EE 219 Project 5
Popularity Prediction on Twitter
Winter 2018

Jui Chang 804506544
Wenyang Zhu 904947071
Xiaohan Wang 405033965
Yang Tang 505036001

Introduction

The public discussion attribute of Twitter provides a good platform to perform popularity prediction analysis. In this project, we collected available Twitter data by querying popular hashtags related to the 2015 Super Bowl ranged from 2 weeks before the game to a week after the game. Then we trained a regression model and created a predictor for new samples. The test data consists of tweets containing a hashtag in a specified time window. Next, we used our regression model to predict number of tweets containing the hashtag posted within one hour immediately following the given time window. In the last part, we defined our problem with the knowing data, made analysis and tried to implement our idea.

Part 1 Popularity Prediction

Problem 1.1

In this part, we downloaded the original training tweet data and calculate three statistics for each hashtag. Below are our statistics results for each hashtag:

Table 1. statistics results for each hashtag

Hashtag	Avg. tweets per hour	Avg. followers per tweets	Avg. retweets
#gohawks	325.372	2203.932	2.015
#gopatriots	45.695	1401.896	1.400
#nfl	441.323	4653.252	1.539
#patriots	834.556	3309.979	1.783
#sb49	1419.888	10267.317	2.511
#superbowl	2302.500	8858.975	2.388

The plots of “number of tweets in hour” over time for #SuperBowl and #NFL are as below:

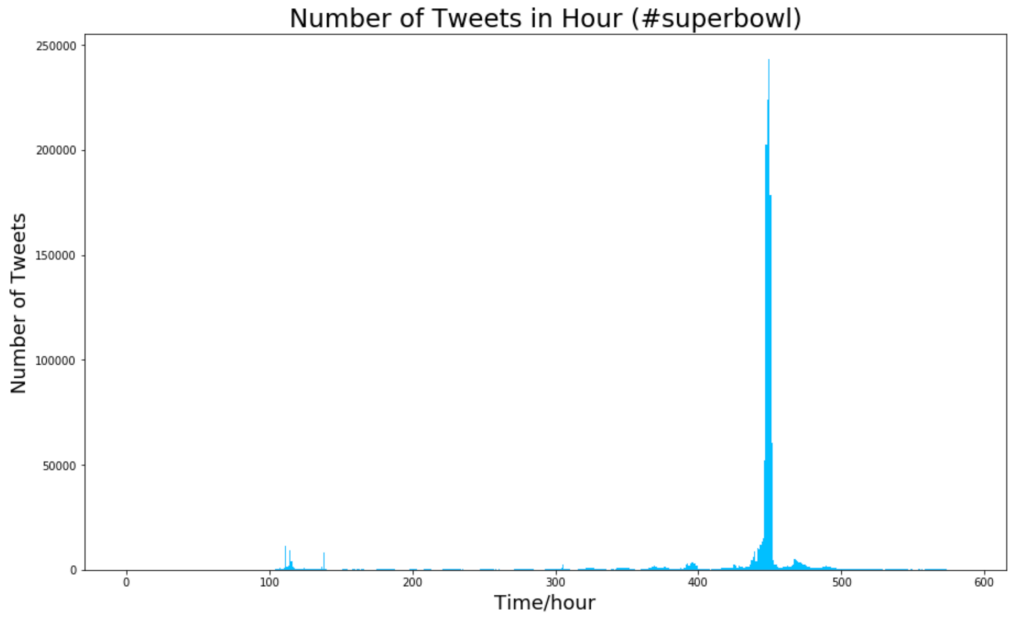

Figure 1. number of tweets in hour (#SuperBowl)

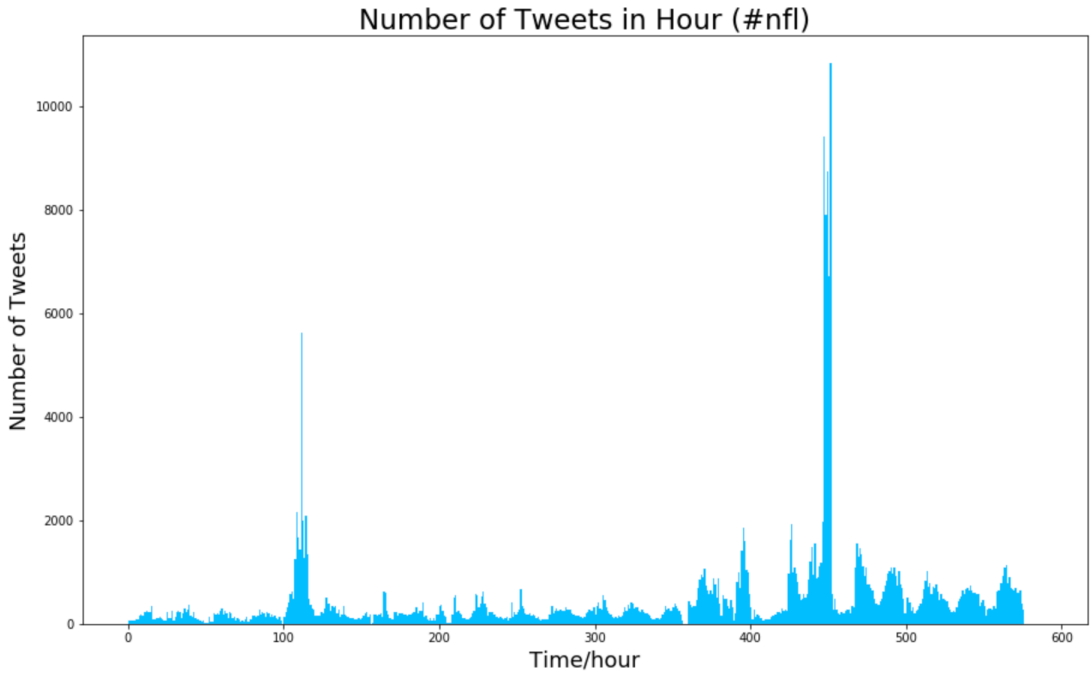

Figure 2. number of tweets in hour (#NFL)

From the above figures, we can find that there are two peaks in those two hashtags, one appeared in 110 hours, another appeared in 450 hours. So we can deduce that there were two big events happening at that time, during which people would engage more with the events and post more relative tweets.

Problem 1.2

In this part, we used the 5 features below in the previous hour to train and fit a Linear Regression model, and then predicted number of tweets in the next hour.

- Number of tweets
- Total number of retweets
- Sum of the number of followers of the users (tweets)
- Maximum number of followers of the users (tweets)
- Time of the day (scalar coding into 24 hours)

For each hashtag, first, we need to process the data. We converted the timestamp into readable date and time in Pacific time zone. Then grouped all data into 1-hour interval windows. Since the tweets might not be posted in all hours, we need to fill in the blank hour with all-zero values. Next, we used statsmodel.api to train and fit a Linear Regression model, and calculated RMSE and R-squared values of the model, and used t-test and P-value to analyze the significance of each feature. Below are our results:

Table 2. RMSEs and R-squared of Linear Regression model for each hashtag

Hashtag	RMSE	R-squared
#gohawks	972.50	0.473
#gopatriots	185.02	0.632
#nfl	581.42	0.564
#patriots	2526.28	0.670
#sb49	4470.45	0.805
#superbowl	8003.56	0.802

From the table above, we find that RMSEs are all very high and have no relationship with R-squared values. This is because that the number of tweets are very large, and the formula of RMSE is

$$RMSE = \sqrt{\frac{\sum_{i=1}^{n}(X_{obs,i} - X_{model,i})^2}{n}}$$

the value of RMSE is relative to the value of the original data. So we should not use RMSE to compare the performance of the model for each hashtag. R-squared is a number that indicates the proportion of the variance in the dependent variable that is predictable from independent variables. For Linear Regression model, the value of R-squared is the square of correlation coefficient of the sample. The formula is

Where SSE is the residual sum of squares, and SST is total sum of squares. The closer the R-squared to 1, the better the model is fitted. Thus, we can use R-squared to evaluate the accuracy of the model for each hashtag.

From the table, we can see that #sb49 and #superbowl both have relatively higher R-squared value close to 1, which indicates that the Linear Regression model fitted the data in these two hashtags. #gohawks has lowest R-squared, showing that the model or the features we chose is not suitable for its data. Below are the detailed model fittings:

1. #gohawks

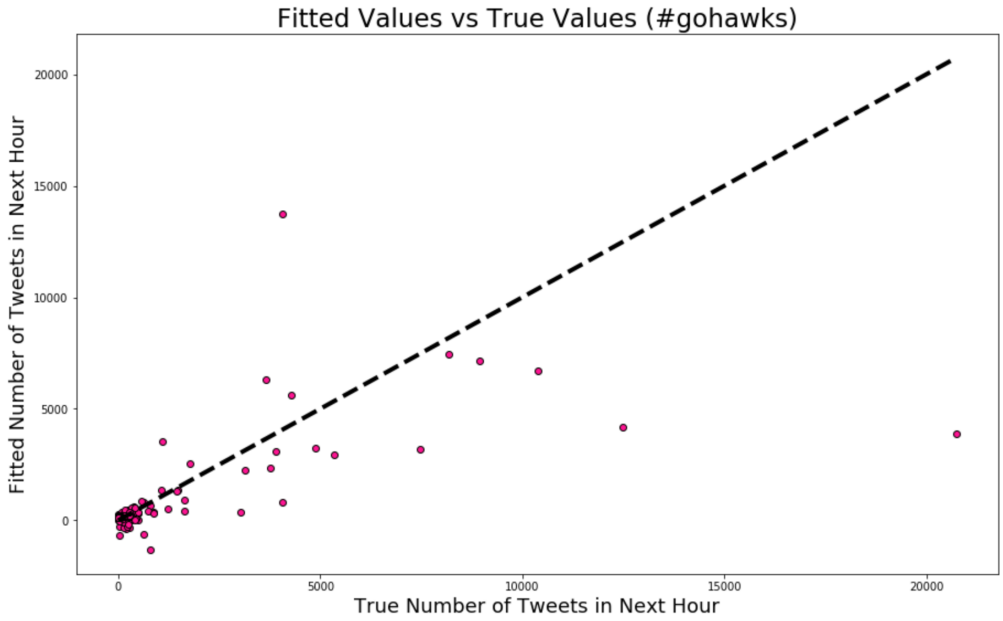

Figure 3. fitted values v.s. true values (#gohawks)

OLS Regression Results

Dep. Variable:	y	R-squared:	0.473
Model:	OLS	Adj. R-squared:	0.468
Method:	Least Squares	F-statistic:	102.6
Date:	Mon, 12 Mar 2018	Prob (F-statistic):	3.70e-77
Time:	15:06:52	Log-Likelihood:	-4796.7
No. Observations:	578	AIC:	9605.
Df Residuals:	572	BIC:	9632.
Df Model:	5		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	107.7373	78.781	1.368	0.172	-46.998	262.473
x1	2.0470	5.951	0.344	0.731	-9.642	13.736
x2	1.2305	0.170	7.250	0.000	0.897	1.564
x3	-0.1272	0.044	-2.886	0.004	-0.214	-0.041
x4	-0.0002	8.51e-05	-2.064	0.039	-0.000	-8.54e-06
x5	1.807e-05	0.000	0.112	0.911	-0.000	0.000

Omnibus:	916.664	Durbin-Watson:	2.222
Prob(Omnibus):	0.000	Jarque-Bera (JB):	789278.497
Skew:	8.688	Prob(JB):	0.00
Kurtosis:	183.197	Cond. No.	5.56e+06

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 5.56e+06. This might indicate that there are strong multicollinearity or other numerical problems.

Figure 4. OLS Regression Results (#gohawks)

2. #gopatriots

Figure 5. fitted values v.s. true values (#gopatriots)

OLS Regression Results

Dep. Variable:	y	R-squared:	0.632
Model:	OLS	Adj. R-squared:	0.629
Method:	Least Squares	F-statistic:	195.0
Date:	Mon, 12 Mar 2018	Prob (F-statistic):	9.76e-121
Time:	15:06:54	Log-Likelihood:	-3811.0
No. Observations:	574	AIC:	7634.
Df Residuals:	568	BIC:	7660.
Df Model:	5		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	10.9721	15.103	0.726	0.468	-18.692	40.637
x1	-0.1395	1.135	-0.123	0.902	-2.368	2.089
x2	-0.0802	0.255	-0.314	0.754	-0.582	0.421
x3	0.5083	0.223	2.282	0.023	0.071	0.946
x4	0.0002	0.000	1.237	0.217	-0.000	0.001
x5	-0.0004	0.000	-1.908	0.057	-0.001	1.12e-05

Omnibus:	510.963	Durbin-Watson:	1.953
Prob(Omnibus):	0.000	Jarque-Bera (JB):	301082.351
Skew:	2.789	Prob(JB):	0.00
Kurtosis:	115.061	Cond. No.	8.15e+05

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 8.15e+05. This might indicate that there are
strong multicollinearity or other numerical problems.

Figure 6. OLS Regression Results (#gopatriots)

3. #nfl

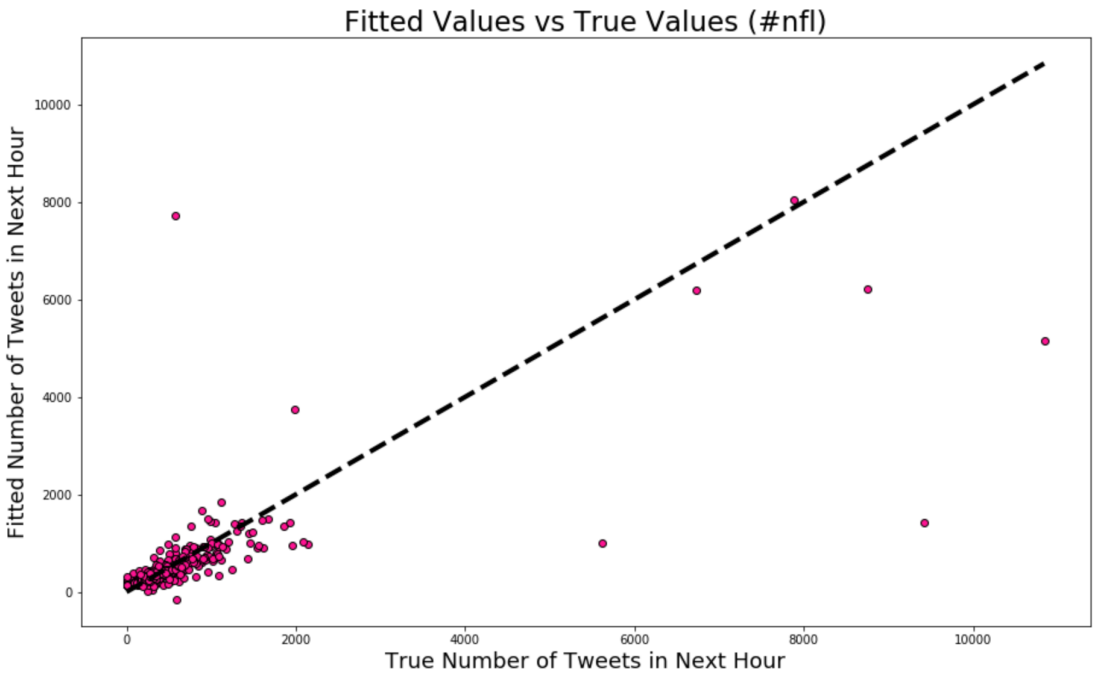

Figure 7. fitted values v.s. true values (#nfl)

OLS Regression Results

Dep. Variable:	y	R-squared:	0.564
Model:	OLS	Adj. R-squared:	0.561
Method:	Least Squares	F-statistic:	150.3
Date:	Mon, 12 Mar 2018	Prob (F-statistic):	3.60e-102
Time:	15:06:59	Log-Likelihood:	-4561.7
No. Observations:	586	AIC:	9135.
Df Residuals:	580	BIC:	9162.
Df Model:	5		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	131.0636	48.067	2.727	0.007	36.656	225.471
x1	-0.0188	3.528	-0.005	0.996	-6.949	6.911
x2	0.6843	0.134	5.103	0.000	0.421	0.948
x3	-0.1663	0.064	-2.606	0.009	-0.292	-0.041
x4	8.651e-05	2.63e-05	3.289	0.001	3.48e-05	0.000
x5	-9.073e-05	3.64e-05	-2.494	0.013	-0.000	-1.93e-05

Omnibus:	617.270	Durbin-Watson:	2.333
Prob(Omnibus):	0.000	Jarque-Bera (JB):	353726.203
Skew:	3.886	Prob(JB):	0.00
Kurtosis:	123.111	Cond. No.	9.36e+06

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 9.36e+06. This might indicate that there are
strong multicollinearity or other numerical problems.

Figure 8. OLS Regression Results (#nfl)

4. #patriots

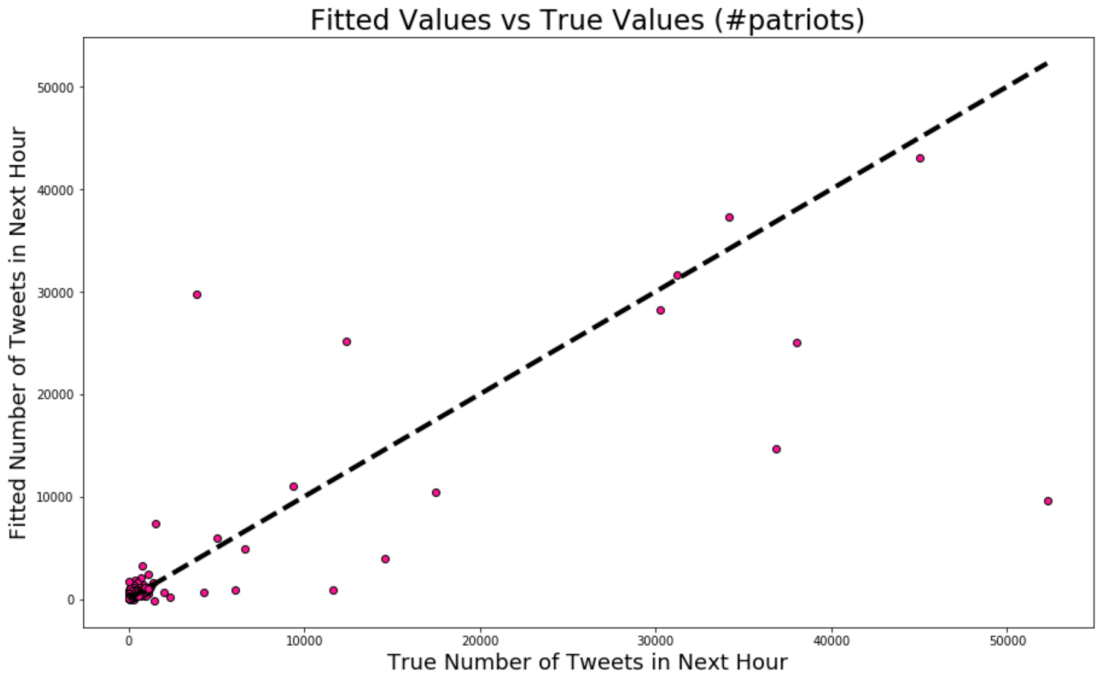

Figure 9. fitted values v.s. true values (#patriots)

OLS Regression Results

Dep. Variable:	y	R-squared:	0.670
Model:	OLS	Adj. R-squared:	0.667
Method:	Least Squares	F-statistic:	235.2
Date:	Mon, 12 Mar 2018	Prob (F-statistic):	6.42e-137
Time:	15:07:06	Log-Likelihood:	-5422.5
No. Observations:	586	AIC:	1.086e+04
Df Residuals:	580	BIC:	1.088e+04
Df Model:	5		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	177.4406	204.638	0.867	0.386	-224.481	579.362
x1	-6.9399	15.278	-0.454	0.650	-36.946	23.067
x2	0.9208	0.072	12.867	0.000	0.780	1.061
x3	-0.0876	0.059	-1.484	0.138	-0.203	0.028
x4	-3.749e-07	2.62e-05	-0.014	0.989	-5.19e-05	5.12e-05
x5	0.0002	0.000	1.615	0.107	-3.59e-05	0.000

Omnibus:	881.415	Durbin-Watson:	1.995
Prob(Omnibus):	0.000	Jarque-Bera (JB):	694137.509
Skew:	7.813	Prob(JB):	0.00
Kurtosis:	170.883	Cond. No.	1.80e+07

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.8e+07. This might indicate that there are
strong multicollinearity or other numerical problems.

Figure 10. OLS Regression Results (#patriots)

5. #sb49

Figure 11. fitted values v.s. true values (#sb49)

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:	Mon, 12 Mar 2018	Prob (F-statistic):	1.09e-201
Time:	15:07:21	Log-Likelihood:	-5717.7
No. Observations:	582	AIC:	1.145e+04
Df Residuals:	576	BIC:	1.147e+04
Df Model:	5		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	228.0606	365.278	0.624	0.533	-489.378	945.500
x1	-17.7694	27.150	-0.654	0.513	-71.094	35.556
x2	1.1878	0.095	12.478	0.000	1.001	1.375
x3	-0.2139	0.088	-2.437	0.015	-0.386	-0.042
x4	1.856e-05	1.4e-05	1.323	0.186	-9e-06	4.61e-05
x5	9.581e-05	4.8e-05	1.997	0.046	1.57e-06	0.000

Omnibus:	1183.066	Durbin-Watson:	1.683
Prob(Omnibus):	0.000	Jarque-Bera (JB):	2241201.699
Skew:	14.686	Prob(JB):	0.00
Kurtosis:	305.585	Cond. No.	1.73e+08

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.73e+08. This might indicate that there are
strong multicollinearity or other numerical problems.

Figure 12. OLS Regression Results (#sb49)

6. #superbowl

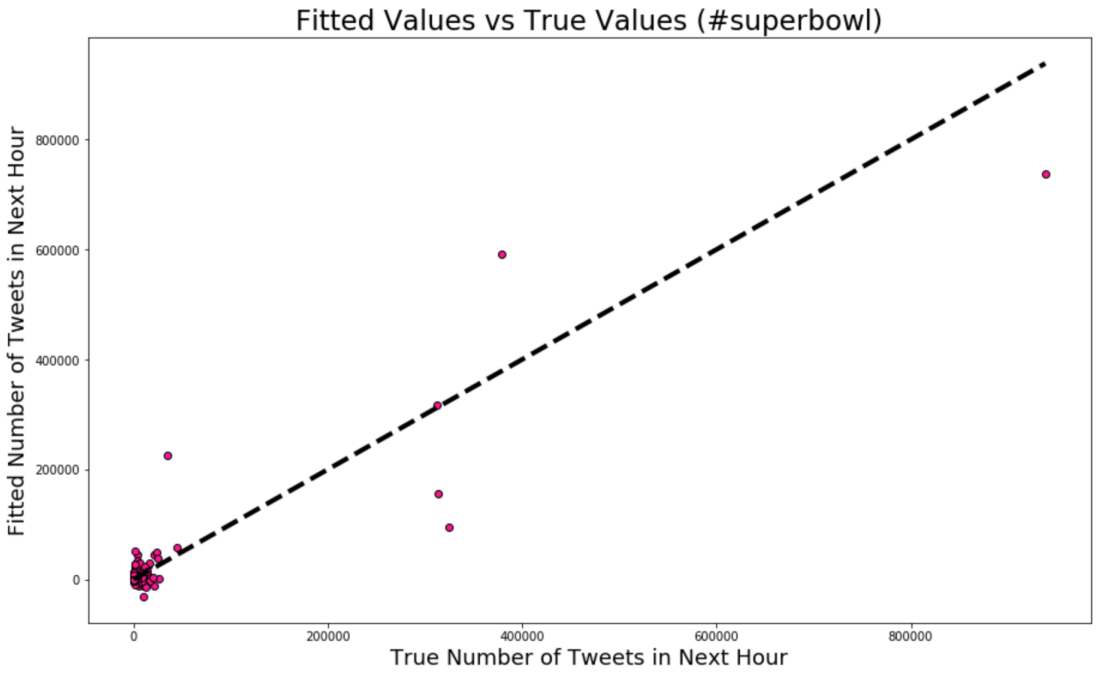

Figure 13. fitted values v.s. true values (#superbowl)

```
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.830
Model:                            OLS   Adj. R-squared:                  0.829
Method:                 Least Squares   F-statistic:                     566.4
Date:                Sun, 11 Mar 2018   Prob (F-statistic):          1.87e-220
Time:                        01:36:10   Log-Likelihood:                -6621.6
No. Observations:                 586   AIC:                         1.326e+04
Df Residuals:                     580   BIC:                         1.328e+04
Df Model:                           5
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
const      -1313.9553   1635.784     -0.803      0.422   -4526.737    1898.826
x1           -78.5801    117.790     -0.667      0.505    -309.927     152.767
x2             6.7886      0.194     34.906      0.000       6.407       7.171
x3            -0.5170      0.088     -5.878      0.000      -0.690      -0.344
x4            -0.0006   4.59e-05    -12.519      0.000      -0.001      -0.000
x5             0.0029      0.000      7.997      0.000       0.002       0.004
==============================================================================
Omnibus:                      402.688   Durbin-Watson:                   3.101
Prob(Omnibus):                  0.000   Jarque-Bera (JB):           222709.874
Skew:                           1.688   Prob(JB):                         0.00
Kurtosis:                      98.445   Cond. No.                     2.46e+08
==============================================================================

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 2.46e+08. This might indicate that there are
strong multicollinearity or other numerical problems.
```

Figure 14. OLS Regression Results (#superbowl)

Table 3. feature significance analysis

features	#gohawks	#gopatriots	#nfl	#patriots	#sb49	#superbowl
x1	0.731	0.902	0.996	0.650	0.513	0.613
x2	0.000	0.754	0.000	0.000	0.000	0.000
x3	0.004	0.023	0.009	0.138	0.015	0.000
x4	0.039	0.217	0.001	0.989	0.186	0.000
x5	0.911	0.057	0.013	0.107	0.046	0.000

Here, x1 – time of a day; x2 – number of tweets; x3 – total number of retweets; x4 – sum of the number of followers; x5 – maximum number of followers.

From OLS Regression Results, we find that features “number of tweets”, “total number of retweets”, “sum of the number of followers”, and “maximum number of followers” are almost close to zero in most hashtags, which indicates that those features made great contribution to linear regression model, and have high significance. While for feature “time of a day”, it’s very large to 1, indicating the less importance to our model.

Problem 1.3

From the last part, we can see that the fitting results are various for each hashtag. To better the fitting results, we delete some features that are not that significant and add 4 new features to our model. The total features are:

- Number of tweets
- Total number of retweets
- Sum of the number of followers of the users (tweets)
- Number of URLs
- Number of unique authors
- Number of mentions
- Total ranking score
- Number of hashtags

Below are our results:

Table 4. RMSEs and R-squared values compared with old models

Hashtag	RMSE		R-squared	
	5 features	8 features	5 features	8 features
#gohawks	972.50	795.76	0.473	0.647
#gopatriots	185.02	104.38	0.632	0.883
#nfl	581.42	453.08	0.564	0.735
#patriots	2526.28	1924.86	0.670	0.808
#sb49	4470.45	3927.35	0.805	0.849
#superbowl	8003.56	6164.33	0.802	0.883

From the above compare, we can find that with the new 8 features combination, our models have improved a lot by the R-squared values closer to 1, especially for #gopatriots and #superbowl hashtag, the R-squared is 0.883 which is very close to 1, indicating that the model fitted well for the dataset.

Below are the fitting values vs true values figures and the OLS analysis of each hashtag:

1. #gohawks

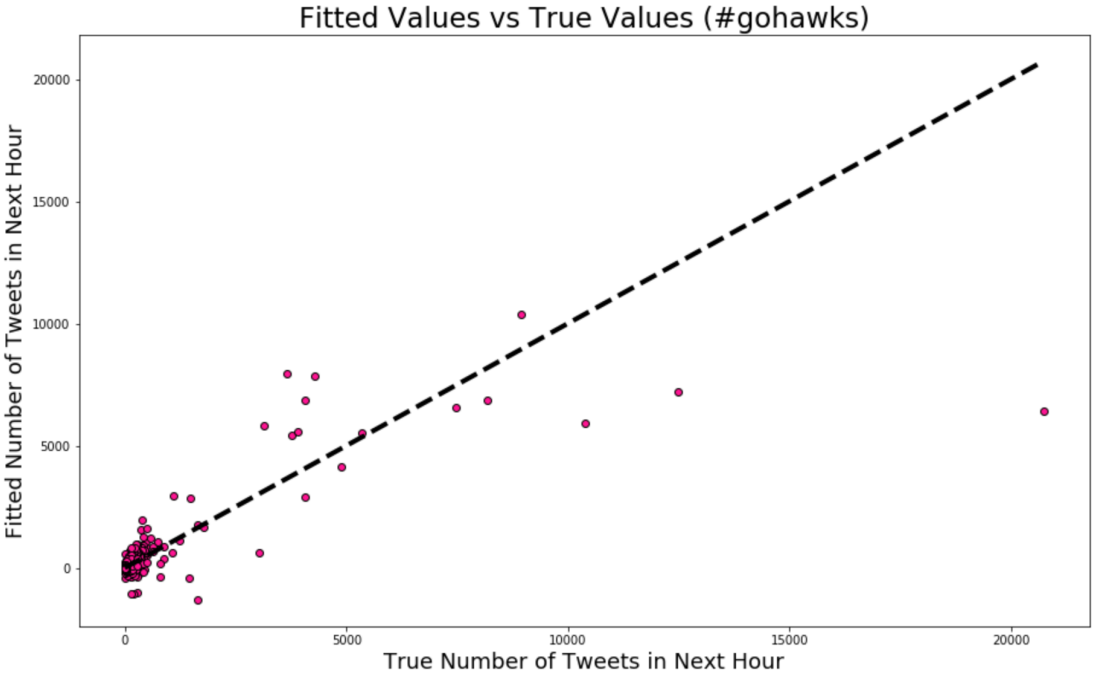

Figure 15. fitted values v.s. true values (#gohawks)

OLS Regression Results

Dep. Variable:	y	R-squared:	0.647
Model:	OLS	Adj. R-squared:	0.642
Method:	Least Squares	F-statistic:	130.4
Date:	Mon, 12 Mar 2018	Prob (F-statistic):	2.36e-123
Time:	15:07:55	Log-Likelihood:	-4680.8
No. Observations:	578	AIC:	9380.
Df Residuals:	569	BIC:	9419.
Df Model:	8		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-10.0099	39.476	-0.254	0.800	-87.547	67.527
x1	-51.3767	4.499	-11.419	0.000	-60.214	-42.540
x2	0.0535	0.039	1.368	0.172	-0.023	0.130
x3	-0.0004	5e-05	-7.633	0.000	-0.000	-0.000
x4	5.4772	1.531	3.578	0.000	2.470	8.484
x5	4.3376	0.787	5.509	0.000	2.791	5.884
x6	2.0695	0.491	4.214	0.000	1.105	3.034
x7	10.1967	0.904	11.278	0.000	8.421	11.972
x8	0.5319	0.349	1.526	0.128	-0.153	1.216

Omnibus:	994.761	Durbin-Watson:	2.142
Prob(Omnibus):	0.000	Jarque-Bera (JB):	845899.055
Skew:	10.395	Prob(JB):	0.00
Kurtosis:	189.257	Cond. No.	3.36e+06

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 3.36e+06. This might indicate that there are
strong multicollinearity or other numerical problems.

Figure 16. OLS Regression Results (#gohawks)

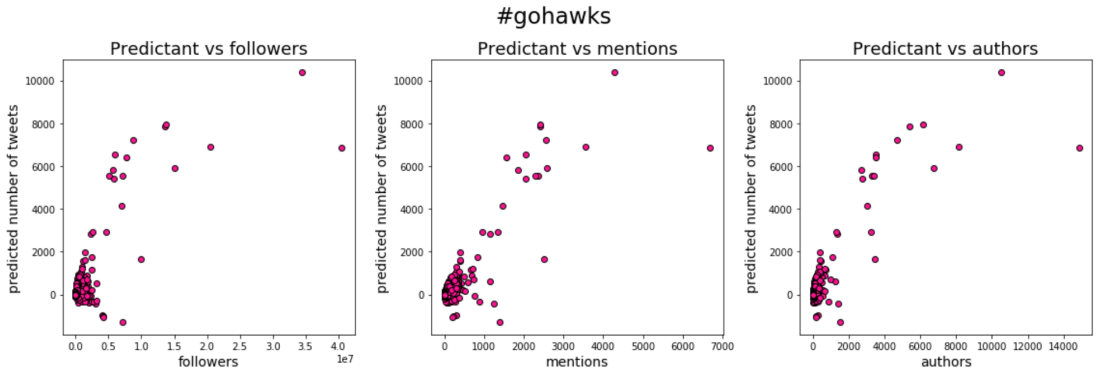

Figure 17. predicted number of tweets v.s. top 3 features(#gohawks)

2. #gopatriots

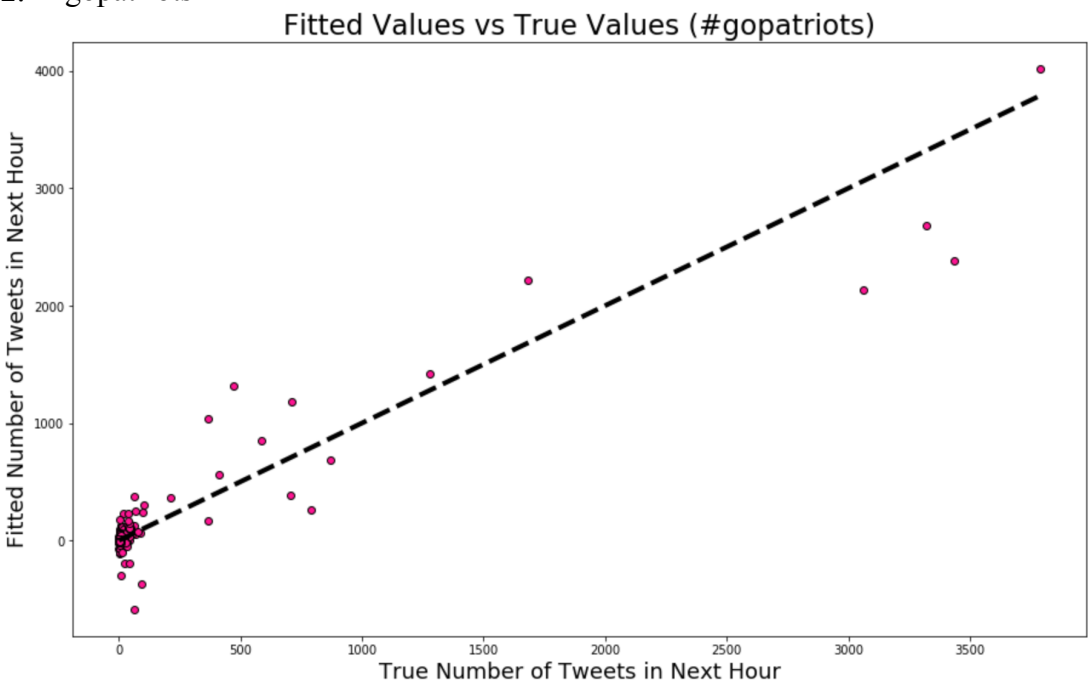

Figure 18. fitted values v.s. true values (#gopatriots)

OLS Regression Results

Dep. Variable:	y	R-squared:	0.883
Model:	OLS	Adj. R-squared:	0.881
Method:	Least Squares	F-statistic:	532.2
Date:	Mon, 12 Mar 2018	Prob (F-statistic):	2.26e-257
Time:	15:07:57	Log-Likelihood:	-3482.5
No. Observations:	574	AIC:	6983.
Df Residuals:	565	BIC:	7022.
Df Model:	8		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-7.1230	4.658	-1.529	0.127	-16.272	2.026
x1	-12.0147	1.964	-6.118	0.000	-15.872	-8.157
x2	-1.9233	0.151	-12.750	0.000	-2.220	-1.627
x3	2.896e-05	3.42e-05	0.846	0.398	-3.82e-05	9.62e-05
x4	9.3970	0.664	14.162	0.000	8.094	10.700
x5	-4.7258	0.375	-12.591	0.000	-5.463	-3.989
x6	5.7821	0.398	14.523	0.000	5.000	6.564
x7	3.1127	0.301	10.334	0.000	2.521	3.704
x8	1.0077	0.355	2.840	0.005	0.311	1.705

Omnibus:	426.854	Durbin-Watson:	1.994
Prob(Omnibus):	0.000	Jarque-Bera (JB):	51040.091
Skew:	2.414	Prob(JB):	0.00
Kurtosis:	48.943	Cond. No.	4.27e+05

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 4.27e+05. This might indicate that there are
strong multicollinearity or other numerical problems.

Figure 19. OLS Regression Results (#gopatriots)

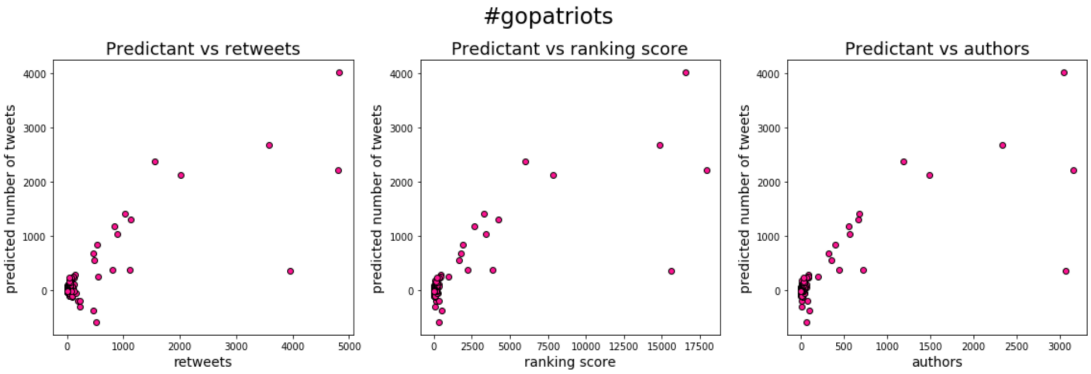

Figure 20. predicted number of tweets v.s. top 3 features(#gopatriots)

3. #nfl

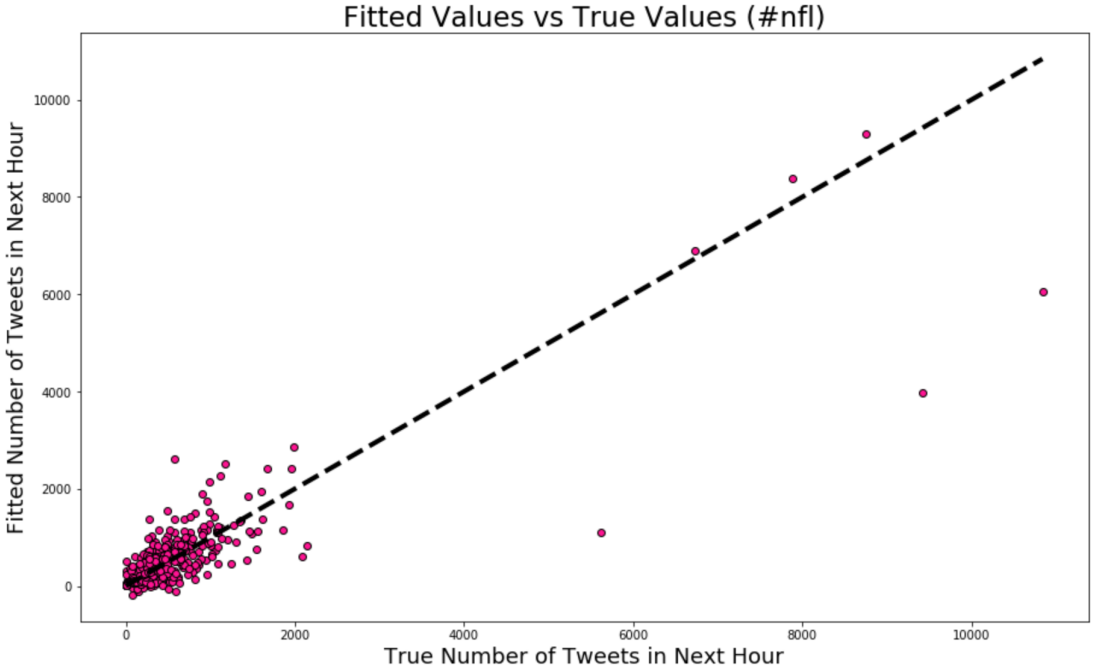

Figure 21. fitted values v.s. true values (#nfl)

OLS Regression Results

Dep. Variable:	y	R-squared:	0.735
Model:	OLS	Adj. R-squared:	0.732
Method:	Least Squares	F-statistic:	200.5
Date:	Mon, 12 Mar 2018	Prob (F-statistic):	4.10e-161
Time:	15:07:59	Log-Likelihood:	-4415.5
No. Observations:	586	AIC:	8849.
Df Residuals:	577	BIC:	8888.
Df Model:	8		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	18.6813	30.879	0.605	0.545	-41.968	79.331
x1	-3.3460	1.442	-2.320	0.021	-6.178	-0.514
x2	-0.0955	0.055	-1.736	0.083	-0.204	0.013
x3	-9.686e-06	1.2e-05	-0.807	0.420	-3.33e-05	1.39e-05
x4	0.0273	0.152	0.179	0.858	-0.272	0.327
x5	-3.5454	0.308	-11.493	0.000	-4.151	-2.940
x6	2.9945	0.551	5.438	0.000	1.913	4.076
x7	0.5986	0.302	1.982	0.048	0.005	1.192
x8	1.1345	0.084	13.515	0.000	0.970	1.299

Omnibus:	759.667	Durbin-Watson:	2.138
Prob(Omnibus):	0.000	Jarque-Bera (JB):	131708.939
Skew:	6.396	Prob(JB):	0.00
Kurtosis:	75.323	Cond. No.	7.52e+06

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 7.52e+06. This might indicate that there are
strong multicollinearity or other numerical problems.

Figure 22. OLS Regression Results (#nfl)

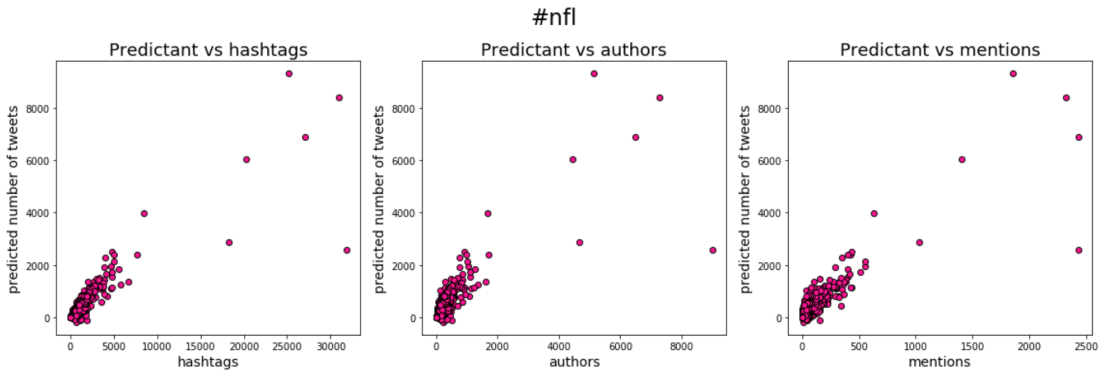

Figure 23. predicted number of tweets v.s. top 3 features(#nfl)

4. #patriots

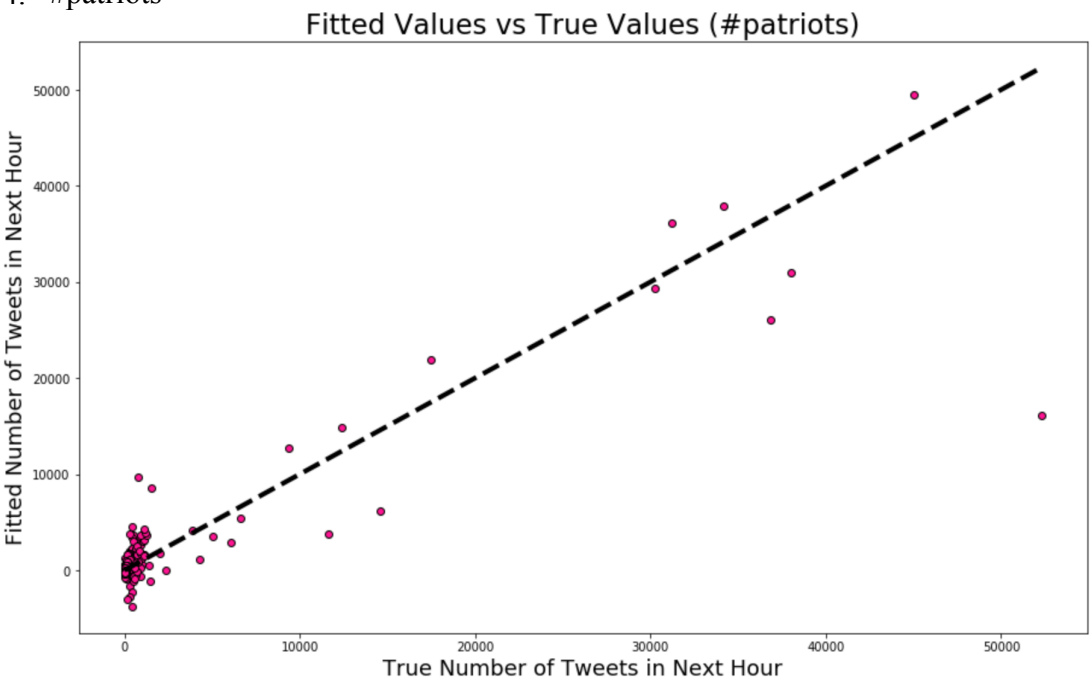

Figure 24. fitted values v.s. true values (#patriots)

OLS Regression Results

Dep. Variable:	y	R-squared:	0.808
Model:	OLS	Adj. R-squared:	0.806
Method:	Least Squares	F-statistic:	304.1
Date:	Mon, 12 Mar 2018	Prob (F-statistic):	2.41e-201
Time:	15:08:02	Log-Likelihood:	-5263.1
No. Observations:	586	AIC:	1.054e+04
Df Residuals:	577	BIC:	1.058e+04
Df Model:	8		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-288.8739	105.919	-2.727	0.007	-496.908	-80.840
x1	-63.9103	4.720	-13.542	0.000	-73.180	-54.641
x2	-0.2570	0.047	-5.418	0.000	-0.350	-0.164
x3	3.201e-05	3.58e-05	0.895	0.371	-3.82e-05	0.000
x4	-4.3191	1.648	-2.622	0.009	-7.555	-1.083
x5	0.5897	0.858	0.687	0.492	-1.096	2.275
x6	6.5548	0.871	7.522	0.000	4.843	8.266
x7	11.8253	0.891	13.279	0.000	10.076	13.574
x8	3.7054	0.375	9.893	0.000	2.970	4.441

Omnibus:	1065.015	Durbin-Watson:	1.809
Prob(Omnibus):	0.000	Jarque-Bera (JB):	1149400.779
Skew:	11.630	Prob(JB):	0.00
Kurtosis:	218.716	Cond. No.	1.21e+07

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.21e+07. This might indicate that there are
strong multicollinearity or other numerical problems.

Figure 25. OLS Regression Results (#patriots)

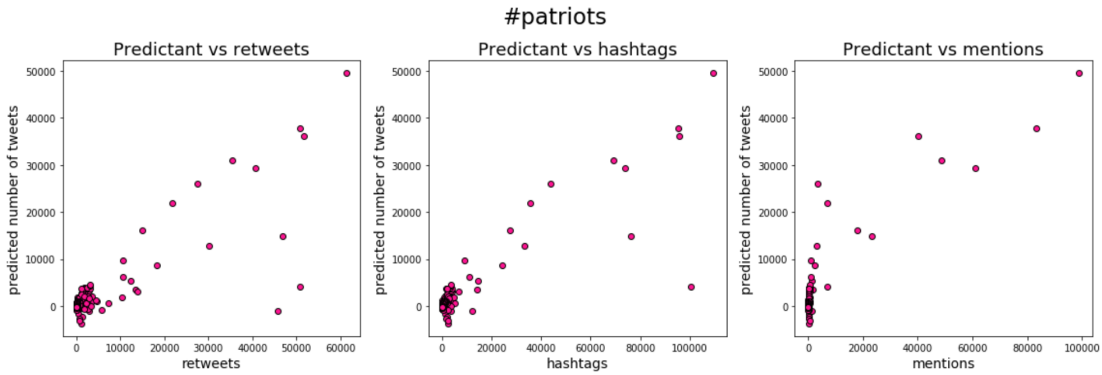

Figure 26. predicted number of tweets v.s. top 3 features(#patriots)

5. #sb49

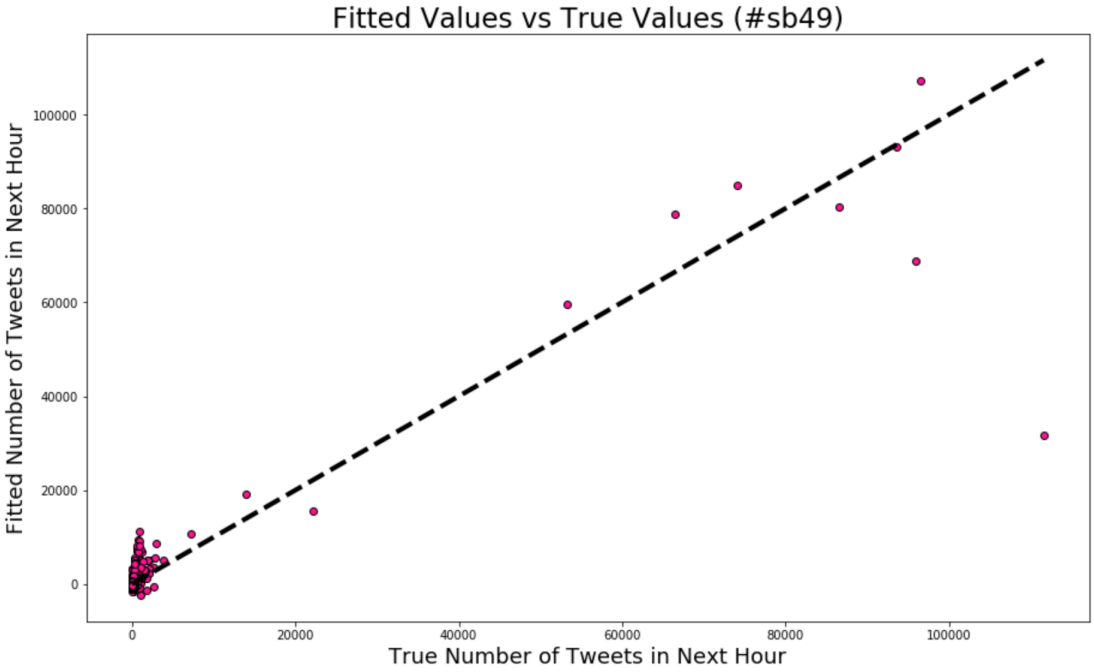

Figure 27. fitted values v.s. true values (#sb49)

OLS Regression Results

Dep. Variable:	y	R-squared:	0.849
Model:	OLS	Adj. R-squared:	0.847
Method:	Least Squares	F-statistic:	404.0
Date:	Mon, 12 Mar 2018	Prob (F-statistic):	6.81e-230
Time:	15:08:10	Log-Likelihood:	-5642.3
No. Observations:	582	AIC:	1.130e+04
Df Residuals:	573	BIC:	1.134e+04
Df Model:	8		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-485.2193	178.873	-2.713	0.007	-836.545	-133.893
x1	-48.1131	8.309	-5.790	0.000	-64.433	-31.793
x2	0.3794	0.098	3.863	0.000	0.186	0.572
x3	0.0001	1.64e-05	6.414	0.000	7.29e-05	0.000
x4	-2.3131	1.332	-1.736	0.083	-4.930	0.304
x5	-2.5243	0.898	-2.810	0.005	-4.289	-0.760
x6	6.0693	0.682	8.898	0.000	4.730	7.409
x7	8.7332	1.727	5.058	0.000	5.342	12.125
x8	2.6960	0.381	7.069	0.000	1.947	3.445

Omnibus:	1185.879	Durbin-Watson:	1.901
Prob(Omnibus):	0.000	Jarque-Bera (JB):	2172531.684
Skew:	14.790	Prob(JB):	0.00
Kurtosis:	300.849	Cond. No.	9.58e+07

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 9.58e+07. This might indicate that there are
strong multicollinearity or other numerical problems.

Figure 28. OLS Regression Results (#sb49)

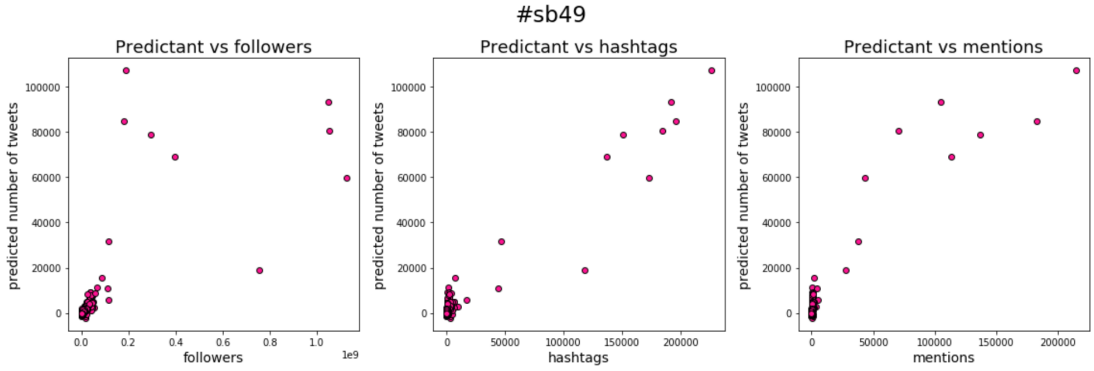

Figure 29. predicted number of tweets v.s. top 3 features(#sb49)

6. #superbowl

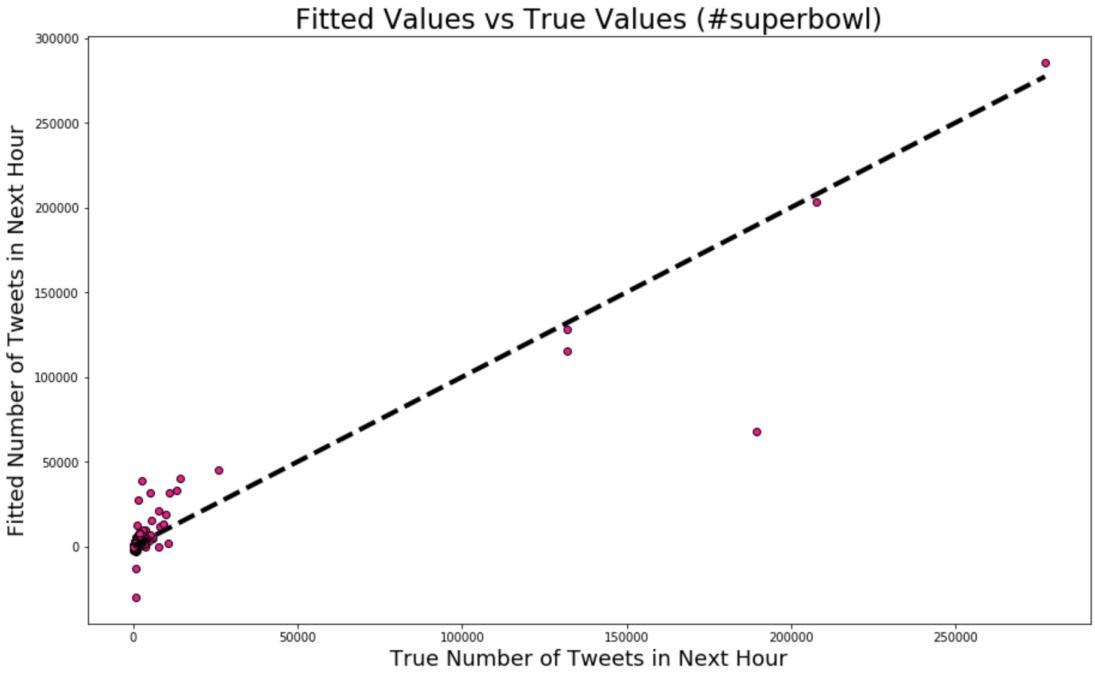

Figure 30. fitted values v.s. true values (#superbowl)

OLS Regression Results

Dep. Variable:	y	R-squared:	0.883
Model:	OLS	Adj. R-squared:	0.881
Method:	Least Squares	F-statistic:	542.5
Date:	Mon, 12 Mar 2018	Prob (F-statistic):	9.75e-263
Time:	15:37:38	Log-Likelihood:	-5945.2
No. Observations:	586	AIC:	1.191e+04
Df Residuals:	577	BIC:	1.195e+04
Df Model:	8		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-652.4337	302.118	-2.160	0.031	-1245.818	-59.050
x1	-41.8699	7.004	-5.978	0.000	-55.627	-28.113
x2	-0.6643	0.085	-7.780	0.000	-0.832	-0.497
x3	-5.44e-05	1.7e-05	-3.200	0.001	-8.78e-05	-2.1e-05
x4	-4.0067	1.342	-2.986	0.003	-6.642	-1.372
x5	0.8540	0.688	1.242	0.215	-0.496	2.205
x6	5.8267	1.929	3.021	0.003	2.038	9.615
x7	8.1056	1.432	5.661	0.000	5.294	10.918
x8	2.9944	0.451	6.642	0.000	2.109	3.880

Omnibus:	1100.036	Durbin-Watson:	1.906
Prob(Omnibus):	0.000	Jarque-Bera (JB):	1651497.222
Skew:	12.330	Prob(JB):	0.00
Kurtosis:	261.902	Cond. No.	1.44e+08

Warnings:
[1] Standard Errors assume that the covariance matrix of the errors is correctly specified.
[2] The condition number is large, 1.44e+08. This might indicate that there are
strong multicollinearity or other numerical problems.

Figure 31. OLS Regression Results (#superbowl)

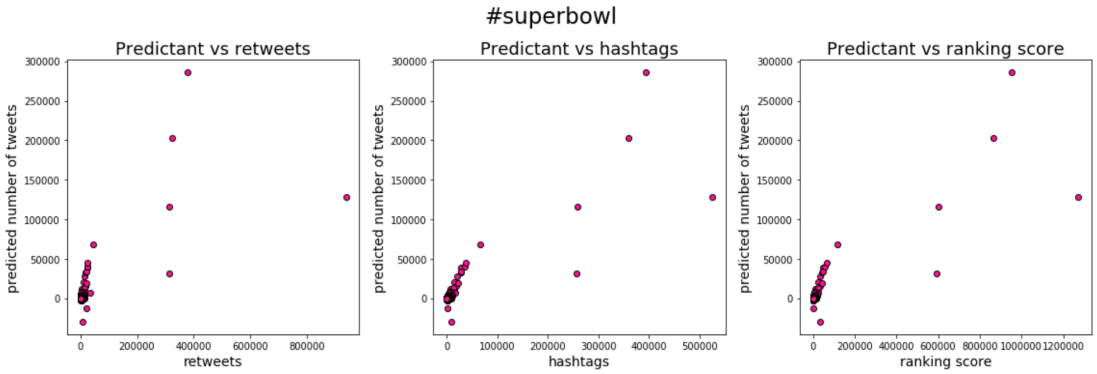

Figure 32. predicted number of tweets v.s. top 3 features(#superbowl)

From the figures above, we can see the improvement of the model visually and clearly. Except a few points, the trends of the scatter points are linearly distributed, which shows that the linear regression model is reasonable and suitable. Among those six hashtags, the predict results of #gohawks are relatively worse than others, while the linear regression model of #superbowl achieves the best performance.

Among 8 features, “number of retweets”, “number of hashtags”, “number of URLs”, “sum of followers”, “ranking score”, etc. have strong significance to the model. Besides, the effect of 8-feature model performs better than the effect of 5-feature model. This is because with the increment of the dimension of the training model, the target value can be predicted from more aspects, which makes the performances more accurate.

Problem 1.4

Problem 1.4 – 1

In this part, we used 10-fold cross-validation on the models same as that in the last part (8 features in 1-hour window). First, we split the data into three periods in each hashtag: before SuperBowl, in SuperBowl and after SuperBowl. Next, in each period, we tried to train 3 models (Linear Regression Model, Neural Network Model and SVM) and evaluate its performance with 10-fold cross-validation and got the average absolute error . Below are our results:

Table 5. average cross-validation errors (each hashtag)

#gohawks	Linear Regression	Neural Network	SVM
Before	375.531934	8314.903891	255.007611
Between	5027.973001	180303.544592	6778.950000
After	25.505607	357.586270	32.091667
#gopatriots	**Linear Regression**	**Neural Network**	**SVM**
Before	17.744468	182.638253	14.457082
Between	569.792956	73859.194300	2007.050000
After	2.749368	28.267953	4.732692
#nfl	**Linear Regression**	**Neural Network**	**SVM**
Before	119.813958	5042.333790	189.861681
Between	5224.187969	308582.744264	6023.100000
After	108.621711	87831.278276	592.968681
#patriots	**Linear Regression**	**Neural Network**	**SVM**
Before	252.338981	13027.094352	291.941173
Between	92370.687362	198878.623067	27055.250000
After	65.207328	6103.759251	149.131868
#sb49	**Linear Regression**	**Neural Network**	**SVM**
Before	46.745358	4.718992e+03	106.868182
Between	93722.963115	3.788425e+06	50906.600000
After	99.323845	4.614280e+05	323.721978

#superbowl	Linear Regression	Neural Network	SVM
Before	362.654305	6.997877e+04	451.030761
Between	264251.105849	3.357778e+06	183828.150000
After	167.808674	9.122700e+05	857.338462

From the tables above, we can find that the absolute errors during SuperBowl period (second period) are very large compared to those in the other two periods. There are mainly 3 reasons. First, there are only 10 hours' data in the second period, which is much smaller than the other two periods of about hundreds of data. The smaller the dataset is, the less accurate the result would be. Second, during the SuperBowl period, there would be a burst of tweets which greatly increase the difficulty to find the patterns. Finally, the results are absolute error instead of relative error. There is a burst of tweets during the event time, which contributes to a large base amount. So the absolute error would be large even though the relative error may be small.

Besides, we can conclude the best model for each hashtag during each period as below:

Table 6. the best fitted model for each hashtag during each period

	Before	Between	After
#gohawks	SVM	Linear Regression	Linear Regression
#gopatriots	SVM	Linear Regression	Linear Regression
#nfl	Linear Regression	Linear Regression	Linear Regression
#patriots	Linear Regression	SVM	Linear Regression
#sb49	Linear Regression	SVM	Linear Regression
#superbowl	Linear Regression	SVM	Linear Regression

From the table, we find that Linear Regression model performs the best. And sometimes, SVM can also achieve good performance. While Neural Network model performs much worse than the other two models.

Problem 1.4 – 2

In this part, we aggregated the data of all hashtags, then used the same model as the previous part to predict the number of tweets in the next hour on the aggregated data. Below are the average absolute errors on the aggregated data comparing with the separated data in each hashtag:

Table 7. average cross-validation errors (aggregated data)

#combine	Linear Regression	Neural Network	SVM
Before	732.172941	8.306819e+04	1460.007611
Between	160531.399686	7.754142e+06	201545.000000
After	436.631025	1.421777e+06	8042.515934

From the above table, we can see that Linear Regression model performs better for all periods. Compared with models trained for individual hashtags, the absolute errors are generally higher for aggregated dataset. This indicates that the patterns in each hashtag are different and we should analyze them individually.

Problem 1.5

In this part, instead of predicting the number of tweets on the train data, we did on the test data. There are 10 test datasets and each file contains a hashtag's tweets for a 6-hour window based on "firstpost_date".

To make more accurate predictions, we used 5-hour window instead of 1-hour window. First, we chose the best model for each period in 1.4-2. Next, we used aggregated dataset to train and fit the model. The features are as below:

- Number of tweets
- Total number of retweets
- Sum of the number of followers of the users (tweets)
- Number of URLs
- Number of unique authors
- Number of mentions
- Total ranking score

This time, we deleted "number of hashtags" feature because they are all 0. Since we used 5-hour window, the total trained features would be . Then we predicted the number of tweets in the 6th hour in each test file using the previous 5 hours' data with the model corresponding to their period.

There is one thing to be noticed that for sample8_period1, the data spans only in 5 hours, so we need to train two kinds of model for period 1, one with 5-hour window (features) and another with 4-hour window (). According to the results in the last part, the models for each period are all Linear Regression model. Below are our predictions and evaluations for each test file:

Table 8. predicted number of tweets on test data (all Linear Regression)

test file	predicted	true value	relative error
sample1_period1	-576.333912	178.0	4.237831
sample2_period2	36342.457437	82923.0	0.561732
sample3_period3	854.793986	523.0	0.634405
sample4_period1	231.368785	201.0	0.151088
sample5_period1	1093.155754	213.0	4.132187
sample6_period2	60845.643856	37307.0	0.630944
sample7_period3	94.144711	120.0	0.215461
sample8_period1	-150.774339	11.0	14.706758
sample9_period2	1560.533447	2790.0	0.440669
sample10_period3	84.020009	61.0	0.377377

From table 8, we find that Linear Regression model performs good for period 2 and 3. While for some samples with period 1, the relative errors are large. Besides, it appears some negative results which are impossible in theory. So the model cannot fit the data during period 1. Next, we change model in period 1 to SVM and the results are as below:

Table 9. predicted number of tweets on test data (period 1 with SVM)

test file	predicted	true value	relative error
sample1_period1	868.000000	178.0	3.876404
sample2_period2	36342.457437	82923.0	0.561732
sample3_period3	854.793986	523.0	0.634405
sample4_period1	868.000000	201.0	3.318408
sample5_period1	868.000000	213.0	3.075117
sample6_period2	60845.643856	37307.0	0.630944
sample7_period3	94.144711	120.0	0.215461
sample8_period1	868.000000	11.0	77.909091
sample9_period2	1560.533447	2790.0	0.440669
sample10_period3	84.020009	61.0	0.377377

After substituting Linear Regression model with SVM in period 1, the wrong results have been eliminated. While the relative errors in this period are still very large. It indicates that the patterns of each hashtag in period 1 are very complicate and we better analyze them individually.

Part 2 Fan Base Prediction

The textual content of a tweet can reveal some information about the author. Recognizing that supporting a sport team has a lot to do with the user location, in this part, we used the textual content of the tweet posted by a user to predict their location.

Specifically, we used #superbowl hashtag to do the binary classification. The classes are two states of the US, “Washington” and “Massachusetts”. We tried three classification method: SVM, Naïve Bayes Algorithms and Logistic Regression with l2 norm regularization. Then plotted the ROC curve, confusion matrix and calculated the accuracy, precision and recall of our model.

The brief processing steps are as below:

Step 1: extracted useful data from raw #superbowl file. The input data is the title of the tweet, and the output data should be the location of the user who posted this tweet. Thus, we need to extract the “title” and the “location” (in user object of tweet object) attributes. We mapped location to two labels, Washington as 1 and Massachusetts as -1, and threw out other unspecific locations. To be noticed, when matching with regular expression, we need to eliminate those with “Washington DC” or “D.C.”.

```
total number of tweets: 1348767
extracted titles: 32931
extracted locations: 32931
```

Step 2: feature extractions. We used Term Frequency-Inverse Document Frequency (TFxIDF) matrix to capture the importance of a word to a title. Then we use Latent Semantic Indexing (LSI) to find the optimal representation of the data in a lower dimensional space. We use TruncatedSVD to decompose the vectors with 100 as the number of elements. Therefore, we get the selected features for our learning algorithms.

Step 3: split train and test dataset. We use train_test_split method to split the whole data into train part and test part with 90%-10% percentage.

Step 4: train three classifiers and predict the test result. Evaluate and compare their performance.

2.1 Classification with SVM

1. ROC Curve

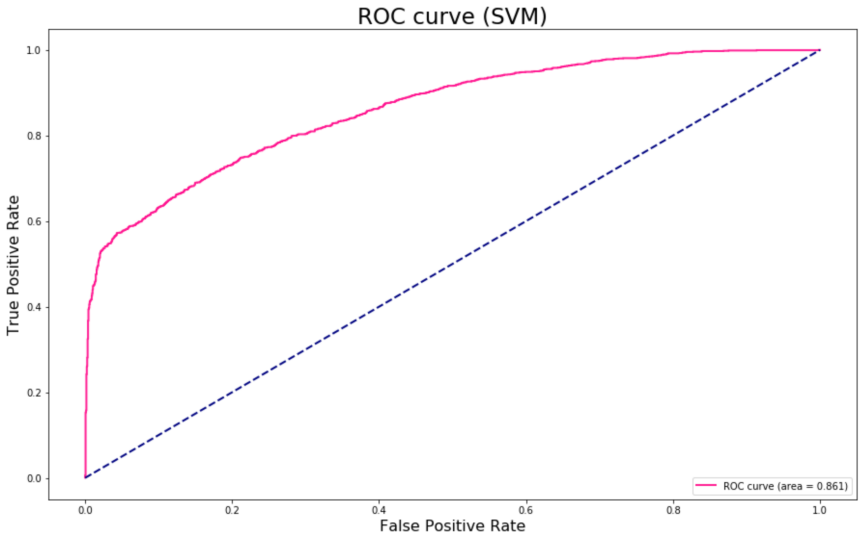

Figure 33. ROC curve with SVM classifier

2. Confusion Matrix

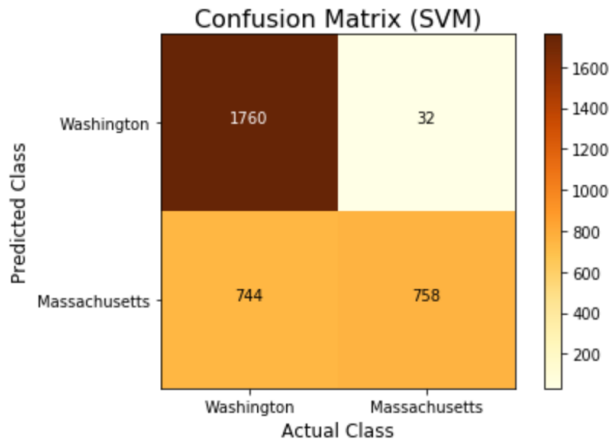

Figure 34. confusion matrix with SVM classifier

3. Accuracy, precision and recall

Table 10. accuracy, precision and recall with SVM classifier

classifier	accuracy	precision	Recall
SVM	0.764420157863	0.959493670886	0.50466045273

2.2 Classification with Naïve Bayes Algorithms

1. ROC curve

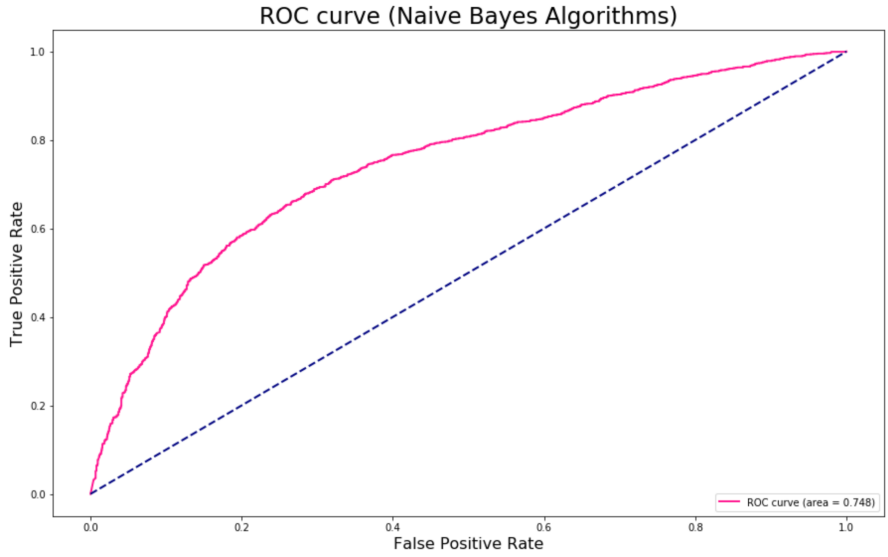

Figure 35. ROC curve with Naïve Bayes classifier

2. Confusion matrix

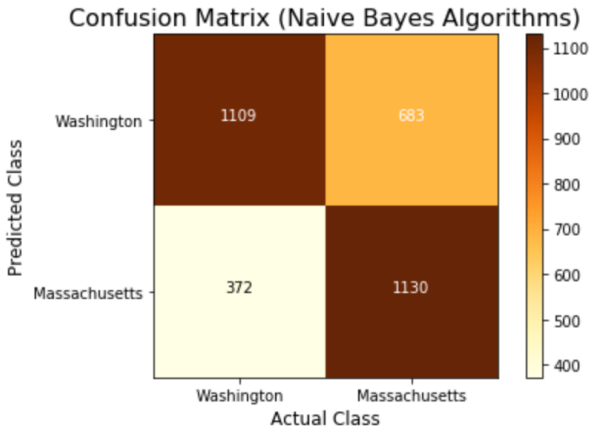

Figure 36. confusion matrix with Naïve Bayes classifier

3. Accuracy, precision and recall

Table 11. accuracy, precision and recall with Naïve Bayes classifier

classifier	accuracy	precision	Recall
Naïve Bayes	0.679720704311	0.623276337562	0.752330226365

2.3 Classification with Logistic Regression

1. ROC curve

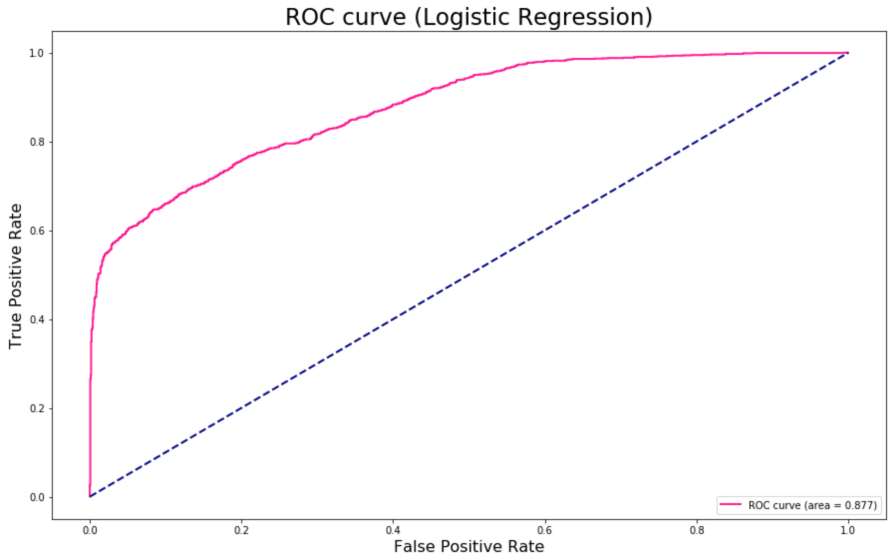

Figure 37. ROC curve with Logistic Regression classifier

2. Confusion matrix

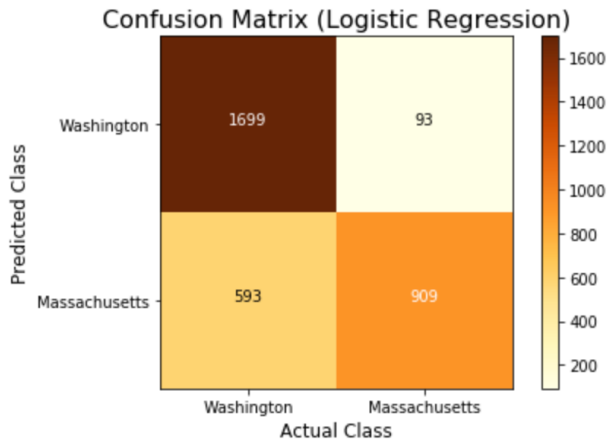

Figure 38. confusion matrix with Logistic Regression classifier

3. Accuracy, precision and recall

Table 12. accuracy, precision and recall with Logistic Regression classifier

classifier	accuracy	precision	Recall
Logistic Regression	0.791742562234	0.907185628743	0.605193075899

2.4 Compare the Performances

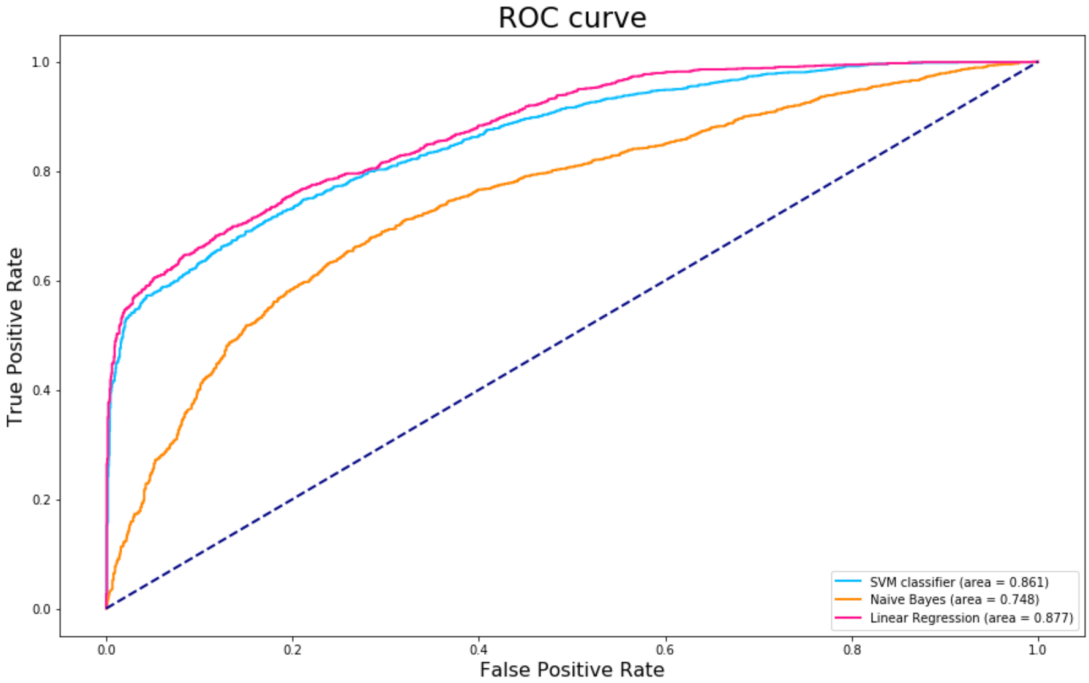

Figure 39. ROC curve for three classifiers

Table 13. accuracy, precision and recall for three classifiers

classifier	accuracy	precision	recall
SVM	0.764420	0.959494	0.504660
Naive Bayes Algorithms	0.679721	0.623276	0.752330
Logistic Regression	0.791743	0.907186	0.605193

From above results, we can see that Logistic Regression has the highest accuracy among three methods. SVM has the best precision values and Naïve Bayes Algorithms has the highest recall. And for ROC curve, Linear Regression classifier has highest AUC area which indicates a stronger ability of classification. In real cases, we need to choose the classification method based on the needs.

Part 3 Define Your Own Project: Sentiment Analysis of Opponent team

In this part, I do sentiment analysis of the two fans of hawks and patriots in the match, so I select the time period from February 1st, 2015, from 1pm(PST) and last for 8 hours, in which there are most twitter comments because the game is in that period.

From the below figure, we can observe in timestamp 10 to 36, there are most people twittering and hash tag two teams, which means that is the hot period people mention two teams, and it approximately equal to the super bowl gaming time, which is 6:30 pm and last for 3hr 37 min in 2015 matching, which is timestamp 15 to 36.

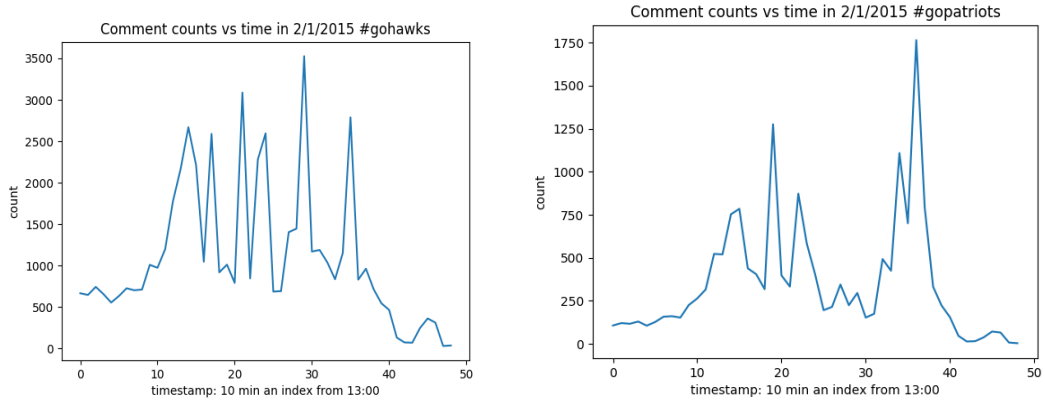

Figure 40. Comment count vs timestamps

To do sentiment analysis, I first clean the comment by delete url, hastag, symbol, stopwords, and stemmed words. Then, I use built-in function nltk.sentiment.vader.SentimentIntensityAnalyzer. I see each comment as an unit, if the positive portion is more than negative portion, I judge it as positive sentiment and set positive sentiment as 1 and negative sentiment as 0, vice versa, and if in one comment, both positive and negative portion is 0, I set both positive and negative sentiment as 0.

For each time period, I take average of all positive/negative sentiment as the representation of that time period, and the result is shown in below figures:

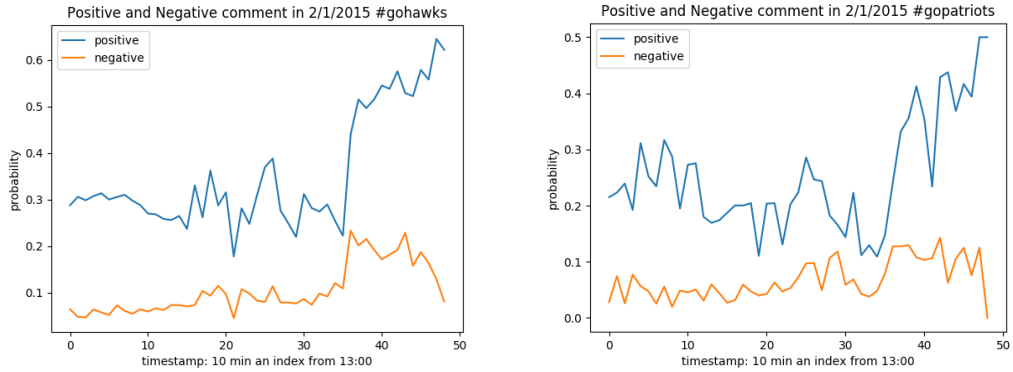

Figure 41. positive and negative comment vs timestamp

In the above figures, we can see for both teams, positive rate is higher than negative rate, because it is the twitter of the fans, the fans would leave more positive comments than negative. For positive rate, it has both higher average and standard deviation than negative rate. To compare with positive and negative rate fairly I take normalization of the two lines of two figures, and it is shown below:

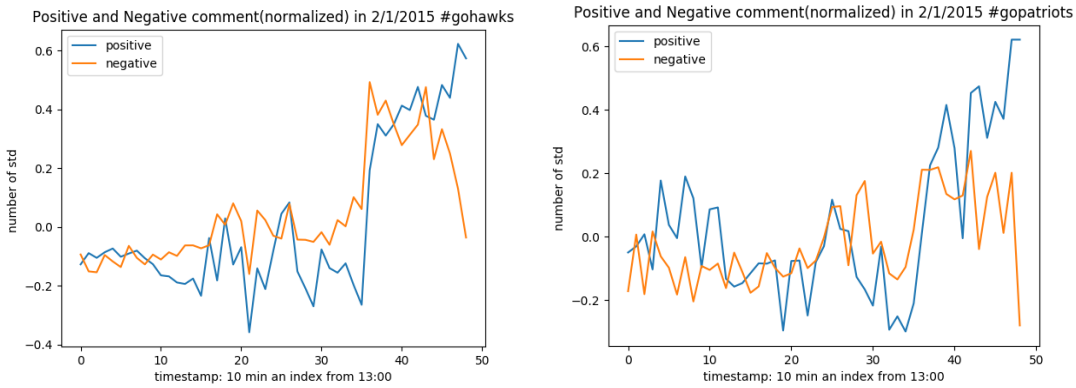

Figure 42. positive and negative comment(normalized) vs timestamp

In the figure, the y axis means the number of standard deviation the points have compared to the mean value. For example, for positive line in #gohawks, if it has y=0.2, it means the positive point is 0.2 standard deviation above the positive mean of all the dataset in #gohawks.

To compare the sentiment change of the fans of two teams, I take normalized positive minus normalized negative of each team to represents the sentiment, and the figure is shown below:

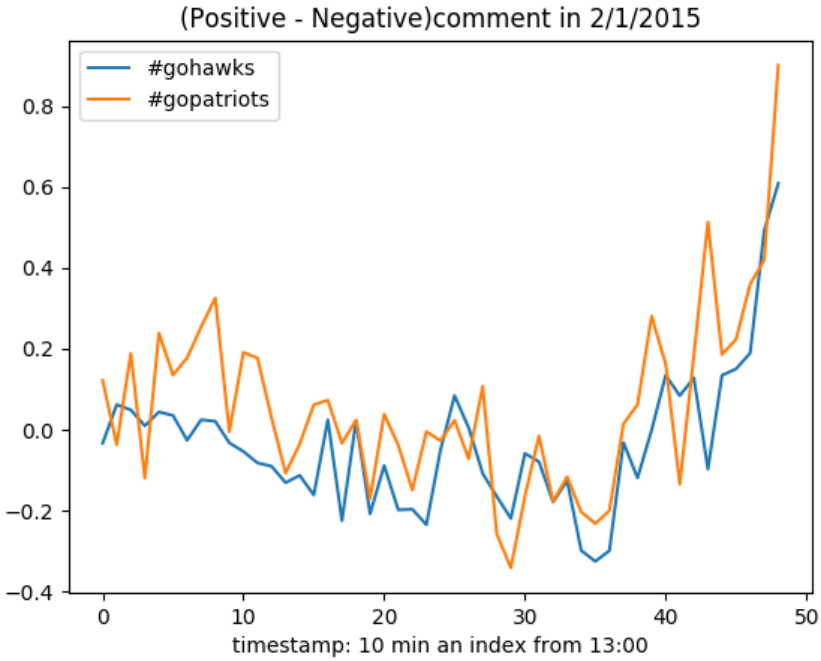

Figure 43. positive minus negative comment(normalized) vs timestamp

We want to analyze the sentiment change, so I take the sign of sentiment at time=i+1 minus the sentiment at time=i to create "sentiment change" arrays for #gohawks and #gopatriots: +1 means audience's sentiment becomes positive, and -1 means audience's sentiment become negative. The "sentiment change" for two teams are different in these time index:
array([0, 1, 5, 7, 9, 14, 21, 22, 23, 26, 30, 37, 39, 42, 43])

There are 48 time periods in the "sentiment change" array, but there are only 15 time periods where two team fans have different sentiment change, which means that two team fans mostly have the same trend of sentiment change: both sentiment becomes positive or negative together, but when a team scores, audience of that team have positive sentiment change, and the audience of opposite team have negative sentiment change. In other words, in the matching time, which is in time period #15 to #36, the sentiment changes for the two team fans have different in time period #21, #22, #23, #26, #30, which means one team scores at these time period or 1 time period before because fans may watch his/her team scores in this time period and comment on twitter in next time period. Additionally, in the time period after the game, which is time period #37 to #49, fans of patriots mostly have more positive sentiment than the fans of hawks, which corresponds to the result of the game: patriots win the match.