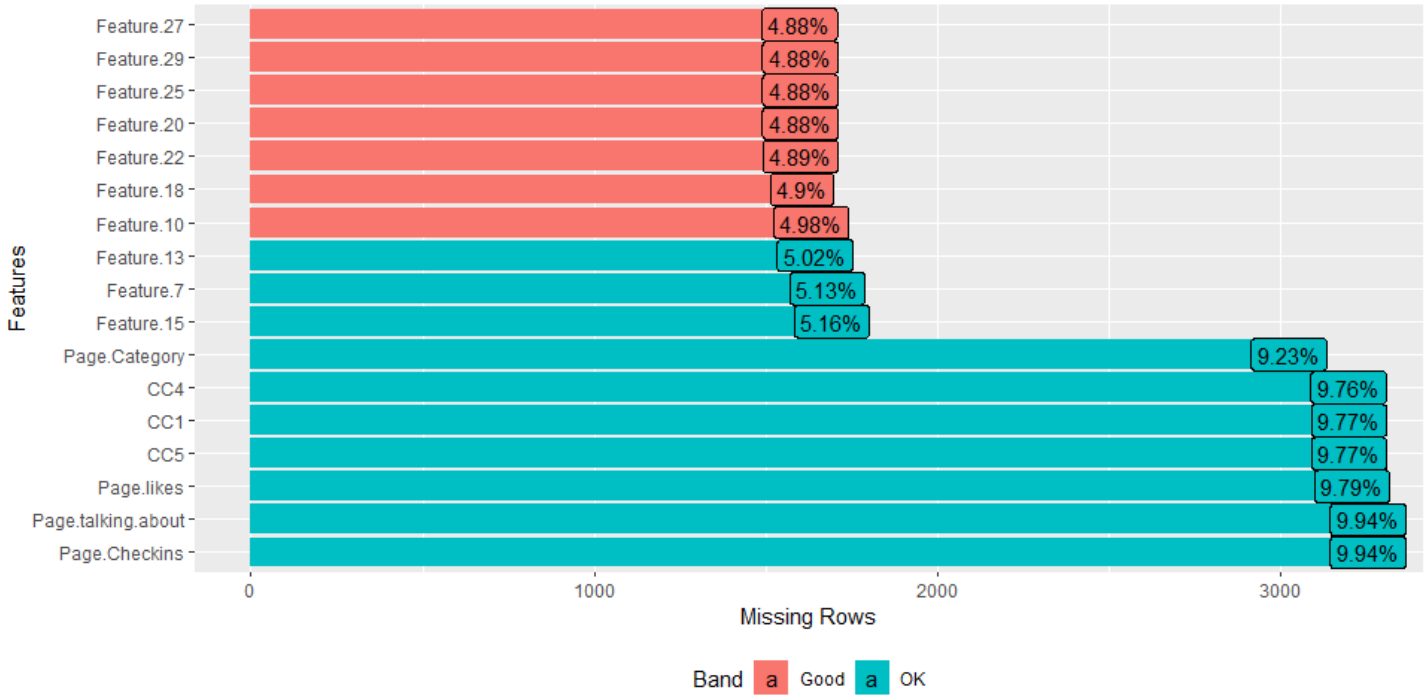

Figure 25 - Missing Values

We have **17 variables** with **38589 rows** which have missing values. A lot of the variables have more than 9% missing values. Three of them have around 5% missing values. All others have less than 5% missing values. We need to impute these values.

For imputation of these missing values, we will use **aregImpute** command from **Hmsic** package. The advantage of this function is that it allows mean imputation using additive regression, bootstrapping, and predictive mean matching.

In bootstrapping, different bootstrap resamples are used for each of multiple imputations. Then, a flexible additive model (non parametric regression method) is fitted on samples taken with replacements from original data and missing values (acts as dependent variable) are predicted using non-missing values (independent variable).

Then, it uses predictive mean matching (default) to impute missing values. Predictive mean matching works well for continuous and categorical (binary & multi-level) without the need for computing residuals and maximum likelihood fit.

Here are some important points to note about this function-:

1. It assumes linearity in the variables being predicted.
2. Fisher's optimum scoring method is used for predicting categorical variables.

We shall do 5 imputations here and pick the values obtained after predictive mean matching.

We get the following summary of the imputation model we used-:

```
Multiple Imputation using Bootstrap and PMM

Number of NAs:
         Page.Category Post.published.weekday  Base.DateTime.weekday            Page.likes
                  3024                      0                     0                  3208
         Page.Checkins     Page.talking.about                Feature.5               Feature.6
                  3255                   3255                     0                     0
             Feature.7                Feature.8                Feature.9            Feature.10
                  1679                      0                     0                  1632
            Feature.11               Feature.12               Feature.13            Feature.14
                     0                      0                  1643                     0
            Feature.16               Feature.17               Feature.18            Feature.19
                     0                      0                  1605                     0
            Feature.20               Feature.21               Feature.22            Feature.23
```

```
      1600                0             1601                0
Feature.24      Feature.25      Feature.26      Feature.27
         0            1600               0            1598
Feature.28      Feature.29             CC1             CC2
         0            1600            3199               0
       CC3             CC4             CC5       Base.Time
         0            3198            3200               0
Post.Length Post.Share.Count         H.local  Target.Variable
         0               0               0               0
```

```
Transformation of Target Variables Forced to be Linear

R-squares for Predicting Non-Missing Values for Each Variable
Using Last Imputations of Predictors
     Page.Category        Page.likes   Page.Checkins Page.talking.about          Feature.7
             0.492             0.775             0.331             0.839             1.000
        Feature.10        Feature.13        Feature.18        Feature.20        Feature.22
             0.630             0.965             0.896             0.998             1.000
        Feature.25        Feature.27        Feature.29               CC1               CC4
             0.942             0.641             0.994             0.996             0.996
               CC5
             0.385
```

Let us match the density plots of the original values with the predicted values to check if the predicted values are near to the original values or not. We will check it for ***Page.Checkins*** and ***CC5*** as the R-squares is lowest for them.

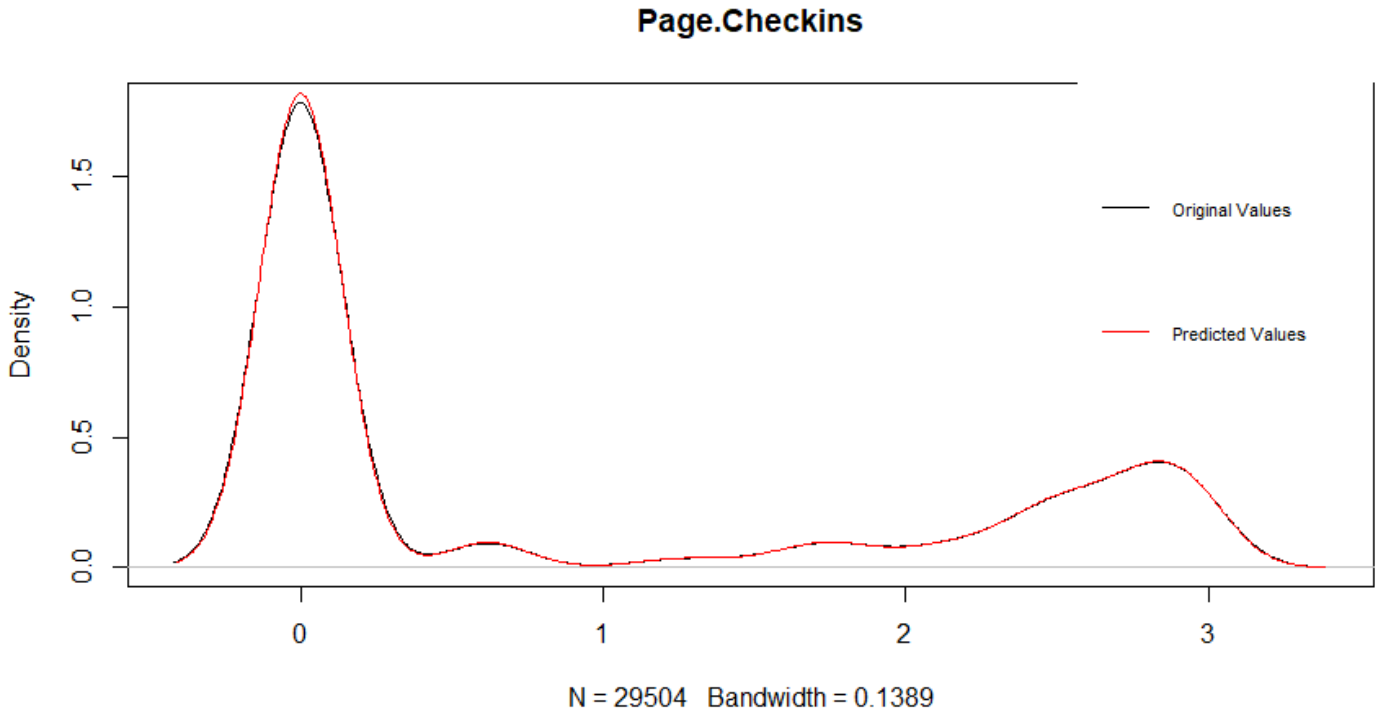

Figure 26 - Density plot comparison of original and predicted NA values of Page.Checkins

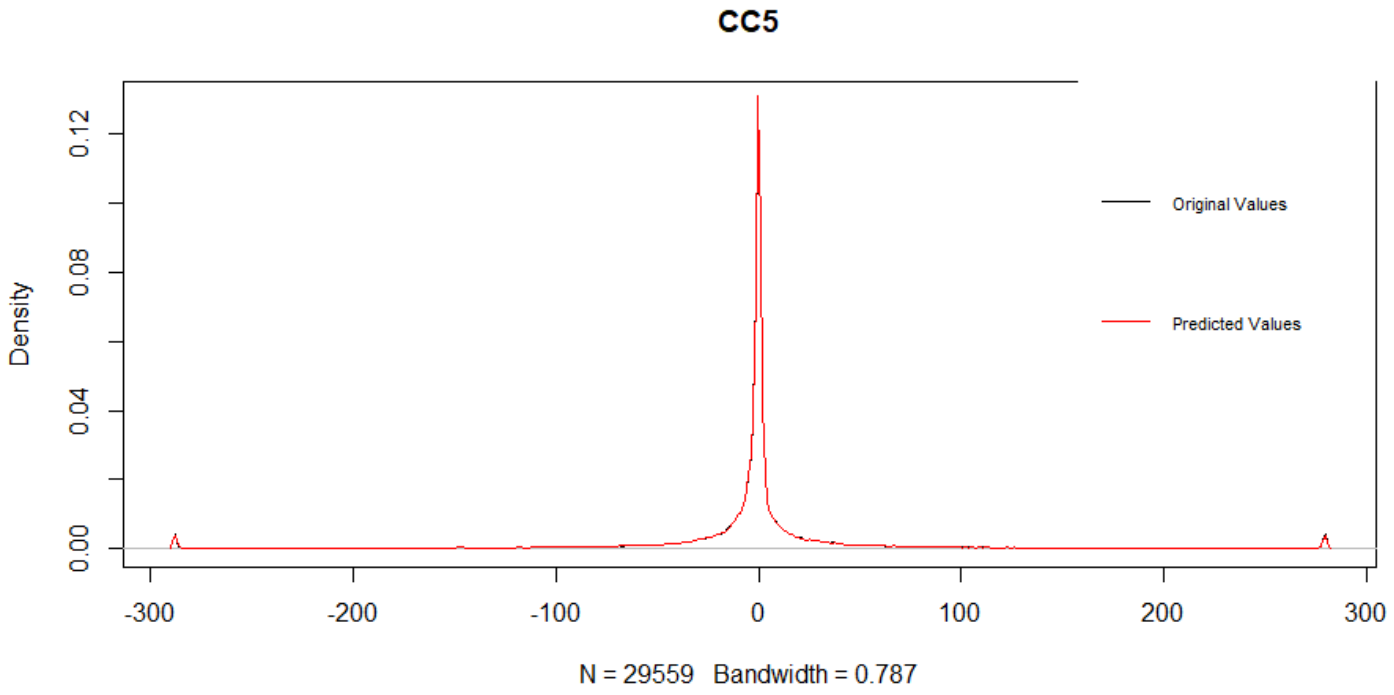

Figure 27 - Density plot comparison of original and predicted NA values of CC5

In both the above graphs, the black and the red line superimpose on each other

Here we see that; the imputation of missing values was good.

9 Modelling Process

9.1 Models to be used

Here we have to predict the number of Facebook comments on a post. Hence it is a regression problem. Following models seem appropriate to use-:

- **Linear Regression** - In statistics, linear regression is a linear approach to modeling the relationship between a scalar response (dependent variable) and one or more explanatory variables (independent variables). In linear regression, the relationships are modeled using linear predictor functions whose unknown model parameters are estimated from the data. Such models are called linear models.
- **XGBoost (Extreme Gradient Boosting)** - Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees. It builds the model in a stage-wise fashion like other boosting methods do, and it generalizes them by allowing optimization of an arbitrary differentiable loss function.
- **Random Forest** - Random forests or random decision forests are an ensemble learning method for classification, regression and other tasks that operates by constructing a multitude of decision trees at training time and outputting the class that is the mode of the classes (classification) or mean prediction (regression) of the individual trees. Random decision forests correct for decision trees' habit of overfitting to their training set.

We see that all the above algorithms use different approaches.

Neural Networks *algorithm was also used but due to the highly skewed nature of the data, a satisfactory model couldn't be built.*

9.2 REMOVING MULTICOLLINEARITY

Also, we need to check the presence of multicollinearity in the dataset. We shall use **Variance Inflation Factor (VIF)** to check and remove multicollinearity in the dataset.

Before finding VIF, we got to know there are some aliases or perfectly correlated variables. We need to remove them before we check VIF.

- ***Feature.27*** seems to be derived from and as a result, perfectly correlated with ***Feature.12*** and ***Feature.17***. We need to remove ***Feature.12*** and ***Feature.17.***
- ***CC5*** seems to be derived from and as a result, perfectly correlated with ***CC2*** and ***CC3***. We need to remove ***CC2*** and ***CC3.***

Following values of VIF we got for each variable in a linear model-:

```
      Page.likes      Page.Checkins Page.talking.about      Page.Category
        1.935855           1.207917           2.972881           1.141809

       Feature.5          Feature.6          Feature.7          Feature.8
      660.192412         469.368604        5314.843576         983.691452

       Feature.9         Feature.10         Feature.11         Feature.13
     2097.695865           6.490887         226.059706          15.311694

      Feature.14         Feature.15         Feature.16         Feature.18
      209.141696           4.848674          88.378255          12.934371

      Feature.19         Feature.20         Feature.21         Feature.22
      120.811249         638.637506         462.034807        4876.214994

      Feature.23         Feature.24         Feature.25         Feature.26
      861.591287        2029.242907          55.883974         187.586166

      Feature.27         Feature.28         Feature.29                CC1
       16.318243           5.252181         190.506122         240.118244

             CC4                CC5          Base.Time       Post.Length
      237.764400           1.372332           1.086071           1.010135

Post.Share.Count            H.local
        1.457652           1.044551
```

Hence there is a definite presence of multicollinearity. We shall this problem by factor analysis.

9.2.1 Factor Analysis

We shall first check if factor analysis can be performed on the dataset. To do that, we will perform **KMO test.**

Results of the KMO test-:

```
Kaiser-Meyer-Olkin factor adequacy
Call: KMO(r = modelData)
Overall MSA =  0.93
MSA for each item =
        Page.likes      Page.Checkins Page.talking.about          Feature.5
              0.97               0.69               0.97               0.77
         Feature.6          Feature.7          Feature.8          Feature.9
              0.88               0.95               0.95               0.88
        Feature.10         Feature.11         Feature.12         Feature.13
              0.95               0.92               0.94               0.96
        Feature.14         Feature.16         Feature.17         Feature.18
              0.93               0.94               0.95               0.97
        Feature.19         Feature.20         Feature.21         Feature.22
              0.93               0.78               0.88               0.95
        Feature.23         Feature.24         Feature.25         Feature.26
              0.95               0.88               0.94               0.92
        Feature.27         Feature.28         Feature.29                CC1
```

0.67	0.96	0.98	0.92
CC2	CC3	CC4	CC5
0.93	0.91	0.93	0.55
Base.Time	Post.Length	Post.Share.Count	H.local
0.46	0.91	0.99	0.90

Overall MSA = 0.93, which is large enough to confirm that Factor Analysis can be performed on the dataset.

Let us now decide the number of factors to which all variables are to be converged by looking at the scree plot and the variance explained-:

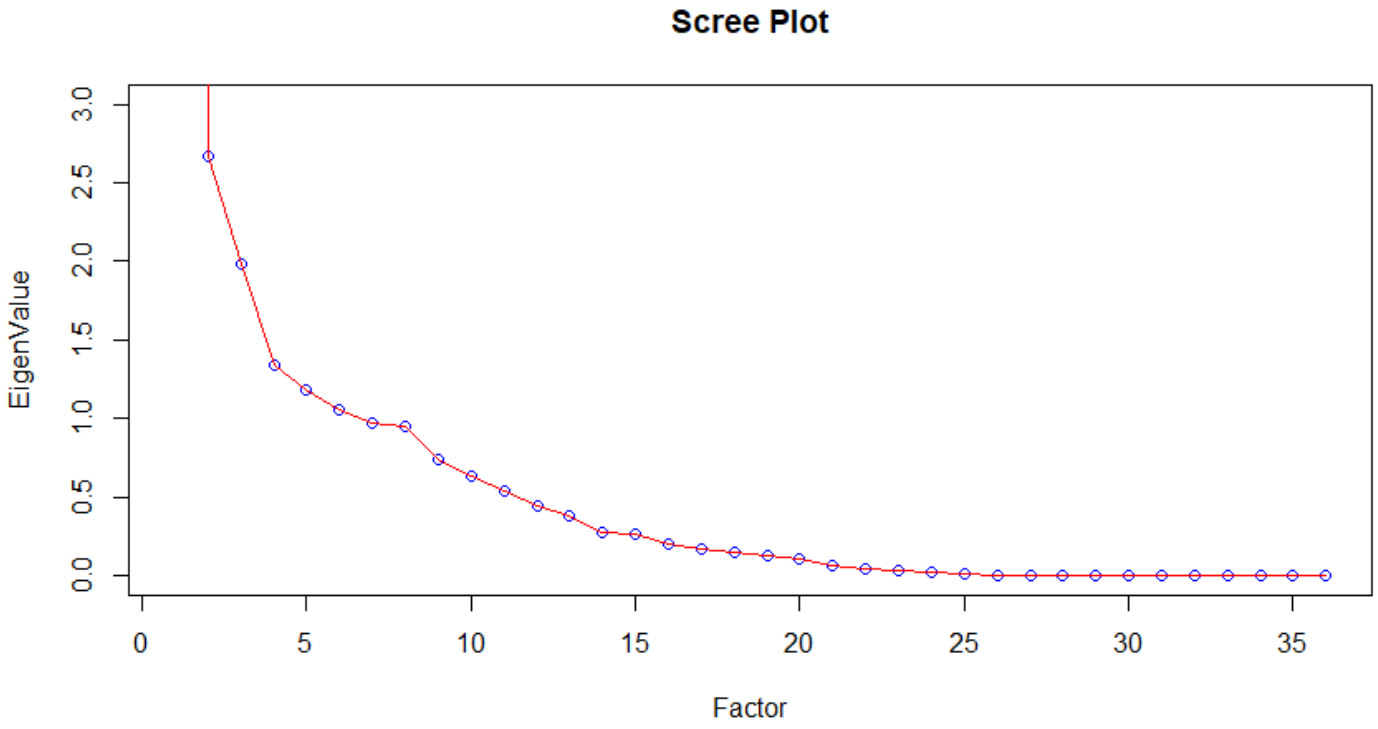

Figure 28 - Scree plot for Factor Analysis

The ideal number of factors seems to be 6.

	PC1	PC2	PC3	PC4	PC5	PC6
Proportion Var	0.600	0.074	0.055	0.037	0.033	0.029
Cumulative Var	**0.600**	**0.674**	**0.729**	**0.766**	**0.799**	**0.828**
Proportion Explained	0.724	0.089	0.067	0.045	0.040	0.036
Cumulative Proportion	0.724	0.813	0.880	0.925	0.964	1.000

Also, we see that the cumulative variance explained by 6 factors is around 83% which is sufficient. Therefore, we rotate the 6 factors and obtain the following Factor Analysis diagram-:

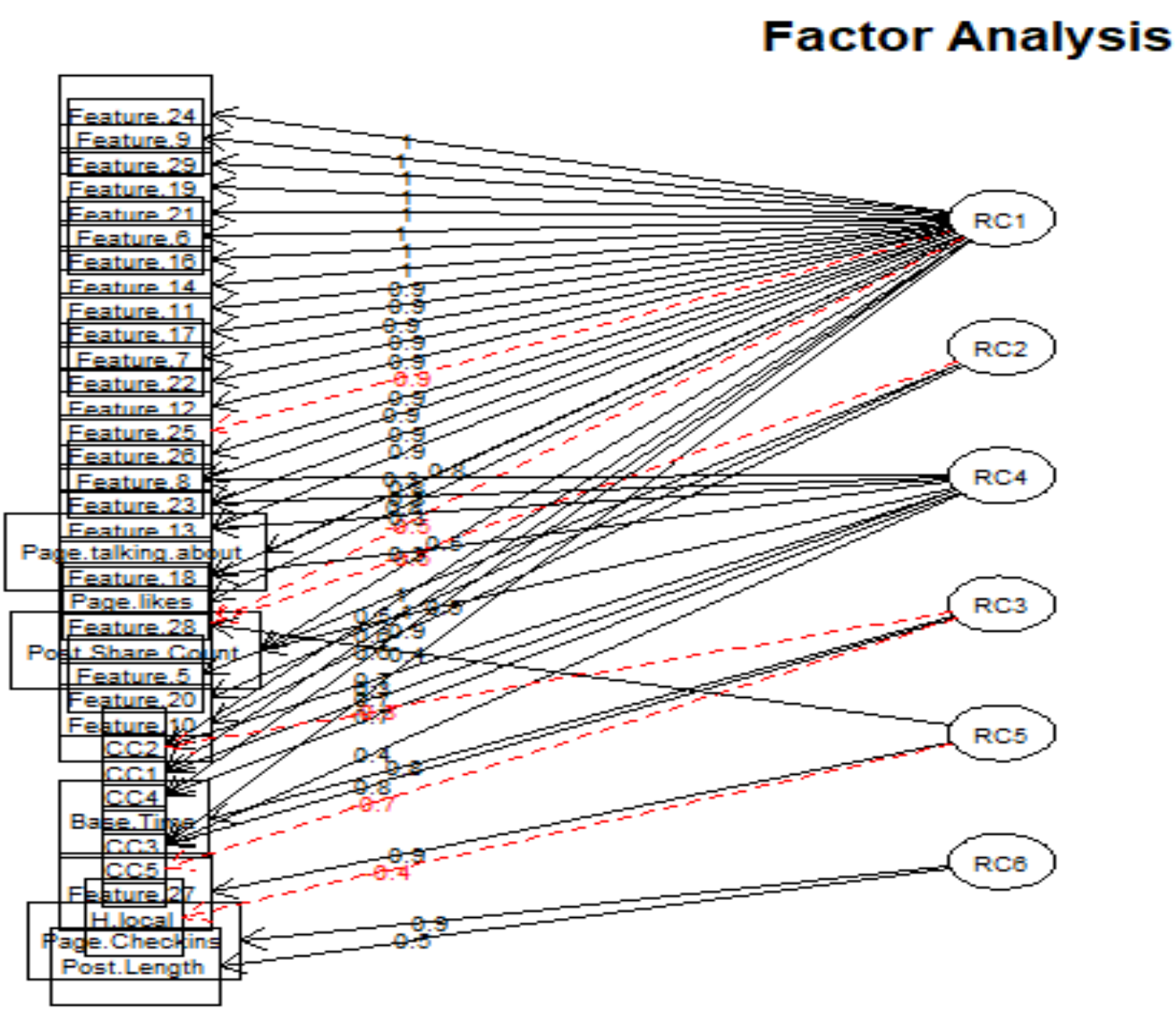

Figure 29 - Factor Analysis Diagram

9.3 Dummy Variables

We had 3 categorical variables-:

Variable	No. of Levels
Page.Category	24
Post.published.weekday	7
Base.DateTime.weekday	7

Table 4 - Categorical Variables with number of levels

We shall convert them all into dummy variables. The number of dummy variables will be k-1 per categorical variable where k is the number of levels of that categorical variable.

9.4 SPLITTING OF DATASET

The dataset will be split into 3:1 ratio.

Train Data	22949 rows
Test Data	9810 rows

Table 5 - Train and test data bifurcation

9.5 LINEAR REGRESSION

Following is the linear model we ran-:

```
Call:
lm(formula = fmla, data = ptrain)


Residual standard error: 0.291 on 22908 degrees of freedom
Multiple R-squared:  0.5804,   Adjusted R-squared:  0.5796
F-statistic:   792 on 40 and 22908 DF,  p-value: < 2.2e-16
```

Let us now apply the model on the test dataset and find out the values of some metrics.

Metric	Value
MAE *(Mean Absolute Error)*	0.23
MSE *(Mean Square Error)*	0.08
RMSE *(Root Mean Square Error)*	0.29
RMSLE *(Root Mean Square Log Error)*	0.21

Table 6 - Linear Regression Model Metric Values

- We got Adj. R-squared as 58%. Hence model is average in its quality. This is due to the highly skewed nature of the data. MAPE is also very high due to same reasons.
- We got RMSE as 0.29 which is pretty good. The reason we have got such a good RMSE is because we had transformed the dataset according to Yeo-Johnson transformation.

9.6 XGBOOST

The XGBoost model was run with the following parameter values -:

- Nrounds = 400
- ETA = 0.1
- Max.Depth = 7
- Min_child_weight=3
- Nfold=3

Here the 1st 3 parameters are the most influential in the model performance.

Following is the metric result of XGBoost Model-:

Metric	Value
MAE *(Mean Absolute Error)*	0.18
MSE *(Mean Square Error)*	0.06
RMSE *(Root Mean Square Error)*	0.25
RMSLE *(Root Mean Square Log Error)*	0.19

Table 7 - XGBoost Model Metric Values

Definitely, this model is better than linear regression in all the metrics.

9.7 RANDOM FOREST

Following is the Random Forest model we ran-:

```
Call:
 randomForest(x = ptrain[, -1], y = ptrain[, 1], ntree = ntree,      do.trace = TRUE)
               Type of random forest: regression
                     Number of trees: 408
No. of variables tried at each split: 13
```

Metric	Value
MAE *(Mean Absolute Error)*	0.18
MSE *(Mean Square Error)*	0.06
RMSE *(Root Mean Square Error)*	0.25
RMSLE *(Root Mean Square Log Error)*	0.19

Table 8 - Random Forest Model Metric Values

This model is better than linear regression and very similar to XGBoost in all the metrics.

10 COMPARISON AMONG THE MODELS

Let us the models we have used so far-:

METRIC	VALUE		
	Linear Model	**XGBoost**	**Random Forest**
MAE *(Mean Absolute Error)*	0.23	0.18	0.18
MSE *(Mean Square Error)*	0.08	0.06	0.06
RMSE *(Root Mean Square Error)*	0.29	0.25	0.25
RMSLE *(Root Mean Square Log Error)*	0.21	0.19	0.19

Table 9 - Comparison of different models

We clearly see that XGBoost is the best model we have got.

- The results show that the model performance is among the best.
- The time taken by the model is also very less than random forest.

Therefore, we shall consider the results of XGBoost model.

11 VARIABLE IMPORTANCE

Let us examine the most important factors in the XGBoost model. Following is the plot of importance of all the factors in the model-:

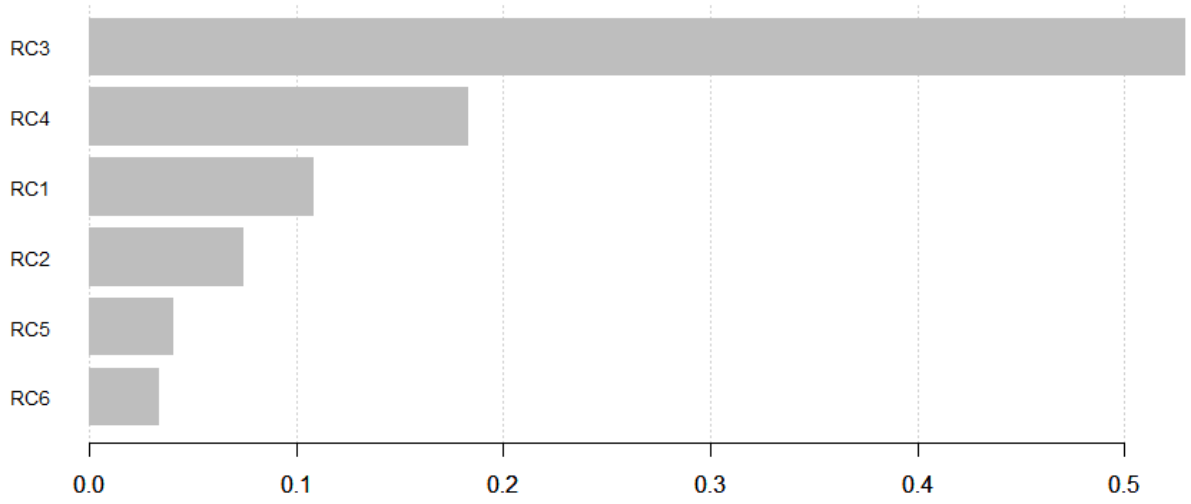

Table 10 -Variable Importance in XGBoost Model

The two most important factors, which are **RC3** and **RC4**, explain almost 70% of the importance. **RC3** explains 5.6% of total variance of all variables whereas **RC4** explains 7.7% of the same. **RC1** explains the most, i.e. 54.2% of the total variance of all variables and is 3rd most important factor in XGBoost regression model.

12 Business Insights

Marketers are always monitoring the user activity in order to drive engagement with the brand. One way to measure the user engagement is the volume of comments on a Facebook post. It is also a measure of how successful a digital marketing campaign was. It is clear from our analysis, that two factors are majorly important in determining the user engagement of the brand on Facebook.

- The number of comments in the last 24 hours of the time during when the Facebook post was published or advertised is the most important factor in determining the total volume of comments. Therefore, the marketers should focus on the comments in the last 24 hours, relative to the base date/time.
  This means that the marketers should carefully choose the day on which the post is being published. In our earlier analysis, we had found out that the posts published on Wednesday saw the most activity by the users. A reason for this maybe that the last 24 hours or the last day of these posts are weekends. Users are more active on weekends simply due to being free from their usual weekday work. Hence these posts see a lot of engagement in the last 24 hours. We can see that ***CC2*** has a positive estimate in the linear model. Thus more are the number of comment sin the last 24 hours, more is the total volume of comments.

- The ***Base.Time*** or the selected time in order to simulate the scenario is the 2nd most important variable. The ***Base.Time*** in other words is the time for which the Facebook marketing campaign ran. There is a negative relation between the ***Base.Time*** and the ***Target.Variable*** as the ***Base.Time*** has negative estimate in the linear model***.*** Marketers have to figure out an optimal time for which the scenario of the marketing campaign should be simulated.

- ***Post.Share.Count*** is also a factor which should be taken care by the digital marketers. A widely shared post may drive greater engagement with the customers which in turn may lead to a high volume of comment on the Facebook post. ***Post.Share.Count*** has positive coefficient in the linear model. Thus, more are the post shares, greater is the comments volume. Therefore, the marketers should make their posts as much shareable as possible.

Let us check the effect of certain key variables in the regression equation. We applied a linear model and checked the summary of that model. Following are the coefficients of some variables which can be understood-:

```
Coefficients:
                    Estimate
(Intercept)         3.150e-02
Page.likes         -4.432e-04
Page.Checkins      -7.685e-03
Page.talking.about -8.505e-04
Base.Time          -1.845e-02
Post.Length        -7.235e-04
Post.Share.Count    5.488e-02
H.local             1.012e-02
```

We draw the following inferences-:

- ***Page.likes*** is inversely proportional to the ***Target.Variable.*** More the number of ***Page.likes,*** less is the number o f comments on the post and vice-versa. The brands should first decide what is important between the likes and c omments as the users like to do only one of them.
- ***Page.Checkins*** is inversely proportional to the ***Target.Variable.*** More the number of ***Page.Checkins,*** less is the n umber of comments on the post and vice-versa. A user checks the brand page from the post if he/she hasn't hea rd about the brand. This means that the first-time viewers of the post who are unaware about the brand don't c omment on the post and rather check out the page. Less number of comments on a post can mean the brand is not known to many people and a post by such a brand can increase the first-time brand exposure.
- ***Page.talking.about*** is also inversely proportional to the ***Target.Variable.*** More the number of people talking ab out the page***,*** less is the number of comments on the post and vice-versa. This means the Facebook users will eit her comment on the post or they would talk about the brand somewhere else. Either ways, the brand is getting exposure.
- ***Post.Length*** is also inversely proportional to the ***Target.Variable.*** Greater is the length of the post***,*** less is the nu mber of comments on the post and vice-versa. Due to a low attention span of social media users nowadays, peo ple tend to skip longer posts and hence don't engage with the brand in that post. Hence it is always wiser to post shorter messages drive greater engagement with the audience. The long-written texts can be replaced by image s or videos to communicate more and more information in a shorter attention span.

Focusing on above factors, the businesses can gain a lot of exposure to their brands and in turn increase their profits.