# Capstone Project

## Facebook Comment Volume Prediction

Yash Malik

# Index:

# Introduction:

- Problem Statement:

The goal is to predict how many comments a user-generated post is expected to receive in the given set of hours.

- Need of the study/project:
    - Information like comments/hr can be used by marketing companies or digital marketing agencies to determine the kind of content that attracts more traffic
    - Such information can also be useful for psychologists, who are studying or trying to understand audience behavior.

- Understanding how data was collected in terms of time, frequency and methodology:
    - The data was collected over the time period of 1 to 24hr, some entries we can see the comments have been monitored for 24hrs whereas in some cases they were monitored only for 1-3 hours.
    - Data is also distributed between different days of the week.
- Visual inspection of data (rows, columns, descriptive details)
    - In the data set, there are 32759 obs. of  43 variables.

# EDA:

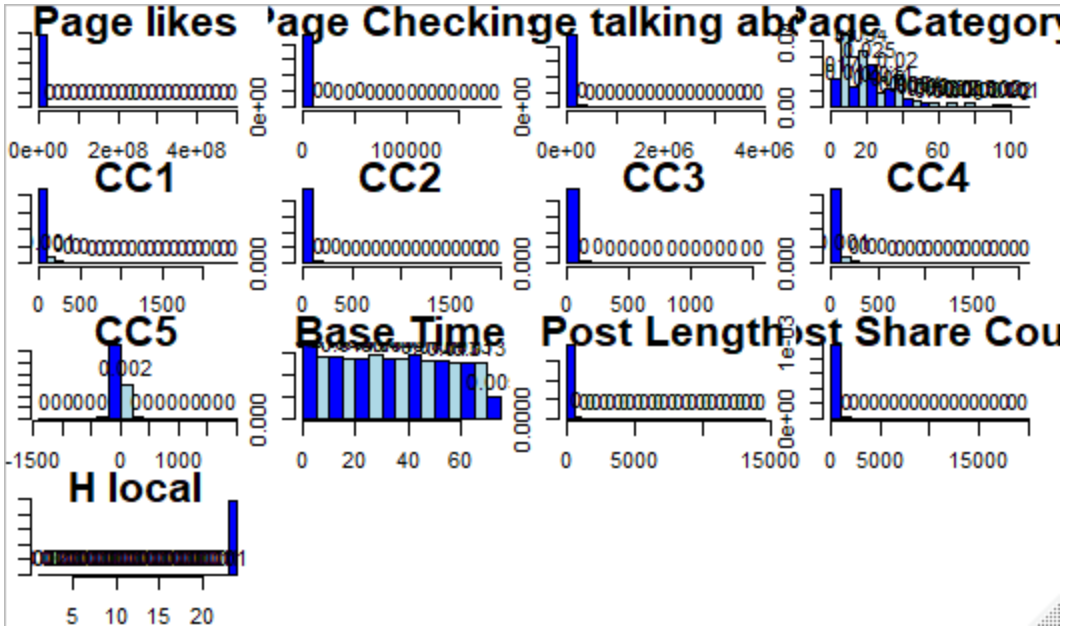
## Exploring and Understanding the features:

- Converting, Post Promotions, Post published weekday and Bas date-time weekday into Categorical variables

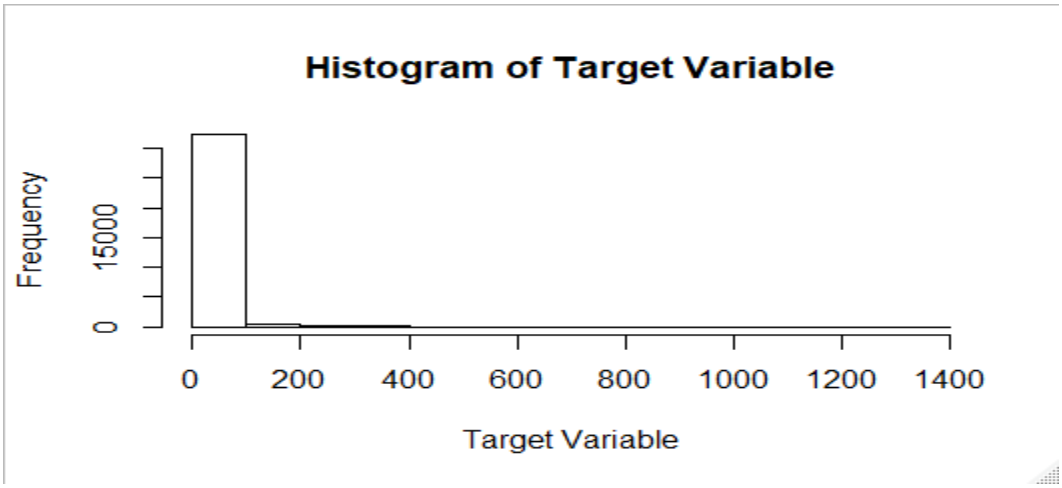
| Variable Name | Data Type |
|---|---|
| Page Popularity/likes | Numerical |
| Page Checkins | Numerical |
| Page talking about | Numerical |
| Page Category | Numerical |
| Feature 5 – Feature 29 | Numerical |
| CC1 | Numerical |
| CC2 | Numerical |
| CC3 | Numerical |
| CC4 | Numerical |
| CC5 | Numerical |
| Base time | Numerical |
| Post length | Numerical |
| Post Share Count | Numerical |
| Post Promotion Status | Categorical |
| H Local | Numerical |
| Post published weekday | Categorical |
| Base Date Time weekday | Categorical |
| Comments | Target Variable |

## Univariate Analysis:

- Distribution of all the continuous variables using the multi.hist():
  - Mostly all the variables are right-skewed
  - H local variable is left-skewed
  - Base Time looks evenly distributed



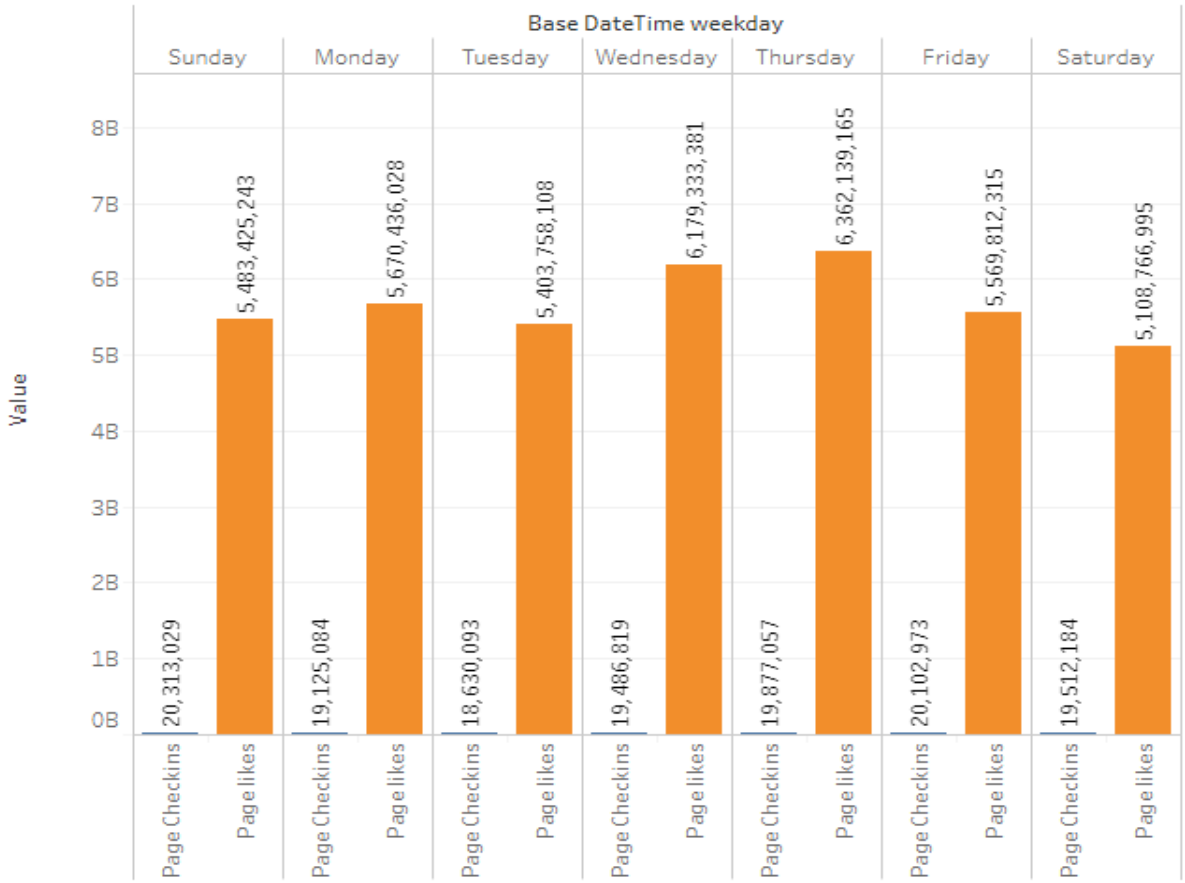

Distribution of Target Variable:




## Bi-Variate Analysis:

BaseDateTime weekday vs PageCheckins&Pagelikes:

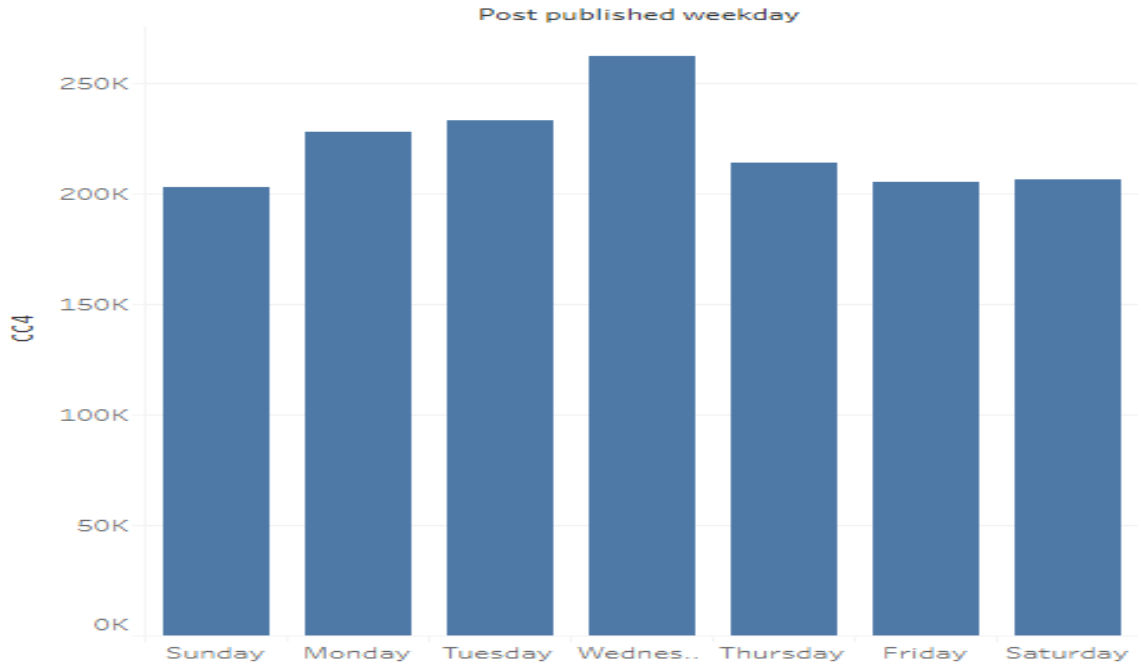
- Page Likes are Maximum on Thursday.
- Page Check Ins are maximum on Friday and Sunday.
- Clearly shows posts posted on Thursday are expected to receive more check ins and likes, which can eventually translate to more comments.




Post published weekday vs Page likes:

- This graphs shows similar trends, likes are highest on Thursday.


Post published weekday vs CC4 ():


Post published weekday
CC4
250K
200K
150K
100K
50K
0K
Sunday
Monday
Tuesday
Wednes..
Thursday
Friday
Saturday

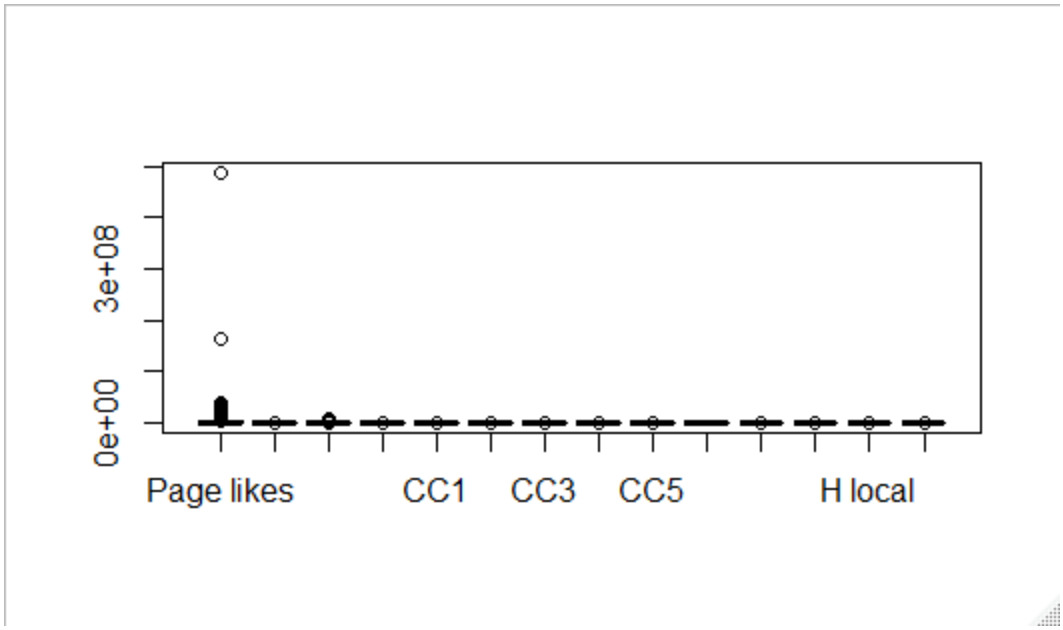
# Data Cleansing and Pre-Processing:

## Missing Values:

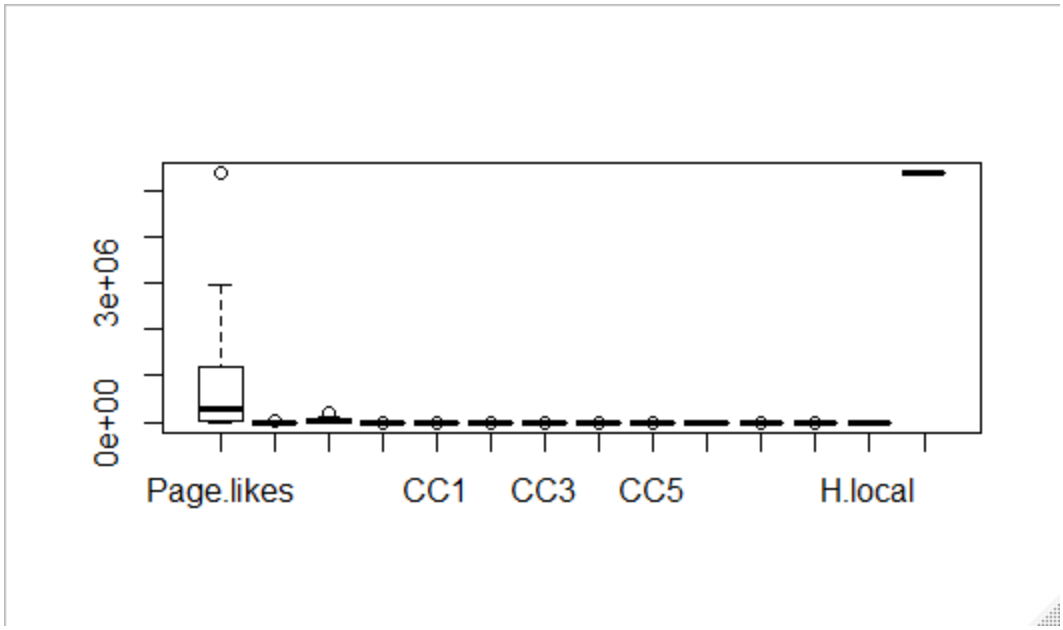
1. The data has 32,759 observations and 43 factors.
2. All the NA values have been removed, bringing the no. of observations down to 17,550. Using the na.omit() function.
3. Outliers will be treated using the iqr() function.

## Outlier Treatment:

Evident oulies in Page likes:

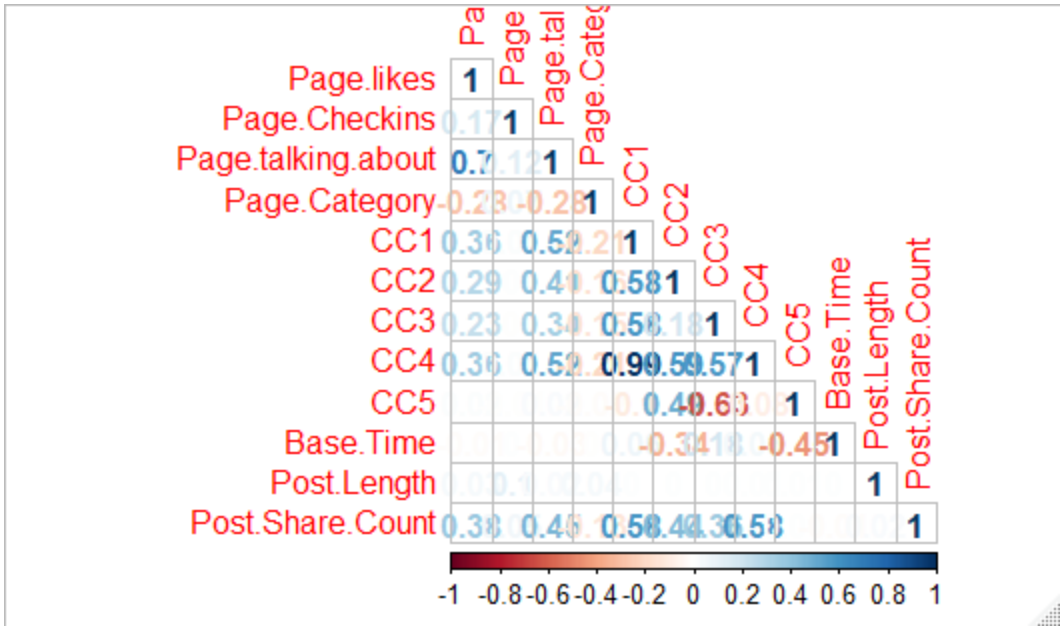



After removing outliers:



## Variable Transformation and Removal:

1. Features 5-29 and ID have been removed.
2. Post published weekday and Bas data-time weekday will not be used as converting them into categorical variables won't solve the problem. 14 new categorical variables are created, 7 for Post published weekday and 7 for Base data-time weekday, each indicating the day of the week with 0 or 1
3. CC4 has also been removed due to high correlation with CC1.



# Model Building:

We are treating our dependent variable to be a continuous variable, so we have a regression problem. The models we will use to predict are as follows:

- Multiple Linear Regression
- CART (Classification and Regression Trees)
- Random Forest

To compare the models amongst themselves we will look at Root Means Square Error(RMSE) and Mean Absolute Error (MAE).

We have split our data into Train and Test, for the train data we will remove certain variables and try to improve our models.

## Multiple Linear Regression:

We will first fit our model with all the variables in the train data (excluding Feature 5 - Feature29 , Post Promotion Status as all values are 0 and CC4 due to high correlation with CC1)

```
multimodel = lm(Target.Variable~.,traindata )
```

We get a very low R-squared value of 0.17 and looking at the variables, we can improve our model by removing the insignificant variables.

Multi Liner Model with Significant Variables:

```
### Removing factors with low importance
  multimodel2 =
lm(Target.Variable~Page.talking.about+Page.Checkins+CC1+CC3+CC5+Base.Time+Post.Share.Count)
```

## RMSE/MAE:

We managed to slightly improve the R-squared value, RMSE = 18.58 and MAE = 7.58

After this work with CART and Random forest.

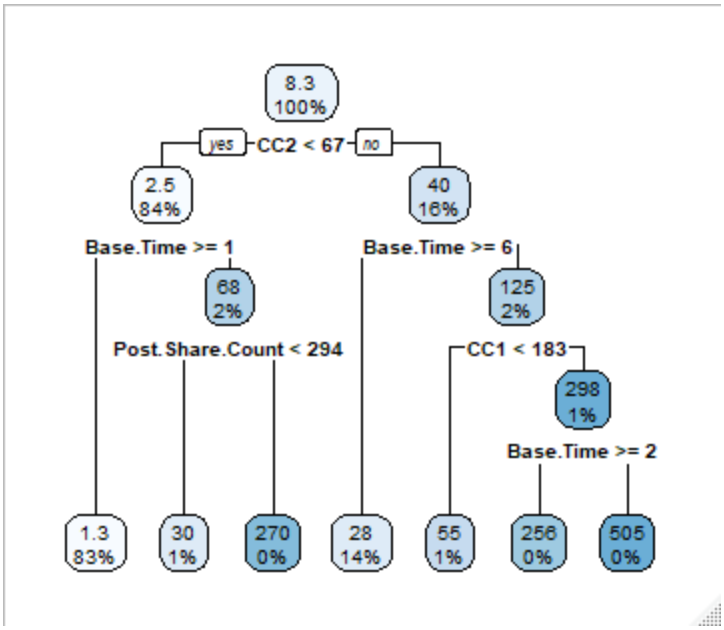
## CART

For CART we have taken a similar approach, where we first use all the variables in the function and then limit them to variables found useful in Multi Regression.

```
tree = rpart(formula = Target.Variable~., data = traindata)
```

We will prune the tree by setting the cp value, to the one with a minimum cross-validation error.

```
ptree = prune(tree,cp=0.025, "CP") ### Minimum cross validation error
 rpart.plot(ptree)
 rsq.rpart(ptree)
```




## RMSE/MAE:

RMSE = 16.520 and MAE = 5.617, the RMSE value is lower than multi. Linear regression.

The restriction of variables in the formula or the Tree doesn't make a significant difference so we will stick to these values for model comparison.

## Random Forest:

We have taken a similar approach where initially random values were assumed for nTress while building the model and later the tree was pruned.

Model with Random Values

```
rf = randomForest(Target.Variable~., data = traindata, ntree = 501, importance = TRUE)
print(rf)
which.min(rf$mse) ### 183

prf0 = randomForest(Target.Variable~., data = traindata, ntree = 183, importance = TRUE)
prf0
### Predicting on Test
predrf0 = predict(prf0,testdata)
RMSErf0 = sqrt(mean((predrf0-testdata$Target.Variable)^2))
RMSErf0 # 15.8034
MAErf0 = mean(abs(predrf0-testdata$Target.Variable))
MAErf0 # 4.0914
```

## RMSE/MAE:

RMSE = 15.80 and MAE = 4.09, though the vlaues are lower than CART model but will further try to prune the tree and try to achieve lower values of RMSE.

```
prf = randomForest(Target.Variable~., data = traindata, ntree = 183, mtry = 3, nodesize =10,
importance = TRUE)
prf
### Predicting on Test
predrf = predict(prf,testdata)
RMSErf = sqrt(mean((predrf-testdata$Target.Variable)^2))
RMSErf #13.721
MAErf = mean(abs(predrf-testdata$Target.Variable))
MAErf #4.37
```

By adding, mtry and nodesize, we are able to bring the values of RMSE to a satisfactory level.

## RMSE/MAE:

RMSE = 13.72 and MAE = 4.37

## Model Comparison:

| | RMSE | MAE |
|---|---|---|
| Multiple Linear Regression | 18.58 | 7.58 |
| CART | 16.52 | 5.67 |
| Random Forest | 13.72 | 4.37 |

Looking at the table above we can clearly state that Random Forest performs the best followed by the CART model as Random Forest has the lowest RMSE and MAE values.

# Final Interpretation/recommendation:

1. After tuning the Random Forest model using the ntree, mtry and nodesize parameters we are able to bring the RMSE down to 13.72 which is the lowest amongst all the models.
2. While performing RDA, we were under the impression that post published week, parge likes or page comments will be the important features in predicting but the random forest paints a different picture.
3. Top 5 important Features (detailed graph below):
    a. CC1 (The total number of comments before selected base date/time)
    b. Base Time
    c. Page Category
    d. CC2 (The number of comments in the last 24 hours, relative to base date/time.)
    e. CC5 (The difference between CC2 and CC3)
    f. Page Check Ins
4. We can change the amount of hours the posts were monitored and check the results.