EE219 Project 5

2017 Winter

Popularity Prediction on Twitter Hashtags

Team Member:

Kaiyi Wu 004761345 kaiyiwu@ucla.edu

Xueyin Yu 504741703 soniayu@g.ucla.edu

Ruoxi Zhang 404753334 zhangruoxi@ucla.edu

1. Statistics for each hashtag

1.1 Statistic Results

In the first part, we download the training tweet data and calculate these statistics for each hashtag: average number of tweets per hour, average number of followers of users posting the tweets, and average number of retweets. For number of followers, we use ['author']['followers'], for number of retweets, we use ['tweet']['retweet_count']. The results of each hashtag is shown is table 1-1.

Hashtag	Avg. tweets/hour	Avg. followers	Avg. retweets
#gohawks	193.5556	2203.9318	0.2092
#gopatriots	38.4070	1401.8955	0.0268
#nfl	279.7235	4653.2523	0.0509
#patriots	499.7071	3309.9788	0.0915
#sb49	1420.8780	10267.3168	0.1780
#superbowl	1402.0447	8858.9747	0.1367

Table 1-1 Statistics for each hashtag

From table 1-1 of statistics, we can see the differences among hashtags. The differences mean the popularity of each hashtag is different.

1.2 Plot “Number of tweets in hour”

Then, we plot “number of tweets in hour” over time for #SuperBowl and #NFL, using histogram with 1-hour bins (figure 1-1, figure 1-2).

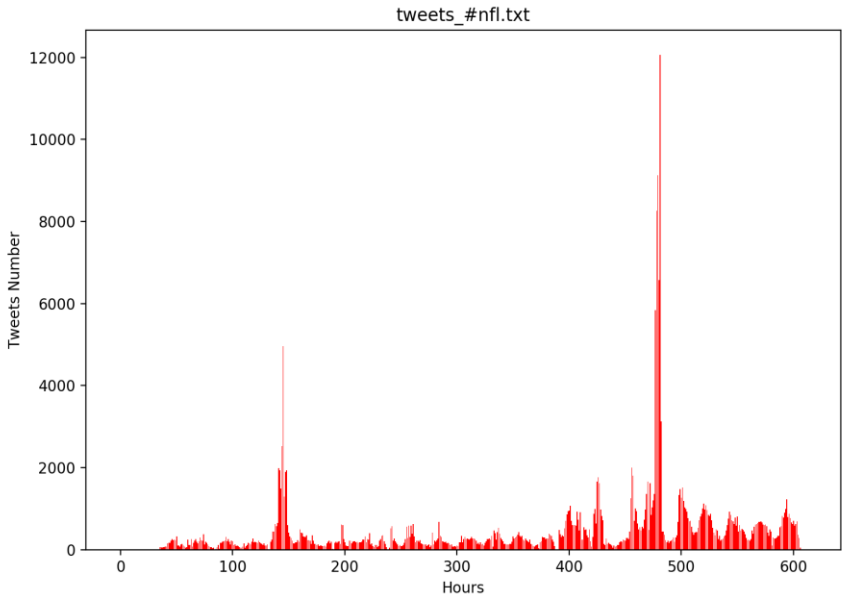


Figure 1-1 “number of tweets in hour” over time for #SuperBowl

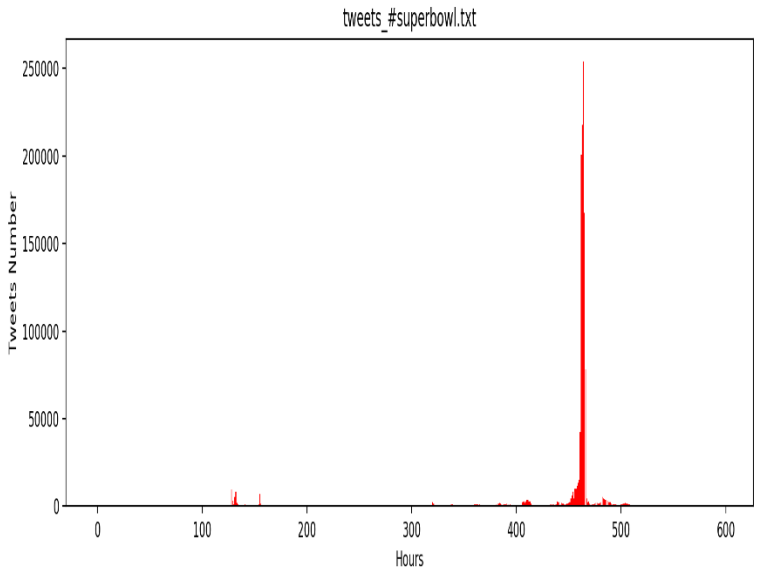


Figure 1-2 "number of tweets in hour" over time for #NFL

Figure 1-1 and figure 1-2 show that #SuperBowl and #NFL have some common burst in some certain time.

2. Linear Regression Model

In this part, we fit a basic linear regression model to predict number of tweets in the next hour, with 5 features extracted. The 5 features are: number of tweets, total number of retweets, sum of the number of followers of the users posting the hashtag, maximum number of the followers of the users posting the hashtag, and time of the day (24 values that represent hours with respect to a given time reference). The 5 features correspond to x1, x2, x3, x4, and x5 in the results.

We use statsmodels in python to finish this task and report the result using summary () method. By seeing "R-squared", we can get the accuracy of each hashtag's prediction. By seeing t-test and P-value, which is "P>|t|" in report, we can find the significance of each feature. If the value "P>|t|" of a certain feature is less than 0.5, we can consider this feature as important. Then by seeing "t" value (t-statistics used in testing whether a given coefficient is significantly different from zero), we can order the importance of these features.

The result of training accuracy is shown in table 2-1.

#gohawks	#gopatriots	#NFL	#patriots	#sb49	#SuperBowl
0.662	0.677	0.721	0.705	0.819	0.847

Table 2-1 Training accuracy for each hashtag

For the significance of features, we have attached the text file including all results, "q2_results.txt". By seeing "t" and "P>|t|", we conclude that:

1) For #gohawks, there are three significant features, ***number of tweets*** (x1), ***total number of retweets*** (x2), and ***time of the day*** (x5), whose "P>|t|" values are all less than 0.5. Among them, ***number of tweets*** is the most significant and ***time of the day*** is the least significant.
2) For #gopatriots, the first four features are all significant, ***number of tweets***, ***total number of retweets***, ***sum of the number of followers***, and ***maximum number of followers***. Among them, ***total number of retweets*** is the most important and ***sum of the number of followers*** the second.
3) For #NFL, ***number of tweets***, ***sum of the number of followers***, ***maximum number of followers***,

and ***time of the day*** are four significant features. Among them, ***sum of the number of followers*** is the most significant, and then ***maximum number of followers***, then ***time of the day***.

4) For #patriots, the first three features are significant, ***number of tweets***, ***total number of retweets***, ***sum of the number of followers***, and the significance decreases among them.
5) For #sb49, the first four features are important, ***number of tweets***, ***total number of retweets***, ***sum of the number of followers***, and ***maximum number of the followers***. The significance decreases among them.
6) For #SuperBowl, same as #sb49, the first four features are important, but ***sum of the number of followers*** is the most significant feature, and then ***maximum number of the followers***, then ***number of tweets*** and ***total number of retweets***.

3. Feature generation and regression model

3.1 Choose 11 features

In this part, we choose other features that may bring significance in prediction. Same as before, we also use *statsmodels* to fit linear regression model and use t-test and P-value to decide the significance of the feature and prediction accuracy. The new features we use are as followed:

x6: average ranking score (per tweet). Ranking score reflect the presence of query keywords and of a certain tweet. We count the total ranking score of tweets in each 1-hour window and do the average ranking score. This feature is extracted from [‘metrics’] [‘ranking_score’].

X7: number of impressions. Impression contains the number of times a certain tweet has been seen. It reflects the actual influence of each tweet to others. We count the total number of impressions in each 1-hour window. This feature is extracted from [‘metrics’] [‘impressions’].

X8: number of momentums. We count the total number of momentums of each 1-hour window. This feature is extracted from [‘metrics’] [‘momentum’].

X9: number of tweets favor. This feature shows how many people like a certain tweet most. We count the total number of it. It is extracted from [‘tweet’] [‘favorite_count’].

X10: number of accelerations. We count the total number of accelerations and this feature is extracted from [‘metrics’] [‘acceleration’].

X11: number of reply: We count the total number of replies of a certain tweet. This feature is extracted from [‘metrics’] [‘citations’] [‘replies’].

Adding the five features mentioned before, we have 11 features now. We then fit the linear regression model of each hashtag. By seeing the summary report of each hashtag. We can gain accuracy and find the top 3 significant features.

3.2 Fit Regression Model and Report Results

Using the 11 features, we fit a linear regression model for each hashtag. The accuracy of each hashtag is shown in table 3-1.

#gohawks	#gopatriots	#NFL	#patriots	#sb49	#SuperBowl
0.709	0.724	0.809	0.682	0.832	0.893

Table 3-1 Training accuracy for each hashtag

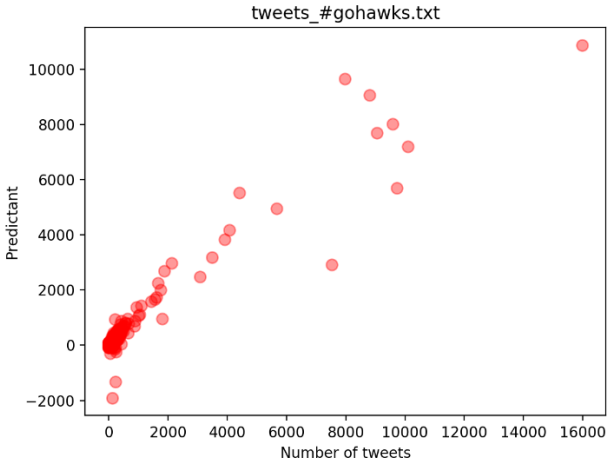
By comparing accuracy here and accuracy using 5 features, we find the accuracy for the first three files and #sb49 has been improved. The accuracy for #patriots and #SuperBowl decrease a little bit. It might because for #sb49, there are other more important features which are not in the 11 features we choose.

For the significance of features, we have attached the text file including all results, "q3_results.txt". By seeing "t" and "P>|t|", we extract the top3 features.

3.3 Draw scatter plot of predictant (number of tweets for next hour)

Since we have found the top 3 features in each hashtag. We draw three scatter plots for each hashtag (The order of plot figures reflects the significance of the 3 features).

For hashtag #gohawks: top 3 features are number of tweet, number of momentums and number of replies.



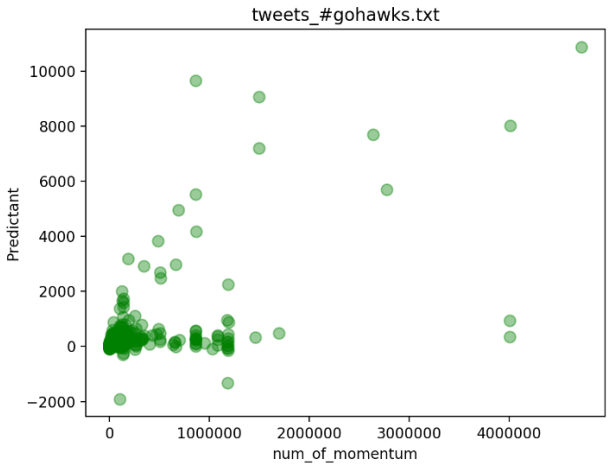


Figure 3-1 (Left) Predictand vs number of tweets for #gohawks

Figure 3-2 (Right) Predictand vs number of momentums for #gohawks

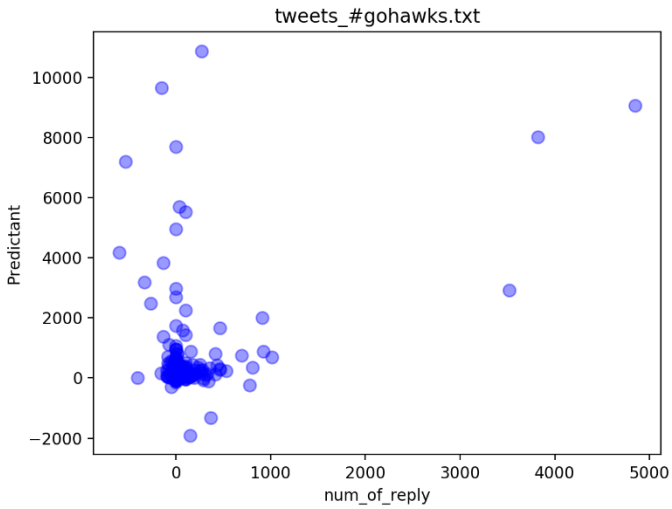
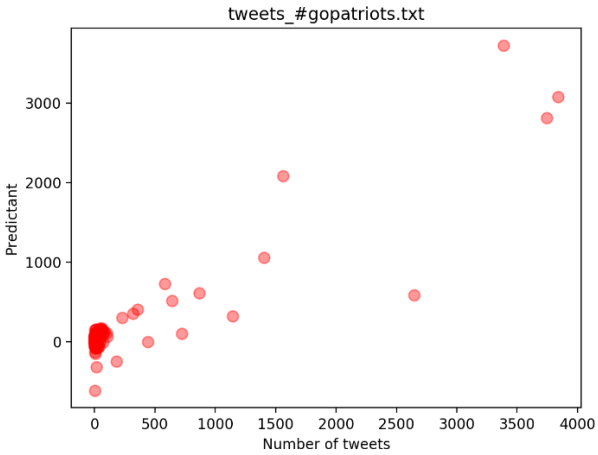


Figure 3-3 Predictand vs number of replies for #gohawks

For hashtag #gopatriots: top 3 features are number of tweets, maximum number of followers and number of accelerations.



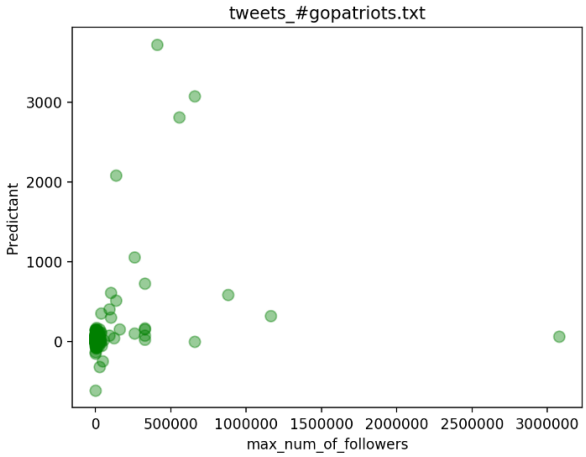


Figure 3-4 (Left) Predictand vs number of tweets for #gopatriots

Figure 3-5 (Right) Predictant vs maximum number of followers for #gopatriots

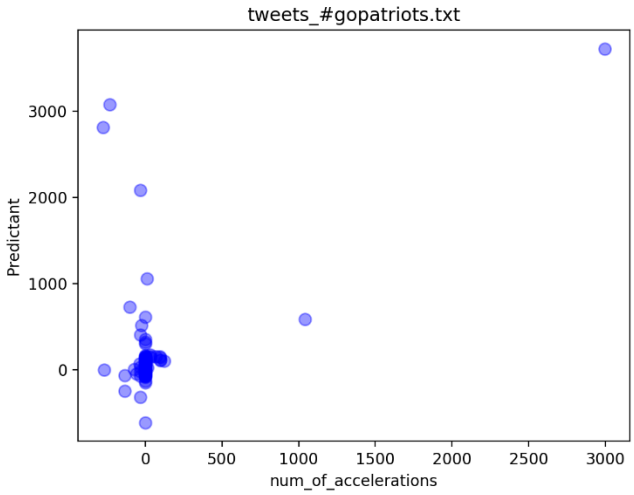
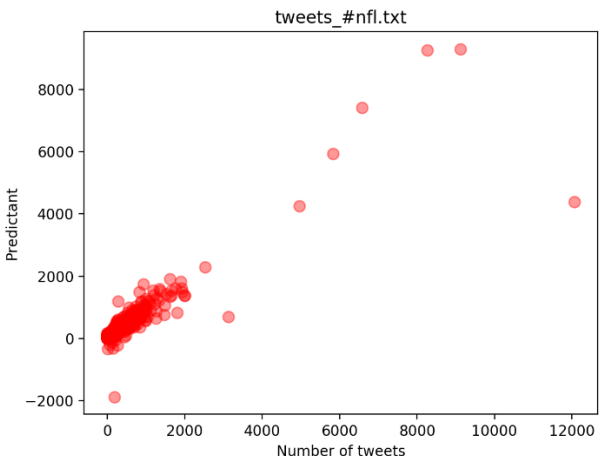


Figure 3-6 Predictand vs number of accelerations for #gopatriots

For hashtag #nfl: top 3 features are number of tweets, number of accelerations and number of followers.



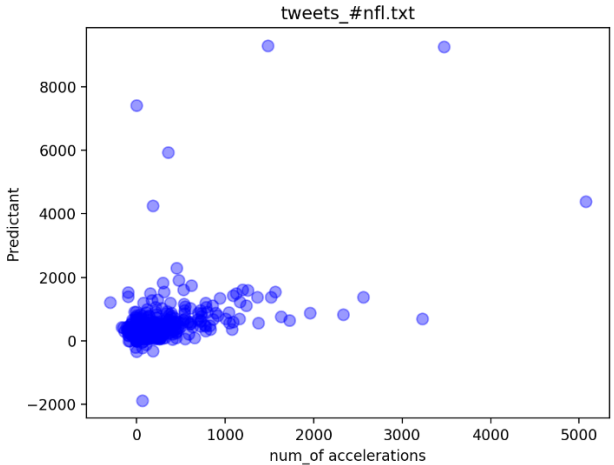


Figure 3-7 (Left) Predictand vs number of tweets for #nfl

Figure 3-8 (Right) Predictand vs number of accelerations for #nfl

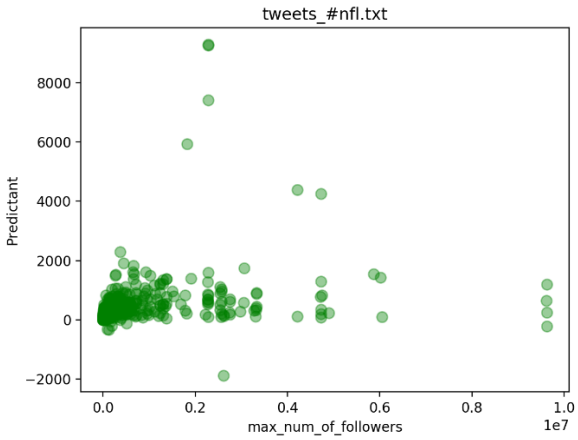
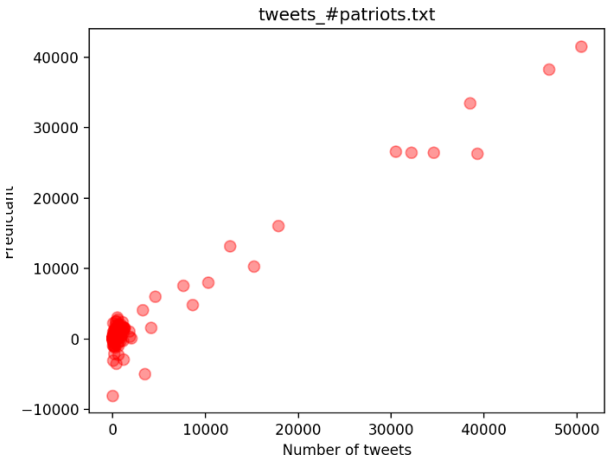


Figure 3-9 Predictand vs maximum number of followers for #nfl

For hashtag #patriots: top 3 features are number of tweets, number of momentums and number of followers.



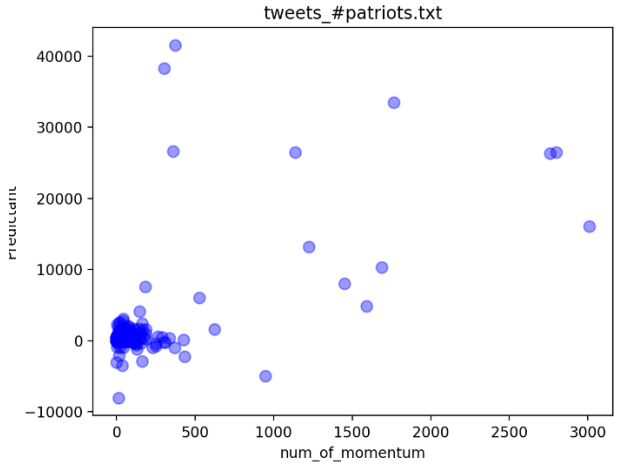


Figure 3-10 (Left) Predictand vs number of tweets for #patriots

Figure 3-11 (Right) Predictand vs number of momentums for #patriots

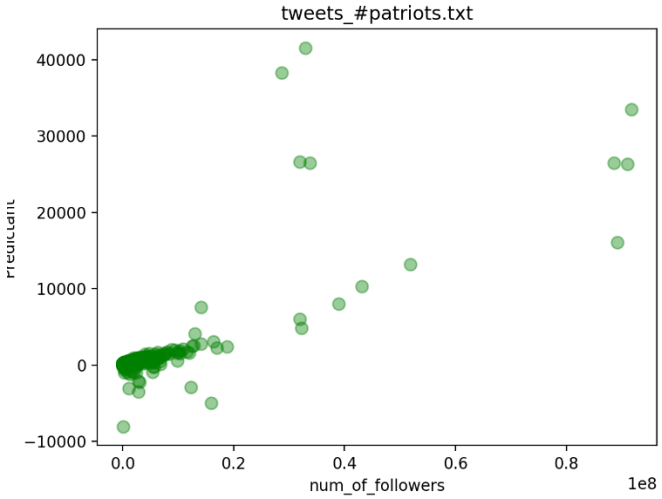
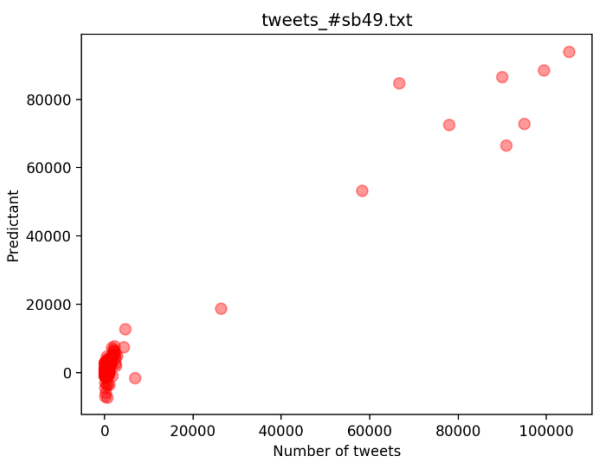


Figure 3-12 Predictand vs number of followers for #patriots

For hashtag #sb49: Top 3 features are number of tweets, number of momentum and maximum number of followers



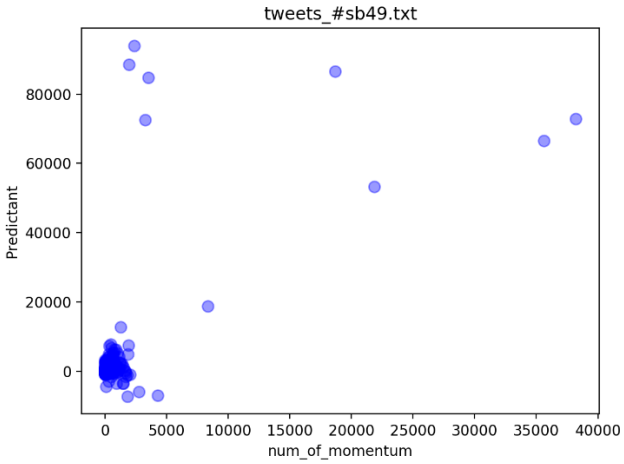


Figure 3-13 (Left) Predictand vs number of tweets for #sb49

Figure 3-14 (Right) Predictand vs number of momentum for #sb49

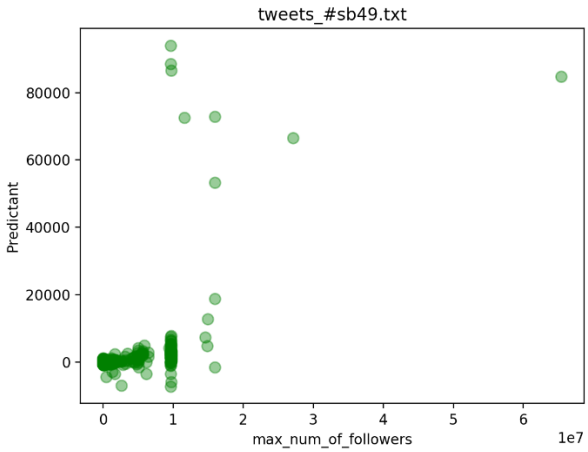
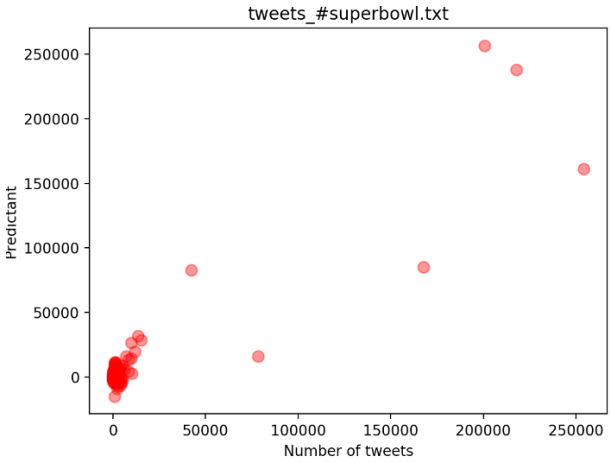


Figure 3-15 Predictand vs maximum number of followers for #sb49

For hashtag #SuperBowl: top 3 features are number of tweets, number of momentums and number of retweets.



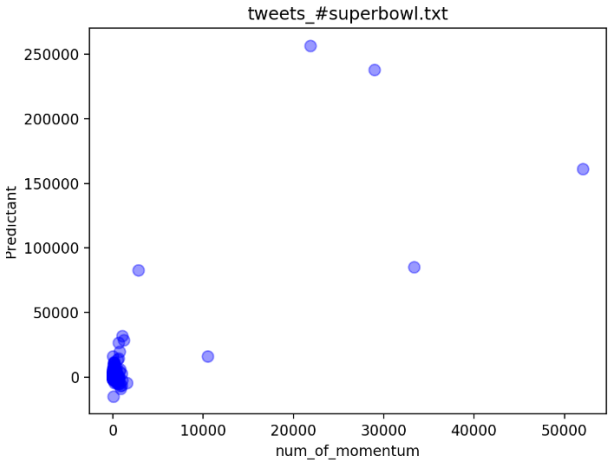


Figure 3-16 (Left) Predictand vs number of tweets for #SuperBowl

Figure 3-17 (Right) Predictand vs number of momentums for #SuperBowl

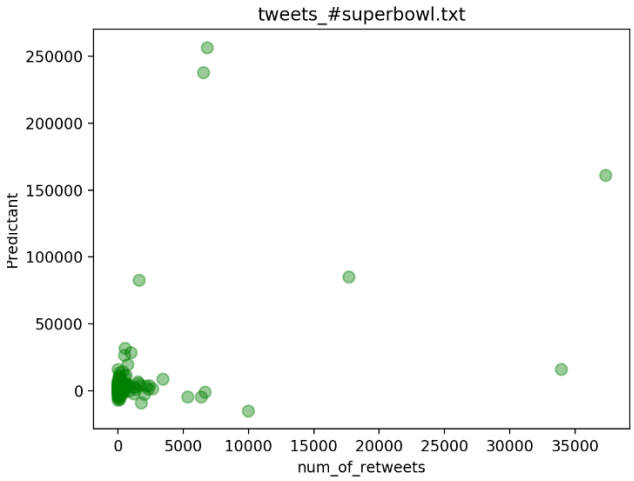


Figure 3-18 Predictand vs number of retweets for #SuperBowl

4. Prediction Models for Different Time Periods

From the above parts, we find all hashtags have a similar pattern that there is a peak at the day Feb 1, 2015. In other words, it is unreasonable to predict the data using same model in the whole time period, which may cause the error becoming large. In this part, we are forced to split the data into three parts, namely, before the game day, during the game day and after the game day. For every part, we use some different kinds of models to train the model and use 10-fold cross validation to verify it. Another thing we should mention is feature selection. For each model, we choose top 10 important features as our inputs. We first use linear model to predict and then choose some different kinds of models. Finally, we find Gboost is a great model for this problem. In the report, for convenience, we just report the linear model and the gradient boost model.

4.1 Before Feb. 1, 8:00 a.m.

In the time period, we use some models to predict data of each hashtag. We can get all results below. First, we use linear model to predict. The results of all hashtags are shown below.

Cross-Validation Errors	gopatriots	gohawks	patriots	nfl	superbowl	sb49

1	72.7	1202.2	209.7	329.4	936.2	55.1
2	8.0	173.5	265.3	93.1	415.6	10.9
3	7.1	97.4	339.9	116.2	112.7	10.8
4	5.1	87.5	350.3	121.5	166.8	12.5
5	7.9	98.2	473.5	103.2	149.2	17.0
6	10	133.6	157.6	49.1	146.2	29.6
7	11.3	209.2	268.3	81.5	258.2	38.6
8	7.3	134.3	265.1	85.9	245.3	56.6
9	8.1	138.2	236.8	351.1	581.9	274.0
10	11.9	141.6	179.6	252.0	530	198.5
Mean Error	15.0	241.6	274.4	158.4	354.2	70.3

Table 4-1 linear model of six hashtags in the first time period

Then we use gradient boost model to predict this time period. The results are shown in the table below:

Cross-Validation Errors	gopatriots	gohawks	patriots	nfl	superbowl	sb49
1	107.3	523.5	284.9	526.0	1496.4	37.2
2	5.1	99.8	337.4	98.2	792.4	11.4
3	4.0	74.9	267.1	121.9	124.6	11.1
4	4.4	62.1	206.3	114.4	102.1	13.4
5	7.5	57.3	237.5	106.6	97.6	15.5
6	4.4	87.4	101.7	66.7	105.6	21.4
7	7.1	123.4	108.3	95.1	257.4	64.7
8	5.7	80.8	112.0	198.2	319.2	59.2
9	4.9	62.9	120.4	262.3	1932.8	417.9
10	4.8	45.3	74.9	276.1	642.9	286.5
Mean Error	14.4	121.8	114.4	186.8	586.9	93.1

Table 4-2 Gradient model of six hashtags in the first time period

4.2 Between Feb. 1, 8:00 a.m. and 8:00 p.m.

In this time period, we also use some models to predict data of each hashtag. We can get all results below. As we are using 3 hours to predict next hour, one problem is that we don't have enough samples to make 10-fold cross validation. In order to get more than 10 samples, we change our data sets to tweets per 20 minutes. So we can use 1:20-4:20 period to predict, for example, 4:20-5:20. In this way, we have more samples and can predict the results better. The results of two different models are shown below:

Cross-Validation Errors	gopatriots	gohawks	patriots	nfl	superbowl	sb49
1	107.3	523.5	284.9	526.0	1496.4	37.2
2	5.1	99.8	337.4	98.2	792.4	11.4
3	4.0	74.9	267.1	121.9	124.6	11.1
4	4.4	62.1	206.3	114.4	102.1	13.4
5	7.5	57.3	237.5	106.6	97.6	15.5
6	4.4	87.4	101.7	66.7	105.6	21.4
7	7.1	123.4	108.3	95.1	257.4	64.7
8	5.7	80.8	112.0	198.2	319.2	59.2
9	4.9	62.9	120.4	262.3	1932.8	417.9
10	4.8	45.3	74.9	276.1	642.9	286.5
Mean Error	14.4	121.8	114.4	186.8	586.9	93.1

Table 4-3 linear model of six hashtags in the second time period

Cross-Validation Errors	gopatriots	gohawks	patriots	nfl	superbowl	sb49
1	247.9	410.6	12764.1	236.1	65114.5	20268.4
2	49.2	240.0	3335.1	1029.6	1834.7	7023.5
3	265.2	1268.3	5387.0	164.8	4854.8	9382.5
4	919.3	2085.2	1513.5	2279.5	32168.6	19550.9
5	405.7	2008.6	1816.3	1148.5	27540.1	5609.9
6	1556.7	1135.6	7616.1	381.3	24898.6	1585.6
7	1596.2	1186.9	7558.5	696.5	44739.5	28633.9
8	872.3	1506.7	24082.8	2578.2	24719.4	19519.0
9	1476.2	5399.2	13955.8	5224.6	46670.3	12626.3
10	2431.2	4223.8	17236.1	1999.1	58159.1	33516.8
Mean Error	882.3	1758.1	7531.7	1434.8	33413.1	14496.1

Table 4-4 Gradient model of six hashtags in the second time period

4.3 After Feb. 1, 8:00 p.m.

At last, in this time period, we use the similar method to predict and we can get the following results:

Cross-Validation Errors	gopatriots	gohawks	patriots	nfl	superbowl	sb49
1	20.0	175.9	238.9	170.7	673.3	540.4
2	3.6	88.4	100.7	292.9	894.4	428.5
3	0.7	7.8	456.1	91.0	270.2	152.0
4	1.3	7.6	16.2	137.7	243.1	122.8
5	0.9	2.3	45.9	155.7	155.6	54.9
6	0.4	6.8	47.1	92.9	98.2	67.8
7	0.2	3.8	11.6	114.0	81.2	21.9
8	0.4	11.28	22.2	100.7	85.6	43.4
9	0.31	5.9	22.7	162.6	135.6	60.8
10	0.29	5.3	40.5	137.0	155.7	56.6
Mean Error	2.9	31.6	101.2	145.6	281.1	189.0

Table 4-5 Linear model of six hashtags in the third time period

Cross-Validation Errors	gopatriots	gohawks	patriots	nfl	superbowl	sb49
1	24.4	172.6	465.2	196.7	796.4	596.5
2	4.8	75.4	78.9	212.5	903.4	694.5
3	0.6	4.8	100.8	295.8	305.4	163.6
4	0.6	3.2	36.6	224.4	234.9	120.7
5	0.7	2.9	57.9	179.3	160.1	92.3
6	0.43	3.8	29.6	118.7	68.3	43.3
7	0.2	2.0	36.4	148.7	83.6	41.8
8	0.3	3.6	27.8	141.6	85.1	39.2
9	0.2	4.5	25.7	178.9	115.6	57.9
10	0.24	2.8	33.8	24.5	100.0	34.9
Mean Error	3.3	27.7	90.5	142.5	276.8	156.3

Table 4-6 Gradient model of six hashtags in the third time period

4.4 Conclusion

From the table above, we can see that gradient boost is better than linear model especially in the application of having lots of datasets. In our results, the error of linear model is much larger than gradient boost model, just as we expected. Besides, when the datasets are linear and small, the difference between gradient boost model and linear model is very negligible. According to all the results above, we can clearly say that our feature selection and model selection are both reasonable, and the predictor can predict the data very well.

5. Prediction of Testing Data

Up to now, all the things we have done are model design, we haven't use other new test data to verify our model. In this part, we are given 10 samples in three different periods, but we don't know which hashtag these samples belong to. So we need first to find which hashtag is dominant in the sample and then use the model of this hashtag to predict the data. The dominant hashtag in each test dataset is as follows:

Hashtag	Dominant Hashtag
sample1.period1	superbowl
sample2.period2	superbowl
sample3.period3	superbowl
sample4.period1	nfl
sample5.period1	patriots
sample6.period2	superbowl
sample7.period3	nfl
sample8.period1	nfl
sample9.period2	superbowl
sample10.period3	nfl

Table 5-1 Dominant Hashtag in each sample

Then, we perform prediction task for each test dataset using model trained by data with corresponding hashtag. We select top ten features and use gradient boost to do the prediction. The prediction results are shown as follow:

Hashtag	Prediction
sample1.period1	242
sample2.period2	119881
sample3.period3	411
sample4.period1	441
sample5.period1	348
sample6.period2	74501
sample7.period3	206
sample8.period1	298
sample9.period2	10867
sample10.period3	72

Table 5-2 Predicted Hashtag in each sample

6. Fan Base Prediction

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 her location. Here we define a list of the keywords which indicates the

location of Washington. Once the user's location contains any of the keywords, it will be assigned to the class 'WA', otherwise it will be classified to 'MA'.

Since the dataset is very huge and takes a lot of time and CPU memory to run, so we randomly chose 1,000,000 set of data as our training data. Based on the dataset, we train three different classifiers to do the prediction: Naïve Bayes, SVM and Logistic Regression.

6.1 Naïve Bayes

The accuracy of the model is:

$$Accuracy = 0.714934$$

The value of precision is:

$$Precision = 0.72$$

The value of recall is:

$$Recall = 0.71$$

The confusion matrix is:

$$\begin{bmatrix} 4562 & 1207 \\ 1116 & 1264 \end{bmatrix}$$

The ROC curve is shown is below:

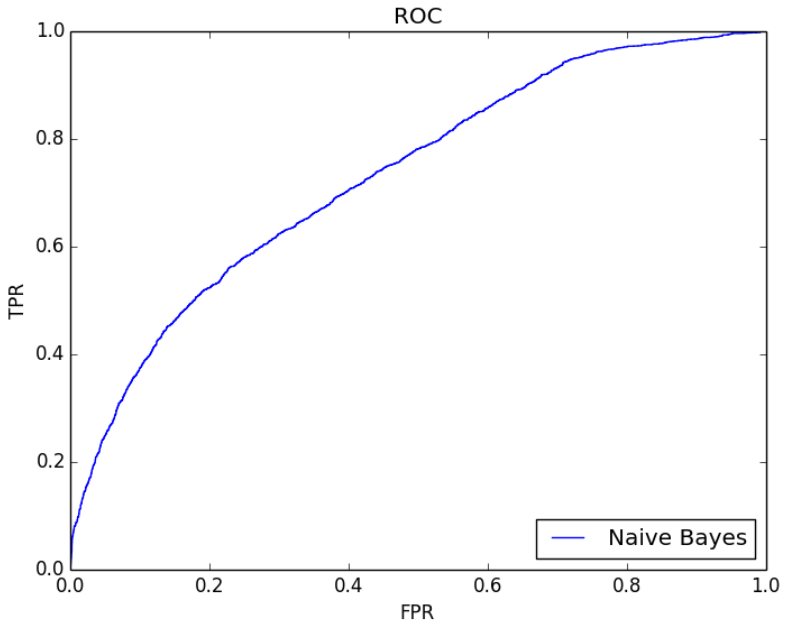


Figure 6-1 ROC curve of Naïve Bayes

6.2 SVM

The accuracy of the model is:

$$Accuracy = 0.788072$$

The value of precision is:

$$Precision = 0.79$$

The value of recall is:

$$Recall = 0.79$$

The confusion matrix is:

$$\begin{bmatrix} 5537 & 232 \\ 1495 & 885 \end{bmatrix}$$

The ROC curve is shown is below:

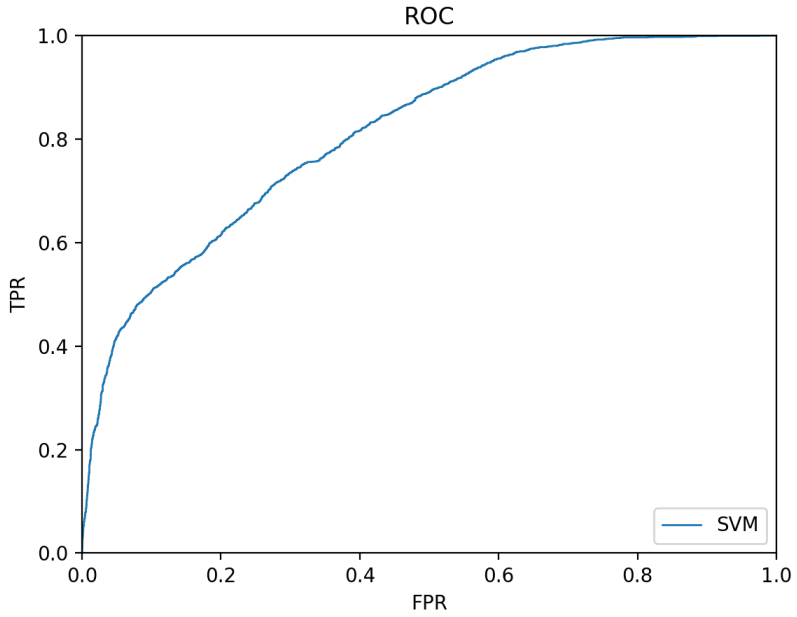


Figure 6-2 ROC curve of SVM

6.3 Logistic Regression

The accuracy of the model is:

$$Accuracy = 0.799239$$

The value of precision is:

$$Precision = 0.81$$

The value of recall is:

$$Recall = 0.80$$

The confusion matrix is:

$$\begin{bmatrix} 5586 & 183 \\ 1453 & 927 \end{bmatrix}$$

The ROC curve is shown is below:

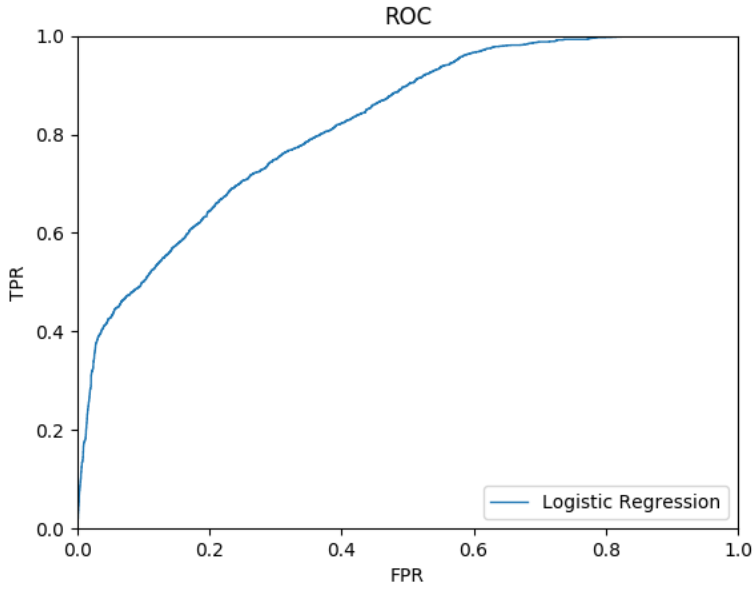


Figure 6-3 ROC curve of Logistic Regression

6.4 Conclusion

From the results shown above, we can find that SVM and Logistic Regression have higher prediction accuracy than Naïve Bayes. However, the ROC curve of all the three models are not satisfactory. In the later work, we will try to fit the model with the entire dataset to see if any of the performance can be improved.

7. Predicting Burst Popularity of Hashtags

7.1 Definition and Clarification

Tweet really gives us a convenient way to keep in touch with fresh news and make comments on them. After publishing the paper [1], the researchers of the same team did some follow-up research on predicting burst tweets phenomenon. In most cases, we can find that in tweets a breaking news is coming out of silence, and then the number of tweets grows rapidly in several hours, but finally it goes back to inactive. Therefore, for some bad breaking news, we may want to predict when a bursting event would happen and the bursting number of tweets so that we can control then ahead. For some good breaking news, we also want to predict the bursting popularity in order to analysis how they are concerned by people. Now the question has been clear, can we predict the bursting value before the burst happens, and more concretely, how can we tell an events will burst or not, and when will it start?

In the previous problem, we have trained models to predict the next hour's tweets corresponding to multiple hashtags. In this part, we are going to deal with the bursting popularity prediction problem using not bursting features. First, we need to define a bursting events and its stages.

7.2 Four Stages of a Bursting Hashtags

According to the research outcome of paper [2], we can divide a bursting hashtag into 4 different phase: action, bursting, off-bursting and inactive.

Active: If t_a is a time interval in the hashtag, C_{ta} is the number of tweets at t_a. Then we define the hashtag is active since t_a if:

$$C_{t_a} + C_{t_{a+1}} + C_{t_{a+2}} + C_{t_{a+3}} + C_{t_{a+4}} > \Phi$$

Here Φ is a parameter to determine the start point of active phase. From lots of statistic observation of large amount of tweet hashtags, Φ = 50.

Bursting: Within the 24 hours after a hashtag becomes active, we define a hashtag will change from active state to bursting state since time t if t first meets the condition of:

$$C_t > \max(C_1 + \delta, 1.5C_1)$$

Here, δ is another parameter to determine bursting state. From lots of statistic observation of large amount of tweet hashtags, δ =50.

Off-Burst: If a hashtag is already bursting, then we define this hashtag is at off-burst state from time interval t' if t' is the first time it meets the following formula:

$$C_{t'} < \max(C_1 + \delta, 1.5C_1)$$

Inactive: If $t_a^{'}$ is a time interval $C_{t_a^{'}}$ is the number of tweets at t_a. Then the hashtag is inactive since t_a if:

$$C_{t_a^{'}} + C_{t_a^{'}+1} + C_{t_a^{'}+2} + C_{t_a^{'}+3} + C_{t_a^{'}+4} < \Phi$$

7.3 Bursting Popularity Prediction

Based on the above mentioned definition of stage, we can do the prediction, and we will do it on the #gopatriots hashtag.

7.3.1 Prediction with one-hour intervals

First we program to find active time points and bursting time points. The time interval is one hour (Source code 7-1). In the "gopatriots" hashtag dataset, we found **2 bursting, 141 active time points and 27 bursting points** in total. Then we divided the time points into 2 sets based on the two bursting.

```
The number of active time points is:
141
The number of bursting time points is:
27
```

Figure 7-3-1 Statistic data of 1-hour prediction

Then we used the features of active points to predict the number of tweets of bursting points. What needs to mention is that as the paper [2] points out, predicting the exact value of the bursting number is extremely difficult and generally not necessary. Therefore we relax the problem and only predict the natural logarithm of the popularity.

The 9 features we used are (all in active state): **1)** number of tweets at active state; **2)** total number of retweets at active state; **3)** sum of the number of followers of the users posting the hashtag at active state; **4)** maximum number of followers of the users posting the hashtag at active state; **5)** number of reply at active state; **6)** number of mention at active state; **7)** ranking at active state; **8)** count of favorite at active state; **9)** impression at active state. In the experiment these features are denoted as x1, x2, … , x9.

Generally, the bursting state appears and disappears very fast, mostly less than 24 hours, thus we choose a max time gap between the active tweet data and bursting tweet data (5 hours) based on the active and bursting time points we figure out, which means that we predict the bursting data with the data 5 hour ahead of it.

In out experiment, we still use OLS regression model. The summary is shown in the figure below:

	coef	std err	t	P>|t|	[0.025	0.975]
x1	0.4586	0.224	2.045	0.057	-0.015	0.932
x2	0.0035	0.003	1.266	0.222	-0.002	0.009
x3	1.203e-05	1.13e-05	1.068	0.300	-1.17e-05	3.58e-05
x4	-1.026e-06	5.17e-06	-0.198	0.845	-1.19e-05	9.89e-06
x5	0.0812	0.132	0.615	0.547	-0.197	0.360
x6	0.0068	0.013	0.516	0.613	-0.021	0.035
x7	-0.0017	0.001	-2.501	0.023	-0.003	-0.000
x8	0.0103	0.042	0.248	0.807	-0.078	0.098
x9	-1.12e-05	6.93e-06	-1.617	0.124	-2.58e-05	3.42e-06
const	3.7004	0.729	5.079	0.000	2.163	5.238

Omnibus:	0.131	Durbin-Watson:	1.587
Prob(Omnibus):	0.937	Jarque-Bera (JB):	0.350
Skew:	0.039	Prob(JB):	0.840
Kurtosis:	2.448	Cond. No.	7.82e+06

Table 7-1 OLS Regression Summary (1-hour Prediction)

RMSE = 0.83 R-Squared = 0.651

From the table we can see that, the P value of the number of tweets and the P value of the total number of retweets are greater than 5%, which means that these two features are statistically not significant for the prediction. This phenomenon is due to the sample size limitation, since there are only 27 pairs of bursting and active time points used in this prediction. If we delete some feature with large P-value, we find that the P-value of some remaining features becomes smaller than 5%, and the fitting degree will also get lower.

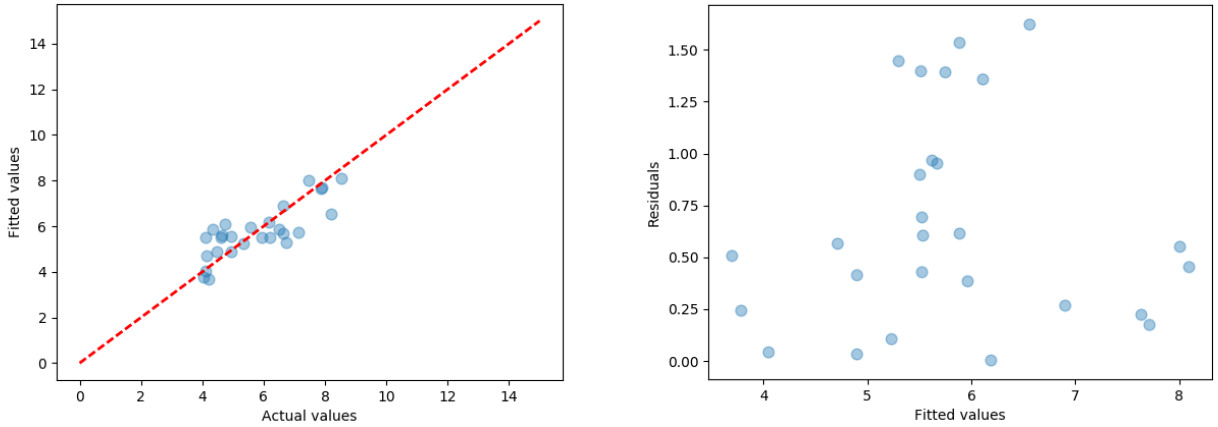


Figure 7-3-2 1-hour Prediction (Left) 1-hour Residual (Right)

From figure 7-3-2, we can see that most of the data points are near or falling around the line of y=x, which means fitted values are nearly the same as the actual values. And R-squared is 0.628. The residuals is also shown above.

7.3.2 Prediction with 30-min intervals

The procedure we used to do the 30-min prediction is nearly the same as that of 1-hour version except that the time interval declines to 30 minutes.

Under this circumstances, in "#gopatriots" dataset, we found **2 bursting, 110 active time points and 35 bursting time points**. Similarly, we divided these time points to two sets based on the two bursting.

```
The number of active time points is:
110
The number of bursting time points is:
35
```

Figure 7-3-3 Statistic data of 0.5-hour prediction

The features and algorithms we used are also the same those in 1-hour version. We choose a max time interval between the active tweet data and bursting tweet data (13*30 minutes) based on the active and bursting time points we figure out, which means that we predict the bursting data with the data 6.5 hour ahead of it.

In out experiment, we still use OLS regression model. The summary is shown in the figure below:

Dep. Variable:	y	R-squared:	0.451
Model:	OLS	Adj. R-squared:	0.253
Method:	Least Squares	F-statistic:	2.279
Date:	Mon, 20 Mar 2017	Prob (F-statistic):	0.0503
Time:	22:42:21	Log-Likelihood:	-43.519
No. Observations:	35	AIC:	107.0
Df Residuals:	25	BIC:	122.6
Df Model:	9		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
x1	-0.0409	0.381	-0.108	0.915	-0.825	0.743
x2	0.0023	0.003	0.825	0.417	-0.003	0.008
x3	1.794e-05	4.98e-05	0.360	0.722	-8.46e-05	0.000
x4	2.01e-05	7.15e-06	2.812	0.009	5.38e-06	3.48e-05
x5	-0.0029	0.316	-0.009	0.993	-0.654	0.648
x6	0.0541	0.024	2.295	0.030	0.006	0.103
x7	-0.0011	0.003	-0.397	0.694	-0.007	0.005
x8	0.0528	0.055	0.955	0.349	-0.061	0.167
x9	-3.905e-05	4.85e-05	-0.805	0.428	-0.000	6.08e-05
const	5.4440	1.068	5.098	0.000	3.244	7.643

Omnibus:	1.241	Durbin-Watson:	0.703
Prob(Omnibus):	0.538	Jarque-Bera (JB):	0.384
Skew:	-0.063	Prob(JB):	0.825
Kurtosis:	3.498	Cond. No.	6.64e+06

Table 7-2 OLS Regression Summary (0.5-hour Prediction)

RMSE = 0.84 R-Squared = 0.451

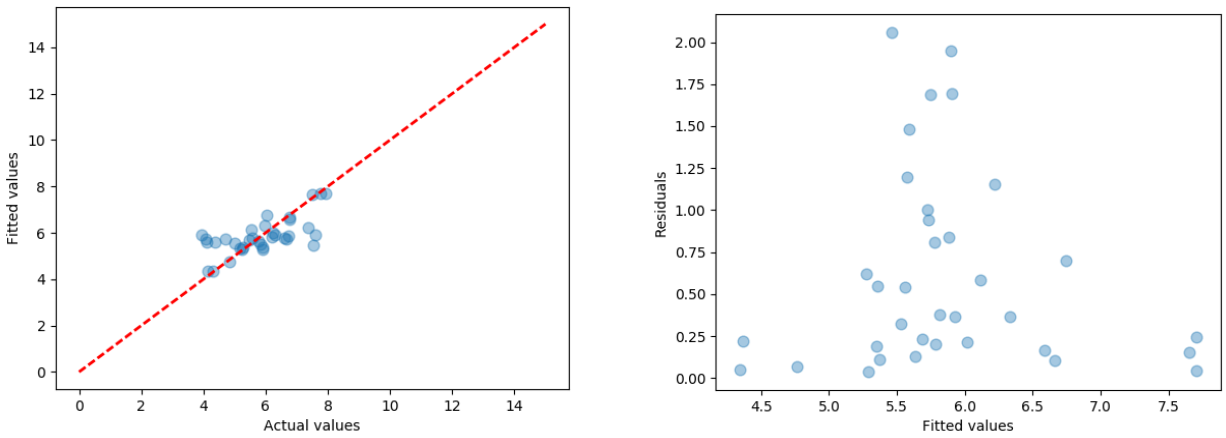


Figure 7-3-4 0.5-hour Prediction (Left) 0.5-hour Residual (Right)

From figure 7-3-4, we can see that most of the points are near or around the line y=x, which means fitted values are nearly the same as the actual value. The residual of fitted values are also shown above.

8. Conclusion

In this project, the main objective is to predict the popularity of hashtags on the Twitter. After observing the characteristics of the data, we select some features to do linear regression. Then, we add some related features in our model to improve the performance. Any way to improve the performance is splitting our data into 3 parts and doing cross-validation on them. Then we can use the test data to test out model. In the end, because we are not satisfied with only predicting the popularity of next hour, we also want to predict some bursting events in the future, thus we raised a problem of predicting upcoming bursting phenomenon using non-bursting data, and used a method in paper [2] to solve it.

References

[1] Kong, Shoubin, et al. "On the Real-time prediction problems of bursting hashtags in twitter." *arXiv preprint arXiv:1401.2018* (2014).

[2] Kong, Shoubin, et al. "Predicting bursts and popularity of hashtags in real-time." Proceedings of the 37th international ACM SIGIR conference on Research & development in information retrieval. ACM, 2014.