REPORT FOR CONTINUOUS ASSESSMENT

1. LINEAR REGRESSION

1.1 BUSINESS UNDERSTANDING

The name of this dataset is “Facebook Metrics Data Set”, downloaded from the UCI machine learning repository. (https://archive.ics.uci.edu/ml/datasets/Facebook+metrics#). This data set is related to posts published on the Facebook page of a renowned cosmetic brand in 2014. The value we are trying to predict from this data set is the “Lifetime Post Total Reach”. This is basically just trying to learn how much attention this post can get.

1.2 DATA UNDERSTANDING AND PREPARATION

This data set contains 19 columns, 7 of which include features known post publication and 12 of which are for evaluating post impact. Each column has 501 rows. The total number of records are : 9515.

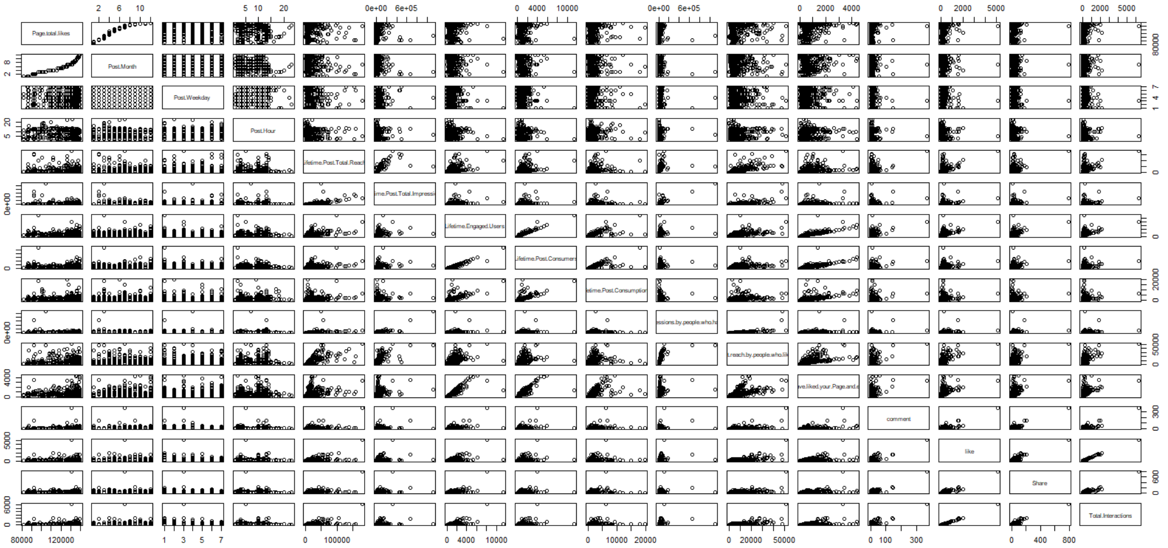
PREPARATION

Not much preparation was done for the data as it was already well prepared from the source. However, some columns that could act as factors were removed. In total 3 columns were removed. They are: Paid, Type and Category columns. The correlation between the dependent variable and all the independent Variables were also checked and their results were noted down.

UNDERSTANDING

Using the “pairs” function, we were able to get a grasp of the type of relationship the independent variables had with each other and with the dependent variable.

Scatter Plot



(Zoom in to see Clearly)

We then split the data set into two : training and test data sets. The train data set consisted of 80 percent of the main data set while the test dataset consisted of the remaining 20 percent.

1.3 MODELLING

The first model was created using all the variables left in the data set. The model set the dependent variable and all the independent variable in a linear model. When the summary of the model was taken, the R and Adjusted R squared values were:

# Multiple R-squared: 0.963, Adjusted R-squared: 0.9617

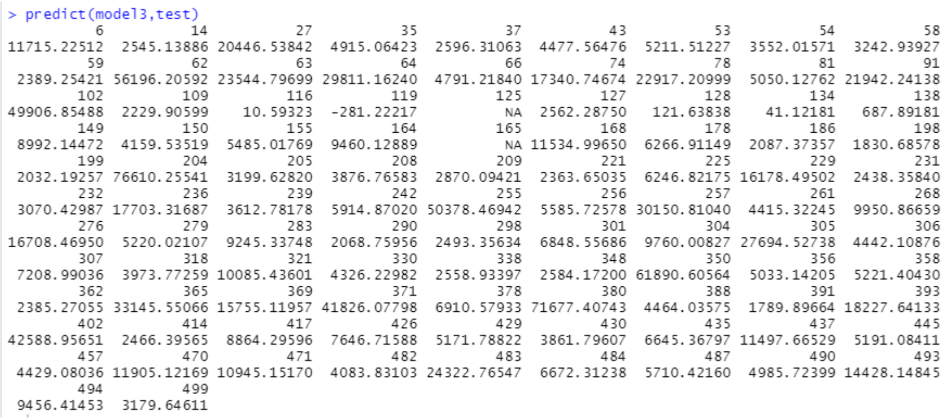
These R squared values were pretty high but we want to optimize the model so we check the summary. After checking the summary, 5(Post.Weekday, Post.Hour, Total.Interaction, Post.consumers and Post.Consumptions) columns had no statistical significance to the dependent variable, Three of these variables were removed and a new model was created based on the remaining variables. When the summary of this new model was taken, the R and Adjusted R values were:

# Multiple R-squared: 0.9624,   Adjusted R-squared: 0.9615

There was almost no deviation and smaller variables so this model was better, however after we removed all the variables that were not statistically significant, we still had the same R and Adjusted R squared values. Therefore the third model was the best. We also created a model containing only the most significant variables. This however had low R and Adjusted R squared values: # Multiple R-squared: 0.5687,   Adjusted R-squared: 0.5654

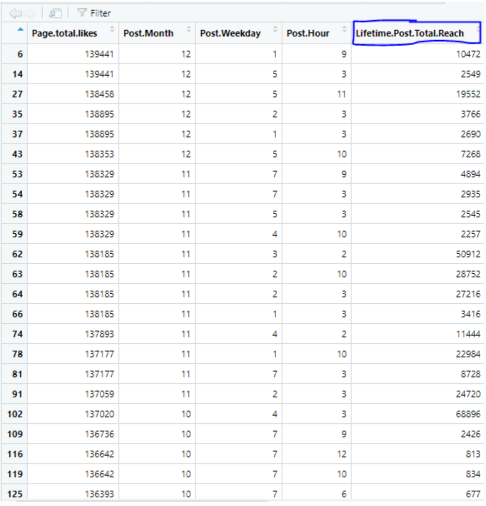
1.4 EVALUATION

In conclusion, the model 3 is better than the other models. This is because it has higher R and Adjusted R squared values thereby making it better in predicting the "Lifetime Post Total Reach". The "predict" function was also used on this model. These are the results:



```
> predict(model3,test)
          6          14          27          35          37          43          53          54          58 
11715.22512  2545.13886 20446.53842  4915.06423  2596.31063  4477.56476  5211.51227  3552.01571  3242.93927 
         59          62          63          64          66          74          78          81          91 
 2389.25421 56196.20592 23544.79699 29811.16240  4791.21840 17340.74674 22917.20999  5050.12762 21942.24138 
        102         109         116         119         125         127         128         134         138 
49906.85488  2229.90599    10.59323  -281.22217          NA  2562.28750   121.63838    41.12181   687.89181 
        149         150         155         164         165         168         178         186         198 
 8992.14472  4159.53519  5485.01769  9460.12889          NA 11534.99650  6266.91149  2087.37357  1830.68578 
        199         204         205         208         209         221         225         229         231 
 2032.19257 76610.25541  3199.62820  3876.76583  2870.09421  2363.65035  6246.82175 16178.49502  2438.35840 
        232         236         239         242         255         256         257         261         268 
 3070.42987 17703.31687  3612.78178  5914.87020 50378.46942  5585.72578 30150.81040  4415.32245  9950.86659 
        276         279         283         290         298         301         304         305         306 
16708.46950  5220.02107  9245.33748  2068.75956  2493.35634  6848.55686  9760.00827 27694.52738  4442.10876 
        307         318         321         330         338         348         350         356         358 
 7208.99036  3973.77259 10085.43601  4326.22982  2558.93397  2584.17200 61890.60564  5033.14205  5221.40430 
        362         365         369         371         378         380         388         391         393 
 2385.27055 33145.55066 15755.11957 41826.07798  6910.57933 71677.40743  4464.03575  1789.89664 18227.64133 
        402         414         417         426         429         430         435         437         445 
42588.95651  2466.39565  8864.29596  7646.71588  5171.78822  3861.79607  6645.36797 11497.66529  5191.08411 
        457         470         471         482         483         484         487         490         493 
 4429.08036 11905.12169 10945.15170  4083.83103 24322.76547  6672.31238  5710.42160  4985.72399 14428.14845 
        494         499 
 9456.41453  3179.64611 
```

Actual values:

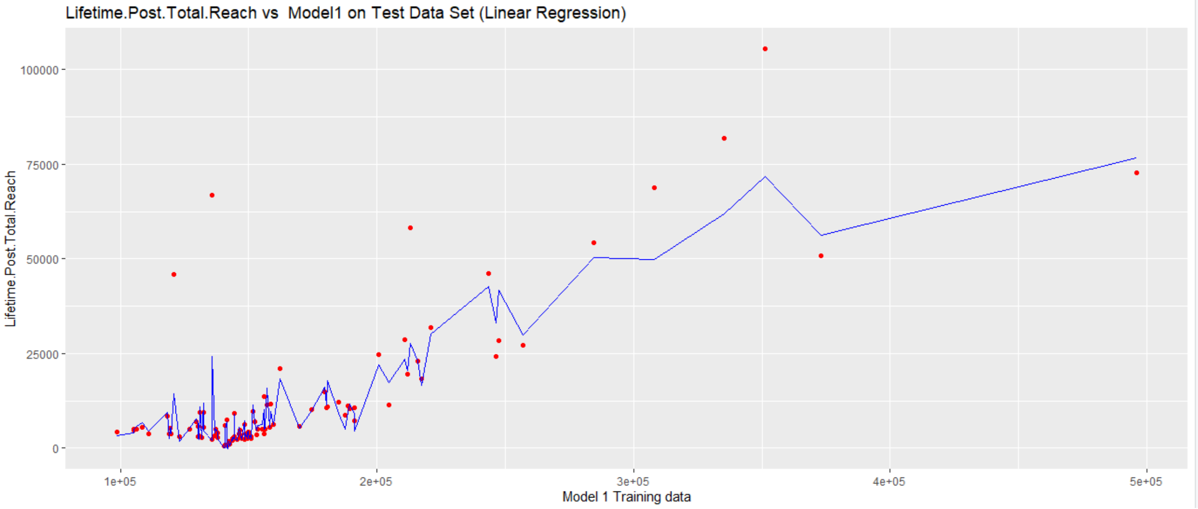


	Page.total.likes	Post.Month	Post.Weekday	Post.Hour	Lifetime.Post.Total.Reach
6	139441	12	1	9	10472
14	139441	12	5	3	2549
27	138458	12	5	11	19552
35	138895	12	2	3	3766
37	138895	12	1	3	2690
43	138353	12	5	10	7268
53	138329	11	7	9	4894
54	138329	11	7	3	2935
58	138329	11	5	3	2545
59	138329	11	4	10	2257
62	138185	11	3	2	50912
63	138185	11	2	10	28752
64	138185	11	2	3	27216
66	138185	11	1	3	3416
74	137893	11	4	2	11444
78	137177	11	1	10	22984
81	137177	11	7	3	8728
91	137059	11	2	3	24720
102	137020	10	4	3	68896
109	136736	10	7	9	2426
116	136642	10	7	12	813
119	136642	10	7	10	834
125	136393	10	7	6	677

The program did say prediction from a rank deficient fit may be misleading though.

We can see from the picture taken that the values predicted are somewhat close to the actual values and are not far off from it.

Prediction of model on test data set



2. POLYNOMIAL REGRESSION

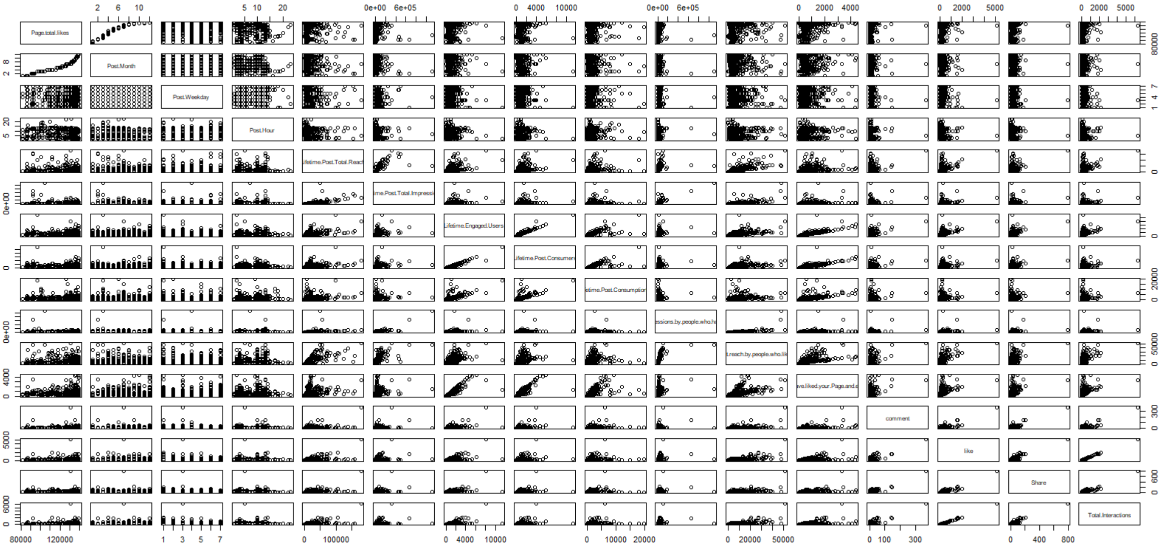
2.1 BUSINESS UNDERSTANDING

The name of this dataset is “Facebook Metrics Data Set”, downloaded from the UCI machine learning repository. (https://archive.ics.uci.edu/ml/datasets/Facebook+metrics#). This data set is related to posts published on the Facebook page of a renowned cosmetic brand in 2014. The value we are trying to predict from this data set is the “Lifetime Post Total Reach”. This is basically just trying to learn how much attention this post can get.

2.2 DATA UNDERSTANDING AND PREPARATION

Since Linear Regression and Polynomial Regression were carried out in the same R file and also were performed on the same data set, the training and testing split sample still existed and all data preparation was intact/ still in place. The pairs function was then used on the main data set to ascertain which variables had possible polynomial relationship with the dependent variable. Below is a review of the scatter plot once again.

Scatter Plot



(Zoom in to see Clearly)

2.3 MODELLING

The models were created from the train data set. This is because we wanted to use to predict function to check if the model works in the end. Based on the scatter plot we took 9 variables(Page.total.likes , Post.Hour ,Lifetime.Post.Total.Impressions , Lifetime.Engaged.Users , Lifetime.Post.Consumers , Lifetime.Post.Consumptions , Lifetime.Post.reach.by.people.who.like.your.Page , Lifetime.People.who.have.liked.your.Page.and.engaged.with.your.post and like) to use for our polynomial model. After creating the model using the poly function we set the degree to 2. The summary of the model gave the following R and Adjusted R squared values : # Multiple R-squared: 0.9632, Adjusted R-squared: 0.9577 . This Value was pretty high, however we went further to see the limits of the model. In the second model, we changed the degree to 3. This gave R and Adjusted R squared values of : # Multiple R-squared: 0.9998, Adjusted R-squared: 0.9996.

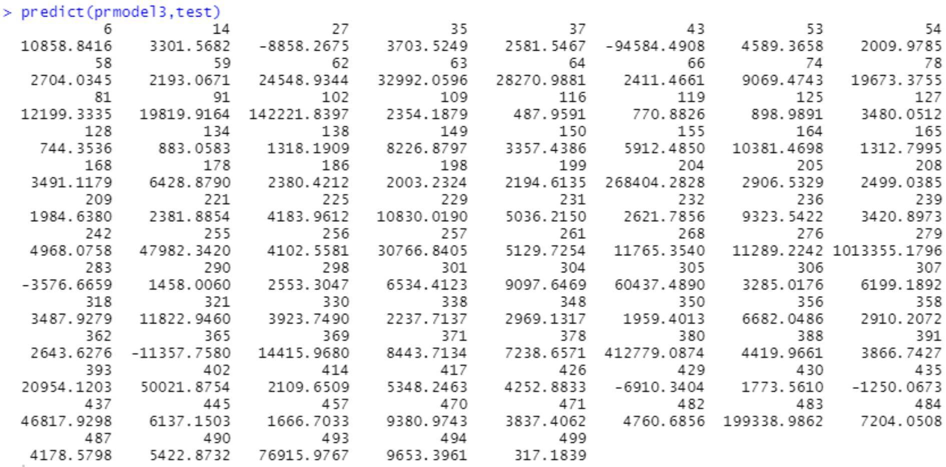
Following this we checked for R and Adjusted R squared values after changing the degree to 4. Here we got: Multiple R-squared: 1, Adjusted R-squared: 1.

2.4 EVALUATION

In conclusion, the third model with degree 4 is in theory a perfect model because it has R and Adjusted R squared values of 1. However, this is suspicious and overfitting may have occurred somewhere. The predict function was also used on this model to predict the test data set and the values predicted were a mess as shown by the second diagram below.

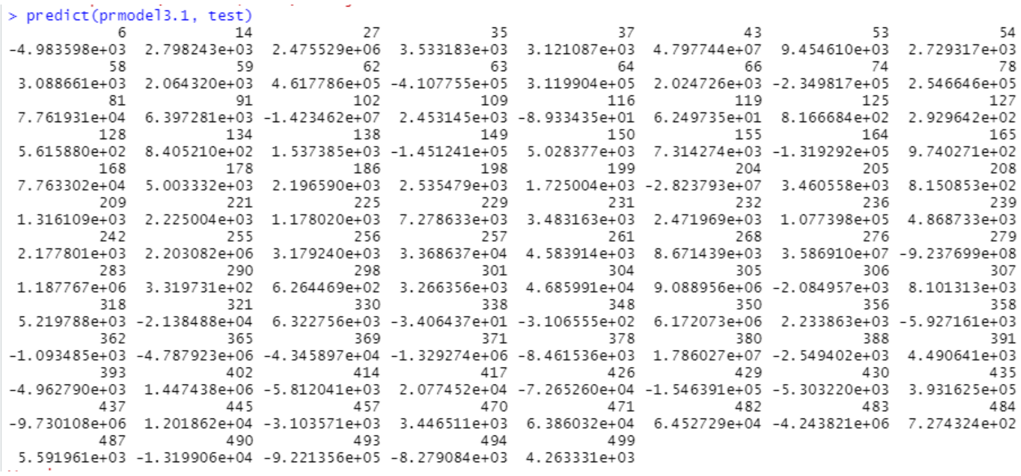
However, when the model was created from the whole dataset, it predicted the test data perfectly(but the full data contains the test data) which nullifies its prediction

Predicted (model based on full data set, degree=4)



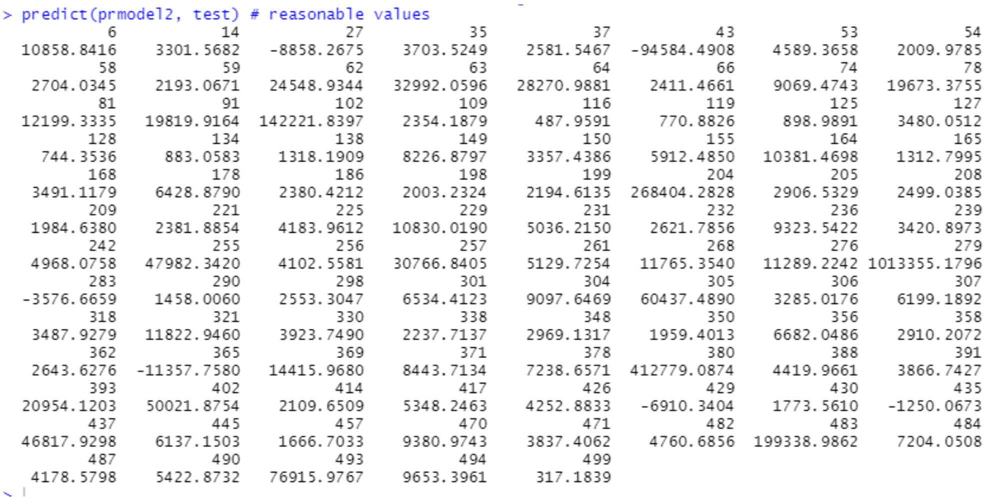
```
> predict(prmodel3,test)
           6           14           27           35           37           43           53           54
  10858.8416    3301.5682   -8858.2675    3703.5249    2581.5467  -94584.4908    4589.3658    2009.9785
          58           59           62           63           64           66           74           78
   2704.0345    2193.0671   24548.9344   32992.0596   28270.9881    2411.4661    9069.4743   19673.3755
          81           91          102          109          116          119          125          127
  12199.3335   19819.9164  142221.8397    2354.1879     487.9591     770.8826     898.9891    3480.0512
         128          134          138          149          150          155          164          165
    744.3536     883.0583    1318.1909    8226.8797    3357.4386    5912.4850   10381.4698    1312.7995
         168          178          186          198          199          204          205          208
   3491.1179    6428.8790    2380.4212    2003.2324    2194.6135  268404.2828    2906.5329    2499.0385
         209          221          225          229          231          232          236          239
   1984.6380    2381.8854    4183.9612   10830.0190    5036.2150    2621.7856    9323.5422    3420.8973
         242          255          256          257          261          268          276          279
   4968.0758   47982.3420    4102.5581   30766.8405    5129.7254   11765.3540   11289.2242 1013355.1796
         283          290          298          301          304          305          306          307
  -3576.6659    1458.0060    2553.3047    6534.4123    9097.6469   60437.4890    3285.0176    6199.1892
         318          321          330          338          348          350          356          358
   3487.9279   11822.9460    3923.7490    2237.7137    2969.1317    1959.4013    6682.0486    2910.2072
         362          365          369          371          378          380          388          391
   2643.6276  -11357.7580   14415.9680    8443.7134    7238.6571  412779.0874    4419.9661    3866.7427
         393          402          414          417          426          429          430          435
  20954.1203   50021.8754    2109.6509    5348.2463    4252.8833   -6910.3404    1773.5610   -1250.0673
         437          445          457          470          471          482          483          484
  46817.9298    6137.1503    1666.7033    9380.9743    3837.4062    4760.6856  199338.9862    7204.0508
         487          490          493          494          499
   4178.5798    5422.8732   76915.9767    9653.3961     317.1839
```

Predicted (model based on train data set, degree =4)



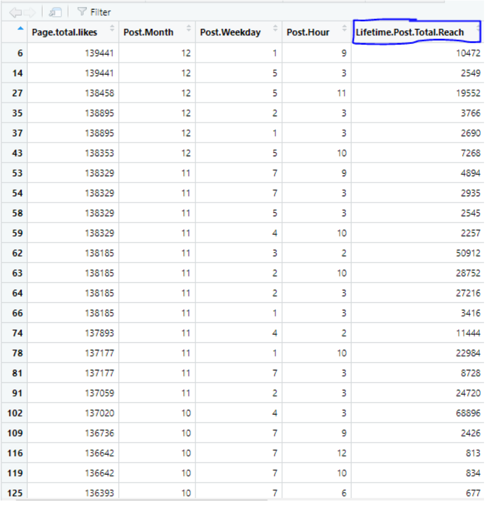
```
> predict(prmodel3.1, test)
            6            14            27            35            37            43            53            54
-4.983598e+03  2.798243e+03  2.475529e+06  3.533183e+03  3.121087e+03  4.797744e+07  9.454610e+03  2.729317e+03
           58            59            62            63            64            66            74            78
 3.088661e+03  2.064320e+03  4.617786e+05 -4.107755e+05  3.119904e+05  2.024726e+03 -2.349817e+05  2.546646e+05
           81            91           102           109           116           119           125           127
 7.761931e+04  6.397281e+03 -1.423462e+07  2.453145e+03 -8.933435e+01  6.249735e+01  8.166684e+02  2.929642e+02
          128           134           138           149           150           155           164           165
 5.615880e+02  8.405210e+02  1.537385e+03 -1.451241e+05  5.028377e+03  7.314274e+03 -1.319292e+05  9.740271e+02
          168           178           186           198           199           204           205           208
 7.763302e+04  5.003332e+03  2.196590e+03  2.535479e+03  1.725004e+03 -2.823793e+07  3.460558e+03  8.150853e+02
          209           221           225           229           231           232           236           239
 1.316109e+03  2.225004e+03  1.178020e+03  7.278633e+03  3.483163e+03  2.471969e+03  1.077398e+05  4.868733e+03
          242           255           256           257           261           268           276           279
 2.177801e+03  2.203082e+06  3.179240e+03  3.368637e+04  4.583914e+03  8.671439e+03  3.586910e+07 -9.237699e+08
          283           290           298           301           304           305           306           307
 1.187767e+06  3.319731e+02  6.264469e+02  3.266356e+03  4.685991e+04  9.088956e+06 -2.084957e+03  8.101313e+03
          318           321           330           338           348           350           356           358
 5.219788e+03 -2.138488e+04  6.322756e+03 -3.406437e+01 -3.106555e+02  6.172073e+06  2.233863e+03 -5.927161e+03
          362           365           369           371           378           380           388           391
-1.093485e+03 -4.787923e+06 -4.345897e+04 -1.329274e+06 -8.461536e+03  1.786027e+07 -2.549402e+03  4.490641e+03
          393           402           414           417           426           429           430           435
-4.962790e+03  1.447438e+06 -5.812041e+03  2.077452e+04 -7.265260e+04 -1.546391e+05 -5.303220e+03  3.931625e+05
          437           445           457           470           471           482           483           484
-9.730108e+06  1.201862e+04 -3.103571e+03  3.446511e+03  6.386032e+04  6.452729e+04 -4.243821e+06  7.274324e+02
          487           490           493           494           499
 5.591961e+03 -1.319906e+04 -9.221356e+05 -8.279084e+03  4.263331e+03
```

Predicted (model based on train data set, degree =3)



```
> predict(prmodel2, test) # reasonable values
           6           14           27           35           37           43           53           54
  10858.8416    3301.5682   -8858.2675    3703.5249    2581.5467  -94584.4908    4589.3658    2009.9785
          58           59           62           63           64           66           74           78
   2704.0345    2193.0671   24548.9344   32992.0596   28270.9881    2411.4661    9069.4743   19673.3755
          81           91          102          109          116          119          125          127
  12199.3335   19819.9164  142221.8397    2354.1879     487.9591     770.8826     898.9891    3480.0512
         128          134          138          149          150          155          164          165
    744.3536     883.0583    1318.1909    8226.8797    3357.4386    5912.4850   10381.4698    1312.7995
         168          178          186          198          199          204          205          208
   3491.1179    6428.8790    2380.4212    2003.2324    2194.6135  268404.2828    2906.5329    2499.0385
         209          221          225          229          231          232          236          239
   1984.6380    2381.8854    4183.9612   10830.0190    5036.2150    2621.7856    9323.5422    3420.8973
         242          255          256          257          261          268          276          279
   4968.0758   47982.3420    4102.5581   30766.8405    5129.7254   11765.3540   11289.2242 1013355.1796
         283          290          298          301          304          305          306          307
  -3576.6659    1458.0060    2553.3047    6534.4123    9097.6469   60437.4890    3285.0176    6199.1892
         318          321          330          338          348          350          356          358
   3487.9279   11822.9460    3923.7490    2237.7137    2969.1317    1959.4013    6682.0486    2910.2072
         362          365          369          371          378          380          388          391
   2643.6276  -11357.7580   14415.9680    8443.7134    7238.6571  412779.0874    4419.9661    3866.7427
         393          402          414          417          426          429          430          435
  20954.1203   50021.8754    2109.6509    5348.2463    4252.8833   -6910.3404    1773.5610   -1250.0673
         437          445          457          470          471          482          483          484
  46817.9298    6137.1503    1666.7033    9380.9743    3837.4062    4760.6856  199338.9862    7204.0508
         487          490          493          494          499
   4178.5798    5422.8732   76915.9767    9653.3961     317.1839
```

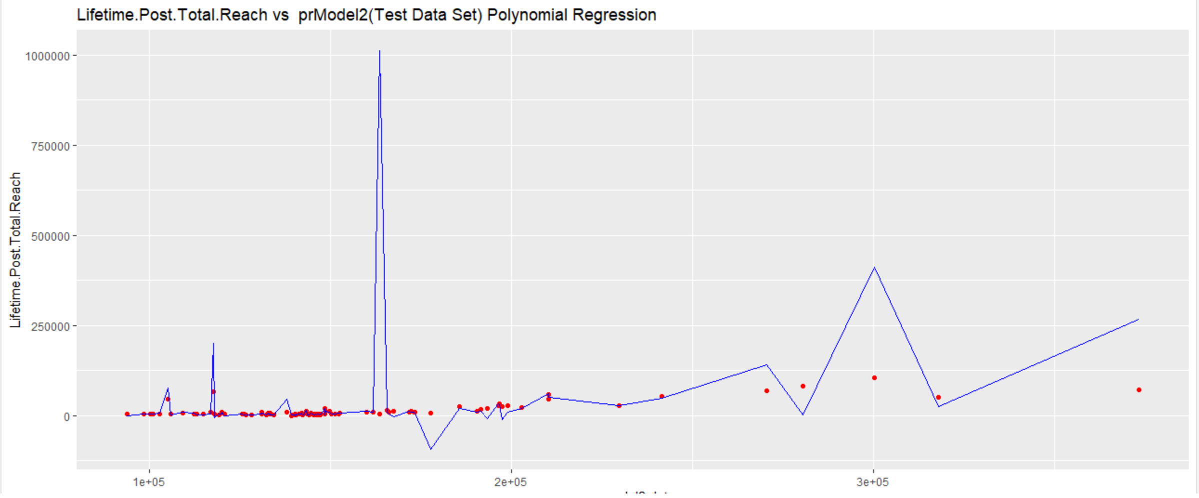
Actual values:



	Page.total.likes	Post.Month	Post.Weekday	Post.Hour	Lifetime.Post.Total.Reach
6	139441	12	1	9	10472
14	139441	12	5	3	2549
27	138458	12	5	11	19552
35	138895	12	2	3	3766
37	138895	12	1	3	2690
43	138353	12	5	10	7268
53	138329	11	7	9	4894
54	138329	11	7	3	2935
58	138329	11	5	3	2545
59	138329	11	4	10	2257
62	138185	11	3	2	50912
63	138185	11	2	10	28752
64	138185	11	2	3	27216
66	138185	11	1	3	3416
74	137893	11	4	2	11444
78	137177	11	1	10	22984
81	137177	11	7	3	8728
91	137059	11	2	3	24720
102	137020	10	4	3	68896
109	136736	10	7	9	2426
116	136642	10	7	12	813
119	136642	10	7	10	834
125	136393	10	7	6	677

Based on these figures, and previous analysis, the second model with degree 3 is the best model as it doesn’t overfit like model 3.

Prediction of model on test dataset



Lifetime.Post.Total.Reach vs prModel2(Test Data Set) Polynomial Regression
Lifetime.Post.Total.Reach
1e+05
2e+05
3e+05