LSDM PROJECT 5 REPORT

Twitter Analysis

A useful practice in social network analysis is to predict future popularity of a subject or event. Twitter, with its public discussion model, is a good platform to perform such analysis. The available Twitter data is collected by querying popular hashtags related to the 2015 Super Bowl spanning a period starting from 2 weeks before the game to a week after the game. We will use data from some of the related hashtags to train a regression model and then use the model to make predictions for other hashtags.

We download the training tweet data. The data consists of 6 text files, each one filled with tweet data from one hashtag as indicated in the filenames. Since this dataset is really large loading the entire thing into memory was taking too long. So, we load the data sequentially, line by line, from each of the files. After reading the tweet individually, we save the relevant features in a list. Then we convert the list to a dictionary with keys as the features. We also convert the timestamp into PST format.

QUESTION 1: Report the following statistics for each hashtag, i.e. each file:

- Average number of tweets per hour
- Average number of followers of users posting the tweets per tweet (to make it simple, we average over the number of tweets; if a users posted twice, we count the user and the user's followers twice as well)
- Average number of retweets per tweet

Average number of tweets/hour for #gohawks : 292.48785062173687
Average number of followers for #gohawks : 2217.9237355281984
Average number of retweets for #gohawks : 2.0132093991319877

Average number of tweets/hour for #gopatriots : 40.95469800606194
Average number of followers for #gopatriots : 1427.2526051635405
Average number of retweets for #gopatriots : 1.4081919101697078

Average number of tweets/hour for #nfl : 397.0213901819841
Average number of followers for #nfl : 4662.37544523693
Average number of retweets for #nfl : 1.5344602655543254

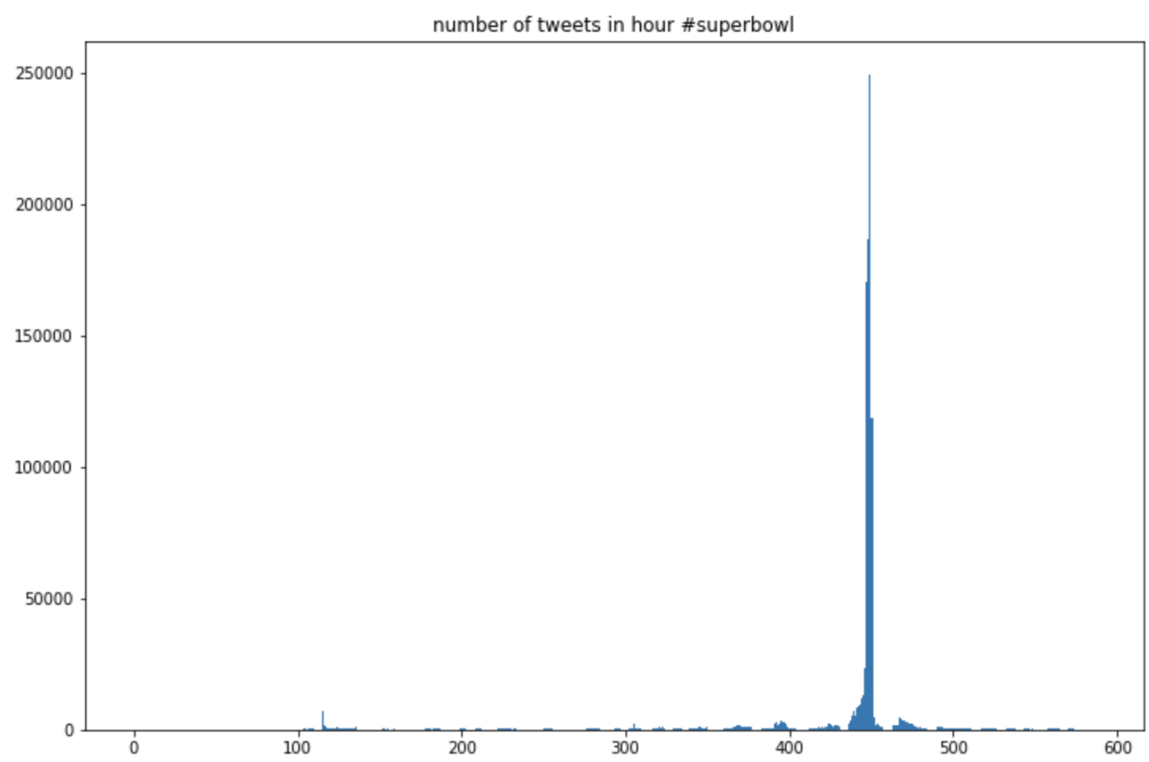
Average number of tweets/hour for #patriots : 750.89426460689
Average number of followers for #patriots : 3280.4635616550277
Average number of retweets for #patriots : 1.7852871288476946

Average number of tweets/hour for #sb49 : 1276.8570598680474
Average number of followers for #sb49 : 10374.160292019487
Average number of retweets for #sb49 : 2.52713444111402

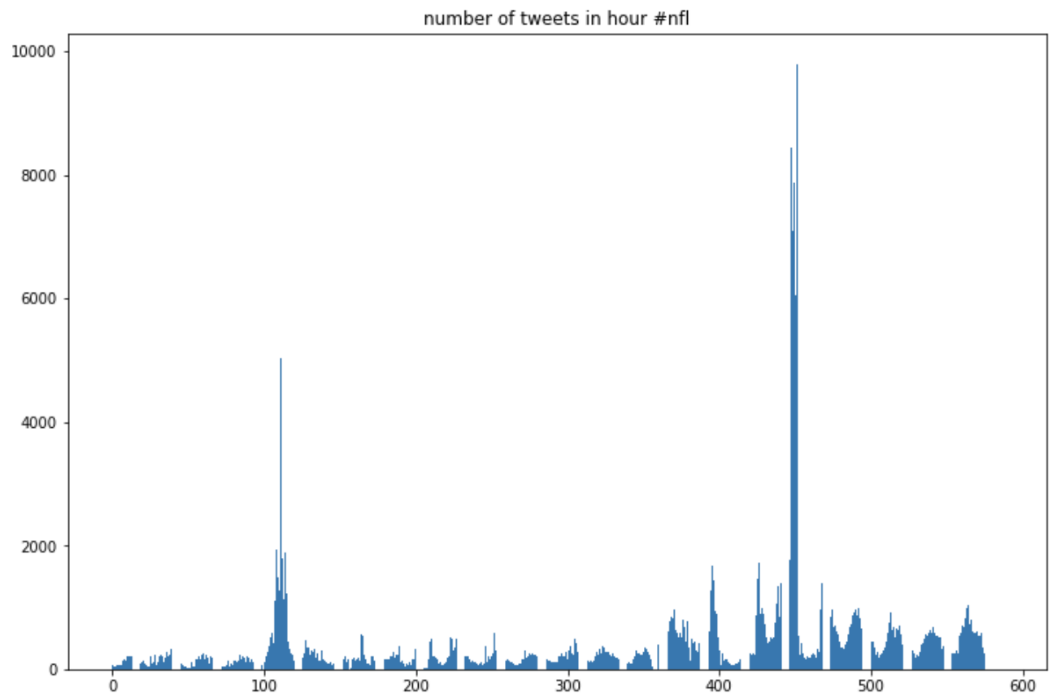
Average number of tweets/hour for #superbowl : 2072.11840170408
Average number of followers for #superbowl : 8814.96799424623
Average number of retweets for #superbowl : 2.3911895819207736

QUESTION 2: Plot "number of tweets in hour" over time for #SuperBowl and #NFL (a histogram with 1-hour bins). The tweets are stored in separate files for different hashtags and files are named as tweet_[#hashtag].txt.

We group the data in the dataframe by the hashtag #superbowl by 60 minutes timestamp and plot the number of tweets.



We group the data in the dataframe by the hashtag #nfl by 60 minutes timestamp and plot the number of tweets.



QUESTION 3: For each of your models, report your model's Mean Squared Error (MSE) and R-squared measure. Also, analyze the significance of each feature using the *t*-test and *p*-value. You may use the OLS in the library statsmodels in Python.

In this question, we fit an OLS estimator to each hashtag giving us 6 models. The data is grouped according to a 1-hour period, and we chose the 5 features below:

- Number of tweets
- Total number of retweets
- Sum of the number of followers of the users posting the hashtag
- Maximum number of followers of the users posting the hashtag
- Time of the day (which could take 24 values that represent hours of the day with respect
  to a given time zone)

#gohawks

```
-------------------------------------------------- #gohawks -------------------------------------
                    OLS Regression Results
==============================================================================
Dep. Variable:         no_of_tweets   R-squared:                 0.504
Model:                          OLS   Adj. R-squared:            0.500
Method:               Least Squares   F-statistic:               116.5
Date:              Thu, 14 Mar 2019   Prob (F-statistic):       6.98e-85
Time:                      22:44:05   Log-Likelihood:          -4733.9
No. Observations:               578   AIC:                      9478.
Df Residuals:                   573   BIC:                      9500.
Df Model:                         5
Covariance Type:          nonrobust
==============================================================================
                    coef    std err        t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
hours              7.6324      2.964      2.575      0.010      1.811      13.453
no_of_tweets         1.2853      0.164      7.842      0.000      0.963      1.607
retweet_count_sum     -0.1379      0.043     -3.170      0.002     -0.223     -0.052
followers_count_sum    -0.0002   8.01e-05     -2.432      0.015     -0.000   -3.74e-05
followers_count_max  7.089e-05      0.000      0.476      0.634     -0.000      0.000
==============================================================================
Omnibus:                    910.819   Durbin-Watson:              2.214
Prob(Omnibus):                0.000   Jarque-Bera (JB):        771500.508
Skew:                         8.576   Prob(JB):                    0.00
Kurtosis:                   181.158   Cond. No.                 2.15e+05
==============================================================================
```

MSE for #gohawks : 767558.4451353371

#gopatriots

```
------------------------------------------------ #gopatriots ------------------------------------------------
                    OLS Regression Results
==============================================================================
Dep. Variable:         no_of_tweets   R-squared:                 0.637
Model:                          OLS   Adj. R-squared:            0.634
Method:               Least Squares   F-statistic:               200.0
Date:              Thu, 14 Mar 2019   Prob (F-statistic):      9.02e-123
Time:                      22:44:05   Log-Likelihood:          -3749.2
No. Observations:               574   AIC:                      7508.
```

```
Df Residuals:                 569   BIC:                      7530.
Df Model:                       5
Covariance Type:        nonrobust
==============================================================================
                     coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
hours                0.5033      0.622      0.810      0.418      -0.718       1.724
no_of_tweets          0.3091      0.284      1.086      0.278      -0.250       0.868
retweet_count_sum      0.4905      0.192      2.559      0.011       0.114       0.867
followers_count_sum   -0.0001      0.000     -0.521      0.602      -0.001       0.000
followers_count_max -1.957e-05      0.000     -0.089      0.929      -0.000       0.000
==============================================================================
Omnibus:                  480.482   Durbin-Watson:                  1.907
Prob(Omnibus):               0.000   Jarque-Bera (JB):          290394.588
Skew:                       2.468   Prob(JB):                        0.00
Kurtosis:                 113.080   Cond. No.                    3.43e+04
==============================================================================
```

MSE for #gopatriots : 27846.061955476795

#nfl

```
------------------------------------------------------ #nfl -----------------------------------------
                            OLS Regression Results
==============================================================================
Dep. Variable:           no_of_tweets   R-squared:                       0.652
Model:                            OLS   Adj. R-squared:                  0.649
Method:                 Least Squares   F-statistic:                     217.8
Date:                Thu, 14 Mar 2019   Prob (F-statistic):          1.21e-130
Time:                        22:44:06   Log-Likelihood:                -4499.9
No. Observations:                 586   AIC:                             9010.
Df Residuals:                     581   BIC:                             9032.
Df Model:                           5
Covariance Type:            nonrobust
==============================================================================
                     coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
hours                7.5806      1.966      3.855      0.000       3.719      11.442
no_of_tweets          0.6315      0.134      4.716      0.000       0.368       0.895
retweet_count_sum     -0.1811      0.064     -2.831      0.005      -0.307      -0.055
followers_count_sum    0.0001    2.5e-05      4.257      0.000    5.73e-05       0.000
followers_count_max -9.968e-05   3.28e-05     -3.040      0.002      -0.000   -3.53e-05
==============================================================================
```

```
Omnibus:                  619.693   Durbin-Watson:               2.363
Prob(Omnibus):              0.000   Jarque-Bera (JB):       342014.314
Skew:                       3.928   Prob(JB):                     0.00
Kurtosis:                 121.092   Cond. No.                 3.91e+05
==============================================================================
```

MSE for #nfl : 276193.94623583014

#patriots

```
------------------------------------------------ #patriots ----------------------------------
                            OLS Regression Results
==============================================================================
Dep. Variable:           no_of_tweets   R-squared:                       0.679
Model:                            OLS   Adj. R-squared:                  0.677
Method:                 Least Squares   F-statistic:                     246.3
Date:                Thu, 14 Mar 2019   Prob (F-statistic):          5.98e-141
Time:                        22:44:08   Log-Likelihood:                -5361.9
No. Observations:                 586   AIC:                         1.073e+04
Df Residuals:                     581   BIC:                         1.076e+04
Df Model:                           5
Covariance Type:            nonrobust
==============================================================================
                       coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------
hours                5.2220      7.843      0.666      0.506     -10.182      20.626
no_of_tweets         0.9148      0.071     12.943      0.000       0.776       1.054
retweet_count_sum   -0.0675      0.058     -1.170      0.243      -0.181       0.046
followers_count_sum -1.156e-05   2.63e-05     -0.439      0.661   -6.32e-05    4.01e-05
followers_count_max    0.0001   9.08e-05      1.489      0.137   -4.31e-05       0.000
==============================================================================
Omnibus:                  884.481   Durbin-Watson:               1.996
Prob(Omnibus):              0.000   Jarque-Bera (JB):       688343.951
Skew:                       7.876   Prob(JB):                     0.00
Kurtosis:                 170.163   Cond. No.                 6.81e+05
==============================================================================
```

MSE for #patriots : 5234121.858285548

#sb49

```
------------------------------------------------- #sb49 ---------------------------------------
                    OLS Regression Results
==============================================================================
Dep. Variable:          no_of_tweets   R-squared:                  0.808
Model:                           OLS   Adj. R-squared:             0.807
Method:                Least Squares   F-statistic:                486.4
Date:               Thu, 14 Mar 2019   Prob (F-statistic):         3.23e-204
Time:                       22:44:12   Log-Likelihood:             -5656.6
No. Observations:                582   AIC:                        1.132e+04
Df Residuals:                    577   BIC:                        1.134e+04
Df Model:                          5
Covariance Type:           nonrobust
==============================================================================
                     coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------
hours             -3.5201     14.260     -0.247      0.805     -31.529     24.488
no_of_tweets       1.1373      0.087     13.040      0.000       0.966      1.309
retweet_count_sum -0.1618      0.079     -2.058      0.040      -0.316     -0.007
followers_count_sum 9.878e-06  1.25e-05   0.790      0.430   -1.47e-05   3.44e-05
followers_count_max 9.885e-05  4.34e-05   2.279      0.023    1.37e-05      0.000
==============================================================================
Omnibus:                    1178.031   Durbin-Watson:              1.673
Prob(Omnibus):                 0.000   Jarque-Bera (JB):           2197143.002
Skew:                         14.548   Prob(JB):                   0.00
Kurtosis:                    302.595   Cond. No.                   6.78e+06
==============================================================================
```

MSE for #sb49 : 16339772.550153121

#superbowl

```
------------------------------------------------- #superbowl ----------------------------------
                    OLS Regression Results
==============================================================================
Dep. Variable:          no_of_tweets   R-squared:                  0.803
Model:                           OLS   Adj. R-squared:             0.801
Method:                Least Squares   F-statistic:                473.8
Date:               Thu, 14 Mar 2019   Prob (F-statistic):         2.80e-202
Time:                       22:44:18   Log-Likelihood:             -6039.9
No. Observations:                586   AIC:                        1.209e+04
Df Residuals:                    581   BIC:                        1.211e+04
Df Model:                          5
Covariance Type:           nonrobust
```

```
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
hours              -29.0126     26.714     -1.086      0.278     -81.480      23.455
no_of_tweets          2.2765      0.080     28.559      0.000       2.120       2.433
retweet_count_sum    -0.2553      0.046     -5.595      0.000      -0.345      -0.166
followers_count_sum  -0.0001   2.19e-05     -6.278      0.000      -0.000   -9.44e-05
followers_count_max   0.0007      0.000      5.013      0.000       0.000       0.001
==============================================================================
Omnibus:                      974.639   Durbin-Watson:                   2.285
Prob(Omnibus):                  0.000   Jarque-Bera (JB):          1789674.506
Skew:                           9.288   Prob(JB):                         0.00
Kurtosis:                     273.097   Cond. No.                     9.75e+06
==============================================================================
```

MSE for #superbowl : 52940707.43276812

Looking at p-value of an OLS estimator, if the p-value is low, it means that if the variable changes, this will have an important change in the response. This means that variables with low p-value have a lot of impact in the prediction. We can also see this in all of the OLS models above. Those with low p-values generally have very high coef. Those with high p-value will have very low coef showing that they don't have much impact during the prediction.

QUESTION 4: Design a regression model using any features from the papers you find or other new features you may find useful for this problem. Fit your model on the data of each hashtag and report fitting MSE and significance of features.

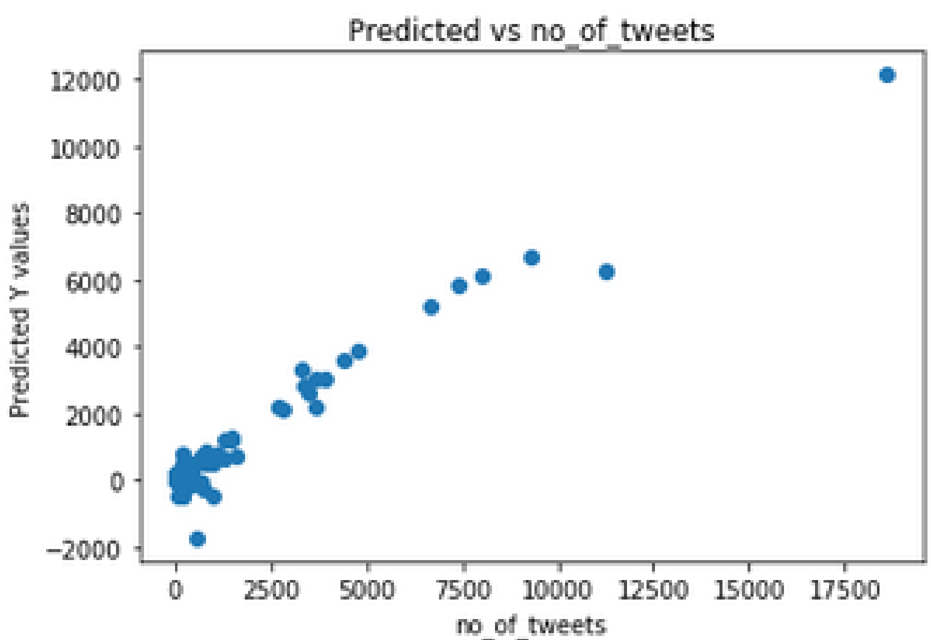
We design a regression model to fit the data of each hashtag. The results are evaluated with the next question.

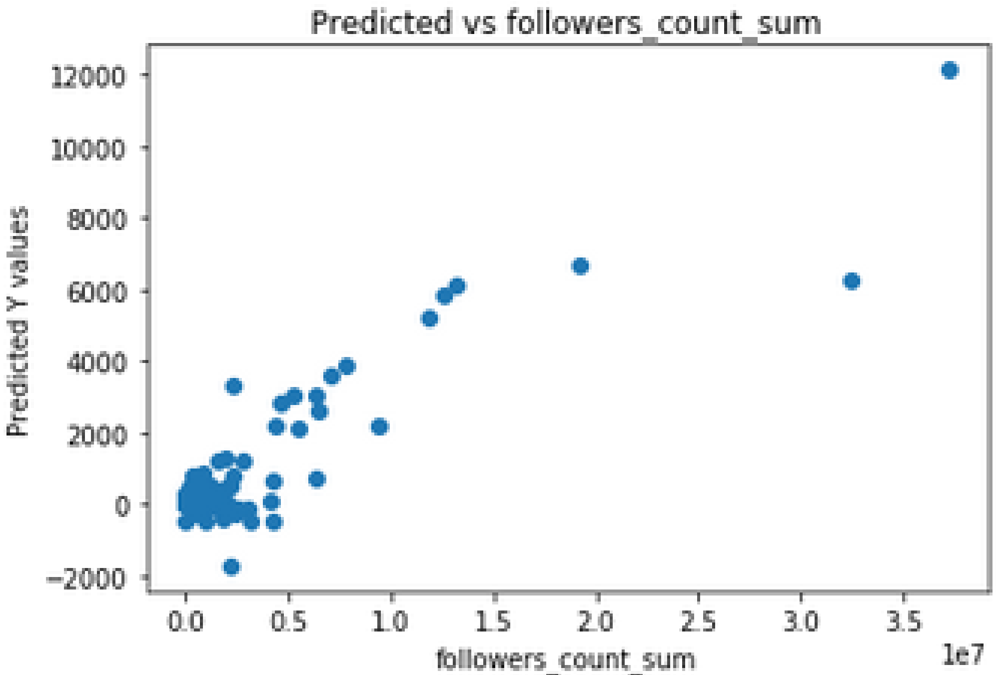
QUESTION 5: For each of the top 3 features (*i.e.* with the smallest *p*-values) in your measurements, draw a scatter plot of predictant (number of tweets for next hour) versus value of that feature, using all the samples you have extracted, and analyze it. Do the regression coefficients agree with the trends in the plots? If not, why?

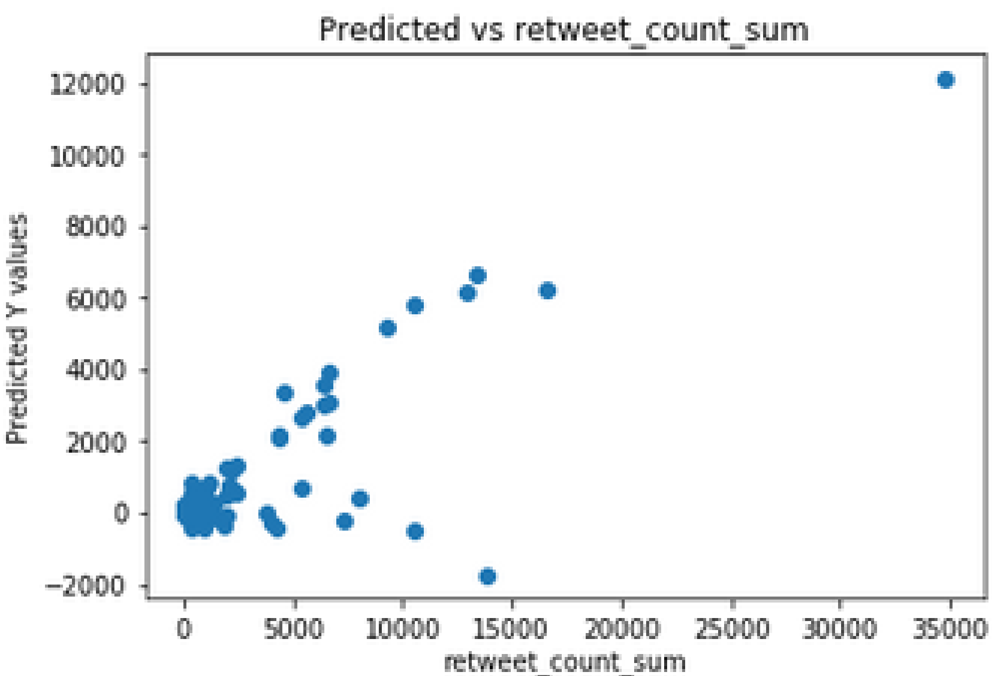
```
------------------------------------------------- #gohawks ------------------------------------
                            OLS Regression Results
==============================================================================
Dep. Variable:           no_of_tweets   R-squared:                       0.509
Model:                            OLS   Adj. R-squared:                  0.502
Method:                 Least Squares   F-statistic:                     73.84
```

```
Date:              Thu, 14 Mar 2019   Prob (F-statistic):          4.98e-83
Time:                      22:47:49   Log-Likelihood:              -4731.1
No. Observations:               578   AIC:                          9478.
Df Residuals:                   570   BIC:                          9513.
Df Model:                         8
Covariance Type:          nonrobust
==============================================================================
                      coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
hours                1.0270      5.278      0.195      0.846      -9.340      11.394
no_of_tweets         1.3369      0.167      8.019      0.000       1.009       1.664
retweet_count_sum   -0.1612      0.046     -3.503      0.000      -0.252      -0.071
followers_count_sum -0.0003      0.000     -2.947      0.003      -0.001      -0.000
followers_count_max 2.449e-05    0.000      0.163      0.870      -0.000       0.000
ranking_mean        23.5268     15.973      1.473      0.141      -7.846      54.899
momentum_mean       17.6156    219.361      0.080      0.936    -413.239     448.471
impressions_sum      0.0001   7.96e-05      1.737      0.083   -1.81e-05       0.000
==============================================================================
Omnibus:                    919.464   Durbin-Watson:                   2.213
Prob(Omnibus):                0.000   Jarque-Bera (JB):           775085.329
Skew:                         8.755   Prob(JB):                         0.00
Kurtosis:                   181.541   Cond. No.                     2.27e+07
==============================================================================
```

MSE for #gohawks : 764136.1928465703




Predicted vs followers_count_sum
Predicted Y values
followers_count_sum
1e7
12000
10000
8000
6000
4000
2000
0
−2000
0.0
0.5
1.0
1.5
2.0
2.5
3.0
3.5


Predicted vs retweet_count_sum
Predicted Y values
retweet_count_sum
12000
10000
8000
6000
4000
2000
0
−2000
0
5000
10000
15000
20000
25000
30000
35000

-- #gopatriots ---------------------------------

OLS Regression Results

==

Dep. Variable:	no_of_tweets	R-squared:	0.642
Model:	OLS	Adj. R-squared:	0.637
Method:	Least Squares	F-statistic:	127.0
Date:	Thu, 14 Mar 2019	Prob (F-statistic):	4.84e-121
Time:	22:47:49	Log-Likelihood:	-3745.3
No. Observations:	574	AIC:	7507.
Df Residuals:	566	BIC:	7541.
Df Model:	8		
Covariance Type:	nonrobust		

==

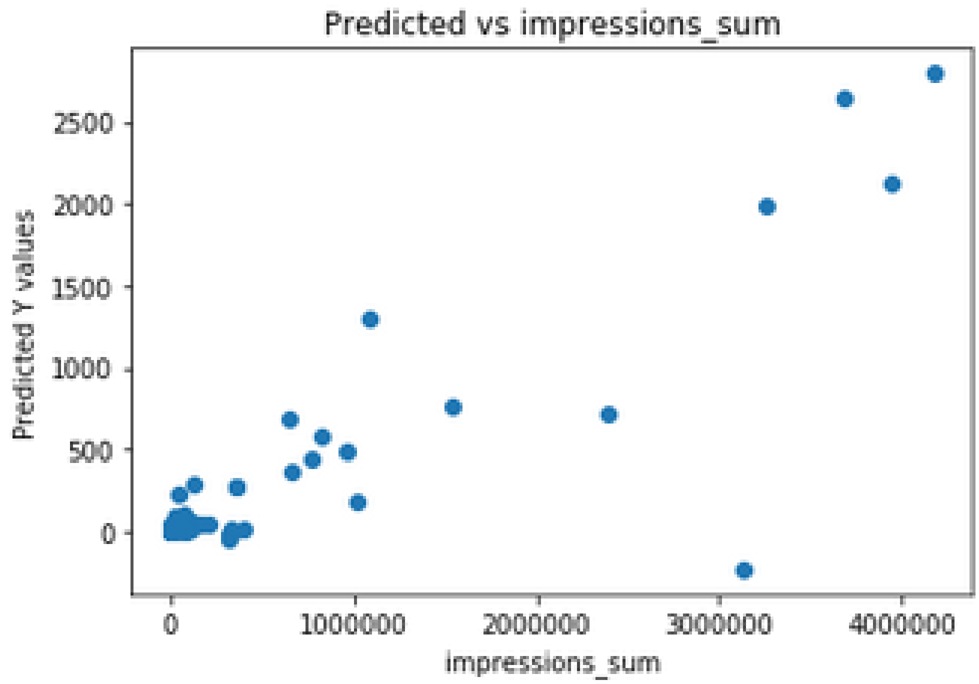
	coef	std err	t	P>|t|	[0.025	0.975]
hours	-0.3988	1.197	-0.333	0.739	-2.750	1.953
no_of_tweets	0.6453	0.314	2.057	0.040	0.029	1.262
retweet_count_sum	0.3861	0.196	1.968	0.050	0.001	0.772
followers_count_sum	-0.0010	0.000	-2.485	0.013	-0.002	-0.000
followers_count_max	0.0002	0.000	0.798	0.425	-0.000	0.001
ranking_mean	3.1103	3.568	0.872	0.384	-3.898	10.119
momentum_mean	-2.3325	10.738	-0.217	0.828	-23.423	18.758
impressions_sum	0.0007	0.000	2.629	0.009	0.000	0.001

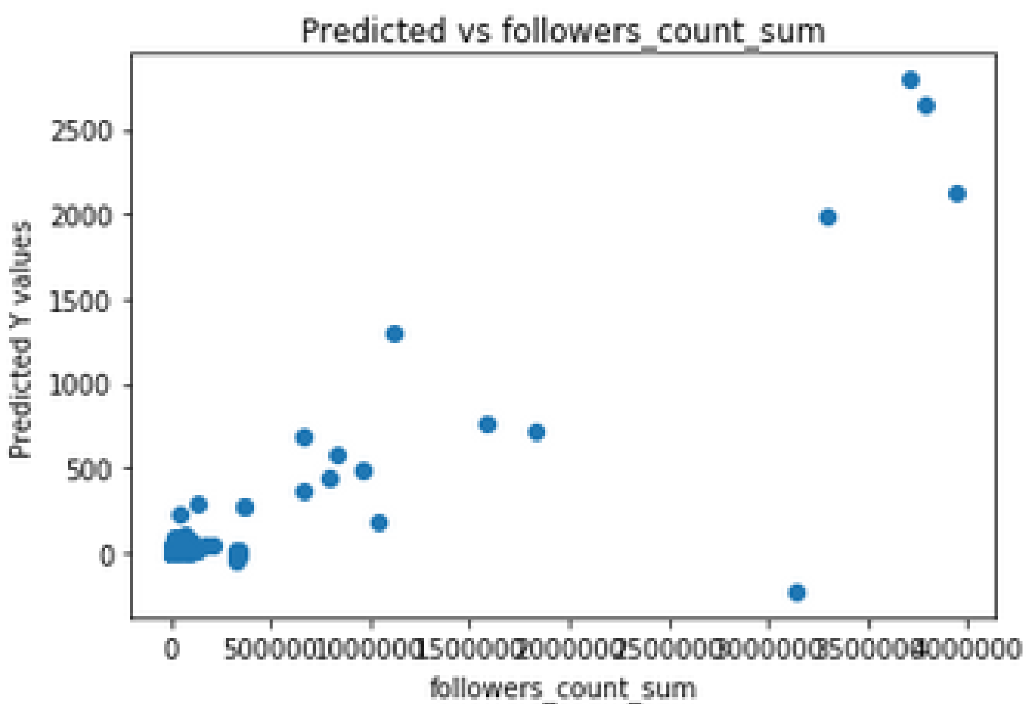
==

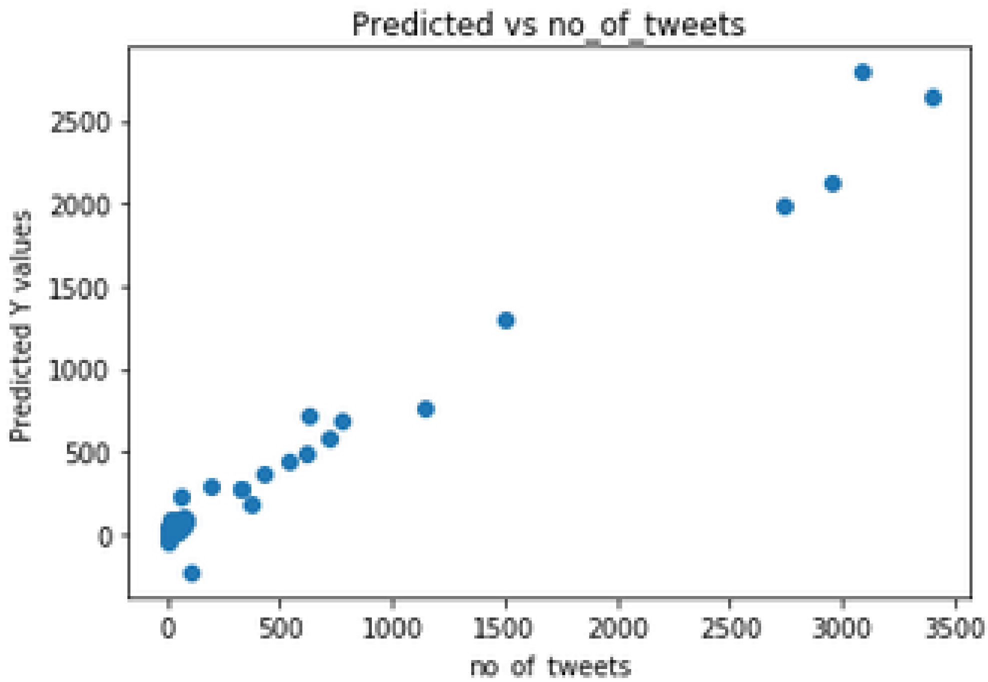
Omnibus:	554.510	Durbin-Watson:	1.891
Prob(Omnibus):	0.000	Jarque-Bera (JB):	311347.384
Skew:	3.282	Prob(JB):	0.00
Kurtosis:	116.908	Cond. No.	8.31e+05

==

MSE for #gopatriots : 27614.845231425377


Predicted vs impressions_sum
Predicted Y values
2500
2000
1500
1000
500
0
0
1000000
2000000
3000000
4000000
impressions_sum

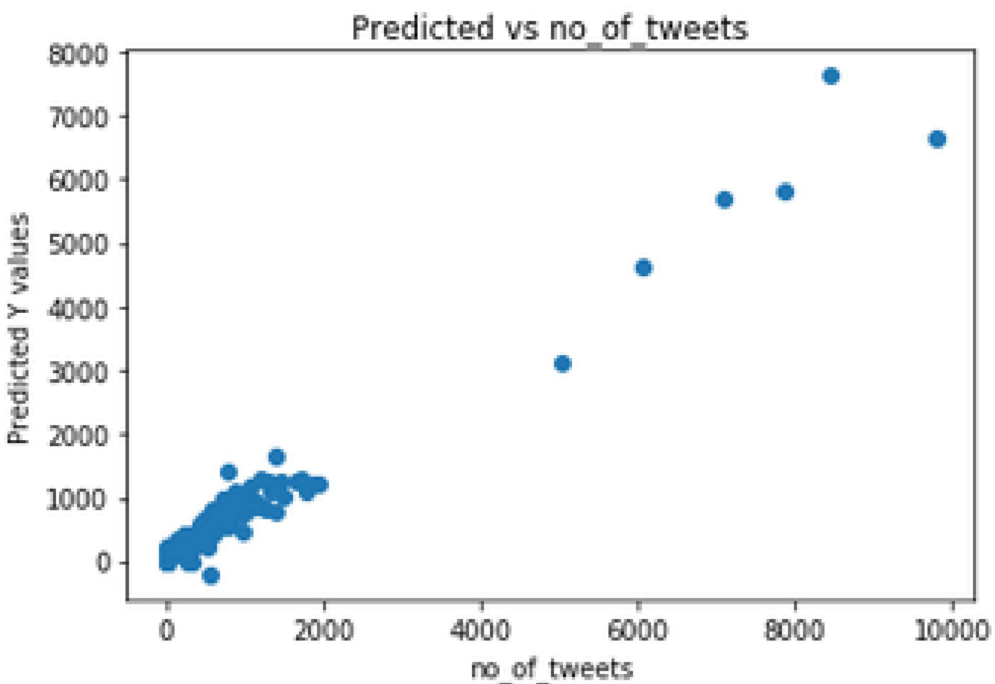

Predicted vs followers_count_sum
Predicted Y values
2500
2000
1500
1000
500
0
0
500000
1000000
1500000
2000000
2500000
3000000
3500000
4000000
followers_count_sum

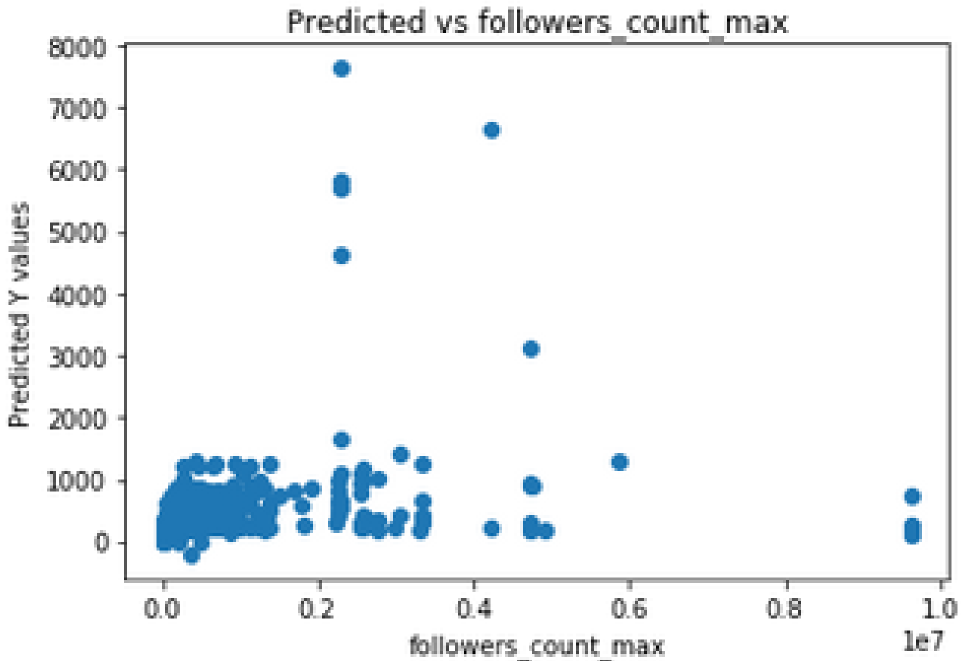



```
------------------------------------------------ #nfl ----------------------------------------
                    OLS Regression Results
==============================================================================
Dep. Variable:          no_of_tweets   R-squared:                       0.658
Model:                           OLS   Adj. R-squared:                  0.653
Method:                Least Squares   F-statistic:                     139.0
Date:               Thu, 14 Mar 2019   Prob (F-statistic):          2.71e-129
Time:                       22:47:51   Log-Likelihood:                -4495.0
No. Observations:                586   AIC:                             9006.
Df Residuals:                    578   BIC:                             9041.
Df Model:                          8
Covariance Type:           nonrobust
```

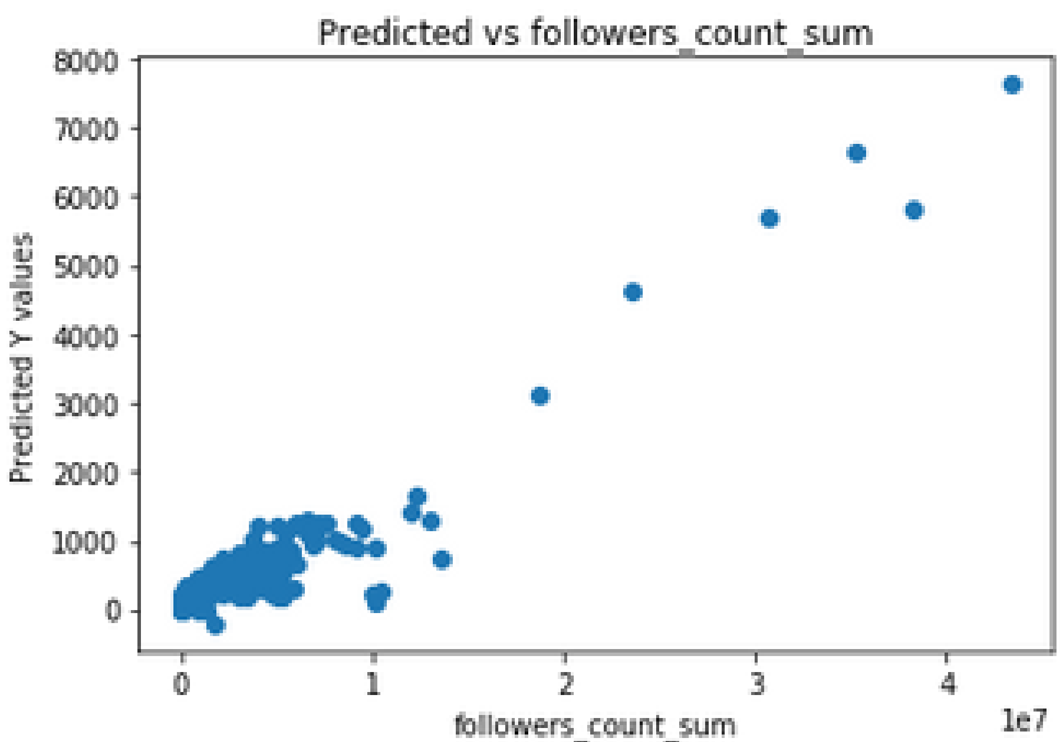
```
==================================================================================
                  coef    std err          t      P>|t|      [0.025      0.975]
----------------------------------------------------------------------------------
hours               -0.0368      3.177     -0.012      0.991     -6.277      6.203
no_of_tweets         0.5699      0.136      4.197      0.000      0.303      0.837
retweet_count_sum   -0.1691      0.065     -2.617      0.009     -0.296     -0.042
followers_count_sum  0.0001   3.91e-05      3.438      0.001   5.76e-05      0.000
followers_count_max -0.0001   3.36e-05     -3.609      0.000     -0.000  -5.53e-05
ranking_mean        29.5472     10.022      2.948      0.003      9.864     49.231
momentum_mean       80.9672    183.670      0.441      0.660   -279.774    441.709
impressions_sum  -1.873e-05   2.88e-05     -0.650      0.516  -7.54e-05   3.79e-05
==================================================================================
Omnibus:                  671.534   Durbin-Watson:                2.371
Prob(Omnibus):              0.000   Jarque-Bera (JB):        350361.388
Skew:                       4.616   Prob(JB):                      0.00
Kurtosis:                 122.432   Cond. No.                  5.21e+07
==================================================================================
```

MSE for #nfl : 272993.3942217459





Predicted vs followers_count_max
Predicted Y values
8000
7000
6000
5000
4000
3000
2000
1000
0
0.0
0.2
0.4
0.6
0.8
1.0
1e7
followers_count_max



Predicted vs followers_count_sum
Predicted Y values
8000
7000
6000
5000
4000
3000
2000
1000
0
0
1
2
3
4
1e7
followers_count_sum

--- #patriots -----------------------------------

OLS Regression Results

==

Dep. Variable:	no_of_tweets	R-squared:	0.681
Model:	OLS	Adj. R-squared:	0.677
Method:	Least Squares	F-statistic:	154.4
Date:	Thu, 14 Mar 2019	Prob (F-statistic):	3.95e-138
Time:	22:47:54	Log-Likelihood:	-5360.2
No. Observations:	586	AIC:	1.074e+04
Df Residuals:	578	BIC:	1.077e+04
Df Model:	8		
Covariance Type:	nonrobust		

==

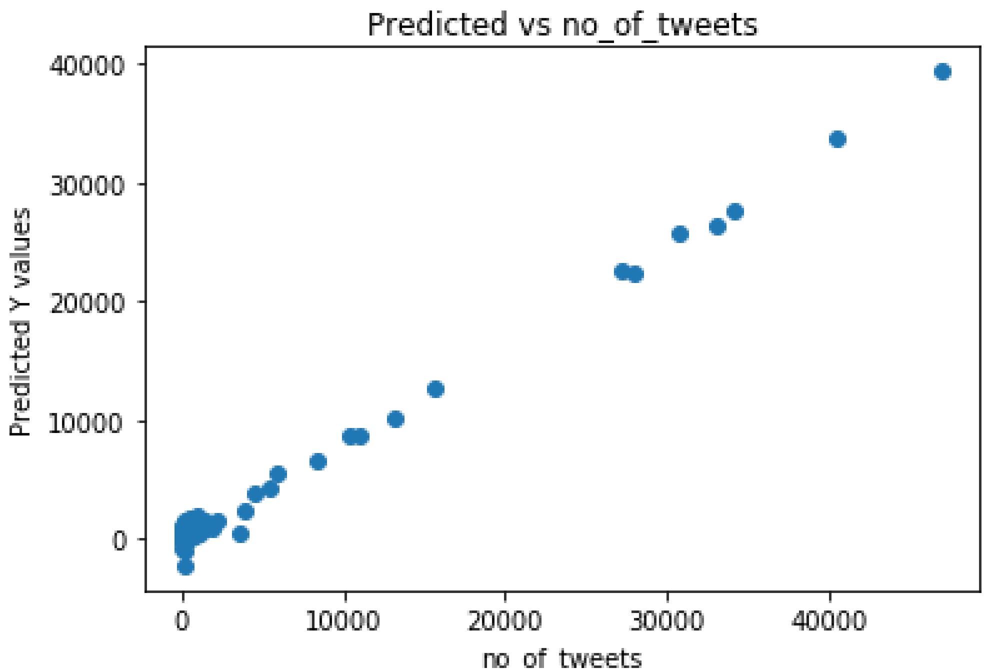
	coef	std err	t	P>|t|	[0.025	0.975]
hours	-5.0062	13.642	-0.367	0.714	-31.800	21.788
no_of_tweets	0.9012	0.071	12.653	0.000	0.761	1.041
retweet_count_sum	-0.0601	0.058	-1.036	0.301	-0.174	0.054
followers_count_sum	0.0003	0.000	1.484	0.138	-9.43e-05	0.001
followers_count_max	0.0001	9.47e-05	1.336	0.182	-5.95e-05	0.000
ranking_mean	42.6384	40.381	1.056	0.291	-36.672	121.949
momentum_mean	-63.0340	386.660	-0.163	0.871	-822.463	696.396
impressions_sum	-0.0003	0.000	-1.558	0.120	-0.001	7.95e-05

==

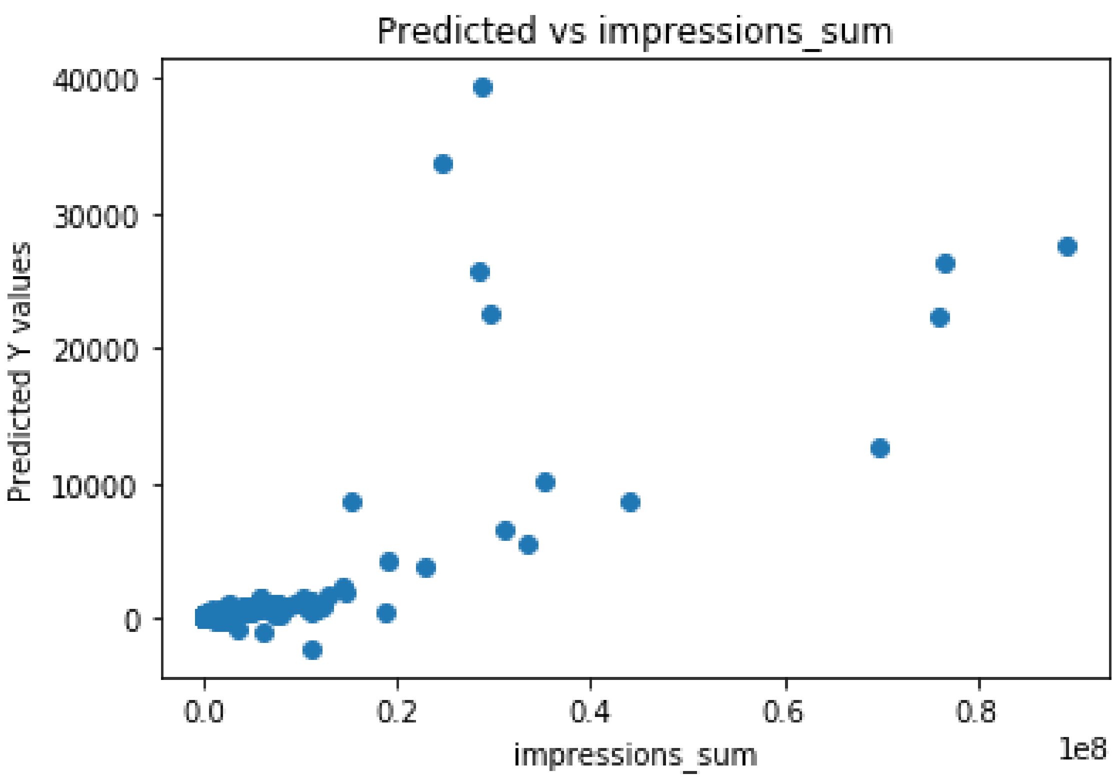
Omnibus:	888.470	Durbin-Watson:	2.002
Prob(Omnibus):	0.000	Jarque-Bera (JB):	690913.644
Skew:	7.952	Prob(JB):	0.00
Kurtosis:	170.463	Cond. No.	4.72e+07

==

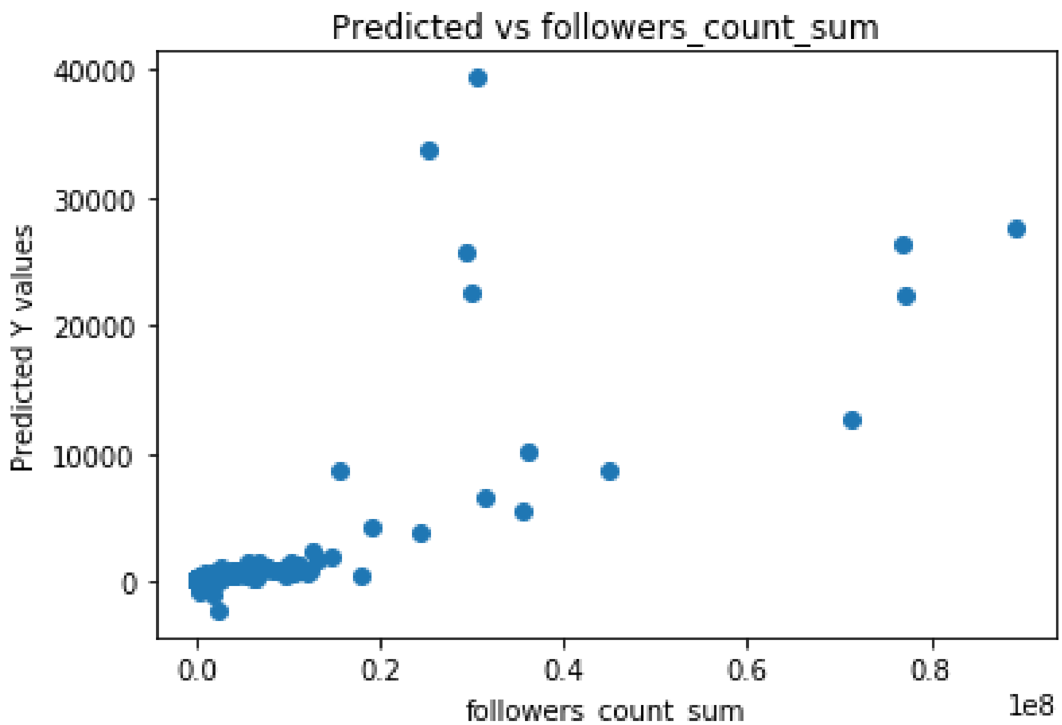
MSE for #patriots : 5231459.668787525



Predicted vs no_of_tweets
Predicted Y values
no_of_tweets
40000
30000
20000
10000
0
0
10000
20000
30000
40000



Predicted vs impressions_sum
Predicted Y values
impressions_sum
1e8
40000
30000
20000
10000
0
0.0
0.2
0.4
0.6
0.8



--- #sb49 ---------------------------------------

OLS Regression Results

==

Dep. Variable: no_of_tweets R-squared: 0.810
Model: OLS Adj. R-squared: 0.807
Method: Least Squares F-statistic: 305.0
Date: Thu, 14 Mar 2019 Prob (F-statistic): 4.06e-201
Time: 22:47:58 Log-Likelihood: -5654.5
No. Observations: 582 AIC: 1.133e+04
Df Residuals: 574 BIC: 1.136e+04
Df Model: 8
Covariance Type: nonrobust

==

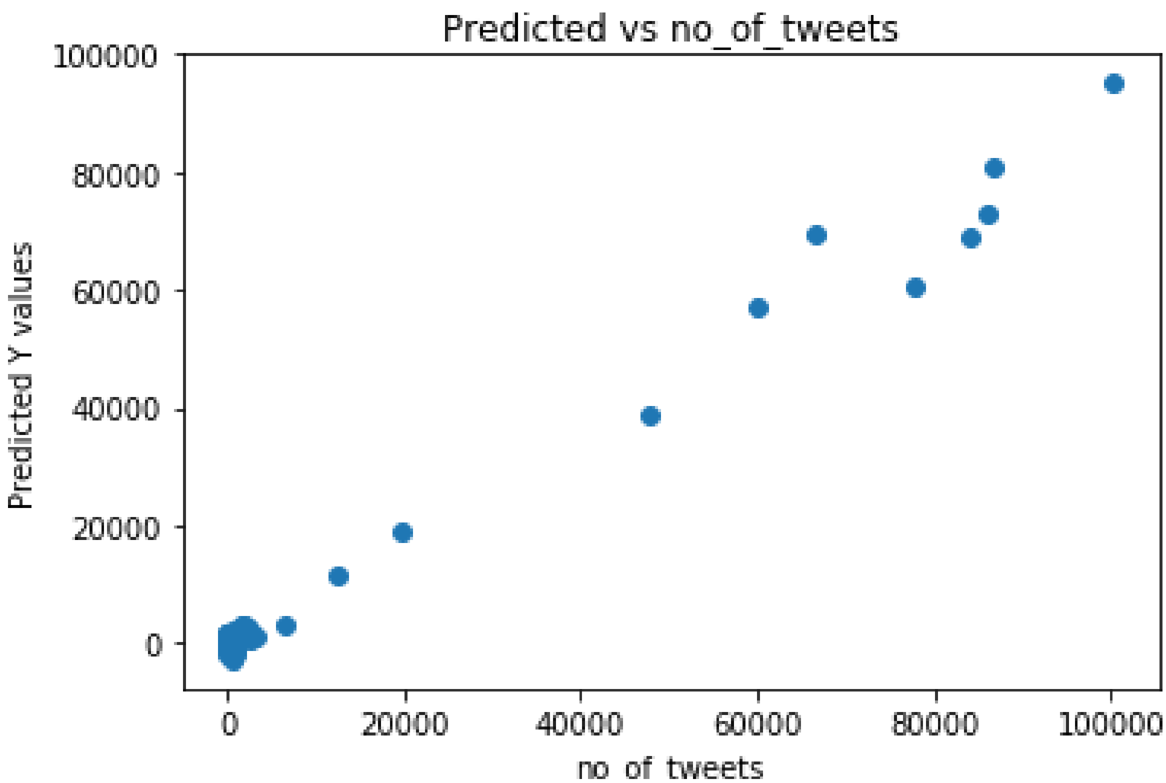
	coef	std err	t	P>|t|	[0.025	0.975]
hours	-13.6049	25.270	-0.538	0.591	-63.238	36.028
no_of_tweets	1.1241	0.088	12.820	0.000	0.952	1.296
retweet_count_sum	-0.1622	0.079	-2.058	0.040	-0.317	-0.007
followers_count_sum	0.0001	7.01e-05	2.003	0.046	2.72e-06	0.000
followers_count_max	0.0001	4.76e-05	2.709	0.007	3.55e-05	0.000
ranking_mean	47.5647	75.278	0.632	0.528	-100.290	195.419
momentum_mean	-22.6204	76.630	-0.295	0.768	-173.130	127.889
impressions_sum	-0.0001	6.97e-05	-1.900	0.058	-0.000	4.48e-06

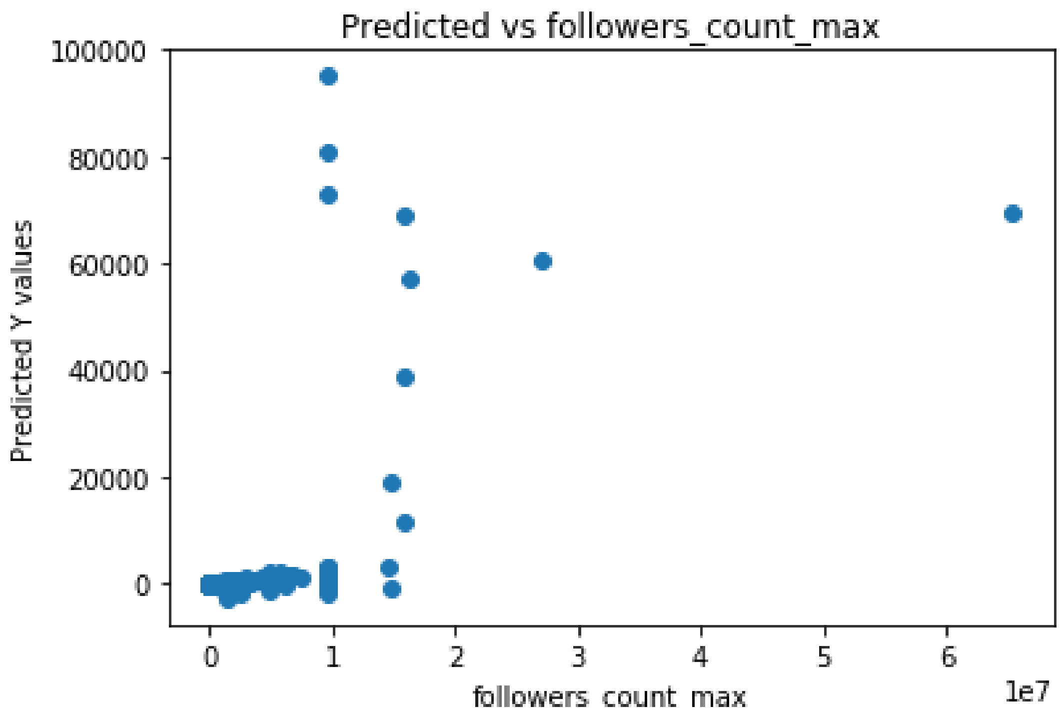
==

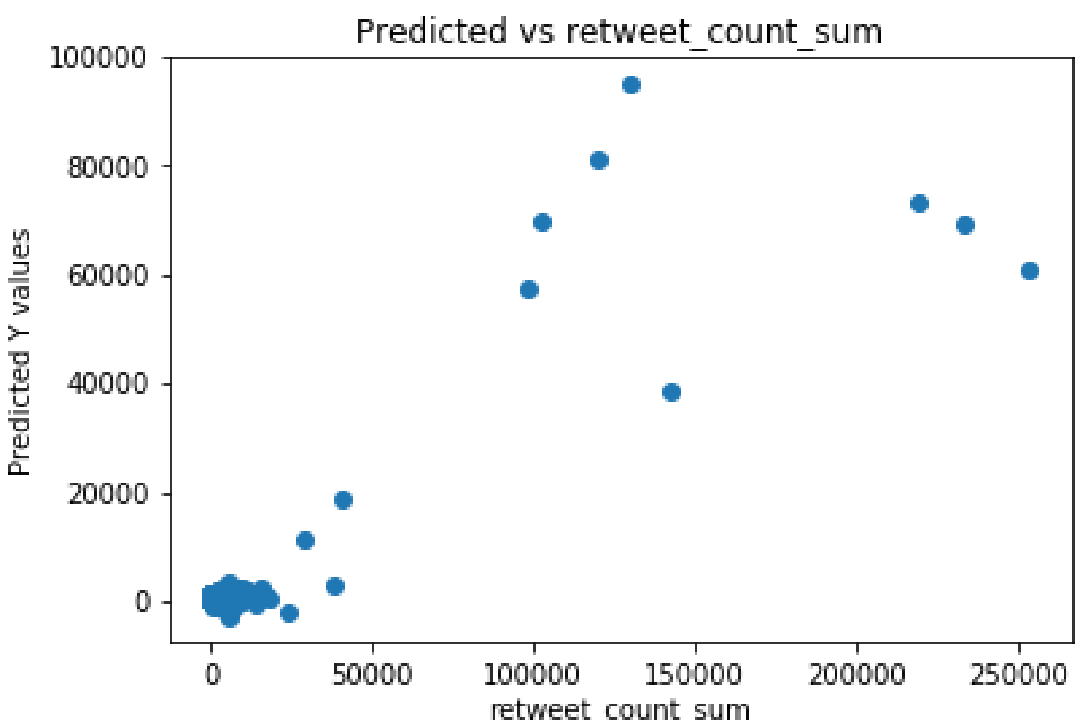
Omnibus:	1177.573	Durbin-Watson:	1.670
Prob(Omnibus):	0.000	Jarque-Bera (JB):	2170824.111
Skew:	14.541	Prob(JB):	0.00
Kurtosis:	300.780	Cond. No.	5.37e+07

==

MSE for #sb49 : 16310145.764435157

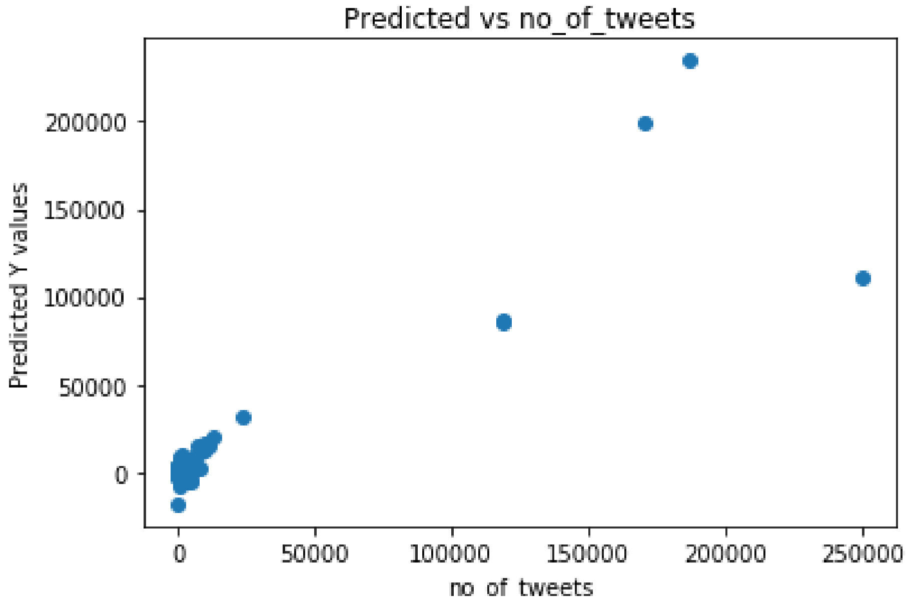


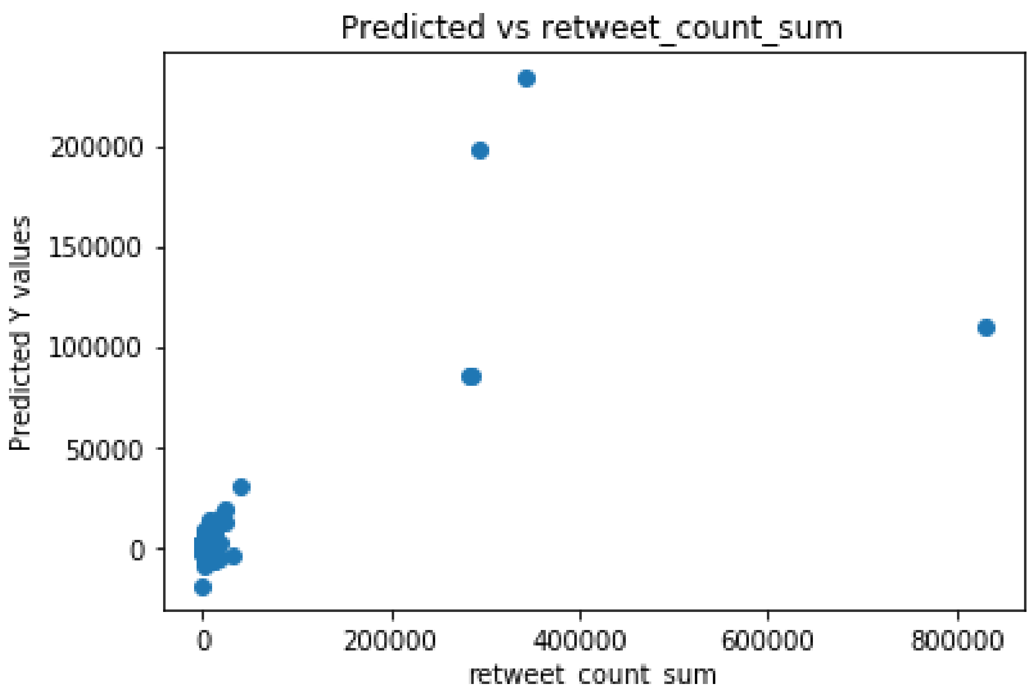

Predicted vs followers_count_max
Predicted Y values
100000
80000
60000
40000
20000
0
0
1
2
3
4
5
6
1e7
followers_count_max

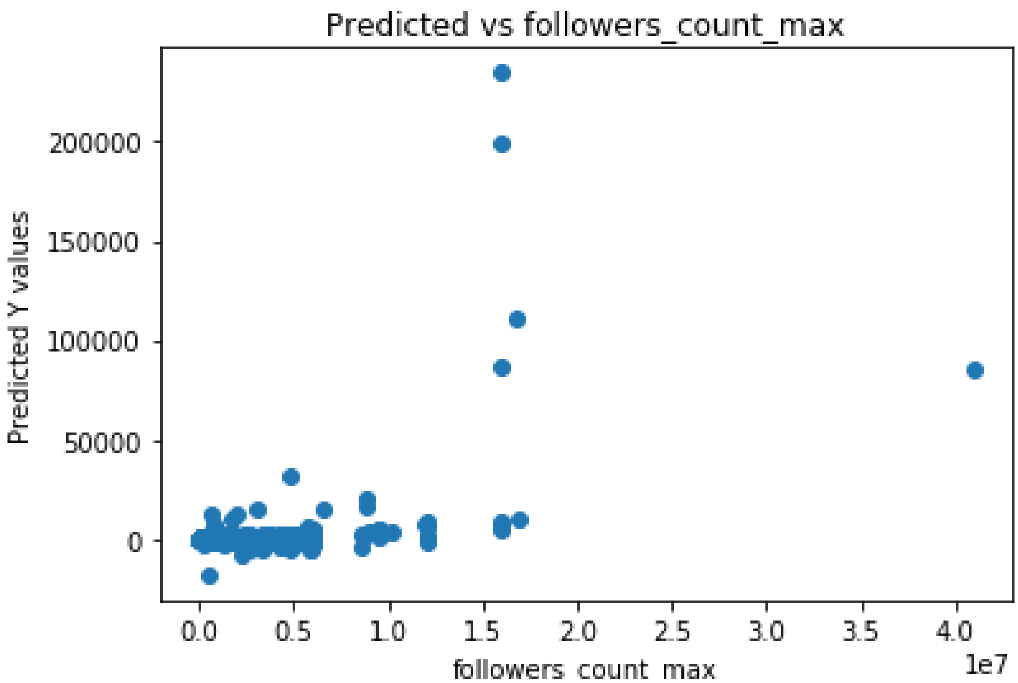

Predicted vs retweet_count_sum
Predicted Y values
100000
80000
60000
40000
20000
0
0
50000
100000
150000
200000
250000
retweet_count_sum

```
------------------------------------------------ #superbowl ----------------------------------
                    OLS Regression Results
==============================================================================
Dep. Variable:          no_of_tweets   R-squared:                       0.809
Model:                           OLS   Adj. R-squared:                  0.806
Method:                Least Squares   F-statistic:                     305.2
Date:               Thu, 14 Mar 2019   Prob (F-statistic):          6.86e-202
Time:                       22:48:05   Log-Likelihood:                -6031.6
No. Observations:                586   AIC:                         1.208e+04
Df Residuals:                    578   BIC:                         1.211e+04
Df Model:                          8
Covariance Type:           nonrobust
==============================================================================
                     coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------
hours                -13.5295     43.152     -0.314      0.754     -98.283      71.224
no_of_tweets           2.2509      0.079     28.477      0.000       2.096       2.406
retweet_count_sum     -0.2671      0.045     -5.882      0.000      -0.356      -0.178
followers_count_sum    0.0008      0.000      3.491      0.001       0.000       0.001
followers_count_max    0.0007      0.000      4.990      0.000       0.000       0.001
ranking_mean           8.3215    139.291      0.060      0.952    -265.257     281.900
momentum_mean        111.3660    873.796      0.127      0.899   -1604.837    1827.569
impressions_sum       -0.0010      0.000     -4.074      0.000      -0.001      -0.001
==============================================================================
Omnibus:                     1012.961   Durbin-Watson:                   2.242
Prob(Omnibus):                  0.000   Jarque-Bera (JB):          1793053.725
Skew:                          10.143   Prob(JB):                         0.00
Kurtosis:                     273.230   Cond. No.                     4.54e+08
==============================================================================
```

MSE for #superbowl : 51720841.903693266


Predicted vs no_of_tweets
Predicted Y values
no_of_tweets
200000
150000
100000
50000
0
50000
100000
150000
200000
250000


Predicted vs retweet_count_sum
Predicted Y values
retweet_count_sum
200000
150000
100000
50000
0
200000
400000
600000
800000



We can see from the graphs that most of the predicted values and the 3 most significant features on each file actually have some trend in some. For instance, in #sb49, we can see that when the value of the feature increase, the predicted y values also increase. This shows that there is some form of trend. This is intuitive because with the most important features, these features would be affecting the equation the most which is why we see this trend. However, for those that are not affecting it as much, this could be because there are also other features that play an important role as well and using just one feature does not really give a trend in the prediction. However, we do see trends in almost all of these top-3 important features we selected based on the p-values.

QUESTION 6: We define three time periods and their corresponding window length as follows:
1. Before Feb. 1, 8:00 a.m.: **1-hour** window
2. Between Feb. 1, 8:00 a.m. and 8:00 p.m.: **5-minute** window
3. After Feb. 1, 8:00 p.m.: **1-hour** window
For each hashtag, train 3 regression models, one for each of these time periods (the times are all in PST).
Report the MSE and R-squared score for each case.

We first divide the data into the period specified above.
Period 1: Not_active
Period 2: Active
Period 3: After_active

For the active period, we group by 5 minutes window instead of the 1-hour window. We then use cross validation with 5 folds in order to choose the best model. After choosing the best model, we predicted it on the whole dataset. This is training each model for each hashtag. This gives us 3x6 which is 18 models because we need 3 models for each period for each hashtag. The MSE and the R2 for each hashtag for each period are shown below:

------------------------ #gohawks active ------------------------
MSE = 73729.51843932953
R2 = 0.48028836807488895
------------------------ #gohawks not_active ------------------------
Fitting 5 folds for each of 1 candidates, totalling 5 fits
MSE = 701388.2912196745
R2 = 0.3196484514434065

------------------------ #gohawks after_active ------------------------
Fitting 5 folds for each of 1 candidates, totalling 5 fits
MSE = 904.7553131763676
R2 = 0.9182293694965499

------------------------ #gopatriots active ------------------------
Fitting 5 folds for each of 1 candidates, totalling 5 fits
MSE = 13822.64803680226
R2 = 0.46181371989285225
------------------------ #gopatriots not_active ------------------------
Fitting 5 folds for each of 1 candidates, totalling 5 fits
MSE = 1730.9947110373832
R2 = 0.5778617518372777
------------------------ #gopatriots after_active ------------------------
Fitting 5 folds for each of 1 candidates, totalling 5 fits
MSE = 33.81709679519066
R2 = 0.8085377714504116

```
------------------------- #nfl active -------------------------
Fitting 5 folds for each of 1 candidates, totalling 5 fits
MSE =  21110.8734010879
R2 =  0.8172525180570319
------------------------- #nfl not_active -------------------------
Fitting 5 folds for each of 1 candidates, totalling 5 fits
MSE =  65364.74238341987
R2 =  0.5158077658401934
------------------------- #nfl after_active -------------------------
Fitting 5 folds for each of 1 candidates, totalling 5 fits
MSE =  16021.246297381975
R2 =  0.813152921829061


------------------------- #patriots active -------------------------
Fitting 5 folds for each of 1 candidates, totalling 5 fits
MSE =  697853.0525191195
R2 =  0.6989371175376538
------------------------- #patriots not_active -------------------------
Fitting 5 folds for each of 1 candidates, totalling 5 fits
MSE =  334661.841273611
R2 =  0.5716534545196443
------------------------- #patriots after_active -------------------------
Fitting 5 folds for each of 1 candidates, totalling 5 fits
MSE =  9945.445755625058
R2 =  0.889416720621399


------------------------- #sb49 active -------------------------
Fitting 5 folds for each of 1 candidates, totalling 5 fits
MSE =  1284650.0745279046
R2 =  0.8663092879264375
------------------------- #sb49 not_active -------------------------
Fitting 5 folds for each of 1 candidates, totalling 5 fits
MSE =  6827.991117529616
R2 =  0.869083362085354
------------------------- #sb49 after_active -------------------------
Fitting 5 folds for each of 1 candidates, totalling 5 fits
MSE =  72191.86528729784
R2 =  0.8031005332334826
```

```
------------------------ #superbowl active ------------------------
Fitting 5 folds for each of 1 candidates, totalling 5 fits
MSE =  6821401.246361589
R2 =  0.8907607286551587
------------------------ #superbowl not_active ------------------------
Fitting 5 folds for each of 1 candidates, totalling 5 fits
MSE =  513625.6650041958
R2 =  0.40478486913255307
------------------------ #superbowl after_active ------------------------
Fitting 5 folds for each of 1 candidates, totalling 5 fits
MSE =  109149.95869667521
R2 =  0.8465692731115256
```

QUESTION 7: Also, aggregate the data of all hashtags, and train 3 models (for the intervals mentioned above) to predict the number of tweets in the next hour on the aggregated data.

Perform the same evaluations on your combined model and compare with models you trained for individual hashtags.

not_active:

```
                    OLS Regression Results
==============================================================================
Dep. Variable:          no_of_tweets   R-squared:                       0.530
Model:                           OLS   Adj. R-squared:                  0.522
Method:                Least Squares   F-statistic:                     60.85
Date:               Mon, 11 Mar 2019   Prob (F-statistic):           4.71e-66
Time:                       00:38:30   Log-Likelihood:                -3981.1
No. Observations:                439   AIC:                             7978.
Df Residuals:                    431   BIC:                             8011.
Df Model:                          8
Covariance Type:           nonrobust
==============================================================================
                      coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------------
hours              -5.8319     14.739     -0.396      0.693     -34.802      23.138
no_of_tweets        0.6153      0.151      4.085      0.000       0.319       0.911
retweet_count_sum  -0.0181      0.077     -0.236      0.814      -0.169       0.133
followers_count_sum -2.509e-05  4.68e-05  -0.537      0.592      -0.000    6.68e-05
followers_count_max -3.161e-05  5.97e-05  -0.529      0.597      -0.000    8.57e-05
ranking_mean      115.5841     48.942      2.362      0.019      19.390     211.779
```

```
momentum_mean      -173.6721    820.299     -0.212      0.832   -1785.956    1438.612
impressions_sum     4.084e-05   4.07e-05     1.004      0.316   -3.91e-05      0.000
==============================================================================
Omnibus:                  791.172   Durbin-Watson:                   2.153
Prob(Omnibus):              0.000   Jarque-Bera (JB):           559190.034
Skew:                      11.066   Prob(JB):                         0.00
Kurtosis:                 176.439   Cond. No.                     3.12e+08
==============================================================================
```

R-Squared = 0.5303890101538111
MSE = 4491114.106272763

active:

```
                            OLS Regression Results
==============================================================================
Dep. Variable:           no_of_tweets   R-squared:                       0.942
Model:                            OLS   Adj. R-squared:                  0.939
Method:                 Least Squares   F-statistic:                     276.5
Date:                Mon, 11 Mar 2019   Prob (F-statistic):           9.02e-80
Time:                        00:38:34   Log-Likelihood:                -1395.2
No. Observations:                 143   AIC:                             2806.
Df Residuals:                     135   BIC:                             2830.
Df Model:                           8
Covariance Type:            nonrobust
==============================================================================
                         coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
5minutes              -6.0147     21.526     -0.279      0.780     -48.587      36.557
no_of_tweets           1.0134      0.090     11.289      0.000       0.836       1.191
retweet_count_sum     -0.0540      0.023     -2.356      0.020      -0.099      -0.009
followers_count_sum 4.974e-05      0.000      0.182      0.856      -0.000       0.001
followers_count_max   5.59e-05   5.82e-05      0.961      0.338   -5.91e-05       0.000
ranking_mean         171.8001    233.732      0.735      0.464    -290.451     634.051
momentum_mean        227.6050   6274.499      0.036      0.971   -1.22e+04    1.26e+04
impressions_sum    -4.694e-05      0.000     -0.169      0.866      -0.001       0.001
==============================================================================
Omnibus:                   27.994   Durbin-Watson:                   1.938
Prob(Omnibus):              0.000   Jarque-Bera (JB):               83.339
Skew:                       0.696   Prob(JB):                     8.00e-19
Kurtosis:                   6.471   Cond. No.                     2.91e+09
==============================================================================
```

R-Squared = 0.9424884620772198
MSE = 18491103.305340026

after_active:

```
                         OLS Regression Results
==============================================================================
Dep. Variable:          no_of_tweets   R-squared:                       0.931
Model:                           OLS   Adj. R-squared:                  0.926
Method:                Least Squares   F-statistic:                     211.9
Date:               Mon, 11 Mar 2019   Prob (F-statistic):           3.08e-69
Time:                       00:38:56   Log-Likelihood:                -1059.6
No. Observations:                134   AIC:                             2135.
Df Residuals:                    126   BIC:                             2158.
Df Model:                          8
Covariance Type:           nonrobust
==============================================================================
                          coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
hours                 -23.0468      8.585     -2.684      0.008     -40.037      -6.057
no_of_tweets            0.6592      0.109      6.026      0.000       0.443       0.876
retweet_count_sum      -0.0441      0.011     -4.135      0.000      -0.065      -0.023
followers_count_sum -2.078e-05   2.26e-05     -0.920      0.360   -6.55e-05    2.39e-05
followers_count_max  2.295e-05   2.09e-05      1.096      0.275   -1.85e-05    6.44e-05
ranking_mean           86.0723     31.502      2.732      0.007      23.731     148.413
momentum_mean         -38.2054     99.909     -0.382      0.703    -235.922     159.511
impressions_sum      4.099e-05   2.16e-05      1.896      0.060   -1.79e-06    8.38e-05
==============================================================================
Omnibus:                     122.073   Durbin-Watson:                   2.222
Prob(Omnibus):                 0.000   Jarque-Bera (JB):             2235.627
Skew:                          3.013   Prob(JB):                         0.00
Kurtosis:                     22.081   Cond. No.                     1.01e+08
==============================================================================
```

R-Squared = 0.9308161698094095
MSE = 459548.3604256475

We can see that when we aggregate the data, our MSE increases. This is intuitive because we are combining all the data and then fitting a model. By fitting one model to each hashtag, we will be training a model to predict values for that particular dataset. If you would like to try, you could try summing up all the MSE for each hashtag, and you would notice that the MSE of the summed up from every model is lower than the aggregated model. However, we can see that the R2 values are actually better when compared to many of the hashtags. R2 tells us how close the data is to our fitted regression line which is very similar to MSE. However, R2 just scales it down. With R2 being better, we can say that the aggregated model fits to the data better because the values are closer to the line. However, it is still very hard to conclude which one is actually better. If you're considering MSE, then splitting them up and training each hashtag is better. If you look at R2, then the aggregated data is better. Personally, we believe that with more data, we should be able to generalize better than with just one particular hashtag and that is also probably why the aggregated model had a better R2 score.

QUESTION 8: Use grid search to find the best parameter set for RandomForestRegressor and GradientBoostingRegressor respectively. Use the following param_grid
{
'max_depth': [10, 20, 40, 60, 80, 100, 200, None],
'max_features': ['auto', 'sqrt'],
'min_samples_leaf': [1, 2, 4],
'min_samples_split': [2, 5, 10],
'n_estimators': [200, 400, 600, 800, 1000,
1200, 1400, 1600, 1800, 2000]
}
Set cv = KFold(5, shuffle=True), scoring='neg_mean_squared_error' for the grid search.
Analyze the result of the grid search. Do the test errors from cross-validation look good? If not, please explain the reason.

RandomForestRegressor:
Best Param = {'max_depth': 60, 'max_features': 'auto', 'min_samples_leaf': 4, 'min_samples_split': 5, 'n_estimators': 1000}
MSE_on_whole_data = 156553106.0123706
R2 = 0.803303237570963
Average Test_neg_MSE from best model across 5 folds = -227819879.19655493

GradientBoostingRegressor:
Best Param = {'max_depth': 80, 'max_features': 'sqrt', 'min_samples_leaf': 2, 'min_samples_split': 2, 'n_estimators': 1200}
MSE_on_whole_data = 9.866646614644697e-08
R2 = 0.9999999999999999
Average Test_neg_MSE from best model across 5 folds = -325997588.5845306

In this part, we perform a gridsearch over those two algorithms with kfold cross-validation where k=5. We obtain the test MSE as above. The test MSE does not look really good. This could be because the test data might not be representative of the training data. This could also be because our dataset is really large. When we have a large dataset, a combination of many small wrong answers square and adding them up will give a really large MSE.

QUESTION 9: Compare the best estimator you found in the grid search with OLS on the entire dataset.

```
                    OLS Regression Results
==============================================================================
Dep. Variable:                  y   R-squared:                   0.839
Model:                        OLS   Adj. R-squared:              0.837
Method:             Least Squares   F-statistic:                 377.1
Date:            Mon, 11 Mar 2019   Prob (F-statistic):      9.66e-224
Time:                    02:23:42   Log-Likelihood:            -6309.4
No. Observations:             586   AIC:                     1.263e+04
Df Residuals:                 578   BIC:                     1.267e+04
Df Model:                       8
Covariance Type:        nonrobust
==============================================================================
            coef    std err        t     P>|t|     [0.025      0.975]
------------------------------------------------------------------------------
x1      -19.6935     69.353   -0.284     0.777   -155.908    116.521
x2        1.5639      0.063   24.993     0.000      1.441      1.687
x3       -0.4864      0.044  -11.133     0.000     -0.572     -0.401
x4        0.0007      0.000    4.876     0.000      0.000      0.001
x5        0.0003      0.000    2.125     0.034   1.98e-05      0.001
x6      105.4242    218.040    0.484     0.629   -322.824    533.672
x7      253.2793   1494.490    0.169     0.865  -2682.013   3188.571
x8       -0.0007      0.000   -4.566     0.000     -0.001     -0.000
==============================================================================
Omnibus:                  758.277   Durbin-Watson:               2.151
Prob(Omnibus):              0.000   Jarque-Bera (JB):       566526.281
Skew:                       5.734   Prob(JB):                     0.00
Kurtosis:                 154.891   Cond. No.                 8.61e+08
==============================================================================
```

R-Squared = 0.8392226806759028
MSE = 133520109.7478503

If you look at the result, OLS actually does better than random forest. Both of these does very inferior compared to gradient boosting. This could be because gradient boosting adds more trees in order to correct the error it made. This would intuitively make the training error really low like in AdaBoosting. However, the testing error as seen above is still very high. The reason OLS does better than random forest could be because of 2 reasons.

1. We haven't searched on the correct parameters of random forest. This is very possible because in general, you would expect random forest to outperform linear regression. If the data is linear, then random forest would be able to find that linear relation, but we just might not be searching on the right parameters.
2. Many of the features have a linear correlation with the output. This could also be the case which shows why OLS is doing better than random forest.

QUESTION 10: For each time period described in Question 6, perform the same grid search above for GradientBoostingRegressor (with corresponding time window length). Does the cross-validation test error change? Are the best parameter set you find in each period agree with those you found above?

Active:
Best Param = {'max_depth': 60, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 10, 'n_estimators': 1200}
MSE_on_whole_data = 9.686402491446615e-08
R2 = 0.9999999999999991
Average Test_neg_MSE from best model across 5 folds = -16444335.403625797

Not_active:
Best Param = {'max_depth': 10, 'max_features': 'sqrt', 'min_samples_leaf': 1, 'min_samples_split': 2, 'n_estimators': 1600}
MSE_on_whole_data = 9.978771956672601e-08
R2 = 0.9999999999999867
Average Test_neg_MSE from best model across 5 folds = -3794649.126084567

After_active:
Best Param = {'max_depth': 100, 'max_features': 'sqrt', 'min_samples_leaf': 4, 'min_samples_split': 2, 'n_estimators': 400}
MSE_on_whole_data = 0.48308918267830486
R2 = 0.9999998498355226
Average Test_neg_MSE from best model across 5 folds = -312957.9584203543

We can see that the value is test MSE is different from that of the aggregated data. We can think of this because we have fewer data, when we have error, the cumulative will be lesser. Another thing we notice is that the parameters for each time period are also different. These are all also different from that of the aggregated data. This shows that these 3 time period could have different kind of data making us converge into different solutions for the parameters. Another reason could be that the number of data points in each period are different. This might make regression harder for some period making them need more complex models.

QUESTION 11: Now try to regress the aggregated data with MLPRegressor. Try different architectures (i.e. the structure of the network) by adjusting hidden_layer_sizes. You should try at least 5 architectures with various numbers of layers and layer sizes. Report the architectures you tried, as well as its MSE of fitting the entire aggregated data.

Architectures tried for the three different time periods (y_active, y_not_active and y_after_active) are as follows:

Architecture 1:

```
mlp_regressor_model = MLPRegressor(hidden_layer_sizes=(5,),
                                   activation='relu',
                                   solver='adam',
                                   learning_rate='adaptive',
                                   max_iter=1000,
                                   learning_rate_init=0.01,
                                   alpha=0.01,
                                   random_state=33,
                                   shuffle = True)
```

Architecture 2:

```
mlp_regressor_model = MLPRegressor(hidden_layer_sizes=(10,),
                                   activation='relu',
                                   solver='adam',
                                   learning_rate='adaptive',
                                   max_iter=10000,
                                   learning_rate_init=0.01,
                                   alpha=0.01,
                                   random_state=33,
                                   shuffle = True)
```

Architecture 3:

```
mlp_regressor_model = MLPRegressor(hidden_layer_sizes=(100,),
                                   activation='relu',
                                   solver='adam',
                                   learning_rate='adaptive',
                                   max_iter=10000,
                                   learning_rate_init=0.01,
                                   alpha=0.01,
                                   random_state=33,
                                   shuffle = True)
```

Architecture 4:

```
mlp_regressor_model = MLPRegressor(hidden_layer_sizes=(50,),
                                   activation='relu',
                                   solver='adam',
                                   learning_rate='adaptive',
                                   max_iter=100000,
                                   learning_rate_init=0.05,
                                   alpha=0.01,
                                   random_state=33,
                                   shuffle = True)
```

Architecture 5:

```
mlp_regressor_model = MLPRegressor(hidden_layer_sizes=(25,),
                                   activation='relu',
                                   solver='adam',
                                   learning_rate='adaptive',
                                   max_iter=10000,
                                   learning_rate_init=0.01,
                                   alpha=0.01,
                                   random_state=33,
                                   shuffle = True)
```

We tried 5 different architectures with hidden_layer_sizes 5, 10, 25, 50 and 100. The MSE and R2 values are as follows:

Architecture 1:

y_active:	MSE = 130011018216.80319	R2 = -1140.4678442059703
y_not_active:	MSE = 125870121.1369884	R2 = -15.785830188295506
y_after_active:	MSE = 13673983381.859936	R2 = -4249.450313490978

Architecture 2:

y_active:	MSE = 6987638579724.154	R2 = -61348.913684910476
y_not_active:	MSE = 206814146.81312934	R2 = -26.580390942535214
y_after_active:	MSE = 2082112076265.9595	R2 = -647207.1821474491

Architecture 3:

y_active:	MSE = 533875334973.1462	R2 = -4686.306784032062
y_not_active:	MSE = 179420092.15116027	R2 = -22.927165335290045
y_after_active:	MSE = 30498662652.293987	R2 = -9479.269692551423

Architecture 4:

y_active:	MSE = 28635905940602.766	R2 = -251415.88965466604
y_not_active:	MSE = 36713345.056442	R2 = -3.896030687783594
y_after_active:	MSE = 180425354238.41098	R2 = -56082.80397051955

Architecture 5:

y_active:	MSE = 12263237981221.438	R2 = -107667.50389608314
y_not_active:	MSE = 44546167.01471574	R2 = -4.940602807831377
y_after_active:	MSE = 7972627811.423561	R2 = -2477.228723414085

QUESTION 12: Use StandardScaler to scale the data before feeding it to MLPRegressor (with the best architecture you got above). Does its performance increase?

Using Standard Scalar to change the mean to 0 and standard deviation to 1 for all the three time periods data.

I used the fifth model to fit all the three time periods arbitrarily since all the models seem to be doing badly and I could not really choose one model which did really well. From the results below we can see that the R2 value has become positive and is very close to 1 for 2 time periods! And the MSE also is lower than the previous results using mdoels without standard scalar.

Architecture 5:

y_active:	MSE = 10332441.352642475	R2 = 0.9092834613715755
y_not_active:	MSE = 4228780.677655294	R2 = 0.4360568360666526
y_after_active:	MSE = 89372.66469139408	R2 = 0.9722192092789139

QUESTION 13: Using grid search, find the best architecture (for scaled data) for each period (with corresponding window length) described in Question 6.

I searched over the following parameters:

```
parameters = {
  'hidden_layer_sizes':[(25,),(50,),(5,),(100,)],
  'activation':['relu'],
  'solver':['sgd', 'adam'],
  'learning_rate':['adaptive'],
  'max_iter':[10000,20000,30000],
  'learning_rate_init':[0.01,0.05,0.001],
}
```

There were 72 candidate combinations for each fit over 5 folds.

y_active:
MSE = 21647506.189987052
R2 = 0.8099397069414975
{'activation': 'relu', 'hidden_layer_sizes': (50,), 'learning_rate': 'adaptive', 'learning_rate_init': 0.01, 'max_iter': 10000, 'solver': 'adam'}

y_not_active:
MSE = 4567079.098533727
R2 = 0.3909419208307032
{'activation': 'relu', 'hidden_layer_sizes': (100,), 'learning_rate': 'adaptive', 'learning_rate_init': 0.001, 'max_iter': 30000, 'solver': 'adam'}

y_after_active:
MSE = 269159.34811946336
R2 = 0.9163339311124871
{'activation': 'relu', 'hidden_layer_sizes': (100,), 'learning_rate': 'adaptive', 'learning_rate_init': 0.001, 'max_iter': 20000, 'solver': 'adam'}

The MSE and R2 values were for the best models chosen. The architectures are as above.

We download the test data. Each file in the test data contains a hashtag's tweets from a 6x-window-length time range. We fit a model on the aggregate of the training data for all hashtags, and predict the number of tweets in the next hour for each test file.

QUESTION 14: Report the model you use. For each test file, provide your predictions on the number of tweets in the next time window.

We compute the aggregate data for train and test using the full dataset into three different categories of active, not active and after active as before. And then predict the number of tweets for the next time window.

The predicted values are as follows for each time_window for each model:

Regression Model:

Not active:

```
Truth:
0    141.0
1    102.0
2    144.0
3    104.0
4     61.0
Predicted
0    632.043427
1    495.871968
2    435.563019
3    475.448692
4    461.106956
```

After active:

```
truth
0    90.0
1    40.0
2    58.0
3    87.0
4    43.0
Predicted
0     736.482965
1     986.672065
2     723.140336
3    3886.330277
4     785.203156
```

Active:

```
Truth
0    19.0
1    25.0
2    27.0
3    29.0
4    28.0
Predicted
0    679.742018
1    584.545688
2    655.463262
3    568.706438
4    656.002781
```

Gradient Boosting:

Not Active:

```
Truth
0    141.0
1    102.0
2    144.0
3    104.0
4     61.0
Predicted
0    507.96329427
1    326.14843536
2    152.94977143
3    168.16142402
4    553.13153223
```

Active

```
truth
0    19.0
1    25.0
2    27.0
3    29.0
4    28.0
Predicted
0    2897.67743257
1    2979.37018827
2    3813.47451105
3    2393.12023284
4    1933.86690294
```

After Active:

```
truth
0    90.0
1    40.0
2    58.0
3    87.0
4    43.0
Predicted
0    65.1334798
1    109.52640157
2    84.29687957
3    556.22096201
4    216.09323515
```

The textual content of a tweet can reveal some information about the author. Recognizing that supporting a sport team has a lot to do with the user location, we try to use the textual content of the tweet posted by a user to predict their location. In order to make the problem more specific, let us consider all the tweets including #superbowl, posted by the users whose specified location is either in the state of Washington (not D.C.!) or Massachusetts.

QUESTION 15:

1. Explain the method you use to determine whether the location is in Washington, Massachusetts or neither. Only use the tweets whose authors belong to either Washington or Massachusetts for the next part.

We read the data line by line and store the two features from the json_object['tweet']['user']['location'] and "text" of the tweet and store it in a dataframe.

We define two sets of places in Massachusetts and Washington including their abreviations, and famous cities.

Washington: seattle, Washington, WA, Kirkland, Spokane, Redmond, Centurylink.

Massachusetts: Bellevue, Boston, Gillette, MA, Massachusetts, Mass, Springfield.

We process the data to get a better analysis of the twitter feed. We convert all the location and text to lower case and remove all the punctuations. Then we take only those tweets that contain #superbowl in them across all the six files and drop the other tweets. We also drop all the duplicate tweets.
Since we are predicted based on location, we drop all the tweets which don't have a location tag to it and reset the index for further analysis. We add a label column to classify the tweets into Massachusetts and Washington as 0 and 1 respectively and do a set intersection between the earlier defined sets and the locations in the dataframe.
We drop all the tweets which don't belong to these two location categories.
We also drop the location and label columns to make our dataset ready for prediction.

2. Train a binary classifier to predict the location of the author of a tweet (Washington or Massachusetts), given only the textual content of the tweet (using the techniques you learnt in project 1). Try different classification algorithms (at least 3). For each, plot ROC curve, report confusion matrix, and calculate accuracy, recall and precision.

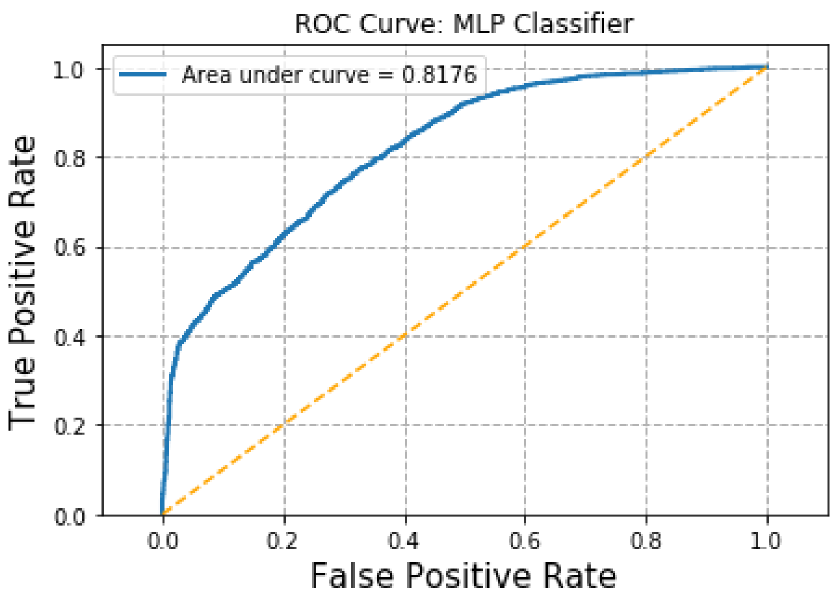
Using the text we perform Count vectorization and TFIDF on the text data of the tweets. We remove all the stop words and punctuations. We then perform Singular Value Decomposition on the resultant data to reduce the dimensions, using n_components=50.

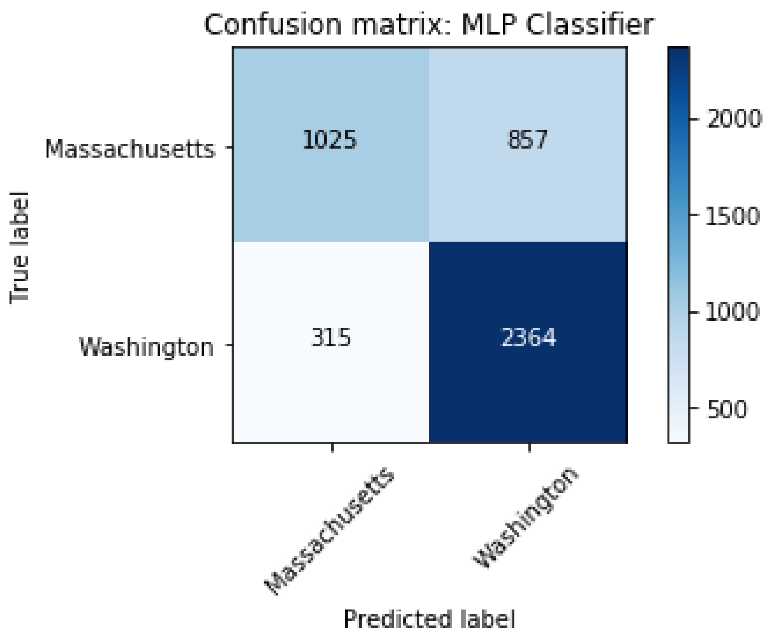
We split the data into test and train of 0.10 and 0.90 respectively.

With the resultant data we train 3 models: Logistic Regression, Random Forest and MLP Classifier to predict the location of the author of the tweet. The results are as follows:

MLP Classifier:

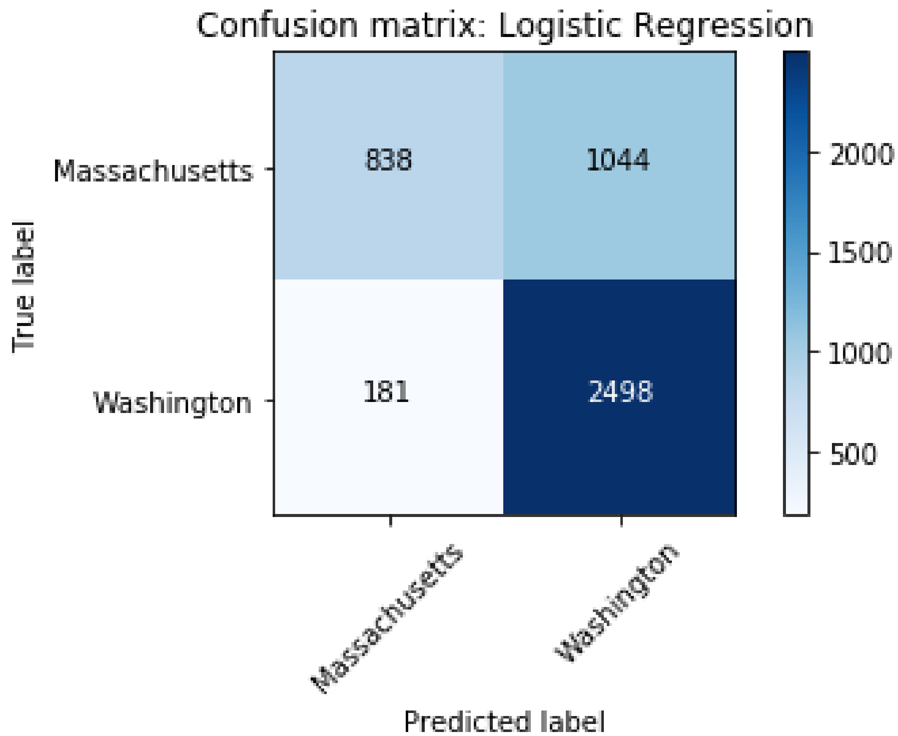
```
Accuracy 0.7430388072791054
Precision 0.7339335610058988
Recall 0.8824188129899216
```





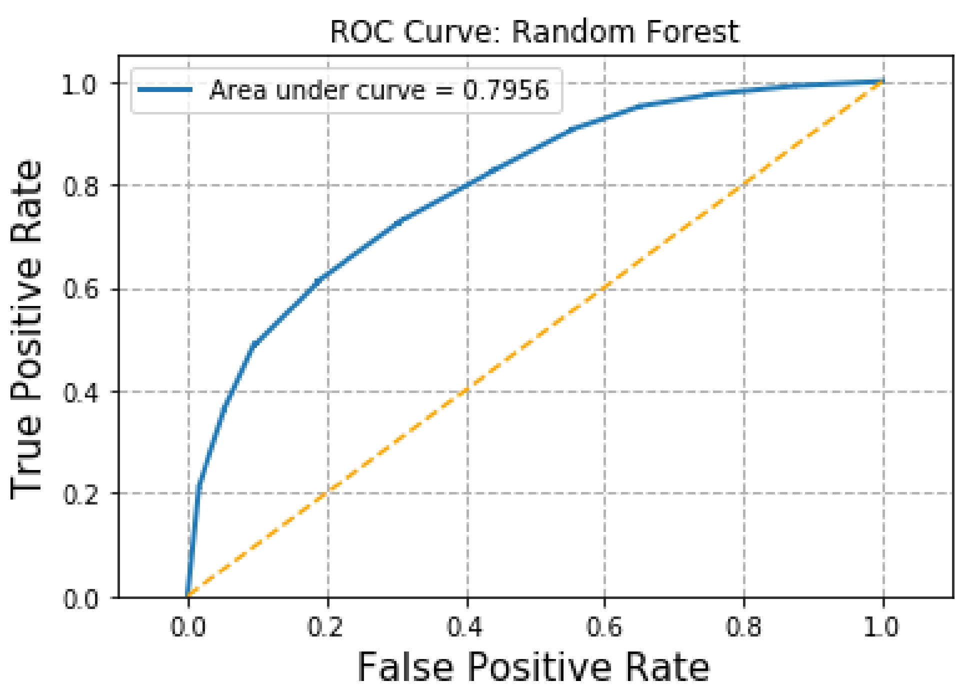
Logistic Regression:

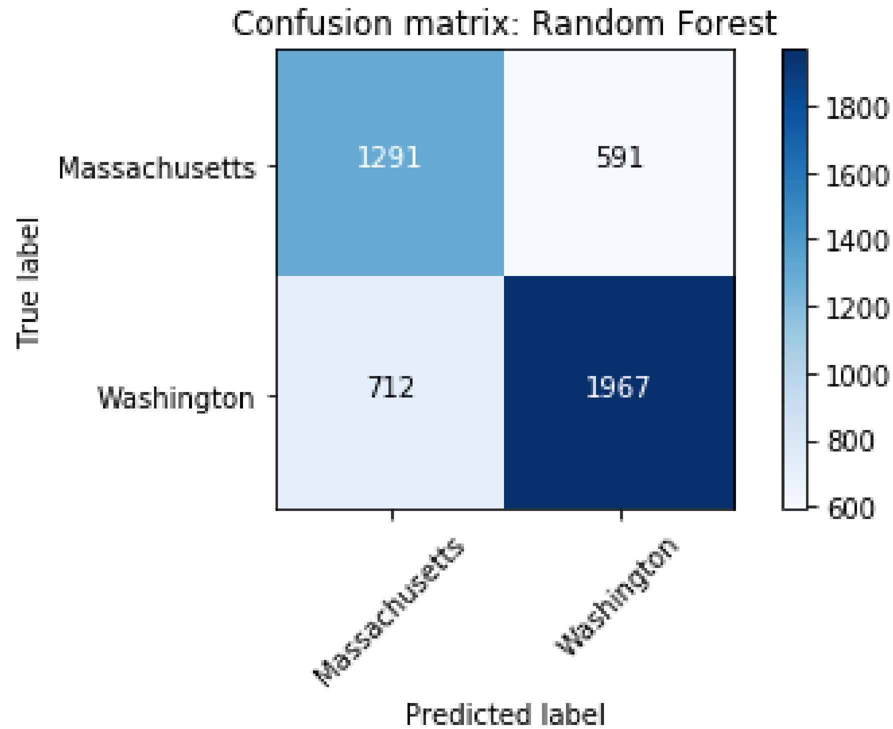
```
Accuracy 0.7314185485639114
Precision 0.7052512704686618
Recall 0.9324374766703994
```



Random Forest:

```
Accuracy 0.7143170357377768
Precision 0.7689601250977326
Recall 0.7342291899962673
```





QUESTION 16: The dataset in hands is rich as there is a lot of metadata to each tweet. Be creative and propose a new problem (something interesting that can be inferred from this dataset) other than the previous parts. You can look into the literature of Twitter data analysis to get some ideas. Implement your idea and show that it works. As a suggestion, you might provide some analysis based on changes of tweet sentiments for fans of the opponent teams participating in the match. You get full credit for bringing in novelty and full or partial implementation of your new ideas.

The twitter data is rich in features and we decided to predict the number of followers a user has by extracting the relevant features from the users' data. Being able to predict this can enable us to gain insight about popular tweet-ers. We do this for commercial companies to target such user to propagandize their products and lead to targeted marketing. For example, a user in this dataset with high number of followers is probably a sports celebrity, and therefore sports companies like Adidas and Nike can target such users with their sports products for advertising.

To accomplish this, we loaded the data sequentially as before. We cleaned up the data by removing duplicate users from the data-frame across different files. While removing duplicates, we take the user value with the highest timestamp as that insures that we take into account the most updated tweet (data). We encode the locations to convert it to a number so that we can use it as a metric for prediction. We drop the users without location.

For prediction, we drop the followers count and make it our label. The features that we use finally for training are:

- profile_use_background_image

- verified
- location
- statuses_count
- friends_count
- favourites_count

We split the data into training and testing. And then further split the training data to training and validation set. We tried various algorithms for prediction and got the best results with a linear regression model with polynomial features. We found the best polynomial feature using cross validation.

We predict the follower count on the test set.

The RMSE Score for the best model found are reported as follows:

```
Validation RMSE 151518.14534382615
Best Polynomial Feature 2
Test MSE 137439.18534455352
```

The predicted statistics are as follows:

```
Min follower count 0
Max follower count 40623398
Mean follower count 9204.20654518565
Std deviation of follower count 149588.73239317065
```

We plot the truth vs predicted values.