EE 219 Project 5

Yifan Shu, Chengshun Zhang, Xuan Yang

Introduction

A useful practice in social network analysis is to predict future popularity of a subject or event. During this project, we are using Twitter data to predict what will become popular. We used the available Twitter data collected by querying popular hashtags related to the 2015 Super Bowl spanning a period starting from two weeks before the game to a week after the game. Besides, we used data from some of the related hashtags to train a regression model and then use the modeltomakepredictionsforotherhashtags. We tested several models, using the training data to train the data, and compare the performance of them by using a test data to make predictions.

Question 1

We first download the data from https://ucla.app.box.com/s/24oxnhsoj6kpxhl6gyvuck25i3s4426d. There are total six files and each file contains information for one hashtag. We then used json library in Python to load each data file one line by one line and got the statistics including

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

The result is shown in the table Table 1 below.

Table 1. Three Statistics for Each Hashtag

Hashtag	Avg # tweets per hour	Avg # followers	Avg # retweets per tweet
#gohawks	292.48785062173687	2217.9237355281984	2.0132093991319877
#gopatriots	40.954698006061946	1427.2526051635405	1.4081919101697078
#nfl	397.0213901819841	4662.37544523693	1.5344602655543254
#patriots	750.89426460689	3280.4635616550277	1.7852871288476946
#sb49	1276.8570598680474	10374.160292019487	2.52713444111402
#superbowl	2072.11840170408	8814.96799424623	2.3911895819207736

Question 2

When we calculated the statistics for each hashtag, we also stored the maximum time and minimum time of each hashtag so that we can use them later. For each hashtag, we divided the total time to one hour interval between maximum time and minimum time and counted the number of tweets in each time interval. We plotted histogram of "number of tweets in hour" over time for #superbowl and #nfl, as shown in Fig 1 and Fig 2 respectively.

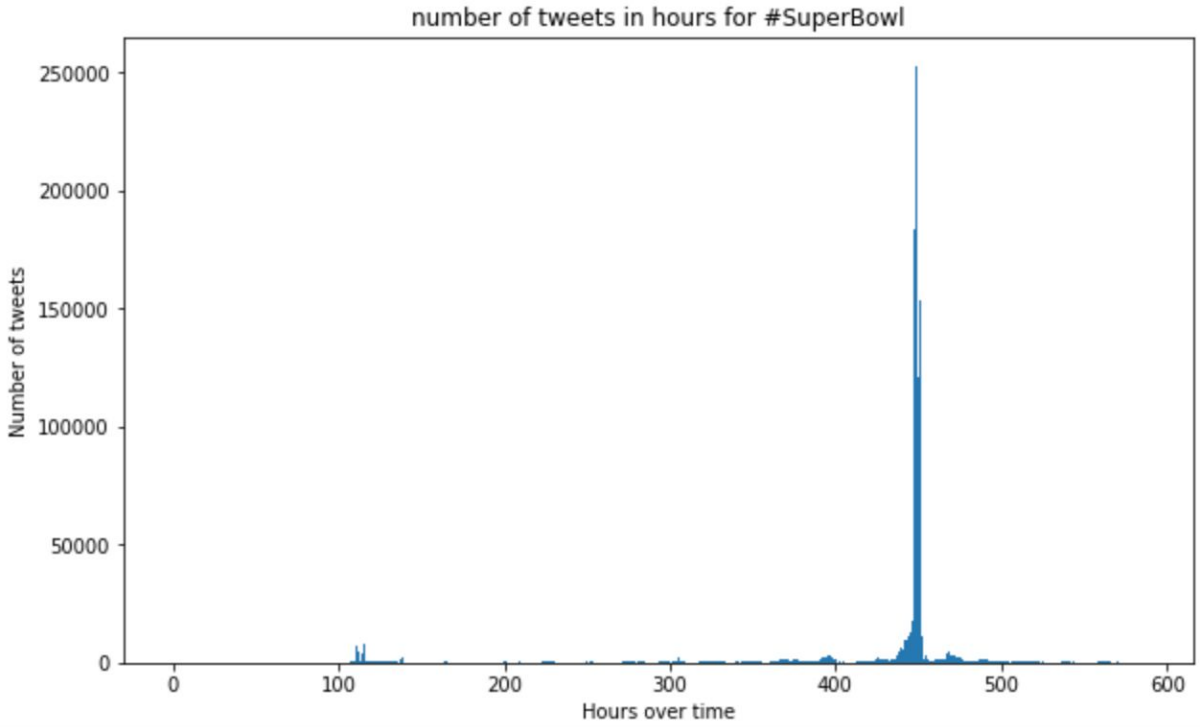

Figure 1. Number of Tweets in hour over time for #superbowl

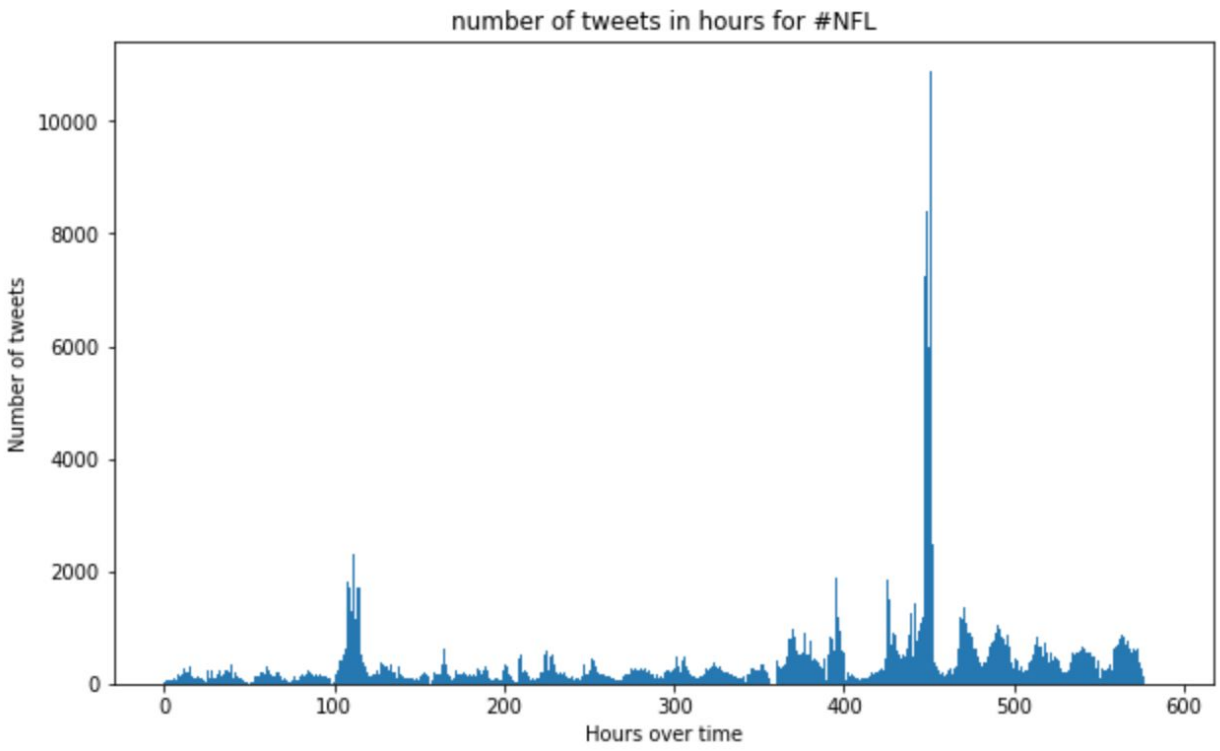

Figure 2. Number of Tweets in hour over time for #nfl

Question 3

We used the OLS in Python's statsmodels library and trained one linear regression model for each hashtag. The features we used are:

- 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 (we took 24 values that represent hours of the day, from 0 to 23, with respect to PST)

We created 1-hour time window and calculated the above features in each time window and ended up getting <# of hours> data points.

The Mean Square Error (MSE) and R-squared measure for each model are shown in Table 2. The summary of each model is shown in Fig 3 to Fig 8. x1 to x5 in each summary corresponds to each feature above in order.

Table 2. MSE and R-squared for each hashtag

Hashtag	MSE	R-squared
#gohawks	389086.61457122036	0.709
#gopatriots	47576.188942486355	0.454

#nfl	209013.3222443561	0.730
#patriots	5466476.76932625	0.667
#sb49	16530174.167058893	0.805
#superbowl	58037195.52955217	0.781

We used t-test and p-value to analyze the significance of each feature. Basically the larger the absolute value of t-test, the more significant the feature is. The smaller the p-value, the more significant the feature is.

From the summary of each hashtag, we can see that *number of tweets* is a significant feature in all hashtags. *Total number of retweets* is also a significant feature in #gohawks, #gopatriots and #superbowl, but is not so significant in other hashtags. *Sum of the number of followers of the users posting the hashtag* is a significant feature in #nfl and #patriots. *Maximum number of followers of the users posting the hashtag* is only significant in #sb49. *Time of the day* is not so significant from the summary, as it at best ranks third in #nfl. The t-test value and p-value for each hashtag is shown in Table 3 to Table 8.

OLS Regression Results

Dep. Variable:	y	R-squared:	0.709
Model:	OLS	Adj. R-squared:	0.707
Method:	Least Squares	F-statistic:	279.5
Date:	Tue, 12 Mar 2019	Prob (F-statistic):	4.57e-151
Time:	19:47:20	Log-Likelihood:	-4540.0
No. Observations:	578	AIC:	9090.
Df Residuals:	573	BIC:	9112.
Df Model:	5		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
x1	1.1645	0.119	9.748	0.000	0.930	1.399
x2	-0.0810	0.033	-2.435	0.015	-0.146	-0.016
x3	-0.0001	6.24e-05	-1.725	0.085	-0.000	1.49e-05
x4	4.212e-05	0.000	0.348	0.728	-0.000	0.000
x5	3.8362	2.141	1.792	0.074	-0.369	8.042

Omnibus:	632.026	Durbin-Watson:	1.864
Prob(Omnibus):	0.000	Jarque-Bera (JB):	222979.332
Skew:	4.342	Prob(JB):	0.00
Kurtosis:	98.829	Cond. No.	2.02e+05

Figure 3. Summary for #gohawks

OLS Regression Results

Dep. Variable:	y	R-squared:	0.454
Model:	OLS	Adj. R-squared:	0.449
Method:	Least Squares	F-statistic:	94.54
Date:	Tue, 12 Mar 2019	Prob (F-statistic):	2.14e-72
Time:	20:03:12	Log-Likelihood:	-3905.5
No. Observations:	574	AIC:	7821.
Df Residuals:	569	BIC:	7843.
Df Model:	5		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
x1	-1.5580	0.298	-5.227	0.000	-2.143	-0.973
x2	1.6997	0.245	6.948	0.000	1.219	2.180
x3	0.0001	0.000	0.581	0.561	-0.000	0.001
x4	-0.0004	0.000	-1.870	0.062	-0.001	1.92e-05
x5	0.4805	0.706	0.680	0.497	-0.907	1.868

Omnibus:	1021.072	Durbin-Watson:	2.438
Prob(Omnibus):	0.000	Jarque-Bera (JB):	1171508.631
Skew:	11.011	Prob(JB):	0.00
Kurtosis:	223.223	Cond. No.	3.11e+04

Figure 4. Summary for #gopatriots

OLS Regression Results

Dep. Variable:	y	R-squared:	0.730
Model:	OLS	Adj. R-squared:	0.728
Method:	Least Squares	F-statistic:	315.4
Date:	Tue, 12 Mar 2019	Prob (F-statistic):	5.03e-163
Time:	20:06:04	Log-Likelihood:	-4428.3
No. Observations:	587	AIC:	8867.
Df Residuals:	582	BIC:	8889.
Df Model:	5		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
x1	0.4814	0.114	4.208	0.000	0.257	0.706
x2	-0.2327	0.060	-3.878	0.000	-0.351	-0.115
x3	0.0002	1.84e-05	10.041	0.000	0.000	0.000
x4	-0.0002	2.92e-05	-7.865	0.000	-0.000	-0.000
x5	8.6207	1.711	5.040	0.000	5.261	11.980

Omnibus:	549.091	Durbin-Watson:	2.070
Prob(Omnibus):	0.000	Jarque-Bera (JB):	149763.175
Skew:	3.321	Prob(JB):	0.00
Kurtosis:	80.968	Cond. No.	3.90e+05

Figure 5. Summary for #nfl

OLS Regression Results

Dep. Variable:	y	R-squared:	0.667
Model:	OLS	Adj. R-squared:	0.664
Method:	Least Squares	F-statistic:	232.8
Date:	Tue, 12 Mar 2019	Prob (F-statistic):	2.92e-136
Time:	20:10:34	Log-Likelihood:	-5386.3
No. Observations:	587	AIC:	1.078e+04
Df Residuals:	582	BIC:	1.080e+04
Df Model:	5		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
x1	1.0036	0.085	11.838	0.000	0.837	1.170
x2	-0.0650	0.072	-0.905	0.366	-0.206	0.076
x3	-7.396e-05	2.79e-05	-2.648	0.008	-0.000	-1.91e-05
x4	8.715e-05	9.37e-05	0.930	0.353	-9.69e-05	0.000
x5	14.0915	8.101	1.739	0.082	-1.820	30.003

Omnibus:	917.929	Durbin-Watson:	2.080
Prob(Omnibus):	0.000	Jarque-Bera (JB):	498211.103
Skew:	8.677	Prob(JB):	0.00
Kurtosis:	144.664	Cond. No.	7.01e+05

Figure 6. Summary for #patriots

OLS Regression Results

Dep. Variable:	y	R-squared:	0.805
Model:	OLS	Adj. R-squared:	0.803
Method:	Least Squares	F-statistic:	475.2
Date:	Tue, 12 Mar 2019	Prob (F-statistic):	7.28e-202
Time:	20:15:48	Log-Likelihood:	-5662.4
No. Observations:	582	AIC:	1.133e+04
Df Residuals:	577	BIC:	1.136e+04
Df Model:	5		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
x1	1.0912	0.088	12.395	0.000	0.918	1.264
x2	-0.1125	0.079	-1.433	0.152	-0.267	0.042
x3	2.786e-06	1.24e-05	0.225	0.822	-2.15e-05	2.71e-05
x4	6.725e-05	4.28e-05	1.570	0.117	-1.69e-05	0.000
x5	0.5488	13.851	0.040	0.968	-26.655	27.753

Omnibus:	1184.527	Durbin-Watson:	1.678
Prob(Omnibus):	0.000	Jarque-Bera (JB):	2245389.731
Skew:	14.729	Prob(JB):	0.00
Kurtosis:	305.862	Cond. No.	6.48e+06

Figure 7. Summary for #sb49

OLS Regression Results

Dep. Variable:	y	R-squared:	0.781
Model:	OLS	Adj. R-squared:	0.779
Method:	Least Squares	F-statistic:	415.1
Date:	Tue, 12 Mar 2019	Prob (F-statistic):	4.34e-189
Time:	20:23:59	Log-Likelihood:	-6069.3
No. Observations:	586	AIC:	1.215e+04
Df Residuals:	581	BIC:	1.217e+04
Df Model:	5		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
x1	2.5533	0.111	23.094	0.000	2.336	2.770
x2	-0.3919	0.040	-9.735	0.000	-0.471	-0.313
x3	-0.0001	2.29e-05	-5.899	0.000	-0.000	-9e-05
x4	0.0009	0.000	6.476	0.000	0.001	0.001
x5	-62.8524	28.245	-2.225	0.026	-118.328	-7.377

Omnibus:	352.078	Durbin-Watson:	2.379
Prob(Omnibus):	0.000	Jarque-Bera (JB):	781423.704
Skew:	0.938	Prob(JB):	0.00
Kurtosis:	181.886	Cond. No.	9.53e+06

Figure 8. Summary for #superbowl

Table 3. T-test and p-value for #gohawks

	t-test	p-value
Number of tweets	9.748	0.000
Total number of retweets	-2.435	0.015
Sum of the number of followers of the users posting the hashtag	-1.725	0.085
Maximum number of followers of the users posting the hashtag	0.348	0.728
Time of the day	1.792	0.074

Table 4. T-test and p-value for #gopatriots

	t-test	p-value
Number of tweets	-5.227	0.000
Total number of retweets	6.948	0.000

Sum of the number of followers of the users posting the hashtag	0.581	0.561
Maximum number of followers of the users posting the hashtag	-1.870	0.062
Time of the day	0.680	0.497

Table 5. T-test and p-value for #nfl

	t-test	p-value
Number of tweets	4.208	0.000
Total number of retweets	-3.878	0.000
Sum of the number of followers of the users posting the hashtag	10.041	0.000
Maximum number of followers of the users posting the hashtag	-7.865	0.000
Time of the day	5.040	0.000

Table 6. T-test and p-value for #patriots

	t-test	p-value
Number of tweets	11.838	0.000
Total number of retweets	-0.905	0.366
Sum of the number of followers of the users posting the hashtag	-2.648	0.008
Maximum number of followers of the users posting the hashtag	0.930	0.353
Time of the day	1.739	0.082

Table 7. T-test and p-value for #sb49

	t-test	p-value
Number of tweets	12.395	0.000

Total number of retweets	-1.433	0.152
Sum of the number of followers of the users posting the hashtag	0.225	0.822
Maximum number of followers of the users posting the hashtag	1.570	0.117
Time of the day	0.040	0.968

Table 8. T-test and p-value for #superbowl

	t-test	p-value
Number of tweets	23.094	0.000
Total number of retweets	-9.735	0.000
Sum of the number of followers of the users posting the hashtag	-5.899	0.000
Maximum number of followers of the users posting the hashtag	6.476	0.000
Time of the day	-2.225	0.026

Question 4

We used the same linear regression model with OLS in statsmodels library, but with different features that we referenced to paper and combined our own understanding. They are:

- Author count: number of unique authors who posted tweets containing the hashtag.
- Mentions count: number of mentions (@).
- Passivity: passivity is defined as the following equation:

$$P_{sv}(u_i) = \frac{N_d(u_i)}{1.0 + N_t(u_i)}$$

  $N_d(u_i)$ denotes the number of days since the user account was created, $N_t(u_i)$ denotes the total number of tweets posted by the user.
- Friends count: number of friends of the users posting the hashtag.
- Ranking score: sum of ranking score of all the posts.

And we also used the 1-hour time window as in Question 3. We then fit the models on the data of each hashtag and got the MSE and R-squared values shown in Table 9.

Table 9. MSE and R-squared for each hashtag

Hashtag	MSE	R-squared
#gohawks	326359.91003858065	0.756
#gopatriots	30790.34299826502	0.646
#nfl	212769.83907092322	0.726
#patriots	4576598.777337125	0.721
#sb49	11081474.184033835	0.869
#superbowl	65094852.30742756	0.755

Again, we used t-test and p-value to evaluate the significance of each feature and the result is shown in Table 10 to Table 15.

Table 10. T-test and p-value for #gohawks

	t-test	p-value
Author count	-1.890	0.059
Mentions count	5.440	0.000
Passivity	-7.500	0.000
Friends count	6.048	0.000
Ranking score	4.960	0.000

Table 11. T-test and p-value for #gopatriots

	t-test	p-value
Author count	9.719	0.000
Mentions count	0.718	0.473
Passivity	6.034	0.000
Friends count	-0.451	0.652
Ranking score	-10.556	0.000

Table 12. T-test and p-value for #nfl

	t-test	p-value
Author count	-0.989	0.323
Mentions count	8.455	0.000
Passivity	-2.852	0.004
Friends count	0.642	0.521
Ranking score	0.929	0.353

Table 13. T-test and p-value for #patriots

	t-test	p-value
Author count	-3.106	0.002
Mentions count	7.810	0.000
Passivity	-0.020	0.984
Friends count	6.625	0.000
Ranking score	1.066	0.287

Table 14. T-test and p-value for #sb49

	t-test	p-value
Author count	3.851	0.000
Mentions count	10.654	0.000
Passivity	3.452	0.001
Friends count	14.351	0.000
Ranking score	-14.821	0.000

Table 15. T-test and p-value for #patriots

	t-test	p-value
Author count	-2.263	0.024
Mentions count	11.243	0.000

Passivity	-4.617	0.000
Friends count	-12.016	0.000
Ranking score	10.311	0.000

Question 5

From the tables in Question 4, we can see that for #gohawks, *mentions count*, *passivity* and *friends count* are top three significant features; for #gopatriots, *author count*, *passivity* and *ranking score* are top three significant features; for #nfl, *author count*, *mentions count* and *passivity* are top three significant features; for #patriots, *author count*, *mentions count* and *friends count* are top three significant features; for #sb49, *mentions count*, *friends count* and *ranking score* are top three significant features; for #superbowl, *mentions count*, *friends count* and *ranking score* are top three significant features.

There is no universal feature that is significant for each hashtag, so we used majority count to find the top three significant features across the six hashtags. The result is

- Mentions count
- Passivity
- Friends count

We drew scatter plots of predicant (predicted number of tweets for next hour) versus value of that feature for each of the above three feature on six hashtags. The results are shown in Fig 9 to Fig 26.

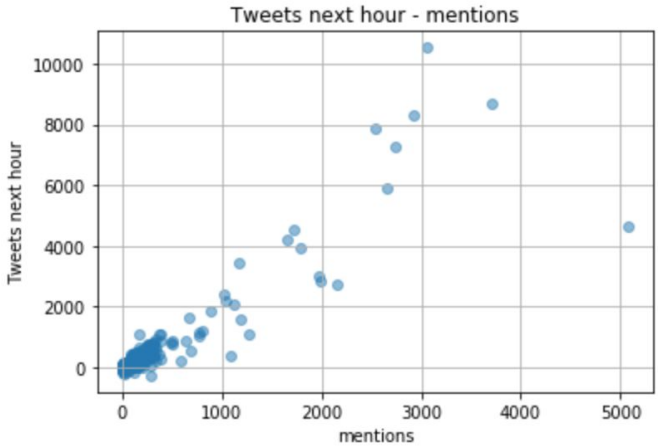

Figure 9. Predicant vs Mentions Count for #gohawks

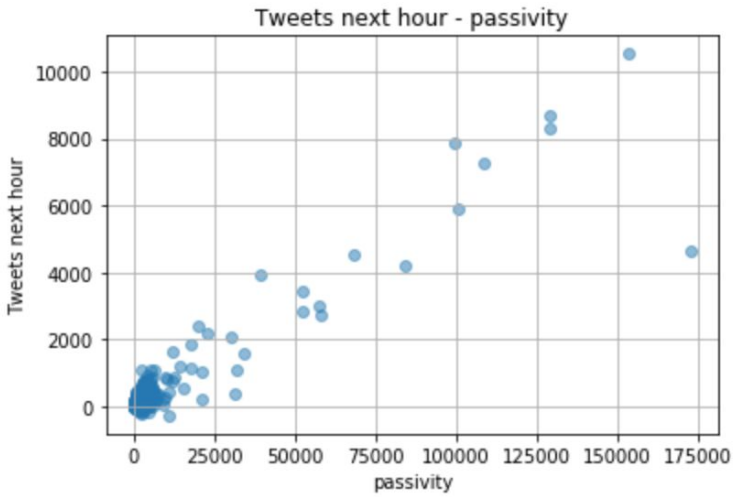

Figure 10. Predicant vs Passivity for #gohawks

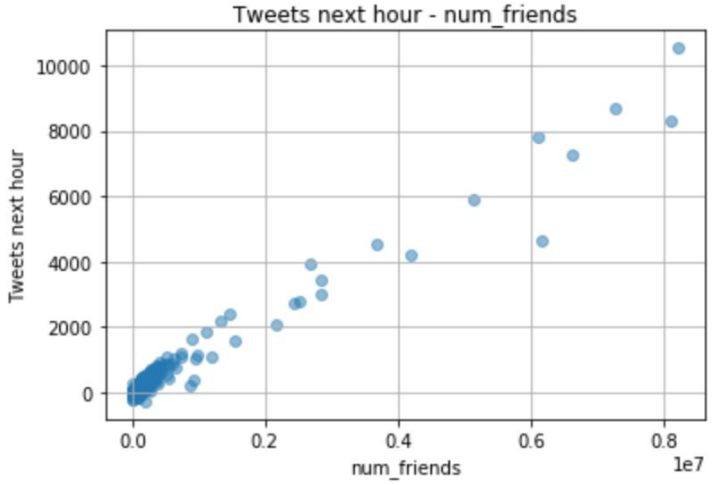

Figure 11. Predicant vs Friends Count for #gohawks

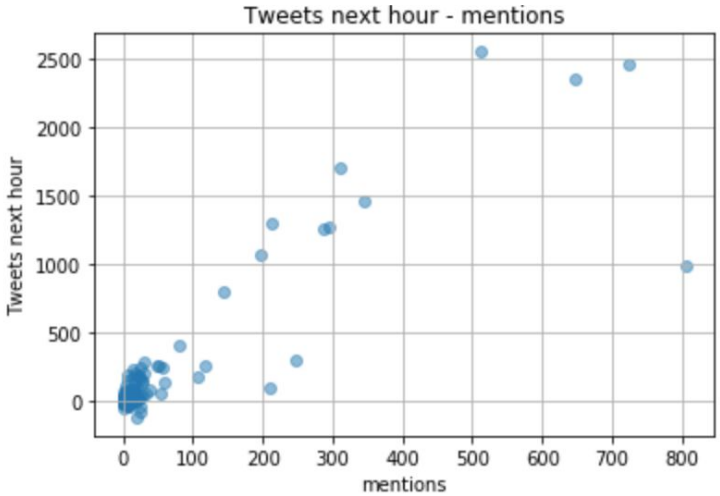

Figure 12. Predicant vs Mentions Count for #gopatriots

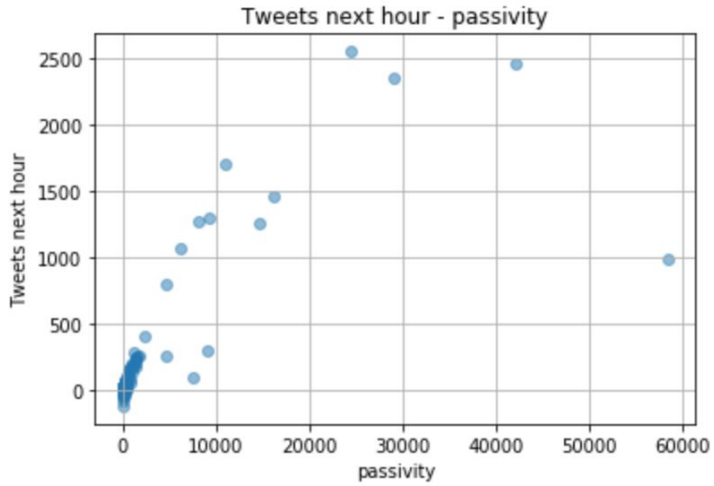

Figure 13. Predicant vs Passivity for #gopatriots

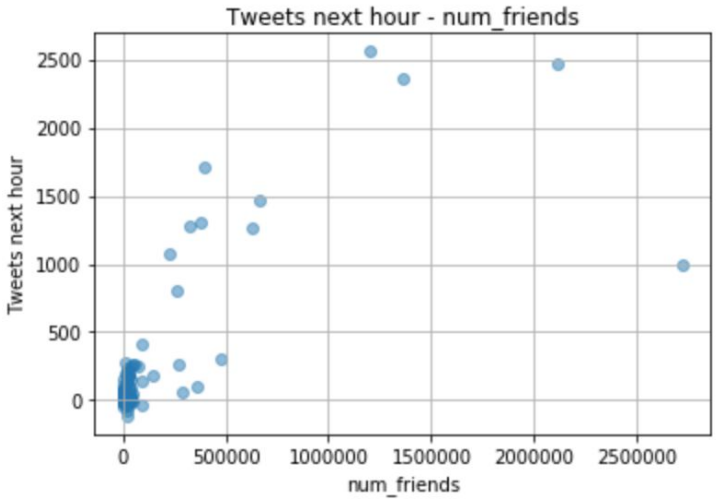

Figure 14. Predicant vs Friends Count for #gopatriots

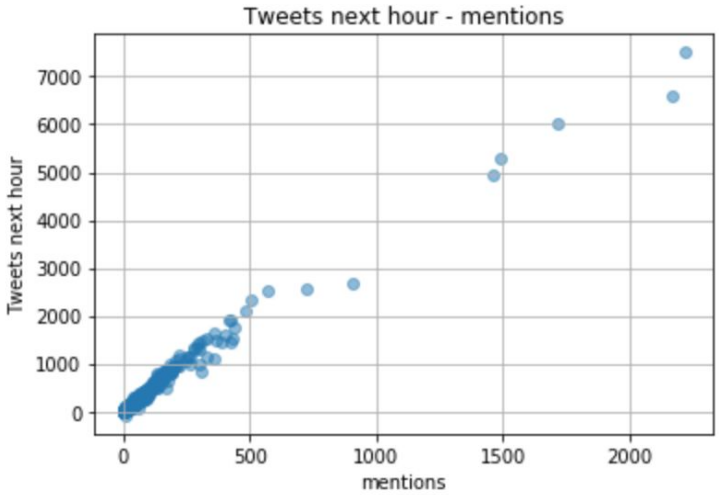

Figure 15. Predicant vs Mentions Count for #nfl

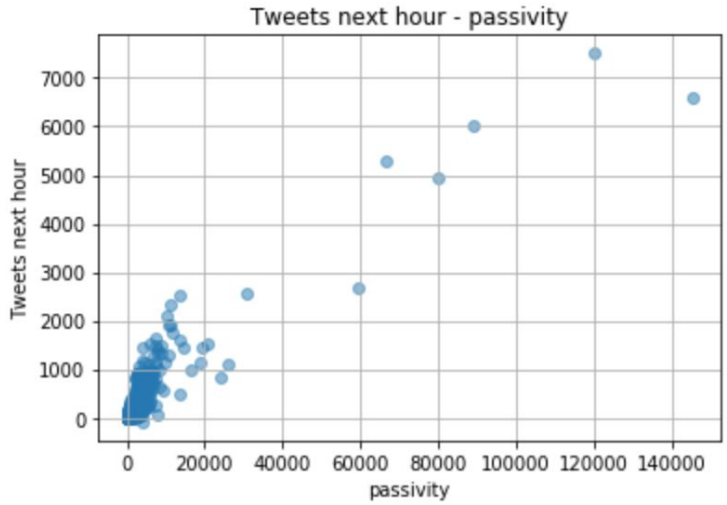

Figure 16. Predicant vs Passivity for #nfl

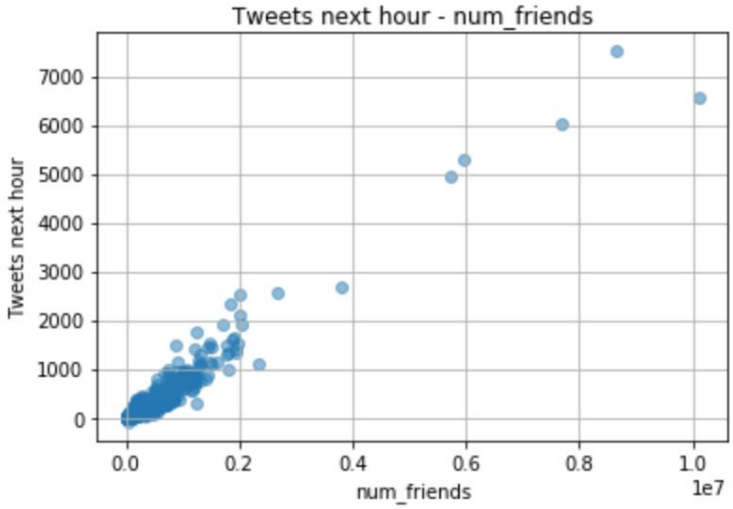

Figure 17. Predicant vs Friends Count for #nfl

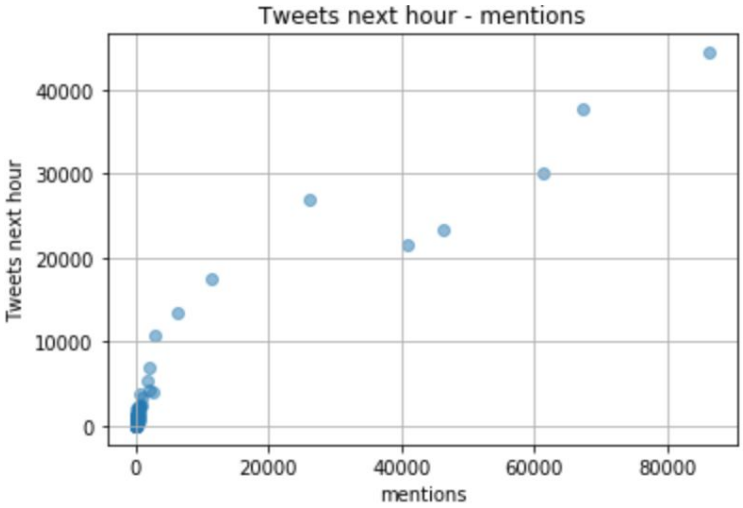

Figure 18. Predicant vs Mentions Count for #patriots

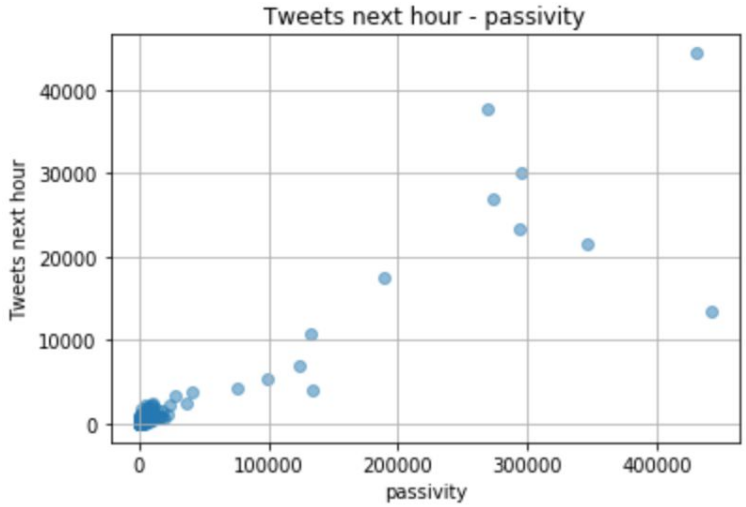

Figure 19. Predicant vs Passivity for #patriots

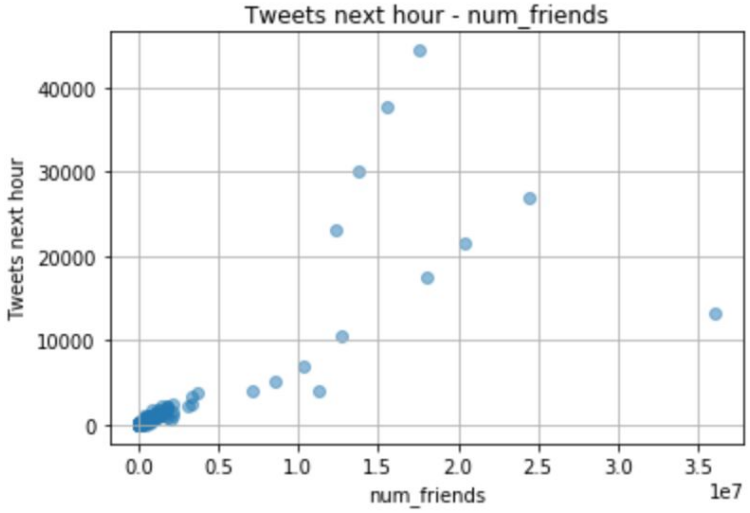

Figure 20. Predicant vs Friends Count for #patriots

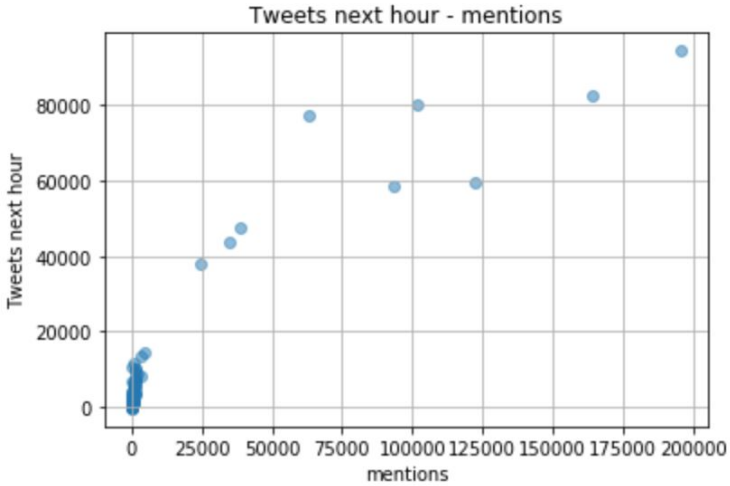

Figure 21. Predicant vs Mentions Count for #sb49

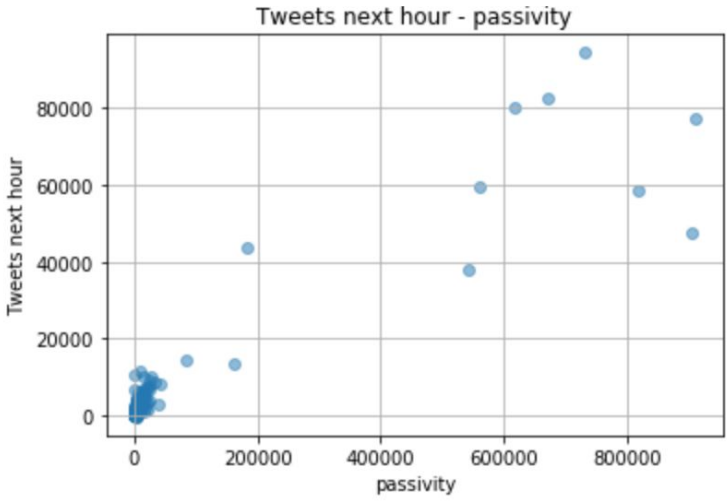

Figure 22. Predicant vs Passivity for #sb49

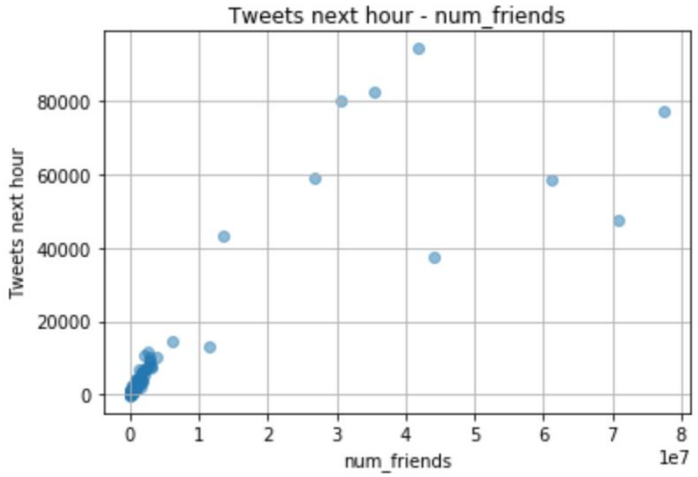

Figure 23. Predicant vs Friends Count for #sb49

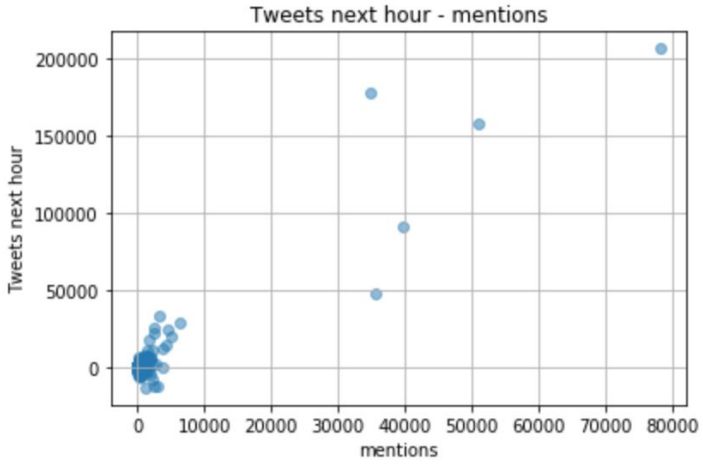

Figure 24. Predicant vs Mentions Count for #superbowl

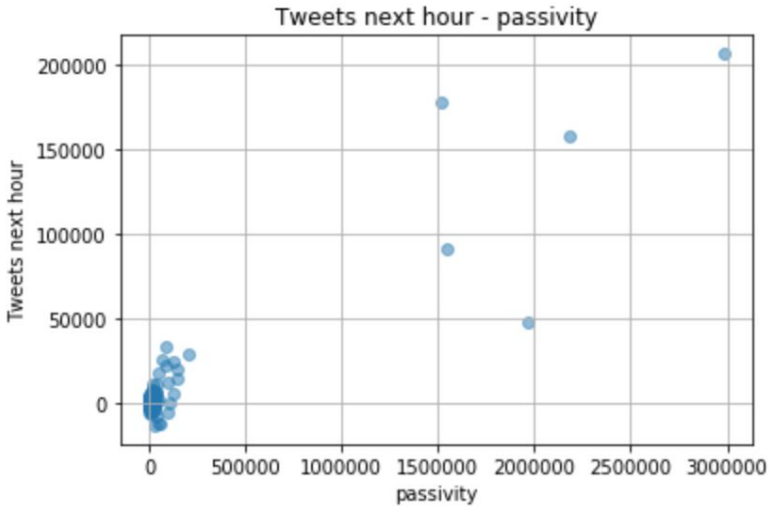

Figure 25. Predicant vs Passivity for #superbowl

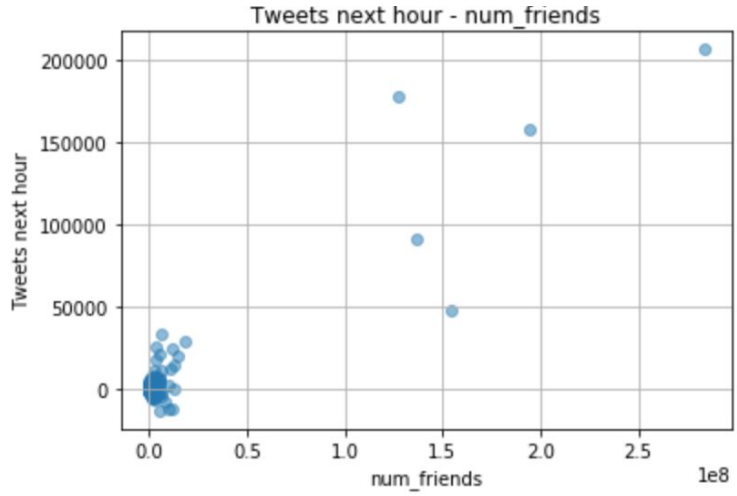

Figure 26. Predicant vs Friends Count for #superbowl

The regression coefficients of the three features for each hashtag are shown in Table 16.

Table 16. Regression Coefficients of Each Feature for Each Hashtag

	Mentions Count	Passivity	Friends Count
#gohawks	1.6139	-0.1236	0.0012
#gopatriots	0.5120	0.2179	-0.0002
#nfl	4.5495	-0.0323	9.657e-05
#patriots	1.1688	-0.0005	0.0018
#sb49	1.8342	0.0947	0.0060

#superbowl	13.5610	-0.1720	-0.0073

From the plots we can see that the slope of the line is always positive, but this is not the case for coefficients as shown in Table 16. This is reasonable because we are not using mono feature. So, the regression coefficients of each feature is the result of intercorrelation with other features. The coefficient can be negative even though the slope of the line that predicant versus that feature alone has a positive slope.

Question 6

We defined three time periods and their corresponding window length as below:

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

We trained 3 regression models for each hashtag, one for each time periods shown above and we used features in Question 3. We predicted the number of tweets for next hour for time period 1 and 3; for next 5 minute for time period 2. The MSE and R-squared score for each model and each hashtag are shown below in Table 17 to Table 22.

Table 17. MSE and R-squared for #gohawks

Time Period	MSE	R-squared
Before Feb.1 8:00 a.m.	290288.97685053764	0.665
Between Feb.1 8:00 a.m. and Feb.1 8:00 p.m.	122111.3256125044	0.628
After Feb.1 8:00 p.m.	2002.4089945633852	0.882

Table 18. MSE and R-squared for #gopatriots

Time Period	MSE	R-squared
Before Feb.1 8:00 a.m.	2672.805976590645	0.463
Between Feb.1 8:00 a.m. and Feb.1 8:00 p.m.	16405.867336461277	0.589
After Feb.1 8:00 p.m.	89.50224958341418	0.770

Table 19. MSE and R-squared for #nfl

Time Period	MSE	R-squared

Before Feb.1 8:00 a.m.	57620.76162516333	0.719
Between Feb.1 8:00 a.m. and Feb.1 8:00 p.m.	23019.976901364116	0.896
After Feb.1 8:00 p.m.	16706.081053186925	0.947

Table 20. MSE and R-squared for #patriots

Time Period	MSE	R-squared
Before Feb.1 8:00 a.m.	414481.8008567434	0.556
Between Feb.1 8:00 a.m. and Feb.1 8:00 p.m.	660210.518734672	0.891
After Feb.1 8:00 p.m.	11916.856419764157	0.928

Table 21. MSE and R-squared for #sb49

Time Period	MSE	R-squared
Before Feb.1 8:00 a.m.	7136.773182458621	0.883
Between Feb.1 8:00 a.m. and Feb.1 8:00 p.m.	1598766.7936376657	0.947
After Feb.1 8:00 p.m.	49338.55222581286	0.925

Table 22. MSE and R-squared for #superbowl

Time Period	MSE	R-squared
Before Feb.1 8:00 a.m.	384513.33446941635	0.597
Between Feb.1 8:00 a.m. and Feb.1 8:00 p.m.	11630855.203604735	0.889
After Feb.1 8:00 p.m.	85266.10420321378	0.932

The overall result is consistent with the result in Question 3. And we can see that the MSE and R-squared values before Feb.1 8:00 a.m. and after Feb.1 8:00 p.m. are much better than those between Feb.1 8:00 a.m. and after Feb.1 8:00 p.m. One possible reason is that users are not so active before and after the superbowl, thus results in much fewer tweets. Because of smaller dataset, the model is likely to overfit in these two time periods.

Question 7

We aggregated the data of all hashtags and trained 3 models for each time period in Question 6 to predict the number of tweets for next time interval. The MSE and R-squared values are shown below in Table 23.

Table 23. MSE and R-squared for Aggregate Data

Time Period	MSE	R-squared
Before Feb.1 8:00 a.m.	4609787.853991622	0.509
Between Feb.1 8:00 a.m. and Feb.1 8:00 p.m.	29340842.713701054	0.904
After Feb.1 8:00 p.m.	404923.2727548294	0.944

We can see from the table above that both MSE and R-squared values of aggregate data are worse than those for individual hashtags. This again shows the inconsistency across different hashtags. From previous questions, we have seen that most significant features for each hashtag is different. But the trend of results for different time periods is the same, i.e. the result is better for models on time periods before and after superbowl.

Question 8

We used grid search to find the best parameter set for RandomForestRegressor and GradientBoostingRegressor on aggregate data respectively. We used the 1-hour time window and the same features in Question 3. The parameter set is shown below:

```
{
        '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]
}
```

We set cv = KFold(5, shuffle=True) and scoring = 'neg_mean_squared_error' for the grid search.
The best parameters found for RandomForestRegressor is:

```
{
        'max_depth': 60,
        'max_features': 'auto',
        'min_samples_leaf': 4,
        'min_samples_split': 5,
        'n_estimators': 400
}
```

And the MSE and R-squared values are:

MSE = -258331806.56741208

R-squared = 0.8099101702701623

The best parameters found for GradientBoostingRegressor is:

```
{
        'max_depth': 40,
        'max_features': 'sqrt',
        'min_samples_leaf': 4,
        'min_samples_split': 2,
        'n_estimators': 1000
}
```

And the MSE and R-squared values are:

MSE = -166048862.61051157

R-squared = 0.9997371041929909

Because we used negative mean errors, the MSEs are negative as expected. But the absolute values are really large, as we have pointed out earlier that it is because data inconsistency across the six hashtags and we used the aggregate data to train our models.

Question 9

We trained a linear regression model with OLS on the entire dataset and the result is shown below in Fig 27. We used 1-hour time window and the same features in Question 3 to train the model. The MSE and R-squared values are:

MSE = 129742264.57460497

R-squared = 0.842

From Question 8, the best estimator GradientBoostingRegressor. With the best parameter set, the MSE and R-squared values are:

MSE = -166048862.61051157

R-squared = 0.9997371041929909

The absolute value of MSE from GradientBoostingRegressor is larger than that from OLS and the R-squared value from GradientBoostingRegressor is smaller than that from OLS. So we concluded that the best estimator found in Question 8 performs worse than OLS.

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.842
Model:                            OLS   Adj. R-squared:                  0.841
Method:                 Least Squares   F-statistic:                     621.7
Date:                Thu, 14 Mar 2019   Prob (F-statistic):          1.06e-230
Time:                        00:00:07   Log-Likelihood:                -6315.8
No. Observations:                 587   AIC:                         1.264e+04
Df Residuals:                     582   BIC:                         1.266e+04
Df Model:                           5
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
x1             1.7280      0.075     23.036      0.000       1.581       1.875
x2            -0.6075      0.051    -11.865      0.000      -0.708      -0.507
x3          7.349e-05   1.32e-05      5.562      0.000    4.75e-05    9.94e-05
x4             0.0007      0.000      5.868      0.000       0.000       0.001
x5          -138.5015     42.369     -3.269      0.001    -221.717     -55.286
==============================================================================
Omnibus:                      857.298   Durbin-Watson:                   1.859
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           704372.074
Skew:                          -7.325   Prob(JB):                         0.00
Kurtosis:                     172.069   Cond. No.                     1.73e+07
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.73e+07. This might indicate that there are
strong multicollinearity or other numerical problems.
MSE for aggregated data: 129742264.57460497
```

Figure 27. Result of OLS on Entire Dataset

Question 10

For each time period in Question 6 (with the same time window length), we performed the same grid search for GradientBoostingRegressor on the aggregate data. The best parameters found for each time period are shown below:

- Before Feb.1 8:00 a.m.

  {

  ‘max_depth’: 10,
  ‘max_features’: ‘auto’,
  ‘min_samples_leaf’: 1,
  ‘min_samples_split’: 5,
  ‘n_estimators’: 600

  }
- Between Feb.1 8:00 a.m. and Feb.1 8:00 p.m.

  {

  ‘max_depth’: 10,
  ‘max_features’: ‘sqrt’,
  ‘min_samples_leaf’: 1,
  ‘min_samples_split’: 10,
  ‘n_estimators’: 1200

}

- After Feb.1 8:00 p.m.

{

'max_depth': 10,
'max_features': 'sqrt',
'min_samples_leaf': 1,
'min_samples_split': 2,
'n_estimators': 800

}

The MSE and R-squared values for each time period of the models with best parameters found are shown below:

- Before Feb.1 8:00 a.m.
  MSE = -4078104.9024462765
  R-squared = 0.9999999999999855
- Between Feb.1 8:00 a.m. and Feb.1 8:00 p.m.
  MSE = -27688875.155442346
  R-squared =0.9999999999999992
- After Feb.1 8:00 p.m.
  MSE = -469967.5997977643
  R-squared = 0.9999999999999757

We can see that when we divide the total time into three time periods and trained models for each one on the aggregate data, the non-linear model (GradientBoostingRegressor) performed better than OLS (in Question 7). For each time period, the absolute value of MSE is smaller than that of OLS.

Best parameter set found is different across three time periods, also different from that found in Question 8, where we just trained one model for the whole time period. This again suggests that we need to treat each time period differently to make our model perform better.

Question 11

In this part, we tried to regress the data with MLPRegressor with 6 different structures by adjusting hidden_layer_sizes. The structures we tried are one layer of size 10, two layers of size 10, three layers of size 10, four layers of size 10, four layers of size 20 and five layers of size 10. For each of the structure, mean squared error (MSE) of fitting the entire data were calculated. The results are shown as below:

Table 24. MSE of MLPRegressor with different structures

Structure	MSE
(10)	40994919394.45372
(10, 10)	3488706555.6891546
(10, 10, 10)	1702431573.7536132
(10, 10, 10, 10)	5030049024.943863

(20, 20, 20, 20)	660028822.5281397
(10, 10, 10, 10, 10)	391732482498.2709

As a whole, we could see that more layers help to improve the results of MSE and so is bigger size for each layer. But the improvement seems more obvious with more layers than just raise the size of each layer.

Question 12

In this part, we used StandardScaler to scale the data before feeding it to the MLPRegressor. And here we used the structure with best results, which is five layers of size 10. The results are shown as below:

Table 25. MSE of MLPRegressor with or without scaling data before

Five layers of size 10	Scaled	non-Scaled
MSE	244827156.9917313	391732482498.2709

From the results above, we could see that scaling the data before feeding to the MLPRegressor does improve the performance.

Question 13

Here we used GridSearch to find out the best structure of MLPRegressor to each time period of the data (as described in Question 6). Here data were also scaled before feeding to the regressor. The results are shown as below:

Table 26. Best structure of MLPRegressor for each time period

Time period	Before	Between	After
Best structure	(10, 10, 10, 10, 10)	(20, 20, 20, 20)	(20, 20, 20, 20)

Question 14

Here we have more sample files already classified by different time period (as described in Question 6). We used the whole data sets before as the train set, while the new sample files as the test sets. And we used the data from all previous 6 hours to predict the next time window for making more accurate predictions. For each time period, we performed GridSearch on GradientBoostingRegressor, MLPRegressor, Linear Regression models with best parameters got in the previous questions, to find out the most suitable model. The value of MSE were used to evaluate the model. The most suitable models are shown as below:

Table 26. Best training model for each time period

Time Period	Before	Between	After
Best Model	GradientBoostingRegressor(max_depth = 10, max_features = 'sqrt', min_samples_leaf = 1, min_samples_split = 5, n_estimators = 1000)	GradientBoostingRegressor(max_depth = 100, max_features = 'sqrt', min_samples_leaf = 1, min_samples_split = 2, n_estimators = 400)	GradientBoostingRegressor(max_depth = 20, max_features = 'sqrt', min_samples_leaf = 4, min_samples_split = 5, n_estimators = 1000)

Then we used the models shown in Table 26 to predict the number of tweets in the next time window for each time period by using the new samples files as the test set. The predicted results are shown as below:

Table 27. The predicted results for the sample files

File name	Number of tweets in next time window
sample0_period1	367
sample0_period2	2307
sample0_period3	26
sample1_period1	938
sample1_period2	1632
sample1_period3	23
sample2_period1	247
sample2_period2	4069
sample2_period3	122

Question 15

Here we tried to predict the location of the author by the contexts of the tweet. But we only consider about the states of Washington and Massachusetts. First we need to filter out only the locations of Washington and Massachusetts and label them as 0 and 1 respectively, to make the ground truth label. To find out the location of Washington State, we used the method as below:

```
def is_WA(loc):
```

```
    loc = loc.lstrip().rstrip()
    low = loc.lower()
    if "seattle" in low:
        return True
    elif "WA" in loc:
        return True
    elif "washington d" in low:
        return False
    elif "washington" in low:
        return True
    else:
        specs = [",", ".", " ", "|", "/"]
        subs = ["wa", "wash"]
        for sub in subs:
            for spec in specs:
                if low.startswith(sub + spec):
                    return True
                elif low.endswith(spec + sub) or low.endswith(spec + sub + "."):
                    return True
                else:
                    for sp2 in specs:
                        if (spec + sub + sp2) in low:
                            return True
        return False
```

Then we lemmatized the words and used CountVectorizer, TfidfTransformer and TruncatedSVD to preprocess the words and reduce the word features to 50. And we fitted the data to SVM, Logistic Regression with L2 regularization and GaussianNB classifiers. Five-fold cross validation was used to find out the best parameter of each classifier. For each of the classifier, ROC curve was plotted and confusion matrix, accuracy, recall and precision were calculated to evaluate the performance. Here we still used the whole data set used before as train set, while the new sample files were considered as test sets. The results are shown as below:

(1) SVM

Best Parameter: C = 10

ROC curve:

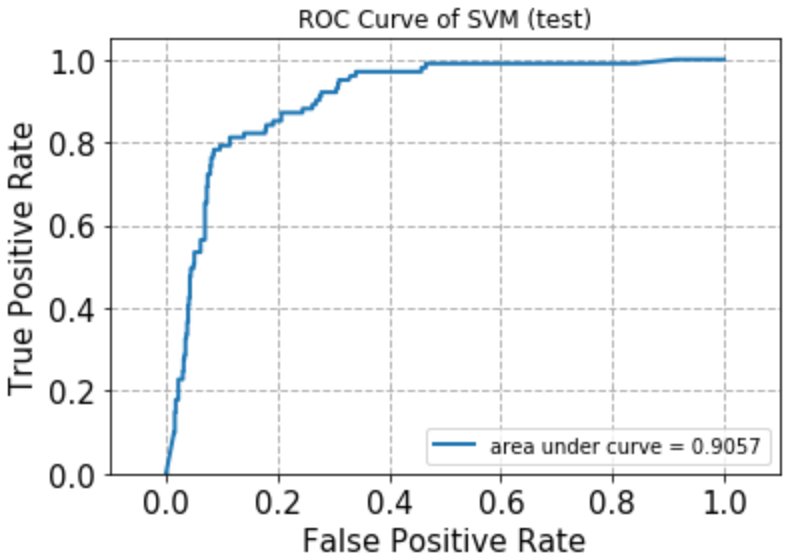

Figure 28. ROC curve of Linear SVC based on test set

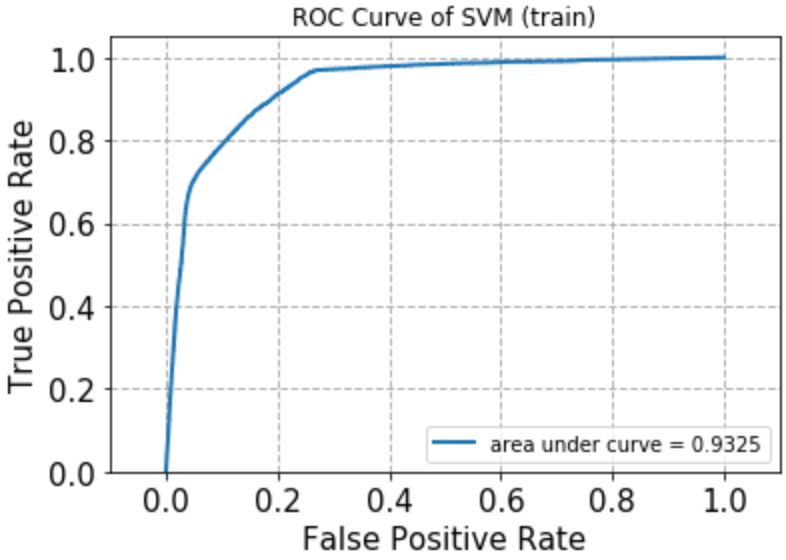

Figure 29. ROC curve of Linear SVC based on train set

Confusion Matrix:

Train set	[[118271 1908] [6291 3267]]
Test set	[[602 27] [58 43]]

Other evaluations:

Train set	Test set
Accuracy: 0.9368029166698782 Recall: 0.3418079096045198 Precision: 0.6313043478260869 F-1 Score: 0.4434941967012828	Accuracy: 0.8835616438356164 Recall: 0.42574257425742573 Precision: 0.6142857142857143 F-1 Score: 0.5029239766081872

(2) Logistic Regression with L2 Regularization
Best Parameter: C = 10
ROC curve:

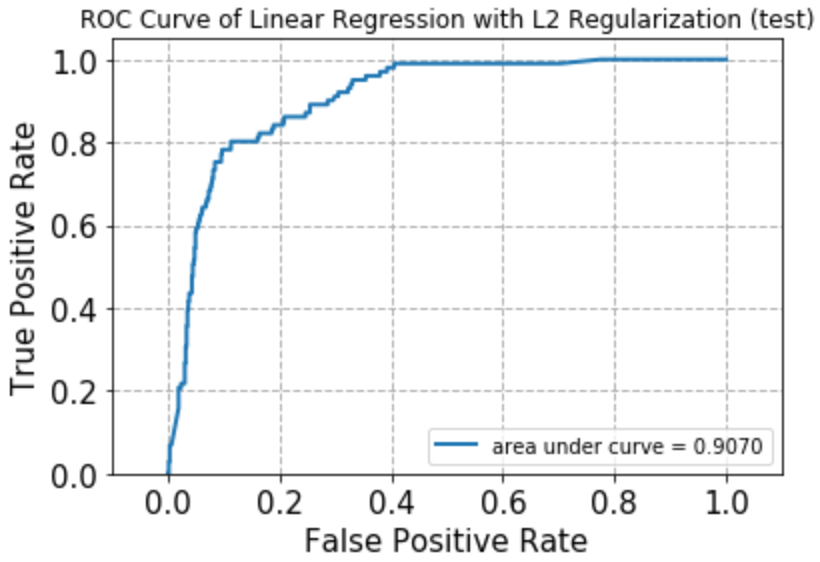

Figure 30. ROC curve of Logistic Regression based on test set

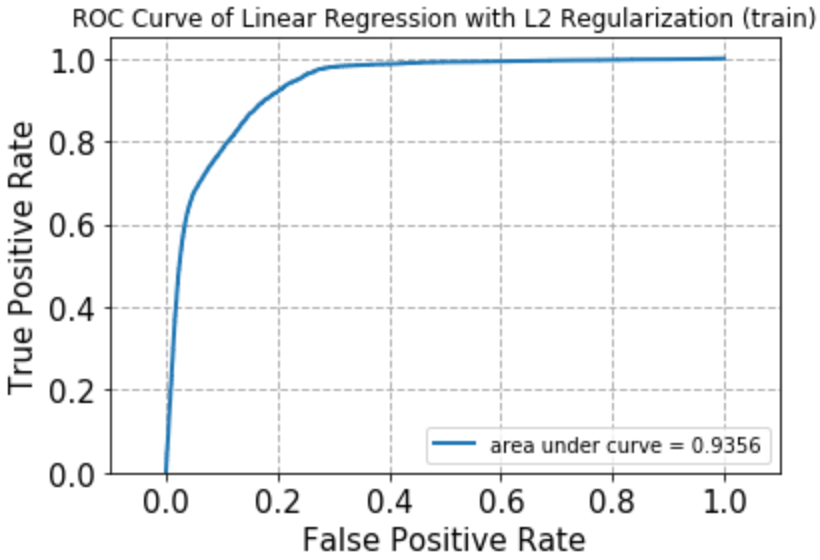

Figure 31. ROC curve of Logistic Regression based on train set

Confusion Matrix:

Train set	[[118297 1882] [6106 3452]]
Test set	[[602 27] [54 47]]

Other evaluations:

Train set	Test set
Accuracy: 0.9384292838588837 Recall: 0.361163423310316 Precision: 0.6471691038620172 F-1 Score: 0.46360461993016383	Accuracy: 0.8890410958904109 Recall: 0.46534653465346537 Precision: 0.6351351351351351 F-1 Score: 0.5371428571428571

(3) GaussianNB

ROC curve:

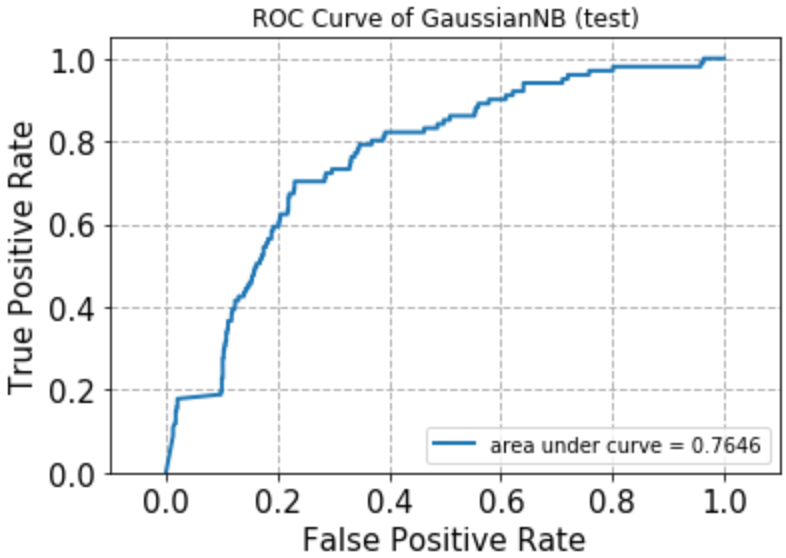

Figure 32. ROC curve of GaussianNB based on test set

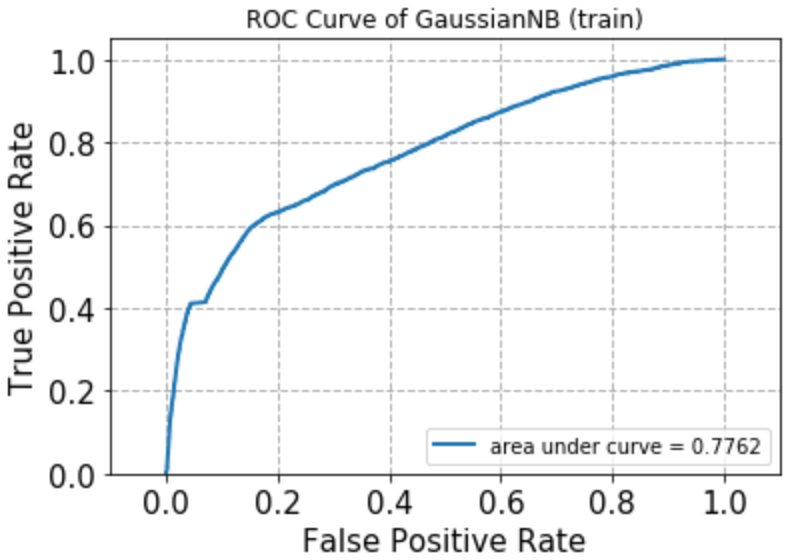

Figure 33. ROC curve of GaussianNB based on train set

Confusion Matrix:

Train set	[[85150 35029] [2968 6590]]
Test set	[[404 225] [21 80]]

Other evaluations:

Train set	Test set
Accuracy: 0.7071228716557343 Recall: 0.6894747855199833 Precision: 0.15834114226675317 F-1 Score: 0.2575375657033433	Accuracy: 0.663013698630137 Recall: 0.7920792079207921 Precision: 0.26229508196721313 F-1 Score: 0.39408866995073893

From the results above, we could see that the SVM classifier with C = 10 performed best to predict the location of the author in Washington or Massachusetts according to the context of the tweet.

Question 16

For this part, we did some sentiment analysis, and we also tried to find the most popular keywords among the tweets among the tweets of different teams' supporters. We proposed two questions given the tweet data as follows:

- How the sentiment changed during the Super Bowl for the fan of Seahawks and Patriots?

- What is the most popular words among the tweets of different teams' supporters?

By answering these two questions we might get some intuition about about whether there is relationship between the sentiment of different teams' supporters and the result of the 2015 Super Bowl as well as whether different teams supporter tends to have different popular keywords.

(a) How the sentiment changed during the Super Bowl for the fan of Seahawks and Patriots?

For this problem, we assumed that the tweets with the tag #gopatriots and #patriots are the supporters for the team Patriots and the tweets with the tag #gohawks are the supporters for the team SeaHawks. Then we first checked out how the sentiment changed during all the time (from two weeks before the match to a week after the match). The result for the SeaHawks supporters can be shown in Fig. 34 and the result for the Patriots supporters can be shown in Fig. 35.

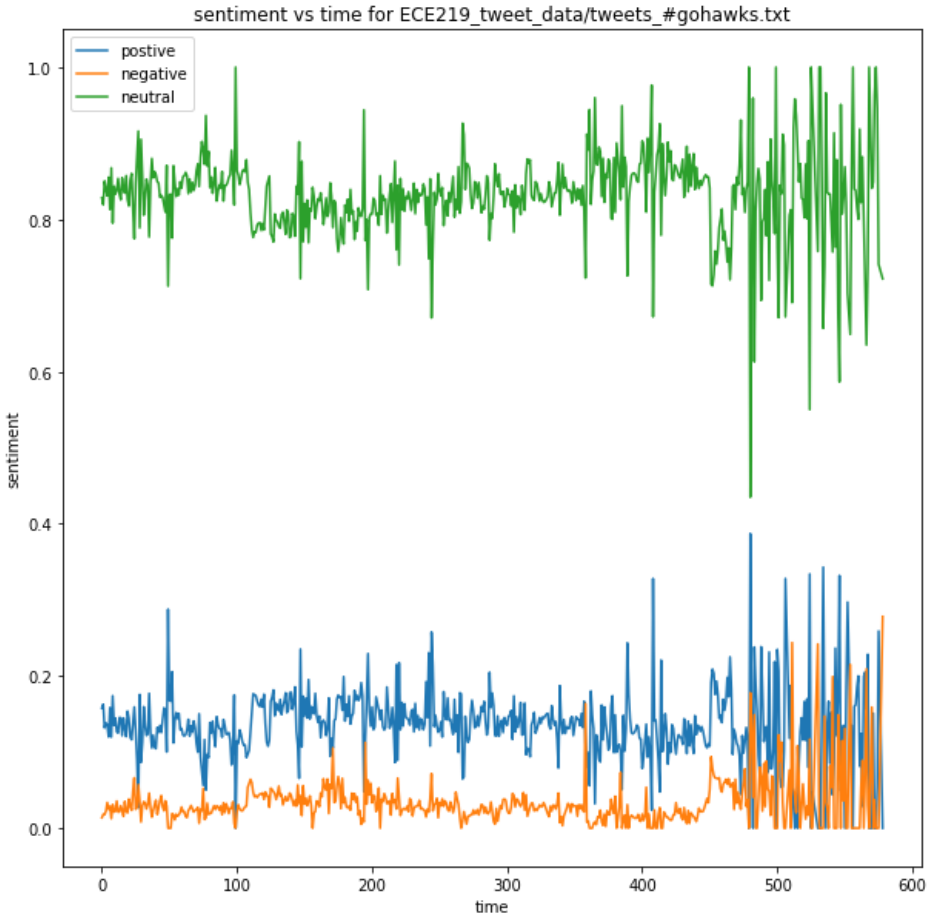

Figure 35. Sentiment Change for SeaHawks' Supporters' Tweet During All the Time (From Two Week Before the Match to a Week After the Match)

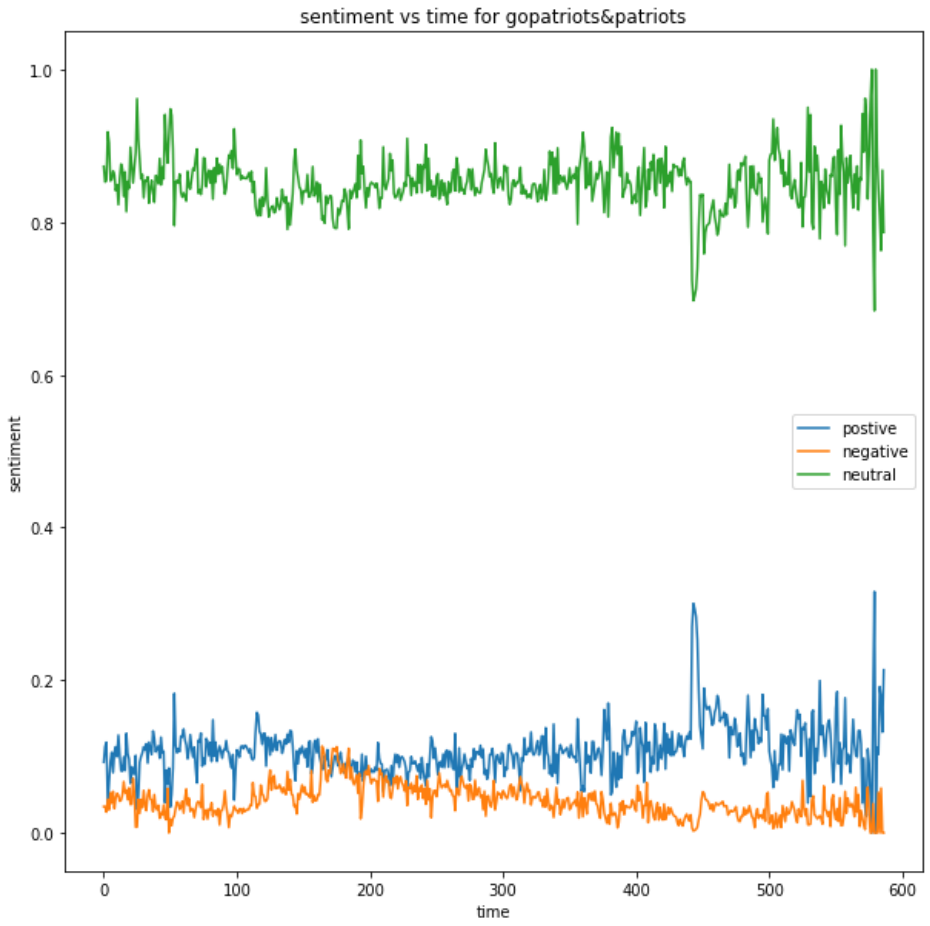

Figure 36. Sentiment Change for Patriots' Supporters' Tweet During All the Time (From Two Week Before the Match to a Week After the Match)

The result of Fig. 35 and Fig. 36 show how the sentiment changed during the time for different team based on their tweets. The x-axis represents how many hours have already passed from the earliest tweet (two weeks before the match). The y-axis represent the sentiment scores. The larger the "neutral" score. The more neutral the supporters' tweets are. The larger the "negative" score, the more passive the supporters' tweets are. The larger the "positive" score, the more positive the supporters' tweets are.

From both Fig.35 and Fig.36, it can be shown that after the match (about 450h), peoples from both team tends to post a non-neutral tweets. For the match, it can be shown that during the match (around 450h), there is a peek in the Fig. 36. This is reasonable because the Patriots won the 2015 Super Bowl, and therefore the positive tweets will increase during the match, while the negative tweets just increase a little as shown in Fig. 36. In contrast, the SeaHawks loss the 2015 Super Bowl, so that though there were a increase in the positive tweets but at the same time there are also a significant increase in the negative tweets.

To further confirm the conclusion, we looked at the how the sentiment changed during the day of Super Bowl (February 1, 2015). The result for the SeaHawks supporters can be shown in Fig. 36 and the result for the Patriots supporters can be shown in Fig. 37.

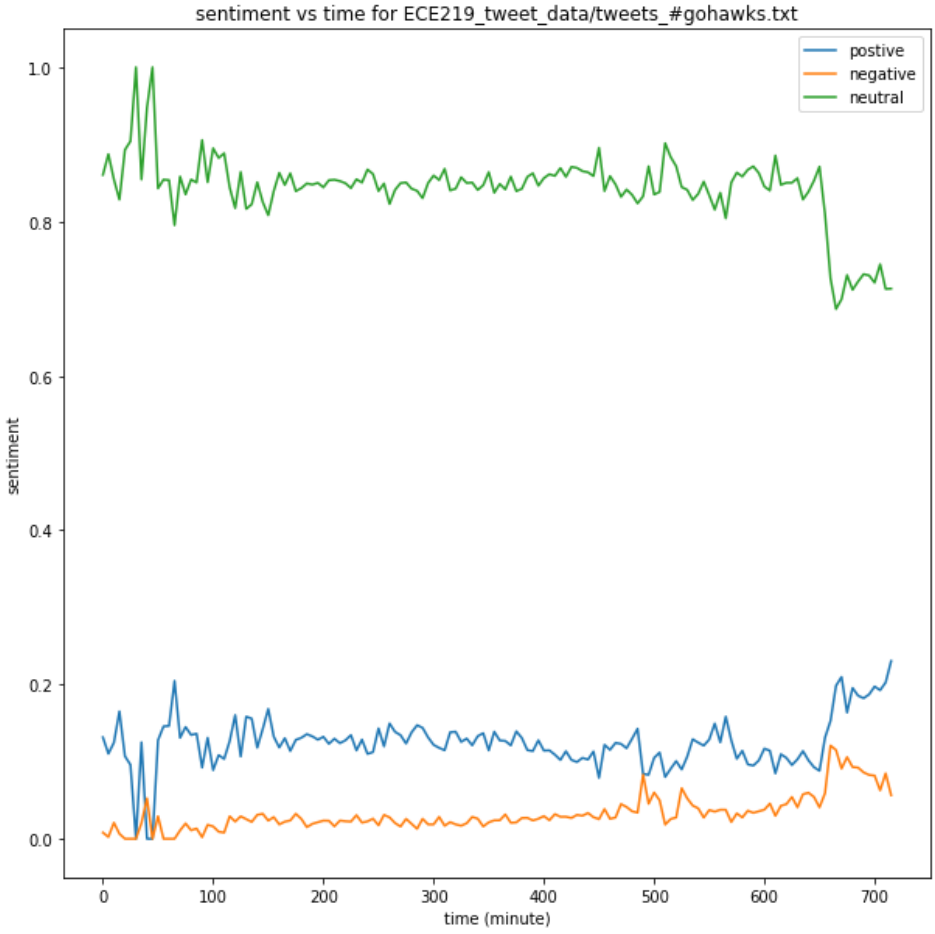

Figure 36. Sentiment Change for SeaHawks' Supporters' Tweet During the Day of Super Bowl (February 1, 2015)

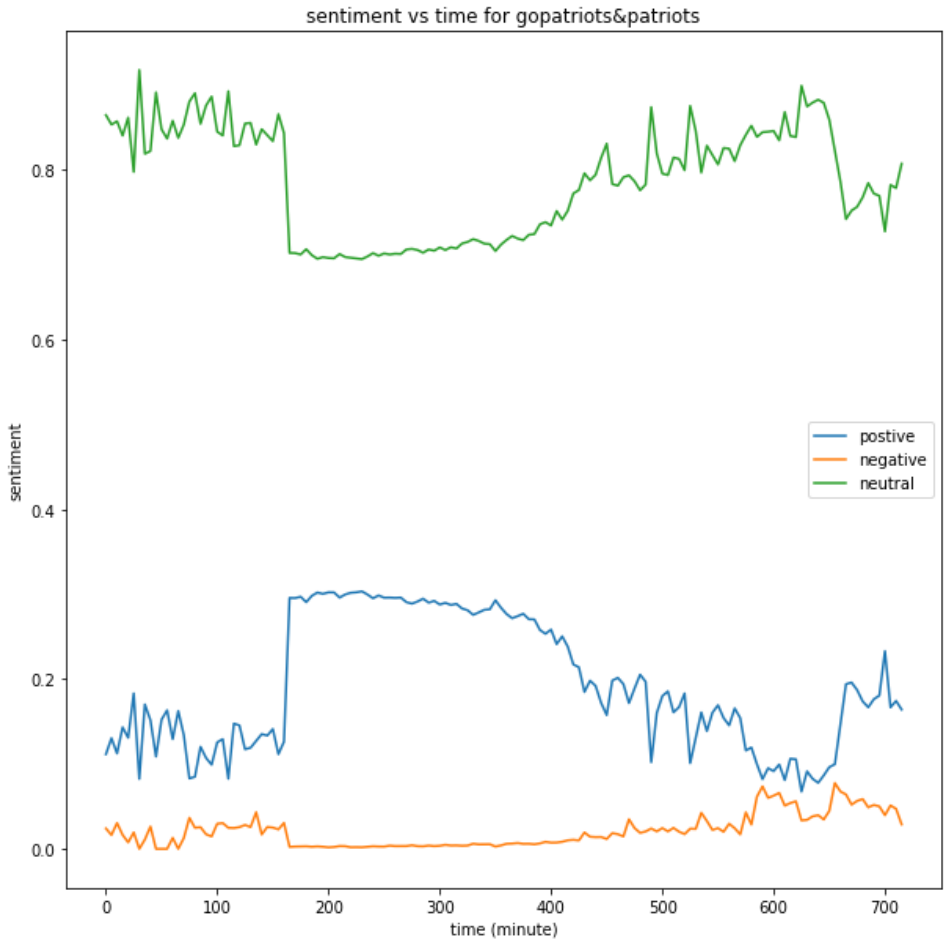

Figure 37. Sentiment Change for Patriots' Supporters' Tweet During the Day of Super Bowl (February 1, 2015)

The result of Fig. 36 and Fig. 37 show how the sentiment changed during the time for different team based on their tweets. The x-axis represents how many minutes have already passed from the Feb. 1st, 8:00 am. The y-axis represent the sentiment scores. The larger the "neutral" score. The more neutral the supporters' tweets are. The larger the "negative" score, the more passive the supporters' tweets are. The larger the "positive" score, the more positive the supporters' tweets are.

We knows that the Super Bowl started at 3:30 pm on Feb, 1st, which is 450 minutes away from Feb. 1st, 8:00 am. From Fig. 36 and Fig. 37, it can be seen that before the match started, the tweets of Patriots' supporters showed more positive sentiment than that of SeaHawks' supporters. Thus, it can be shown that Patriots' supporters have stronger morale.

However, from Fig.36, it shows that for the tweets of SeaHawks' supporter there were a significantly decrease of "positive" scores at 4:00 pm on Feb. 1st, and a significantly increase of "positive" scores around 5:00 pm on Feb. 1st. In contrast, from Fig. 37, it shows that the for the tweets of Patriots', the "positive" score is high until when it came to 5:00 pm on Feb. 1st. After that, the "positive" scores kept decrease until the very end of the game.

The sentiment change we talked previously is reasonable if we looked deeper into the match. The scoring summary can be shown in the Table 28.

Table 28. Scoring Summary of the Super Bowl XLIX

Quarter	Time	Team	Score	
			Patriots	Seahawks
2	9:47	Patriots	7	0
2	2:16	Seahawks	7	7
2	0:31	Patriots	14	7
2	0:02	Seahawks	14	14
3	11:09	Seahawks	14	17
3	4:54	Seahawks	14	24
4	7:55	Patriots	21	24
4	2:02	Patriots	28	24

From the scoring summary, it can be seen that before the third quarters, the Patriots always led the game, this is the reason why the tweet of Patriots' supporters show positive sentiment before 5:00 pm on Feb. 1st and the tweet of Seahawks supporters did not change much in "positive" score. However, when it came to the third quarters, with getting 10 points, the scores not only came back but also led the game. This can explain why there was a significant increase for "positive" score for Seahawks' supporters and that of Patriots' supporters kept decreasing. Then it was the fourth quarter. In the fourth quarter, until the last two minute of the game, the Patriots finally came back and won the game. Thus, at the end of the game, there were a significant increase of "positive" score for the Patriots' supporters.

Thus, for this question, it can be shown that from then sentiment change of different team's supporters, it is possible to learn how the match is going on. For example, we can know which team finally win the game, or which team is leading the team in a specific time period.

(b) What is the most popular words among the tweets of different teams' supporters?

For this problem, we also assumed that the tweets with the tag #gopatriots and #patriots are the supporters for the team Patriots and the tweets with the tag #gohawks are the supporters for the team SeaHawks. Then we used TfidfVectorizer to extract the most popular 50 words among the tweets of Seahawks' supporters and Patriots' supporters. We also checked the most popular 50 words among all the tweets in our dataset (include #gohawks, #gopatriots, #nfl, #patriots, #sb49, and #superbowl).

The result can be summarized into the following table.

Table 29. Summary of the Most Popular 50 Words

The most popular 50 words among the tweets of Seahawks' supporters	dict_keys(['http', 'ready', 'just', 'seahawks', '12s', '12thman', '12', 'superbowl', 'good', 'year', 'seattle', 'hawks', 'today', 'love', 'moneylynch', 'beastmode', 'amp', 'game', 'best', 'lob', 'repete', 'team', 'rt', 'right', 'weare12', 'nfl', 'day', 'man', 'going', 'let', 'dangerusswilson', 'time', 'fan', 'got', 'rsherman_25', 'great', 'touchdown', 'win', 'football', 'super', 'bowl', 'gbvssea', 'like', 'sunday', 'nfcchampionship', 'come', 'don', 'patriots', 'superbowlxlix', 'sb49'])
The most popular 50 words among the tweets of Patriots' supporters	dict_keys(['gopatriots', 'patriots', 'http', 'game', 'football', 'newengland', 'team', 'rt', 'win', 'super', 'bowl', 'today', 'going', 'time', 'nfl', 'tom', 'brady', 'just', 'don', 'vs', 'colts', 'pats', 'patsnation', 'like', 'seahawks', 'let', 'el', 'amp', 'fans', 'good', 'patriotsnation', 'gopats', 'winning', 've', 'got', 'superbowlxlix', 'seattle', 'new', 'england', 'tombrady', 'los', 'belichick', 'touchdown', 'superbowl', 'patriotsvsseahawks', 'sb49', 'deflategate', 'balls', 'deflated', 'patriotswin'])
The most popular 50 words among all the tweets in our dataset	dict_keys(['gohawks', 'http', 'just', 'seahawks', 'superbowl', 'good', 'year', 'new', 'seattle', 'love', 'amp', 'game', 'best', 'rt', 'nfl', 've', 'going', 'let', 'time', 'got', 'great', 'touchdown', 'win', 'football', 'watch', 'super', 'bowl', 'like', 'que', 'https', 'patriots', 'en', 'la', 'el', 'commercial', 'winning', 'colts', 'pats', 'superbowlxlix', 'tom', 'brady', 'sb49', 'commercials', 'perry', 'halftime', 'katyperry', 'katy', 'seahawkswin', 'superbowlcommercials', 'patriotswin'])

By removing the keyword in common, we can get table 30.

Table 30. Summary of the Most Popular 50 Words After Removing the Keywords in Common

The most popular 50 words among the tweets of Seahawks' supporters	['lob', 'gbvssea', 'dangerusswilson', 'right', 'beastmode', 'repete', 'nfcchampionship', 'weare12', 'moneylynch', 'day', 'sunday', '12s', '12thman', 'ready', 'rsherman_25', '12', 'come', 'fan', 'hawks', 'man']
The most popular 50 words among the tweets of Patriots' supporters	['patsnation', 'balls', 'england', 'newengland', 'deflategate', 'los', 'fans', 'vs', 'belichick', 'deflated', 'gopatriots', 'gopats', 'patriotsnation', 'tombrady', 'patriotsvsseahawks']
The most popular 50 words among all the tweets in our dataset	['katy', 'en', 'katyperry', 'la', 'watch', 'commercials', 'commercial', 'perry', 'que', 'https', 'gohawks', 'halftime', 'superbowlcommercials', 'seahawkswin']

By checking these distinct popular words, we can get some intuition about the match as follows:

- From Seahawks' most popular keywords, it can be shown "12" and "lob" have a special meaning in Seahawks. 'Dangerusswilson', 'lob', and "rsherman" are team members of Seahawks.
- From Patriots' most popular keywords, it can be shown that "'belichick'" and "'tombrady'" are team member of Patriots. We can also learn that, besides from #gopatriots and #patriots tags, the supporters' of Patriots also create some similar tag like "'patsnation'", "england", "newengland", "gopats", and so on.
- From the most popular 50 words among all the tweets in our dataset, we can know that Katy Perry was the singer for the halftime show and people really enjoyed watching it.

To sum up, by checking the most common word from different team, it can also imply some information about the event.