ECE 219
Large Scale Data Mining: Models and Algorithm

Project 5
Popularity Prediction on Twitter

Haitao Wang (UID: 504294402)
Weiqian Xu (UID: 404297854)
Xinyi Gu (UID: 205034975)
Yutan Gu(UID: 005034976)

1. Popularity Prediction

1.1 Calculate Statistics and Histogram

The task in this section is to calculate the average number of tweets per hour, average number of followers of users, and average number of retweets for each hashtags. All data is loaded using JSON. First, to get the average number of tweets per hour, it is necessary to record the timestamp of each tweet. The timestamps can be accessed from data['citation date'], and are converted from seconds to hours after the loading process. Since each tweet has its unique timestamp, the number of tweets in total is just the total number of timestamps. So the average number of tweets per hour is the total amount of timestamps divided by the total hours.

Then, the average number of followers can be calculated by using total number of followers divides total number of users. The followers of each user can be accessed from data['author']['followers'], and sum them up. The total number of users is just the total amount of tweets. Although one user can submit multiple tweets, multiple tweets are not counted as duplicate since this parameter will be used to predict the number of tweets later.

Similar to the previous step, the average number of retweets can be calculated by using total number of retweets divides the total number of tweets. The number of retweets can be accessed from data['metrics']['citations']['total'].

The result summary of each hashtag is shown in the following:

	avg. of tweets per hour	average followers	average retweets
#gohawks	325.371591	2203.931767	2.014617
#gopatriots	45.694511	1401.895509	1.400084
#nfl	441.323431	4653.252286	1.538533
#patriots	834.555509	3309.978828	1.782816
#sb49	1419.887907	10267.316849	2.511149
#superbowl	2302.500402	8858.974663	2.388272

Table 1.1 Statistics of each hashtags

The last step of this section is to plot the histogram of the number of tweets per hour using hashtags '#nfl' and '#superbowl'. To record the number of tweets in every hour, an index indicates hour is calculated first. Then tweets are stored in the particular time slot according to their timestamps. The results are shown in the following pictures. From the histograms, the tweets distribution of hashtag '#nfl' are relatively even compared to the tweets distribution of '#superbowl'. Both two hashtags has a sharp peak between hour 400-500. This period may be the time of Super Bowl game.

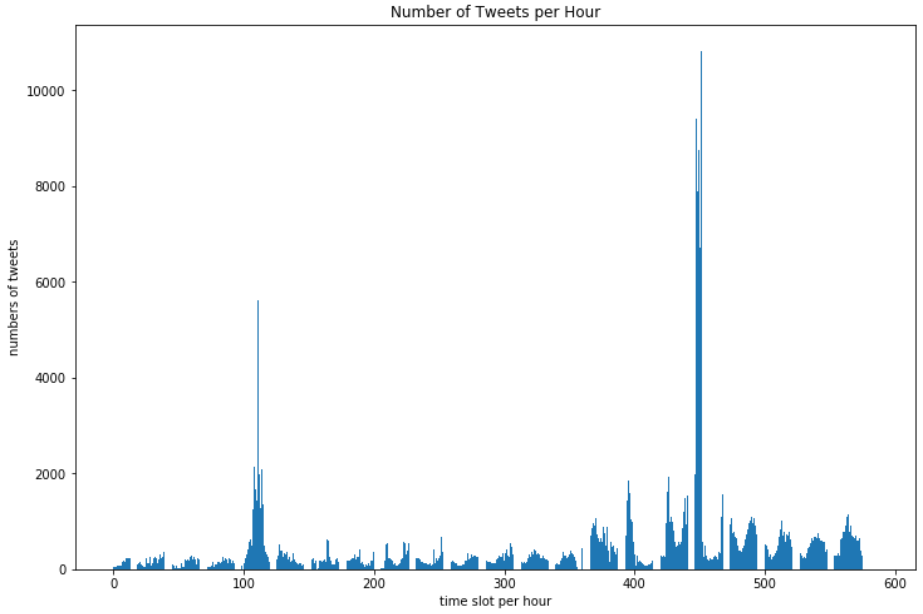


Figure 1.1 Number of Tweets per Hour (#nfl)

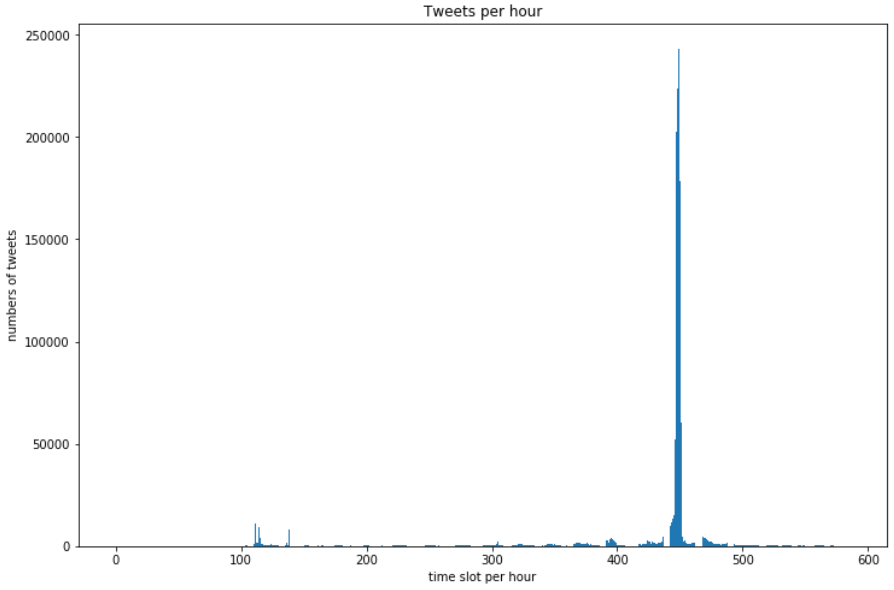


Figure 1.2 Number of Tweets per Hour (#superbowl)

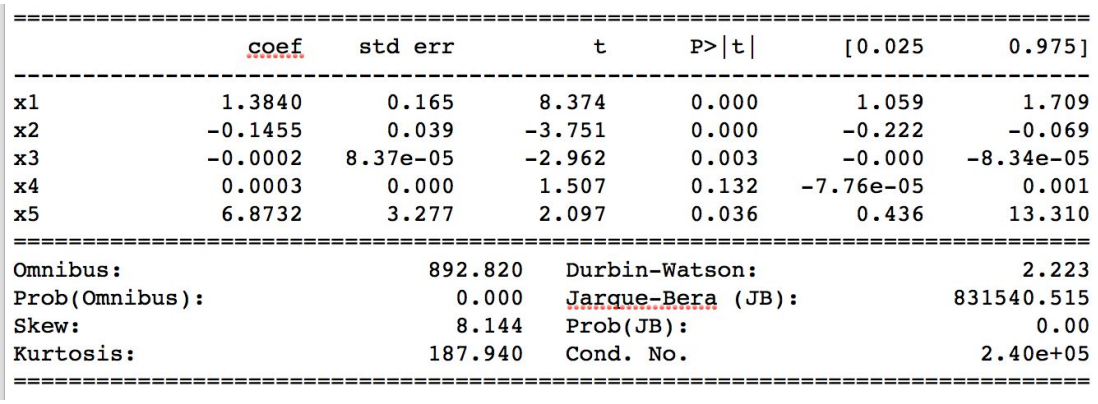
1.2 Fit a Linear Regression Model

The main purpose of this section is to fit a linear regression model. 5 features are extracted from the data to predict the number of tweets in the next hour. In other words, features from the first hour are used to predict the number of tweets in the second hour, and features from the second hour are used to predict the number of tweets in the third hour and so on. 5 features used in this section are: Number of tweets, Total number of retweets, Sum of the number of followers of the users, Maximum number of followers of the users, Time of the day. All features are stored in the unit of hour. Time of the day has 24 values to indicate which hour the tweets' timestamps belong to. Each hashtag is trained in a separate model, and we have calculate the RMSE to evaluate the fit

performance. Moreover, we also use R-squared value to evaluate the model performance, and P values to analyze the significance of each feature. The statistical results of each hashtag is shown in the following:

#gohawks	
RMSE	949.160291
R-squared value	0.519
Significance of feature:	number of tweets
	total number of retweets
	sum of the number of followers
	time of the day
	maximum followers

Table 1.2 statistical summary of #gohawks



	coef	std err	t	P>|t|	[0.025	0.975]
x1	1.3840	0.165	8.374	0.000	1.059	1.709
x2	-0.1455	0.039	-3.751	0.000	-0.222	-0.069
x3	-0.0002	8.37e-05	-2.962	0.003	-0.000	-8.34e-05
x4	0.0003	0.000	1.507	0.132	-7.76e-05	0.001
x5	6.8732	3.277	2.097	0.036	0.436	13.310

Omnibus:	892.820	Durbin-Watson:	2.223
Prob(Omnibus):	0.000	Jarque-Bera (JB):	831540.515
Skew:	8.144	Prob(JB):	0.00
Kurtosis:	187.940	Cond. No.	2.40e+05

Figure 1.3 t-test and P-value of #gohawks

#gopatriots	
RMSE	194.159482
R-squared value	0.611
Significance of feature:	maximum followers
	sum of the number of followers
	total number of retweets
	number of tweets
	time of the day

Table 1.3 statistical summary of #gopatriots

	coef	std err	t	P>|t|	[0.025	0.975]
x1	-0.4223	0.264	-1.601	0.110	-0.940	0.096
x2	0.4597	0.230	1.997	0.046	0.008	0.912
x3	0.0006	0.000	3.168	0.002	0.000	0.001
x4	-0.0007	0.000	-3.741	0.000	-0.001	-0.000
x5	0.8828	0.737	1.198	0.231	-0.564	2.330

Omnibus:	448.732	Durbin-Watson:	2.086
Prob(Omnibus):	0.000	Jarque-Bera (JB):	346360.433
Skew:	2.088	Prob(JB):	0.00
Kurtosis:	123.164	Cond. No.	3.82e+04

Figure 1.4 t-test and P-value of #gopatriots

#nfl	
RMSE	581.692518
R-squared value	0.646
Significance of feature:	number of tweets
	time of the day
	sum of the number of followers
	total number of retweets
	maximum followers

Table 1.4 statistical summary of #nfl

	coef	std err	t	P>|t|	[0.025	0.975]
x1	0.7610	0.135	5.624	0.000	0.495	1.027
x2	-0.1736	0.066	-2.635	0.009	-0.303	-0.044
x3	7.181e-05	2.62e-05	2.742	0.006	2.04e-05	0.000
x4	-6.806e-05	3.59e-05	-1.896	0.058	-0.000	2.44e-06
x5	7.4596	2.203	3.386	0.001	3.132	11.787

Omnibus:	562.004	Durbin-Watson:	2.328
Prob(Omnibus):	0.000	Jarque-Bera (JB):	352595.554
Skew:	3.204	Prob(JB):	0.00
Kurtosis:	122.896	Cond. No.	4.26e+05

Figure 1.5 t-test and P-value of #nfl

#patriots	
RMSE	2368.900207
R-squared value	0.716
Significance of feature:	number of tweets
	total number of retweets
	maximum followers
	sum of the number of followers
	time of the day

Table 1.5 statistical summary of #patriots

	coef	std err	t	P>|t|	[0.025	0.975]
x1	1.2160	0.079	15.394	0.000	1.061	1.371
x2	-0.3386	0.068	-4.945	0.000	-0.473	-0.204
x3	3.507e-05	2.62e-05	1.336	0.182	-1.65e-05	8.66e-05
x4	0.0002	9.49e-05	1.653	0.099	-2.95e-05	0.000
x5	7.8002	8.226	0.948	0.343	-8.357	23.957

Omnibus:	1019.171	Durbin-Watson:	1.949
Prob(Omnibus):	0.000	Jarque-Bera (JB):	973670.954
Skew:	10.563	Prob(JB):	0.00
Kurtosis:	201.401	Cond. No.	7.71e+05

Figure 1.6 t-test and P-value of #patriots

#sb49	
RMSE	4007.561062
R-squared value	0.844
Significance of feature:	number of tweets
	maximum followers
	total number of retweets
	sum of the number of followers
	time of the day

Table 1.6 statistical summary of #sb49

	coef	std err	t	P>|t|	[0.025	0.975]
x1	1.2907	0.095	13.557	0.000	1.104	1.478
x2	-0.2969	0.087	-3.398	0.001	-0.469	-0.125
x3	2.893e-05	1.38e-05	2.092	0.037	1.77e-06	5.61e-05
x4	0.0002	4.12e-05	4.218	0.000	9.29e-05	0.000
x5	-14.7066	13.894	-1.058	0.290	-41.996	12.583

Omnibus:	959.832	Durbin-Watson:	1.400
Prob(Omnibus):	0.000	Jarque-Bera (JB):	715084.377
Skew:	9.510	Prob(JB):	0.00
Kurtosis:	173.516	Cond. No.	7.17e+06

Figure 1.7 t-test and P-value of #sb49

#superbowl	
RMSE	6519.787283
R-squared value	0.869
Significance of feature:	number of tweets
	total number of retweets
	sum of the number of followers
	maximum followers
	time of the day

Table 1.7 statistical summary of #superbowl

	coef	std err	t	P>|t|	[0.025	0.975]
x1	2.5465	0.107	23.765	0.000	2.336	2.757
x2	-0.1547	0.035	-4.387	0.000	-0.224	-0.085
x3	-0.0002	1.08e-05	-20.237	0.000	-0.000	-0.000
x4	0.0011	0.000	10.433	0.000	0.001	0.001
x5	-55.8322	24.147	-2.312	0.021	-103.258	-8.407

Omnibus:	1138.770	Durbin-Watson:	1.845
Prob(Omnibus):	0.000	Jarque-Bera (JB):	1944105.490
Skew:	13.283	Prob(JB):	0.00
Kurtosis:	283.920	Cond. No.	1.08e+07

Figure 1.8 t-test and P-value of #superbowl

Compared the result of each hashtag, we can conclude that larger dataset gives better R squared value but larger RMSE. For example, #gopatriots has only 26232 data, so its RMSE is as small as 194.16, but its R squared value is only 0.611. #superbowl has 1348767 data, so its RMSE is as large as 6519.79, but its R squared value is 0.869, which is much better than that of #gopatriots. Each hashtag has its unique significance of features. But summarize the p values of all hashtags, we can see that feature Time of the day does not contribute too much on the model performance since it has little significance on most of the hashtags.

1.3 Fit a Linear Regression Model using features we choose

From section 1.2, we have found out that feature 'Time of the day' does not contribute good on the performance of linear regression model. In this section, we decide to remove the feature 'Time of the day', and add 3 more features we find useful for the model fit: favourite count, maximum retweets, and rank scores. Favourite count can be accessed from data['tweet']['user']['favourites_count'], and rank scores can be accessed from data['metrics']['ranking_score']. Similar to section 1.2, each hashtag is trained in a separate model, and we have calculate the RMSE to evaluate the fit performance. Moreover, we also use R-squared value to evaluate the model performance, and P values to analyze the significance of each feature. The statistical results of each hashtag is shown in the following tables. For the top 3 features of each hashtags, we also draw a scatter plot of predicted values vs. feature values. The plots are also shown below.

#gohawks	
RMSE	772.238208
R-squared value	0.682
Significance of feature:	total number of retweets
	sum of the number of followers
	favourite counts
	maximum retweets
	maximum followers
	ranking scores
	number of tweets

Table 1.8 statistical summary of #gohawks

	coef	std err	t	P>|t|	[0.025	0.975]
x1	1.0008	1.985	0.504	0.614	-2.898	4.900
x2	-0.4082	0.105	-3.870	0.000	-0.615	-0.201
x3	-0.0003	6.77e-05	-4.254	0.000	-0.000	-0.000
x4	0.0004	0.000	2.709	0.007	0.000	0.001
x5	0.0016	0.000	12.901	0.000	0.001	0.002
x6	0.4939	0.175	2.826	0.005	0.151	0.837
x7	-0.6612	0.437	-1.513	0.131	-1.519	0.197

Omnibus:	931.982	Durbin-Watson:	1.831
Prob(Omnibus):	0.000	Jarque-Bera (JB):	622422.618
Skew:	9.102	Prob(JB):	0.00
Kurtosis:	162.589	Cond. No.	2.75e+05

Figure 1.9 t-test and P-value of #gohawks

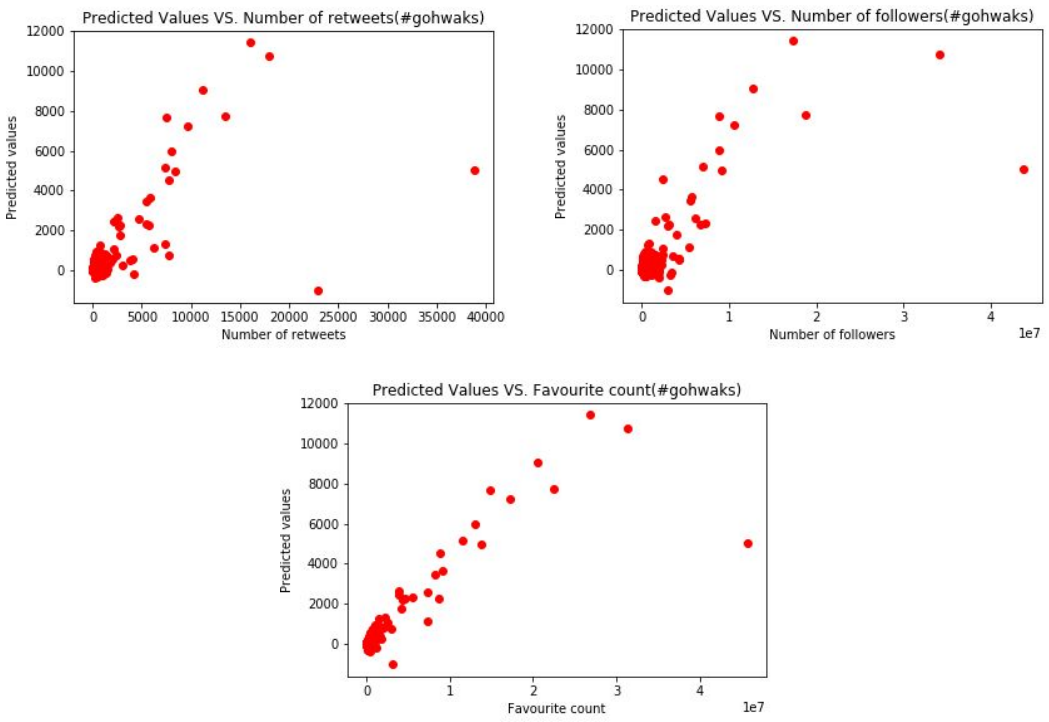


Figure 1.10 top 3 features scatter plot of #gohawks

#gopatriots	
RMSE	129.714748
R-squared value	0.820
Significance of feature:	number of tweets
	total number of retweets
	favourite counts
	maximum retweets
	sum of the number of followers
	ranking scores
	maximum followers

Table 1.9 statistical summary of #gopatriots

	coef	std err	t	P>|t|	[0.025	0.975]
x1	5.7498	1.466	3.923	0.000	2.871	8.629
x2	1.8163	0.378	4.807	0.000	1.074	2.558
x3	0.0004	0.000	2.707	0.007	0.000	0.001
x4	-0.0002	0.000	-1.548	0.122	-0.001	6e-05
x5	-0.0027	0.000	-25.179	0.000	-0.003	-0.003
x6	-2.2443	0.558	-4.022	0.000	-3.340	-1.148
x7	-0.5078	0.284	-1.787	0.074	-1.066	0.050

Omnibus:	633.655	Durbin-Watson:	1.945
Prob(Omnibus):	0.000	Jarque-Bera (JB):	103740.219
Skew:	4.705	Prob(JB):	0.00
Kurtosis:	68.127	Cond. No.	1.93e+05

Figure 1.11 t-test and P-value of #gopatriots

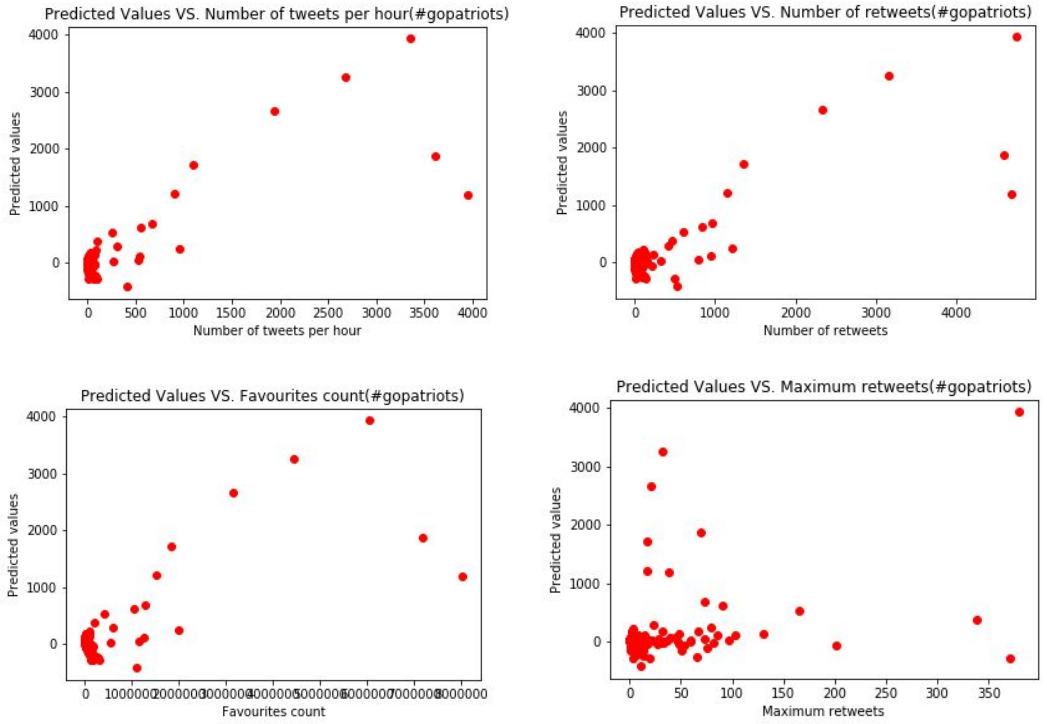


Figure 1.12 top 4 features scatter plot of #gopatriots

#nfl	
RMSE	577.004537
R-squared value	0.649

Significance of feature:	number of tweets
	ranking scores
	total number of retweets favourite counts
	sum of the number of followers
	maximum retweets
	maximum followers
	favourite counts

Table 1.10 statistical summary of #nfl

	coef	std err	t	P>|t|	[0.025	0.975]
x1	6.7772	1.671	4.056	0.000	3.495	10.059
x2	-0.2236	0.095	-2.356	0.019	-0.410	-0.037
x3	5.727e-05	2.66e-05	2.152	0.032	5.01e-06	0.000
x4	-3.55e-05	3.62e-05	-0.982	0.327	-0.000	3.55e-05
x5	-2.802e-05	0.000	-0.254	0.800	-0.000	0.000
x6	0.1509	0.149	1.012	0.312	-0.142	0.444
x7	-1.2944	0.378	-3.422	0.001	-2.037	-0.551

Omnibus:	540.894	Durbin-Watson:	2.386
Prob(Omnibus):	0.000	Jarque-Bera (JB):	355866.900
Skew:	2.961	Prob(JB):	0.00
Kurtosis:	123.478	Cond. No.	3.38e+05

Figure 1.13 t-test and P-value of #nfl

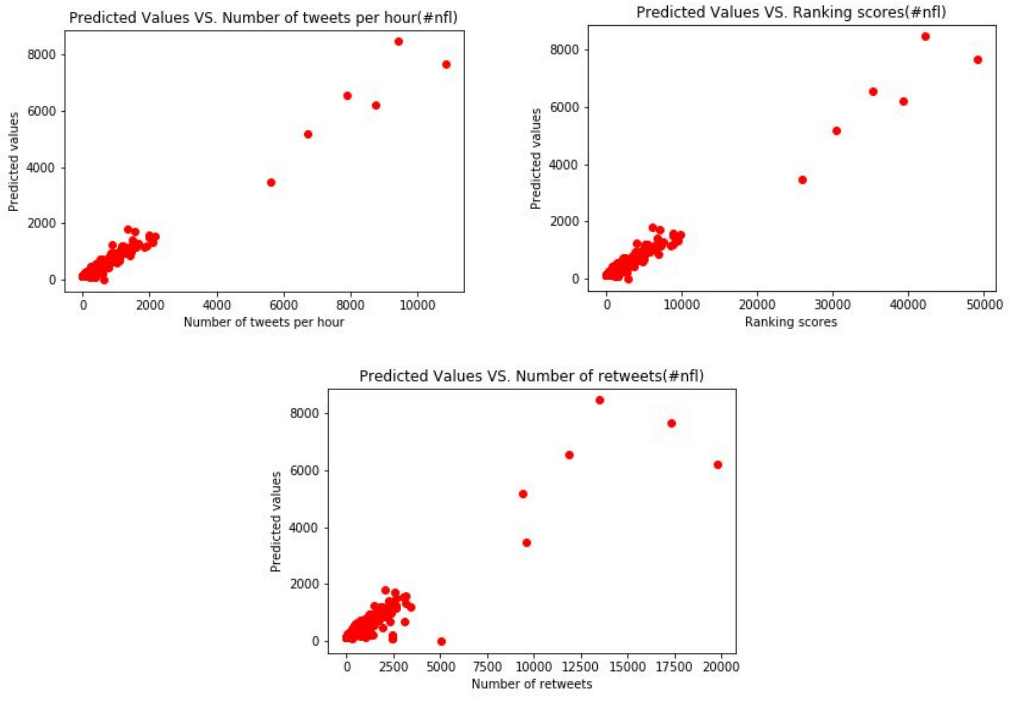


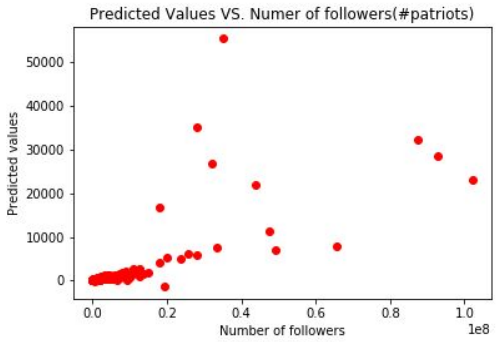
Figure 1.14 top 3 features scatter plot of #nfl

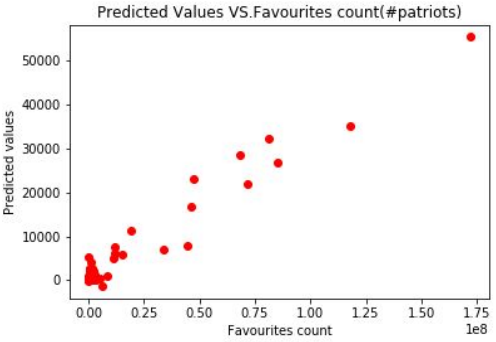
#patriots	
RMSE	2214.981531
R-squared value	0.752
Significance of feature:	sum of the number of followers
	favourite counts
	ranking scores
	total number of retweets
	maximum followers
	number of tweets
	maximum retweets

Table 1.11 statistical summary of #patriots

	coef	std err	t	P>|t|	[0.025	0.975]
x1	3.3243	1.520	2.188	0.029	0.340	6.309
x2	-0.2414	0.096	-2.525	0.012	-0.429	-0.054
x3	0.0003	5.39e-05	6.314	0.000	0.000	0.000
x4	-0.0003	0.000	-2.255	0.025	-0.000	-3.24e-05
x5	0.0005	7.55e-05	6.739	0.000	0.000	0.001
x6	0.2637	0.317	0.832	0.406	-0.359	0.886
x7	-0.9367	0.358	-2.619	0.009	-1.639	-0.234

Figure 1.15 t-test and P-value of #patriots





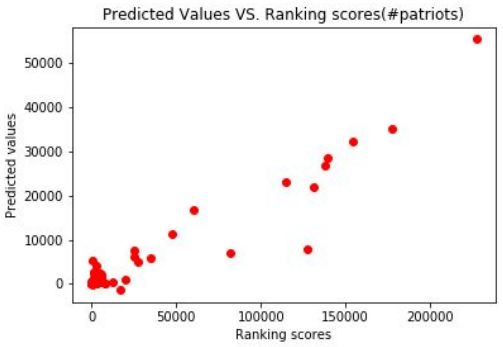


Figure 1.16 top 3 features scatter plot of #patriots

#sb49	
RMSE	3586.569732
R-squared value	0.875
Significance of feature:	number of tweets
	sum of the number of followers
	favourite counts
	ranking scores
	maximum followers
	total number of retweets
	maximum retweets

Table 1.12 statistical summary of #sb49

	coef	std err	t	P>|t|	[0.025	0.975]
x1	11.6445	1.503	7.749	0.000	8.693	14.596
x2	-0.1383	0.141	-0.982	0.326	-0.415	0.138
x3	0.0002	1.86e-05	8.350	0.000	0.000	0.000
x4	-0.0001	5.01e-05	-2.317	0.021	-0.000	-1.77e-05
x5	0.0005	6.67e-05	7.226	0.000	0.000	0.001
x6	0.0483	0.219	0.221	0.825	-0.381	0.477
x7	-2.9815	0.386	-7.728	0.000	-3.739	-2.224

Omnibus:	915.795	Durbin-Watson:	1.500
Prob(Omnibus):	0.000	Jarque-Bera (JB):	822681.383
Skew:	8.502	Prob(JB):	0.00
Kurtosis:	186.242	Cond. No.	9.19e+05

Figure 1.17 t-test and P-value of #sb49

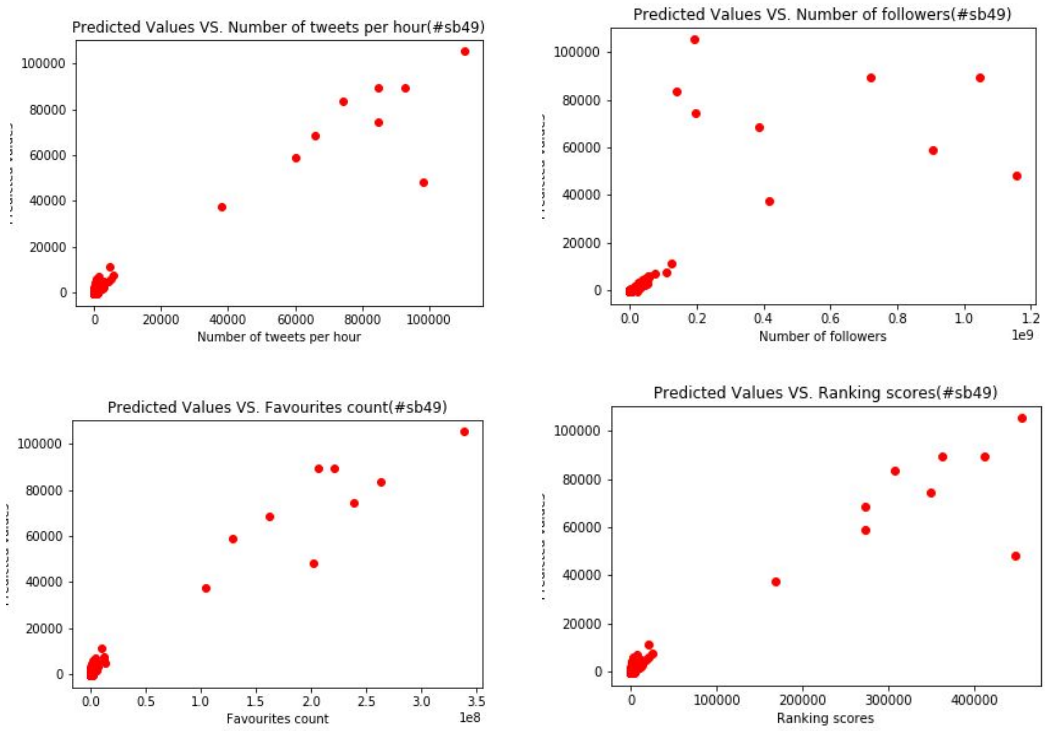


Figure 1.18 top 4 features scatter plot of #sb49

#superbowl	
RMSE	6496.941012
R-squared value	0.869
Significance of feature:	total number of retweets
	sum of the number of followers
	maximum followers
	maximum retweets
	ranking scores
	favourite counts
	number of tweets

Table 1.13 statistical summary of #superbowl

	coef	std err	t	P>|t|	[0.025	0.975]
x1	-1.2384	4.105	-0.302	0.763	-9.302	6.825
x2	-0.1486	0.039	-3.799	0.000	-0.225	-0.072
x3	-0.0002	1.56e-05	-13.415	0.000	-0.000	-0.000
x4	0.0010	8.99e-05	11.005	0.000	0.001	0.001
x5	9.882e-05	0.000	0.362	0.717	-0.000	0.001
x6	-0.5392	0.307	-1.755	0.080	-1.143	0.064
x7	0.7763	0.974	0.797	0.426	-1.136	2.689

Figure 1.19 t-test and P-value of #superbowl

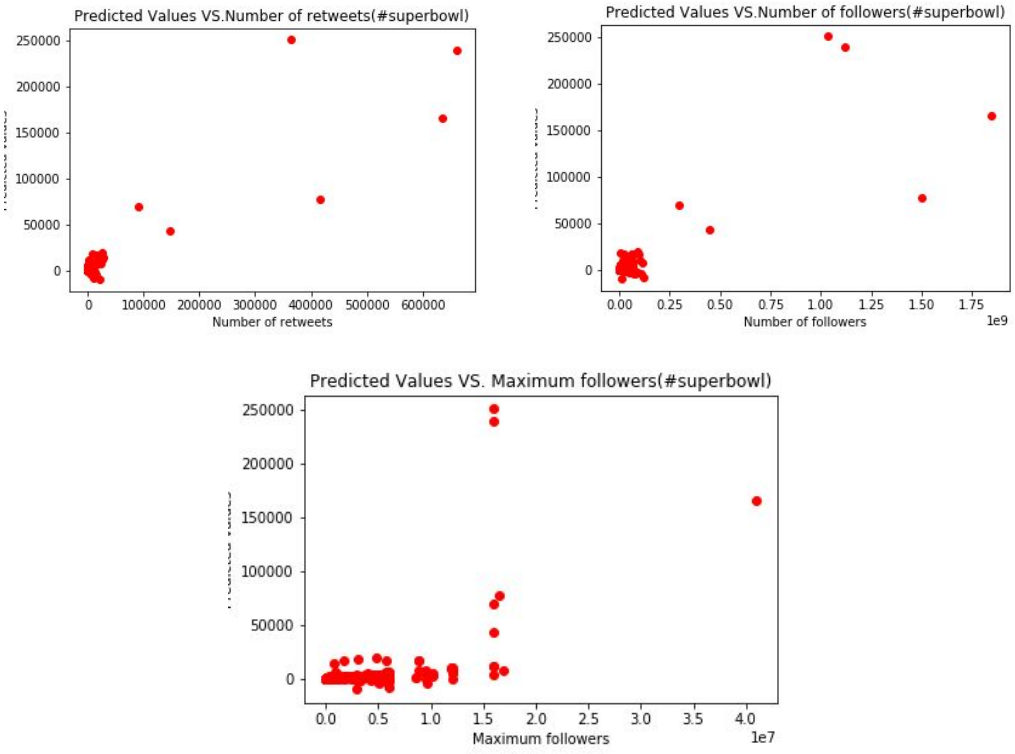


Figure 1.20 top 3 features scatter plot of #superbowl

Compared with the result from section 1.2, we can see that the new features yields a model performance since both RMSE and R-squared values are improved. For example, in section 1.2, RMSE of #gopatriots is 194.16, and R squared value is only 0.611. However, by using new features, RMSE of #gopatriots is 129.71, improved by 33%. R squared values is 0.820, improved by 34%.

For the scatter plot of predicted values vs. features values, plots from all hashtags has a relatively linear relationship. For #gopatriots and #sb49, since 4 features has p-values of 0, so we choose to draw 4 scatter plots. For the third plot of #superbowl, it is not quite obvious to see the linear relationship in the figure. The reason of this problem is that p-value of the third feature(maximum followers) is relatively large. However, since the RMSE of #superbowl drops more than 1000 compared with the result in section 1.2, it is still reasonable to say that new features improves the performance of linear regression model.

1.4 Model Comparison

For this part, we want to introduce more regression models to the task, especially these nonlinear ones. Moreover, we are going to use 10-fold cross validation to evaluate the model.

To see the performance of each model in different situations, we split the whole time span into 3 periods. The two split points of these three periods are Feb. 1, 8:00 a.m. and Feb. 1 8:00 p.m. Thus, the first period indicates one day or more before the superbowl game, during which the tweets are relatively inactive. The second period is exactly in or close to the game, during which the tweets are active. Finally, the tweets will be less and less.

For the features, we will use all the features as we discussed in the 1.3 section above. These features are num_tweets_hour, num_retweets_hour, num_followers_hour, maximum_follower, favourites_count, maximum_retweets and num_ranking_scores.

For the models, we decide to use linear regression model, KNN model and random forest regression model. The latter two models are nonlinear ones. In KNN, we set the number of neighbors to 5. In random forest regression, we set the number of trees to 10 and max_features to 7, which is exactly all the features we use.

Thus, we have 3 different models which will be applied to 3 different periods in 6 different hashtags. Thus, we have overall 54 MAE results after the experiment. The results are shown as following:

	Linear	KNN	RF
#gohawks	275.960716	127.125254	175.267378
#gopatriots	22.644594	11.545645	11.905293
#nfl	123.758905	123.499841	119.219170
#patriots	333.595942	176.492410	222.078911
#sb49	40.410492	61.854651	45.143820
#superbowl	302.266136	318.356945	277.074630

Table 1.14 Average Test MAE using 10-fold CV (Before Feb. 1, 8:00 a.m.)

	Linear	KNN	RF
#gohawks	5629.302606	2184.250000	2173.825000
#gopatriots	1375.506178	852.280000	763.925000
#nfl	4932.455134	2962.430000	2493.260000
#patriots	35532.004309	16463.970000	18302.700000

#sb49	60173.428458	33339.480000	31783.620000
#superbowl	486602.065735	60151.860000	67685.320000

Table 1.15 Average Test MAE using 10-fold CV (Between 8:00 a.m. and 8:00 p.m.)

	Linear	KNN	RF
#gohawks	130.150709	19.904487	18.644120
#gopatriots	29.890780	2.037692	1.496957
#nfl	240.889135	140.111099	112.216011
#patriots	285.728559	74.038571	71.262363
#sb49	405.549713	169.958791	109.577143
#superbowl	1391.395157	280.094396	245.104670

Table 1.16 Average Test MAE using 10-fold CV (After Feb. 1, 8:00 p.m.)

Combining these three tables, we can see that the Average MAE of the second period is significantly larger than the other two periods. There might be three factors that contributes to this phenomenon. First of all, from part 1.1, we notice that the peak number of tweets during the second period is significantly larger than the other two periods. This may surely result in larger absolute error. Secondly, the time span of the second period is less than the other periods, which means that we have less data points to do the training. Thus, the model cannot be fully trained with adequate data points, which may cause larger error. Finally, due to the starting of the superbowl game, there might be an irregular large increasement of the number of tweets, making the model much harder to predict.

Comparing these three models, it seems that in most cases, random forest performs a little better than KNN, while the linear regression model performs much worse than the latter two. This result does make intuitive sense since the the latter two models are both nonlinear while the dataset itself is certainly nonlinear. It is difficult for a linear model to predict a nonlinear dataset with low error, especially when irregular peak appears in the second period.

Next, we aggregate all the hashtags to get a more general dataset. We first extract the same features for all the hashtags, and then concatenate them into a new aggregated dataset. By aggregating all the hashtags, we thus have more training data in each period. We then make prediction on this extended dataset to see what we can find. We will use the best model we found in the previous experiment, which is random forest regression model.

	Before Active	During Active	After Active
Random Forest	130.622565	866.768182	14.548464

Table 1.17 Average Test MAE on the Aggregated Data

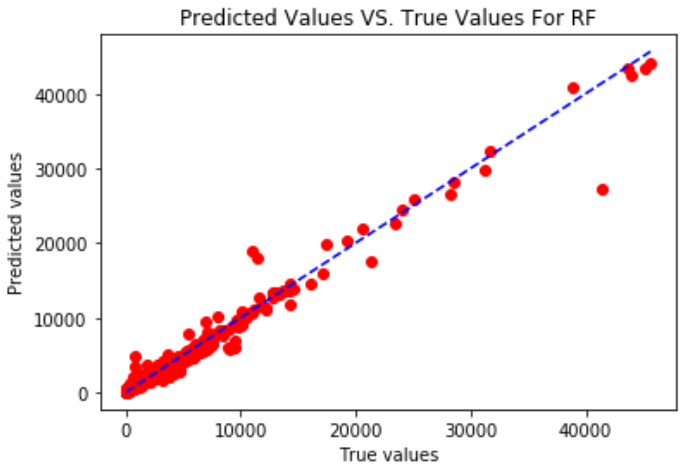


Figure 1.21 Predicted VS True for RF on Aggregated Data (Before Active)

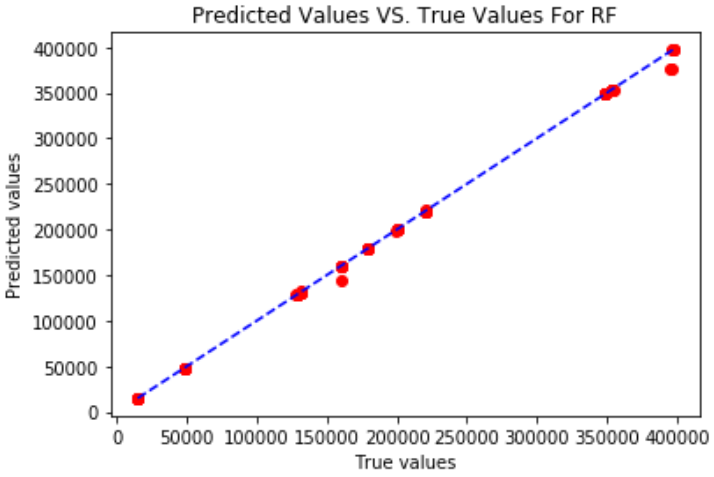


Figure 1.22 Predicted VS True for RF on Aggregated Data (During Active)

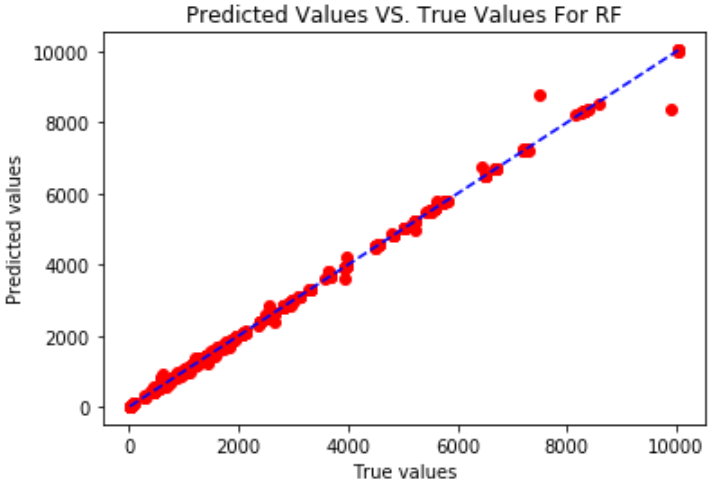


Figure 1.23 Predicted VS True for RF on Aggregated Data (After Active)

Compared to the previous RF model we trained for individual hashtags, it seems that the average MAE of the combined model is smaller than most of individual average MAE of the hashtag. With the predicted vs true plots, we can see that the points are all not far from the true line. The results are good. This is probably because of that by aggregating all the hashtags, we have more data points in the dataset to train the model. The model will then more likely to perform a better result with more training data.

1.5 Apply to Test Data

We now introduce some new test data. Instead of using 10-fold cross validation, we will now use the whole aggregated data to train the model. However, the aggregation method in this part is not quite the same as what we did in part 1.4. We will now aggregate the data in all training hashtags and recalculate the features instead of concatenating the individual features.. Also, to get a more general model, we do not split the data into three periods as before.

Notice that in each test file, the 'first_post_date' attribute spans over a 6-hour window. We now use 'firstpost_date' as the time measure in this part. Our target is to use the features of the first five 1-hour windows(which produce a new 5-hour window) to predict the next hour. In the previous part, we have 7 features in a 1-hour window. Thus, for a 5-hour window, we have a total of 7*5=35 features in each training window.

During the experiment, we find that in 'sample8_period1.txt', there is only a 5-hour time span. Thus, we decide to do a little change when dealing with this special test set. We now use a 4-hour window for training, and predict the fifth hour. For the other 9 test sets, we still use the original 5-hour window.

We use the best model we found in the previous parts, which the Random Forest Regression model, with number of trees = 100, max_features = 35 (or 28 for 'sample8_period1.txt') and max_depth = 10.

The results are shown as follows:

mae (random forest) for sample1_period1.txt is 110.189120
num_tweets_hour = [137. 82. 68. 94. 171. 178.]
predicted_6th_hour = [288.18911962]
mae (random forest) for sample2_period2.txt is 4803.160000
num_tweets_hour = [7591. 9361. 10374. 20066. 81958. 82923.]
predicted_6th_hour = [87726.16]
mae (random forest) for sample3_period3.txt is 264.879700
num_tweets_hour = [442. 549. 610. 888. 616. 523.]
predicted_6th_hour = [787.87970005]
mae (random forest) for sample4_period1.txt is 184.090633
num_tweets_hour = [421. 255. 236. 266. 267. 201.]
predicted_6th_hour = [385.09063301]
mae (random forest) for sample5_period1.txt is 98.110231
num_tweets_hour = [351. 505. 352. 359. 282. 210.]
predicted_6th_hour = [308.11023122]
mae (random forest) for sample6_period2.txt is 11198.540000
num_tweets_hour = [980. 12943. 60627. 52695. 41016. 37293.]
predicted_6th_hour = [26094.46]
mae (random forest) for sample7_period3.txt is 10.294603
num_tweets_hour = [125. 102. 66. 60. 55. 120.]
predicted_6th_hour = [109.70539684]
mae (random forest) for sample8_period1.txt is 89.546920
num_tweets_hour = [49. 72. 56. 41. 11.]

predicted_5th_hour = [100.54692004]
mae (random forest) for sample9_period2.txt is 891.389329
num_tweets_hour = [1729. 1734. 1619. 1582. 1857. 2790.]
predicted_6th_hour = [1898.6106711]
mae (random forest) for sample10_period3.txt is 49.866678
num_tweets_hour = [64. 53. 67. 62. 58. 61.]
predicted_6th_hour = [110.86667828]

From the result, we can see the actual number of tweets per hour over the 6-hour span, along with predicted 6th hour result and the absolute error value. The results are good, giving us a relatively accurate number of how the number of tweets goes in the next hour.

2. Fan Base Prediction

The textual content of a tweet can reveal some information about the author. Here, we are interested in the tweets including #superbowl, posted by the users whose specified location is either in the state of Washington or Massachusetts. The purpose of this part is to predict the location based on texts of tweets. The very first step is to extract most tweets with location specified as Washington or Massachusetts. The process to extract these tweets is as follows:

1. List all unique locations to decrease searching time for following steps
2. Search by regular expressions and filter out irrelevant tweets:

Regex for Washington	Regex for Massachusetts
'[^a-z]wa[^a-z]'	'[^a-z]ma[^a-z]'
'[^a-z]wa$'	'[^a-z]ma$'
'washington$'	'massachusetts$'
'^wa[^a-z]'	'^ma[^a-z]'
'^wa$'	'^ma$'

3. Exact the city names by searching in the filtered tweets with regex as shown below:

Regex for Washington	Regex for Massachusetts
'^(.*), wa$'	'^(.*), ma$'
'^(.*),wa$'	'^(.*),ma$'
'^(.*), wa[.*]'	'^(.*), ma[.*]'
wa[^a-z]'	ma[^a-z]'

City names in Washington ranked by frequency:

'seattle', 'spokane', 'tacoma', 'bellingham', 'vancouver', 'bellevue', 'redmond', 'puyallup', 'olympia', 'kirkland', 'everett', 'renton', 'pullman', 'yakima', 'kennewick', 'woodinville', 'kent', 'perth', 'auburn'

City names in Massachusetts ranked by frequency:

'boston', 'cambridge', 'worcester', 'springfield', 'somerville', 'brookline', 'lowell', 'cape cod', 'quincy', 'natick', 'waltham', 'new bedford', 'watertown', 'beverly',

'yarmouth', 'salem', 'amherst', 'marlborough'

4. Then, use the chosen city names along with the regular expressions in step 2 to find all candidate tweets, which will be used as the dataset for model building.

The total number of tweets is 55246, in which 26584 of tweets are related to Massachusetts State and 28662 of tweets are related to Washington State. In order to predict the location information based on tweet content, we build four different classification models along with two different dimension reduction methods. The models are Support Vector Machine model, Naïve Bayesian model, Logistic Regression model, and Random Forest Classification model. The methods for dimension reduction are Latent Semantic Indexing and Non-negative Matrix Factorization.

2.1 Latent Semantic Indexing(LSI)

The top 50 singular vectors are extracted for projection purpose.

2.1.1 SVM model with LSI

We implement two types of SVM models, hard SVM and soft SVM, separately with C = 1000 and C = 1. The ROC plots are shown in Figure 2.1. The confusion matrices are shown in Figure 2.2.

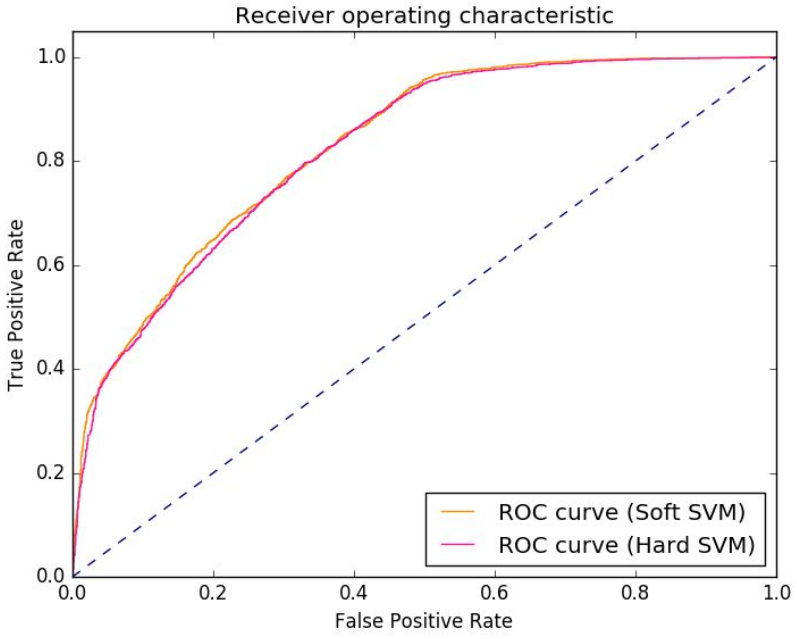


Figure 2.1: ROC curves for SVM

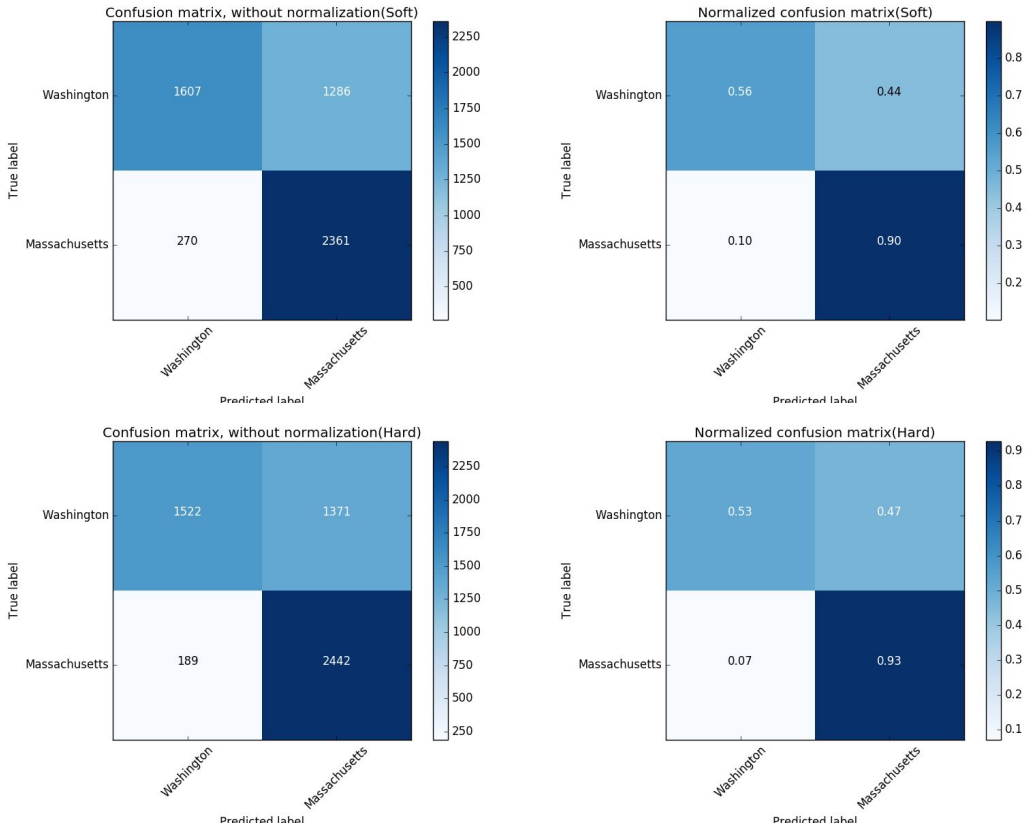


Figure 2.2: Confusion matrices for SVM

Metrics	Average Accuracy	Average Precision	Average Recall
Soft SVM	0.72	0.76	0.72
Hard SVM	0.72	0.77	0.72

Table 2.1: metrics for SVM

2.1.2 NB model with LSI

We implement two types of NB models, Multinomial NB and Gaussian NB. The ROC plots are shown in Figure 2.3. The confusion matrices are shown in Figure 2.4.

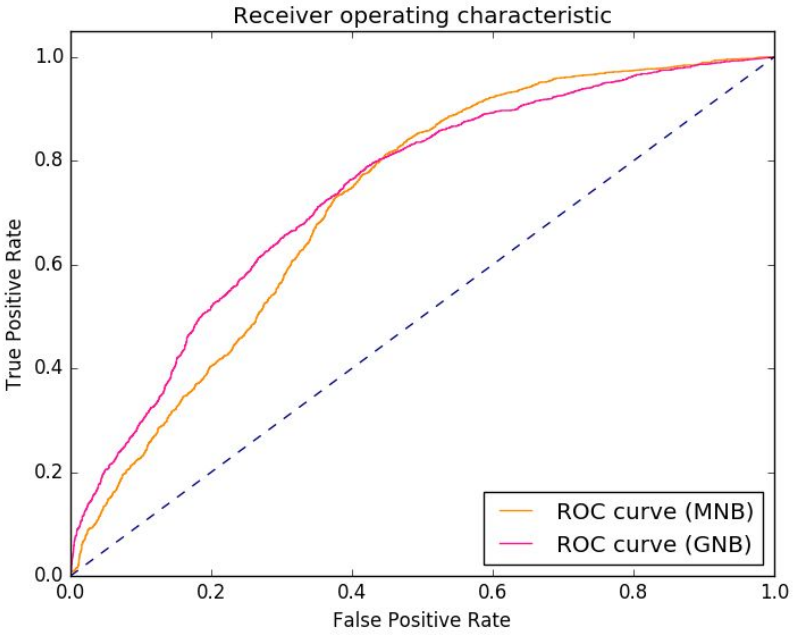


Figure 2.3: ROC curves for Naïve Bayesian models

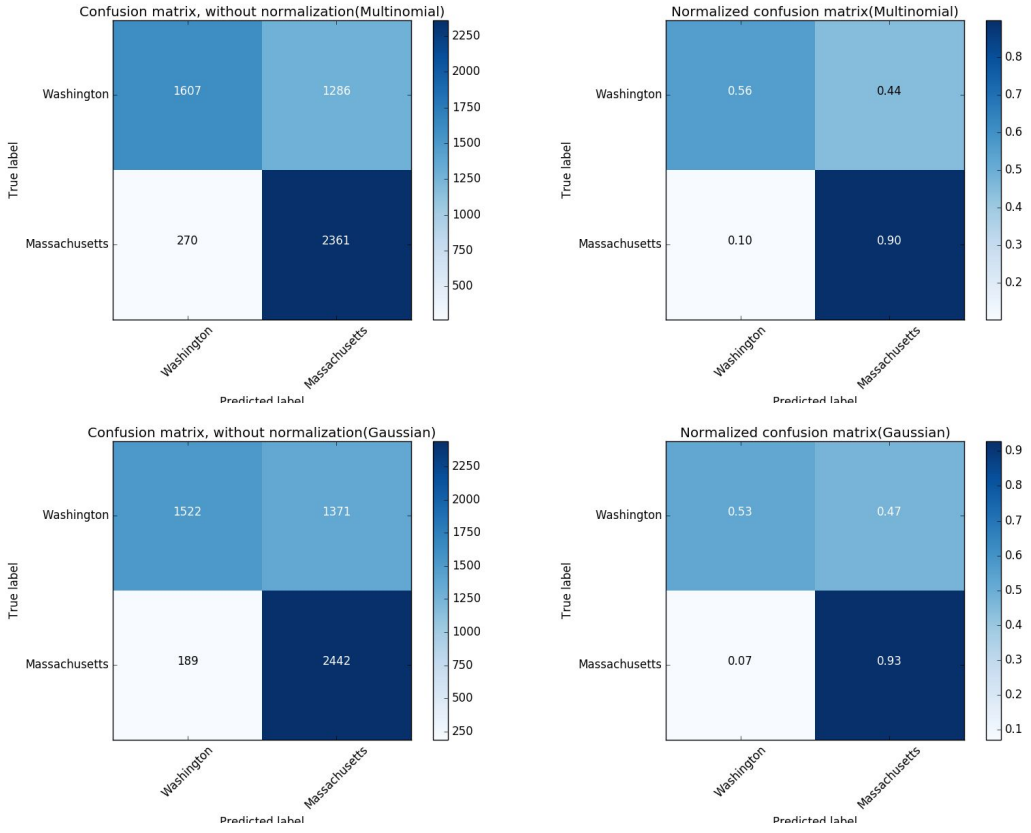


Figure 2.4: Confusion matrices for Naïve Bayesian models

Metrics	Average Accuracy	Average Precision	Average Recall
Multinomial	0.52	0.27	0.52
Gaussian	0.67	0.69	0.67

Table 2.2: metrics for naïve Bayesian model

2.1.3 Logistic Regression model with LSI

We implement logistic regression models with C = 1000. The ROC plot is shown in Figure 2.5. The confusion matrix is shown in Figure 2.6.

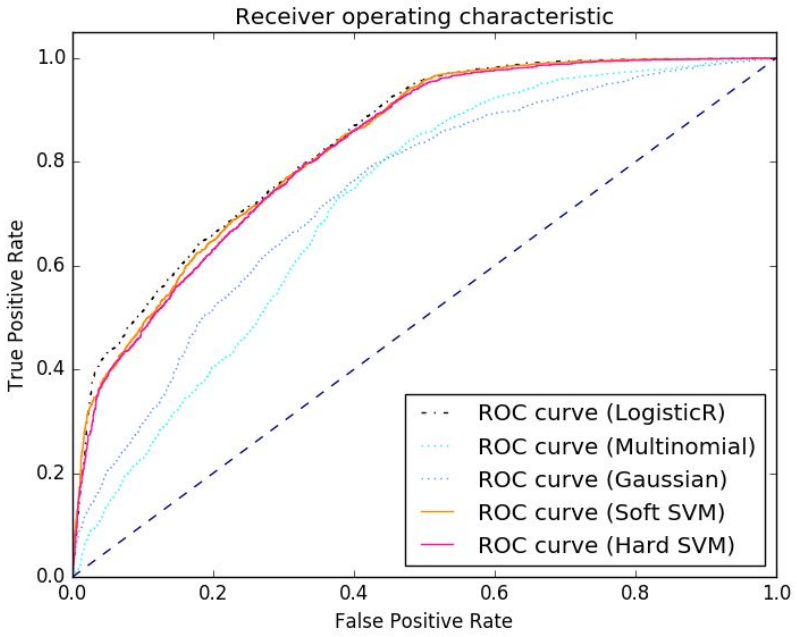


Figure 2.5: ROC curve of logistic regression model along with the models mentioned above

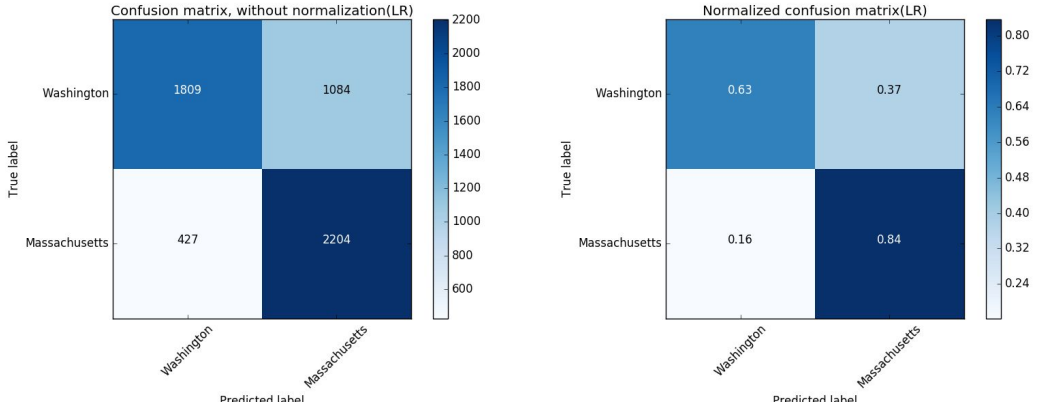


Figure 2.6: Confusion matrices for logistic regression model

Metrics	Average Accuracy	Average Precision	Average Recall
LR	0.73	0.74	0.73

Table 2.3: metrics for logistic regression model

2.1.4 Random Forest classification model with LSI

We implement random forest classification model, with 'max_features' = 5 and Bootstrapping algorithm. The ROC plots are shown in Figure 2.7. The confusion matrix is shown in Figure 2.8.

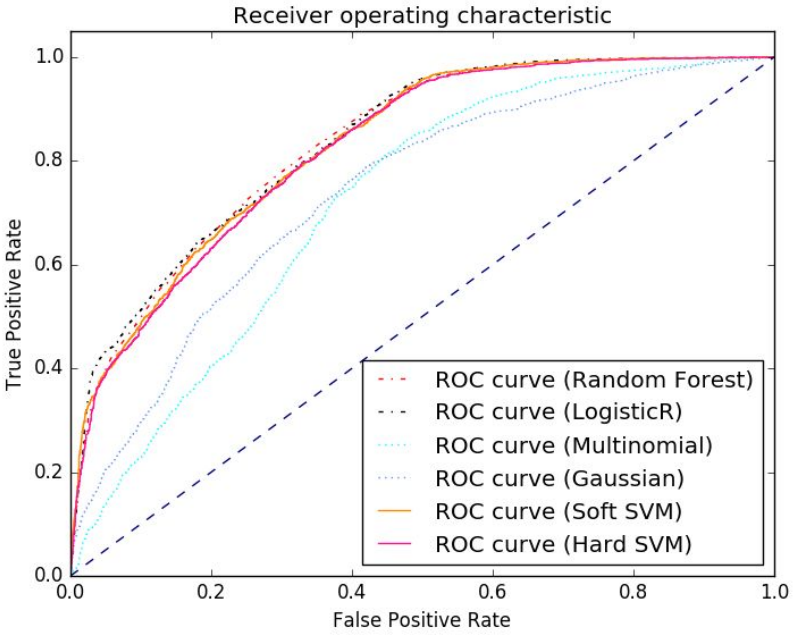


Figure 2.5: ROC curve of random forest classification model along with the models mentioned

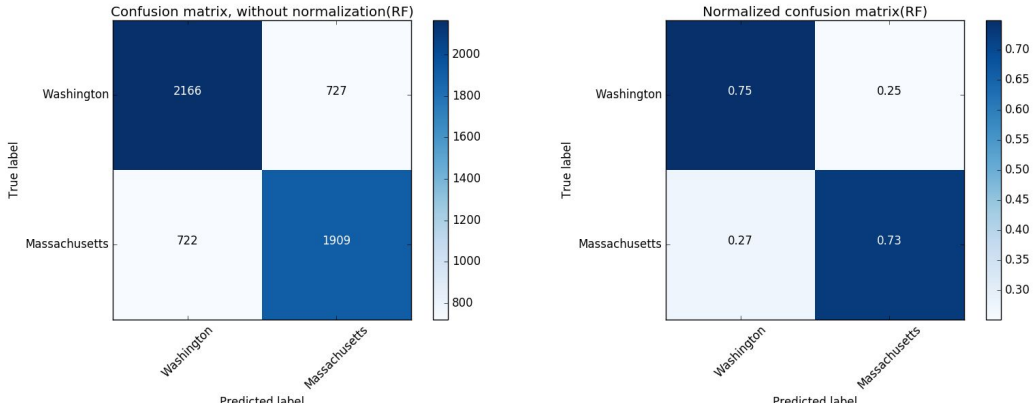


Figure 2.8: Confusion matrices for random forest classification model

Metrics	Average Accuracy	Average Precision	Average Recall
RF	0.72	0.78	0.72

Table 2.4: metrics for random forest classification model

2.2 Non-negative Matrix Factorization(NMF)

The top 50 singular vectors are extracted for projection purpose.

2.2.1 SVM model with NMF

We implement two types of SVM models, hard SVM and soft SVM, separately with C = 1000 and C = 1. The ROC plots are shown in Figure 2.9. The confusion matrices are shown in Figure 2.10.

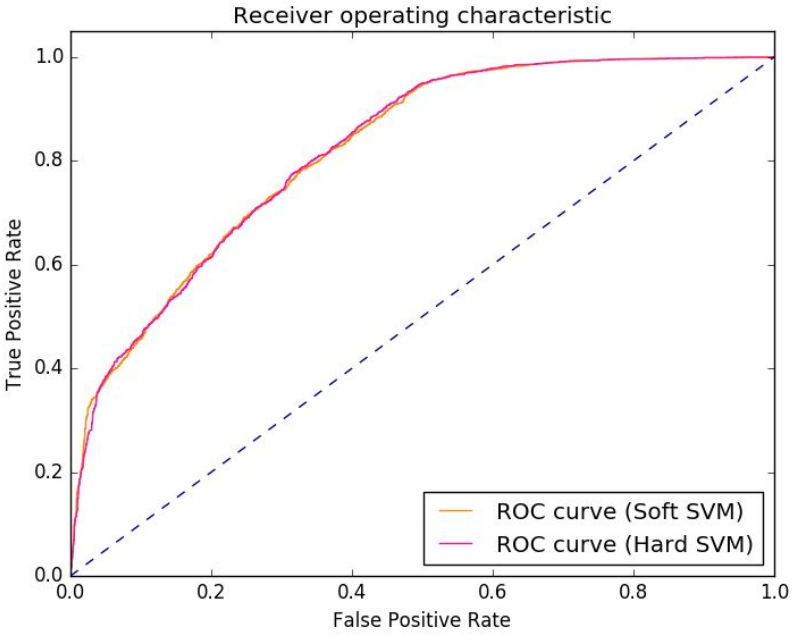


Figure 2.9: ROC curves for SVM

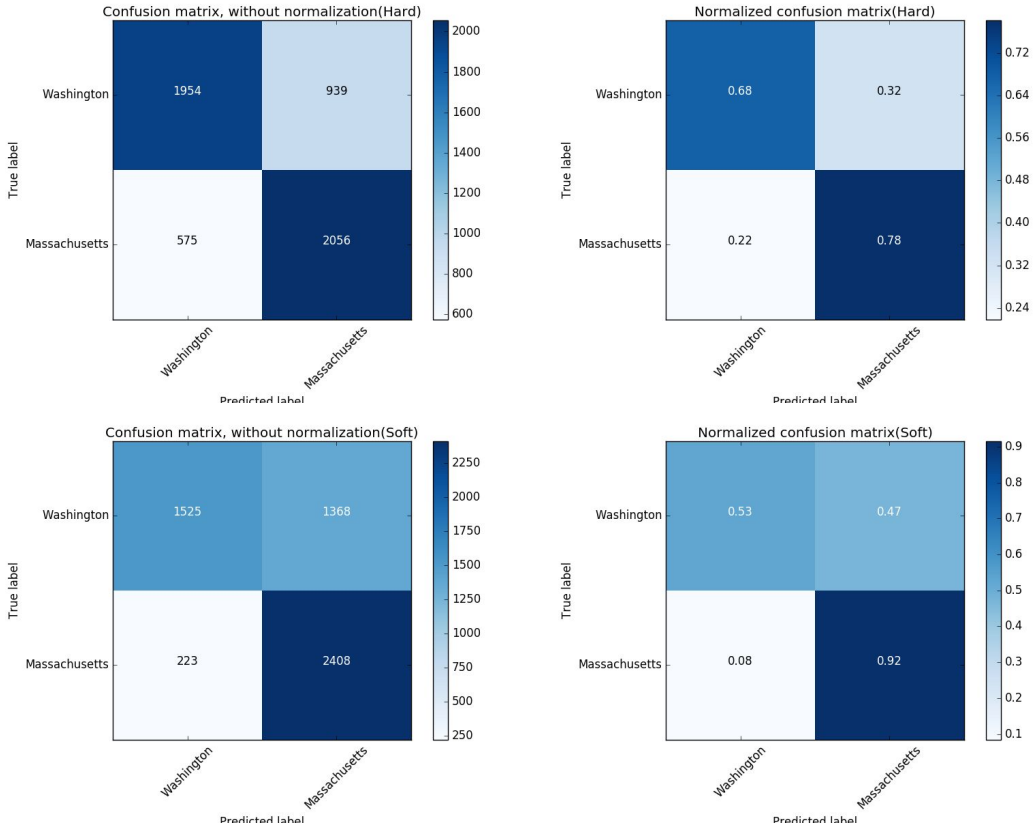


Figure 2.10: Confusion matrices for SVM

Metrics	Average Accuracy	Average Precision	Average Recall
Soft SVM	0.71	0.76	0.71
Hard SVM	0.73	0.73	0.73

Table 2.5: metrics for SVM models

2.2.2 NB model with LSI

We implement two types of NB models, Multinomial NB and Gaussian NB. The ROC plots are shown in Figure 2.11. The confusion matrices are shown in Figure 2.12.

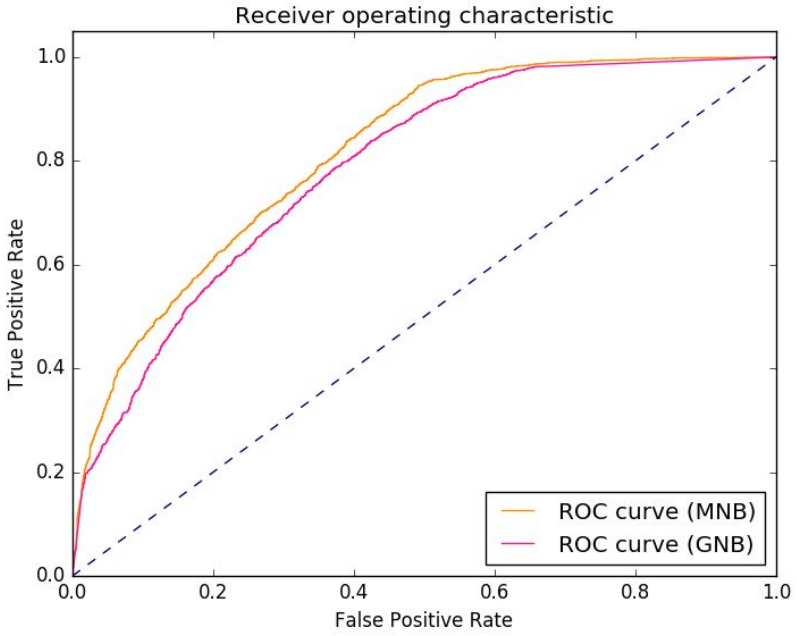


Figure 2.11: ROC curves for Naïve Bayesian models

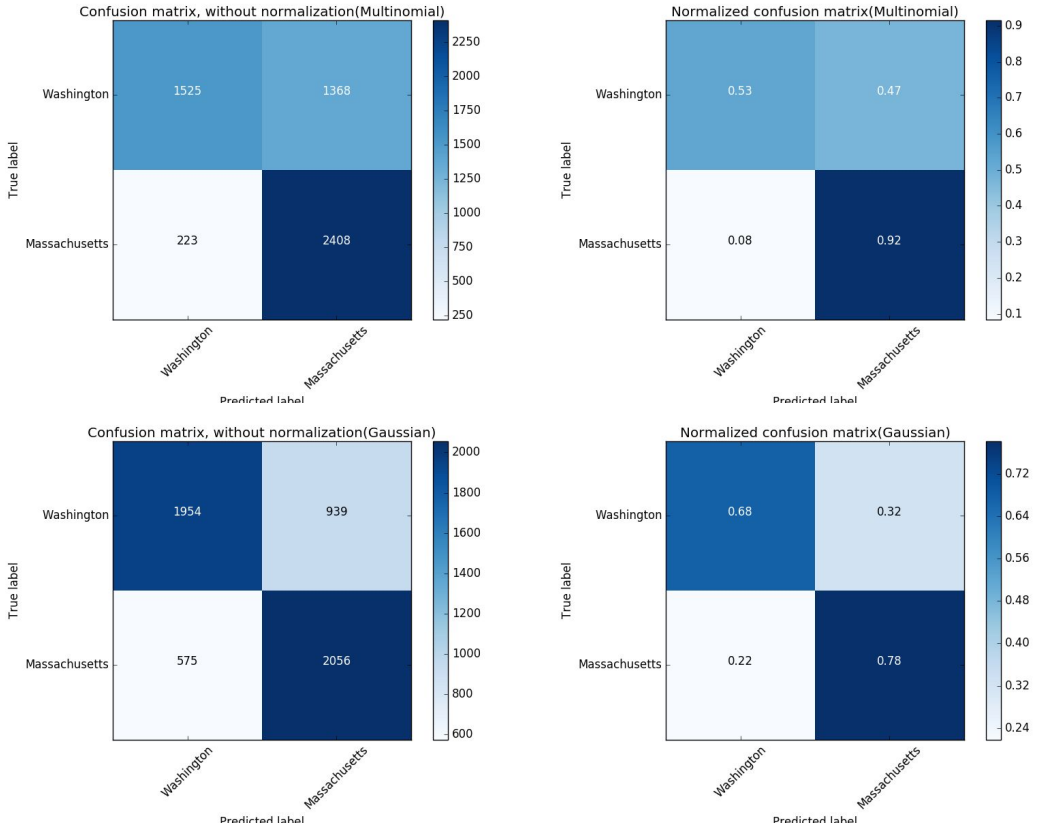


Figure 2.12: Confusion matrices for Naïve Bayesian models

Metrics	Average Accuracy	Average Precision	Average Recall

Multinomial	0.71	0.71	0.71
Gaussian	0.70	0.73	0.69

Table 2.6: metrics for naïve Bayesian model

2.2.3 Logistic Regression model with LSI

We implement logistic regression models with C = 1000. The ROC plot is shown in Figure 2.13. The confusion matrix is shown in Figure 2.14.

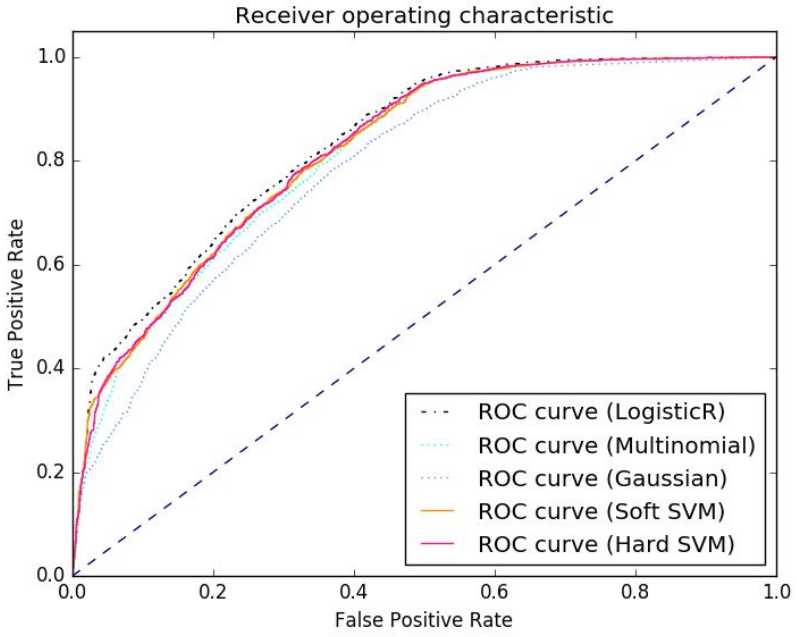


Figure 2.13: ROC curve of logistic regression model along with the models mentioned above

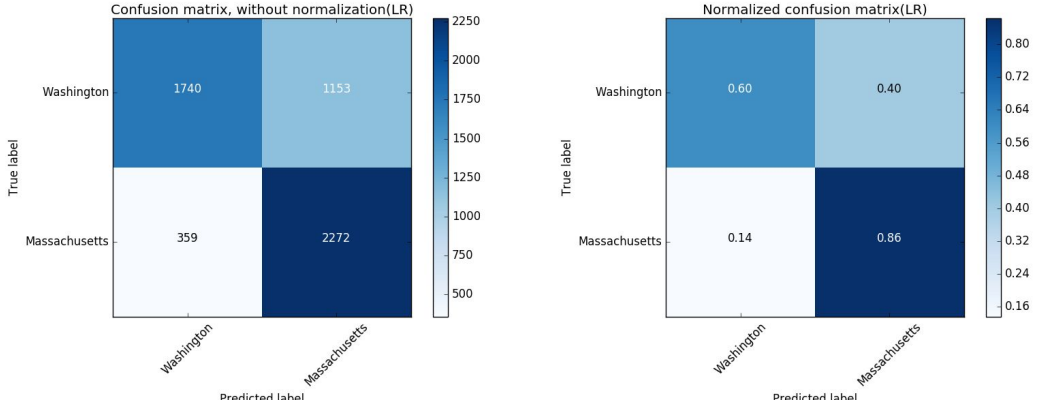


Figure 2.14: Confusion matrices for logistic regression model

Metrics	Average Accuracy	Average Precision	Average Recall
LR	0.73	0.75	0.73

Table 2.7: metrics for logistic regression model

2.2.4 Random Forest classification model with LSI

We implement random forest classification model, with 'max_features' = 5 and Bootstrapping algorithm. The ROC plots are shown in Figure 2.15. The confusion matrix is shown in Figure 2.16.

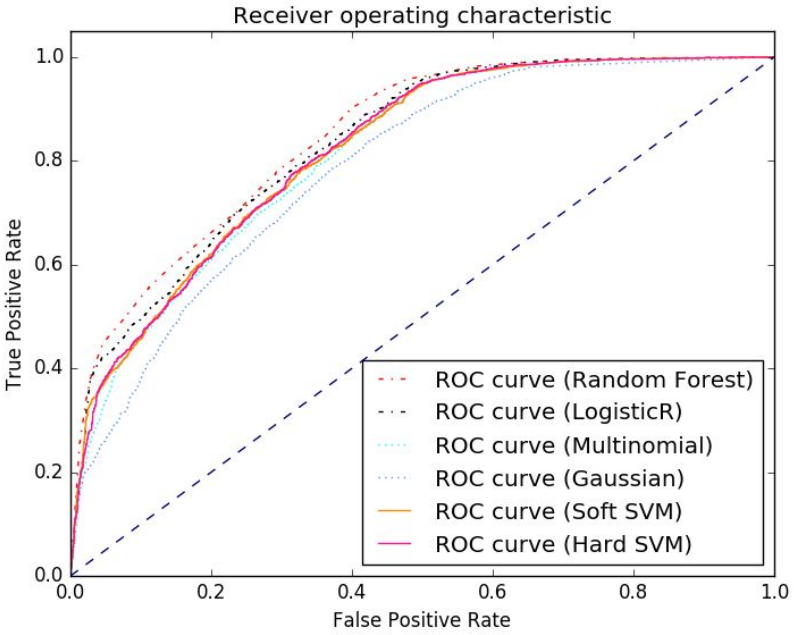


Figure 2.15: ROC curve of random forest classification model along with the models mentioned

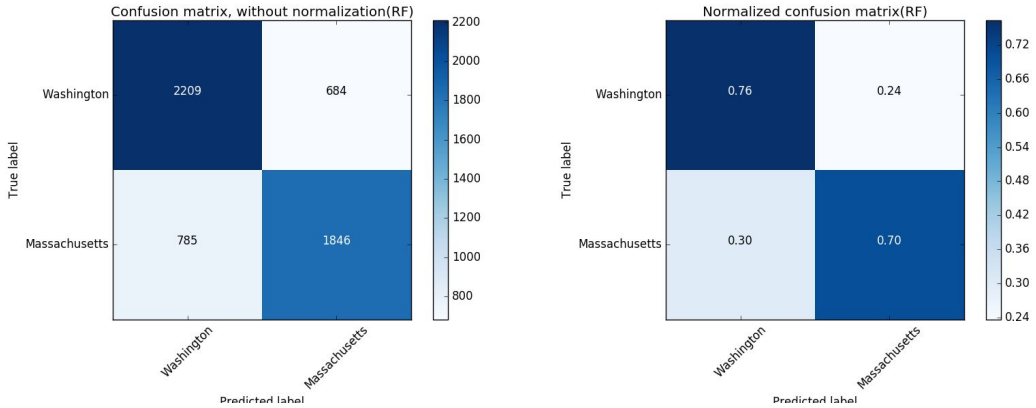


Figure 2.16: Confusion matrices for random forest classification model

Metrics	Average Accuracy	Average Precision	Average Recall
RF	0.70	0.74	0.70

Table 2.8: metrics for random forest model

2.3 Comparison

Among above 12 different models, there are 3 models standing out: hard SVM with NMF, Logistic Regression with LSI and Logistic Regression with NMF. All three achieve the highest accuracy, 0.73. But, if we look at the ROC curves, it is obvious that Random Forest model with either LSI or NMF, has the best performance and approaches the rims of the plot window most closely. This means Random Forest model is the most ideal among the 12 models. Moreover, the Random Forest model with LSI has the highest precision score, 0.77. Besides, hard SVM with NMF, Logistic Regression with LSI and Logistic Regression with NMF have the highest recall score, 0.73. In conclusion, the Naïve Bayesian models perform the worst.

3. Sentiment Analysis & Prediction

In this section, we are going to do some work on analyzing the sentiments of each team's supporters. To be specific, the tasks are including using sentiment classifier/analyzer to analyze tweets with #gopatriots and #gohawks separately, and we are going to extract the tweets with the strongest sentiment involved for each team. We then put all the trends of sentiment together in order to analyze the responses of fans of two teams and at last try to predict future sentiments by using neural network regressor and linear regression model.

3.1 Sentiment Analysis

In this section, we are going to use pure text of tweets to analyze the polarity of the sentiment expressed. We first cleaned the text by removing redundant information such as punctuation marks. Then we analyze the sentiment using the Textblob library which is based on the giant shoulder's of NLTK library and pattern library for natural language processing including sentiment analysis. The polarity of sentiment generated is a number between -1 and 1 with -1 representing the strongest negative sentiment and 1 representing the strongest positive sentiment. Notice that a 0 means the tweet has a neutral sentiment. All data of sentiment is then grouped into a time period of 10 minutes in order to analyze the audiences' sentiment on the game day from 7am to 11pm. Having all the data grouped, we calculate the average of polarity and ratio of positive and negative sentiments separately. The results of tweets with #gohawks are shown below. Notice that the product of x axis reading with **10min/unit** equals to the total time past since 7am on February 1st, 2015. The game started at 6:30pm and last for 3 hours and 34 minutes which means the interval between x equals to **69** and **91** on all the graphs.

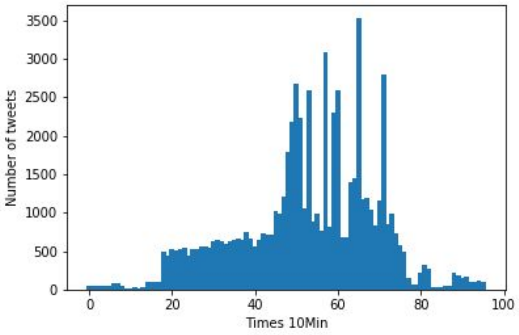


Figure 3.1: Number of tweets with #gohawks

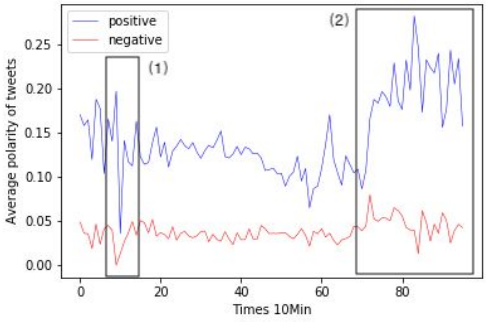


Figure 3.2: Average polarity of tweets

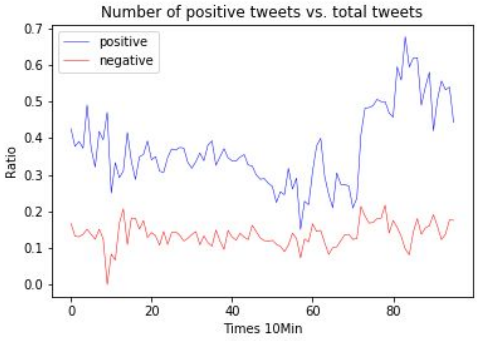


Figure 3.3: Polarity ratio of tweets

In Figure 3.2 one can discover that supporters of Seattle Seahawks started their day with more negative sentiment (portion (1)) and increasing positive sentiment at the start of game and strong fluctuations of positive sentiment during the game(portion (2)). The graph of sentiment ratio (Figure 2.2) can also reflect the truth. On the other hand, the supporters of New England Patriots were having different sentiment patterns as shown below.

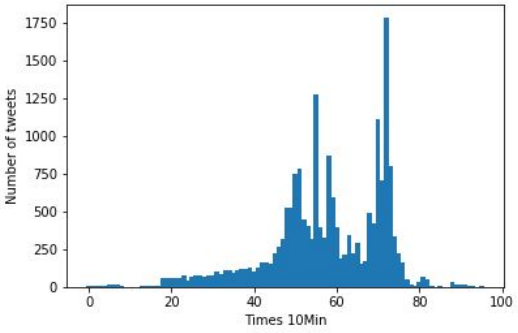


Figure 3.4: Number of tweets with #gopatriots

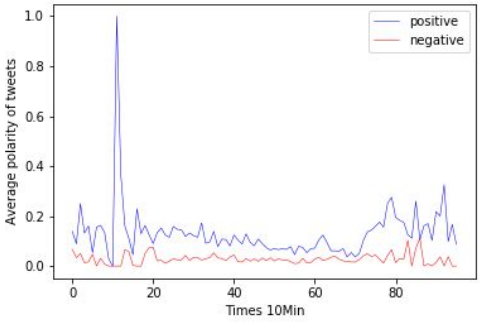


Figure 3.5: Average polarity of tweets

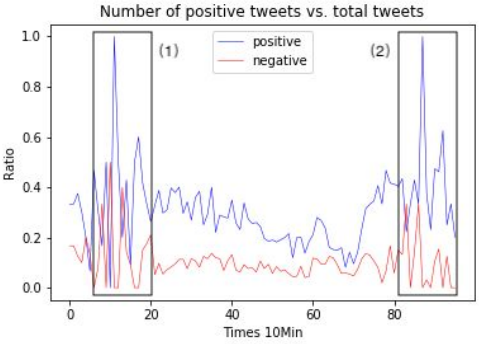


Figure 3.6: Polarity ratio of tweets

As one can discover, the data in portion (1) shows that the supporters of patriots started their day with a spike of strong positivity and similar spikes during the game.

3.2 Sentiment Comparison

In this section, we are going to put all the graphs generated together to analyze whether the sentiment classification makes sense.

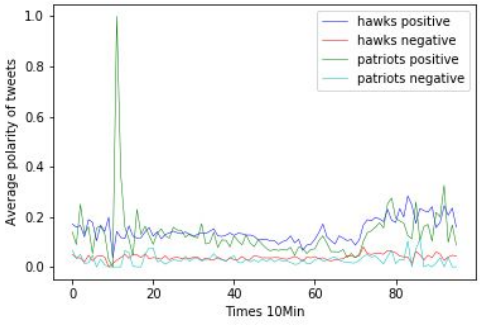


Figure 3.7: Average polarity of tweets

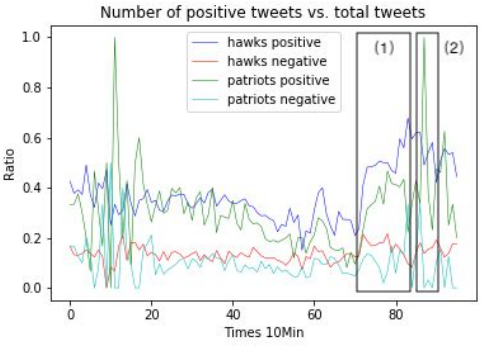


Figure 3.8: Polarity ratio of tweets

In section (1), the ratios of positive tweets and negative tweets of New England Seahawks' supports are higher than those of Seattle Patriots' supporters which means the sentiment of Seattle Patriots' supporters is more neutral. In section (2), there is a spike of ratio of positive sentiments from supporters of Seattle Patriots with a spike of ratio of negative sentiments from supporters of New England Seahawks. This may indicates the moment that Tom Brady pass complete short left to Julian Edelman for 3 yards and a touchdown that makes Seattle Patriots surpass New England Seahawks near the end of the game.

We have also extracted all the tweets with the strongest positive and negative sentiments from the supporters of the two teams. The table below provides examples in the pool of tweets.

Positive	Excellent throw! Awesome touchdown! #GoHawks #BeastMode
	Yes!!!!!!!! Touchdown Seahawks!!!!!!!!!! Excellent!!!!!! #GoHawks
	Chris Matthews!!!!! Your the best!!!! #cfl #SuperBowl #SuperBowl49 #GoHawks
	@Seahawks do best under pressure! That's how it's done! #SB49 #GoHawks #repete
	I can't stop saying awesome. Awesome awesome awesome. #GoHawks
Negative	Another terrible time. #gohawks
	That was #pathetic #LOB!!! #GoHawks #SB49
	Worst play call of the year followed by the worst execution. #GoHawks self destruct
	Omg this is horrible #GoHawks
	Horrible play. Whyyyyy didn't they give the ball to Lynch. #GoHawks

Table 3.1: Tweets with strongest polarity from supporters of New England Seahawks

Positive	The greatest day of the year😍🏈#SB49 #GoPatriots
	Tom Brady. The Greatest of All Time #goPatriots
	@stephmamiii best moment n history of football #GoPatriots
	@StephMcMahon @TripleH Tom Brady and the Patriots ARE best for business and the Authority of the NFL!! #PatriotsWIN #GoPatriots #WWERAW
	Impressive. Almost TOO impressive. 😑😂😂😂 we still won!!!! 28-24!!! courtenay_nelson #Gopatriots… http://t.co/5RWkrt2U4s
Negative	Terrible spot!!! #gopatriots #superbowlXLIX
	This is terrifying #GoPatriots
	Omg! Worst minute ever! #GoPatriots
	What a play, all over! Terrible call tho just give it to lynch! #GoPatriots #SuperBowlXLIX
	Worst.Playcall.ever. #SB49 #nflse #nfl #GoPatriots

Table 3.2: Tweets with strongest polarity from supporters of Seattle Patriots

3.3 Sentiment Prediction

Predicting audiences' sentiment during a game can be very meaningful for advertisers to analyze whether the time that their advertisement played is beneficial or not. If the audience are having positive sentiments, for example, beverage advertisements that people drink when celebrating can be more effective.

In this part we first group all the tweets by hours and then try to use features in current hour to predict the sentiment ratio in the next hour using the MLPRegressor and the Linear Regression Model. However, in the case of MLPRegressor the result is not ideal enough. One needs to comment out the line running MLPRegressor in the source code to show the result. Below is a list of features that we used. The total number of hashtags in tweets sent during a period can potential be a strong evidence of people's sentiment since people may add many hashtags or repetitive hashtags when they feel something is important.

x1: Count of tweets
x2: Count of retweets
x3: Count of followers
x4: Maximum followers
x5: Total number of hashtags
x6: Time of the day
x7: Impression count
x8: Favorite count
x9: Ranking Score
x10: Ratio of tweet with positive sentiment during current time period
x11: Ratio of tweet with positive sentiment from last time period

Table 3.2: List of Features Used

Instead of presenting the result, we came out with several potential reasons why this attempt failed:

1. The features contain not enough information to predict sentiment ratio of the next time period
2. The sample size is too small and cannot be extended due to the fact that for most of the time there are not enough tweets posted related a specific hashtag
3. The sentiment of people related to a sport game is arbitrary to some extend.

However, in the case of Linear Regression Model, the result is better. The results are presented below.

#gohawks	
RMSE	0.052182
R-squared	0.981

Table 3.2: Result of dataset #gohawks

OLS Regression Results

Dep. Variable:	y	R-squared:	0.981
Model:	OLS	Adj. R-squared:	0.978
Method:	Least Squares	F-statistic:	383.1
Date:	Tue, 13 Mar 2018	Prob (F-statistic):	1.38e-65
Time:	22:34:28	Log-Likelihood:	139.87
No. Observations:	93	AIC:	-257.7
Df Residuals:	82	BIC:	-229.9
Df Model:	11		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
x1	0.0013	0.001	1.780	0.079	-0.000	0.003
x2	1.083e-05	1.73e-05	0.626	0.533	-2.36e-05	4.52e-05
x3	-7.287e-08	1.07e-07	-0.683	0.497	-2.85e-07	1.4e-07
x4	-1.541e-08	3.06e-08	-0.503	0.616	-7.63e-08	4.55e-08
x5	-0.0001	4.18e-05	-3.152	0.002	-0.000	-4.87e-05
x6	0.0003	0.002	0.131	0.896	-0.004	0.004
x7	7.504e-08	1.07e-07	0.704	0.483	-1.37e-07	2.87e-07
x8	-2.345e-06	4.76e-06	-0.492	0.624	-1.18e-05	7.13e-06
x9	-0.0002	0.000	-1.519	0.133	-0.000	6.61e-05
x10	0.6224	0.111	5.627	0.000	0.402	0.842
x11	0.3372	0.104	3.227	0.002	0.129	0.545

Omnibus:	10.475	Durbin-Watson:	2.141
Prob(Omnibus):	0.005	Jarque-Bera (JB):	15.082
Skew:	-0.493	Prob(JB):	0.000531
Kurtosis:	4.708	Cond. No.	7.98e+07

Figure 3.9: Result of dataset #gohawks

#gopatriots	
RMSE	0.139894
R-squared	0.797

Table 3.2: Result of dataset #gohawks

OLS Regression Results

Dep. Variable:	y	R-squared:	0.797
Model:	OLS	Adj. R-squared:	0.769
Method:	Least Squares	F-statistic:	29.19
Date:	Tue, 13 Mar 2018	Prob (F-statistic):	7.56e-24
Time:	22:34:50	Log-Likelihood:	41.685
No. Observations:	93	AIC:	-61.37
Df Residuals:	82	BIC:	-33.51
Df Model:	11		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
x1	0.0040	0.005	0.834	0.407	-0.006	0.013
x2	6.222e-05	0.000	0.186	0.853	-0.001	0.001
x3	6.865e-08	8.14e-07	0.084	0.933	-1.55e-06	1.69e-06
x4	-1.185e-08	2.89e-07	-0.041	0.967	-5.87e-07	5.64e-07
x5	-0.0004	0.001	-0.790	0.432	-0.002	0.001
x6	0.0114	0.004	3.178	0.002	0.004	0.019
x7	-4.714e-09	7.27e-07	-0.006	0.995	-1.45e-06	1.44e-06
x8	0.0006	0.002	0.273	0.786	-0.004	0.005
x9	-0.0007	0.001	-0.810	0.420	-0.002	0.001
x10	0.1899	0.110	1.730	0.087	-0.028	0.408
x11	0.2891	0.111	2.604	0.011	0.068	0.510

Omnibus:	50.393	Durbin-Watson:	2.154
Prob(Omnibus):	0.000	Jarque-Bera (JB):	218.733
Skew:	1.698	Prob(JB):	3.18e-48
Kurtosis:	9.702	Cond. No.	4.76e+06

Figure 3.10: Result of dataset #gohawks

As we can see, the prediction has a fairly good R-square value and acceptable RMSE, the performance that may contributed by the large portion of tweets with neutral sentiment. By checking the P value we can determine how significant a feature is. In the case of tweets with #gohawks, count of tweets, total number of hashtags, ratio of tweets with positive sentiment during current period and last period are most significant. In the case of tweets with #gopatriots, ratio of tweets with positive sentiment during current period and last period are most significant.