Project 5
Tweet Popularity Prediction

Shivani Soman - 304946159
Maithili Bhide - 104943331
Satya Vasanth Reddy - 405029459

Part 1: Popularity Prediction

1.1 Statistics

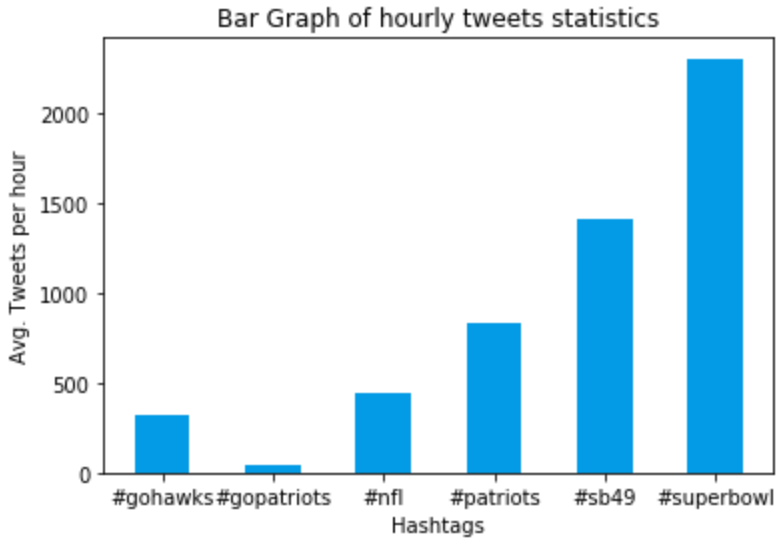
The dataset is very huge compared to the previous projects. There are hashtags with more than million tweets. In order to efficiently load or use the data and answer the required questions, the code we wrote first iterates through each and every <hashtag>.txt and then extracts the information that is required and writes each row corresponding to a line in the txt file into .csv file.

We are then loading the csv file and using pandas to construct a dataframe out of it. Since we are saving the required information as a .csv , whenever we try to load the data next time, we look for the .csv file and use this smaller file. We observed that the size 20 times smaller than the actual file size.

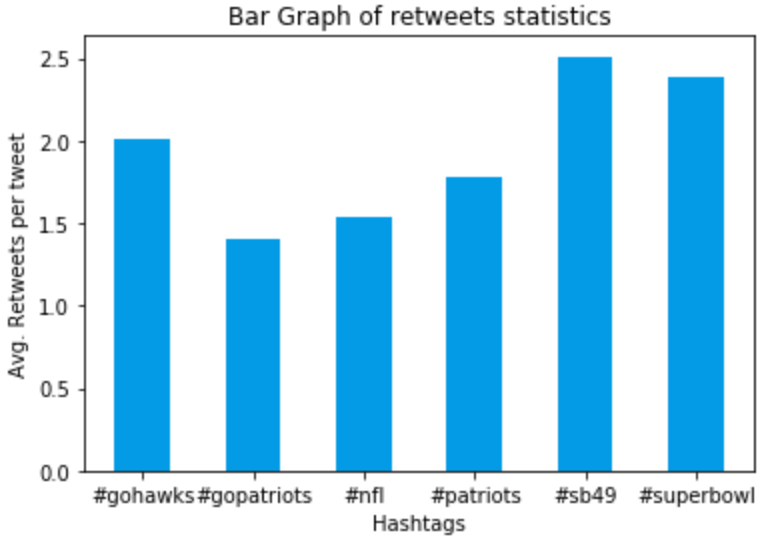
Context : The dataset is the tweets extracted from twitter during the Superbowl game in 2015. The game was played on February 1st between New England patriots and the Seattle Hawks. There are 6 hashtags #patriots and #gopatriots may refer to the tweets that are directed at the New England Patriots team and #gohawks at the Seattle Seahawks team. #superbowl and #sb49 have a greater correlation in the real world as it was the 49th Super Bowl match and sb is the short form of superbowl. #nfl refers to the tournament.

Hashtags	Average hourly tweet count	Average Retweets	Average Followers
#gopatriots	26232	1.400083867	1401.895509
#nfl	259024	1.538533109	4653.252286
#sb49	826951	2.511148786	10267.31685
#gohawks	188136	2.014617086	2203.931767
#patriots	489713	1.782815649	3309.978828
#superbowl	1348761	2.388273386	8858.927767

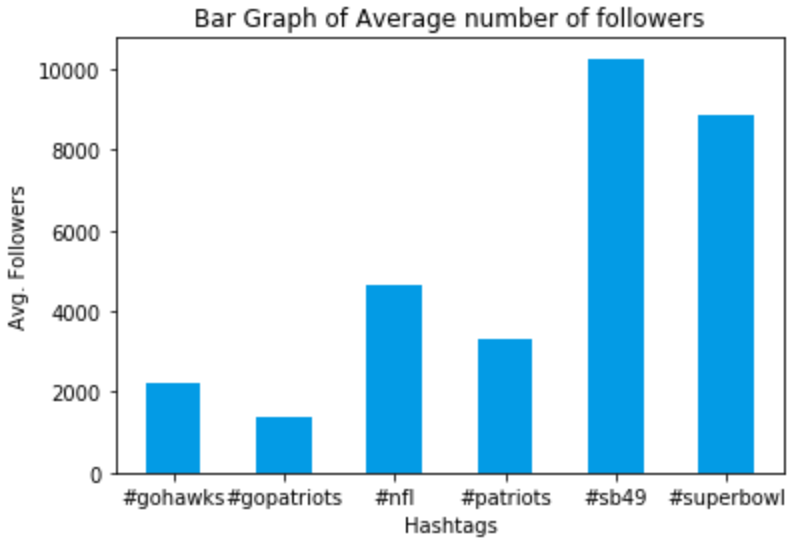
Average number of Tweets per hour



Average number of retweets



Average number of Followers of users posting the tweets



Analysis:

- The average number of tweets for the #superbowl and #sb49 are much higher owing to its a humongous share of audience. The tweets directed at New England Patriots are more in number than the Seattle SeaHawks. #gopatriots sounds like a supportive hashtag but not much can be inferred from the #patriots. This is because the Patriots teams is relatively more popular and often controversial so is the star player Tom Brady.
- The average retweets is similar for all the hashtags and not much can be inferred from the pattern. But if we take the average number of retweets for #gopatriots and #patriots together, it will be smaller than that for #gohawks despite significantly less number of tweets. Since the data contains both tweets and retweets, it can be inferred that the portion of original tweets(not retweets) directed at New England Patriots are more than that of Seattle Seahawks
- The average followers clearly show that popular twitter personalities(high follower count) are tweeting more about #superbowl or #sb49

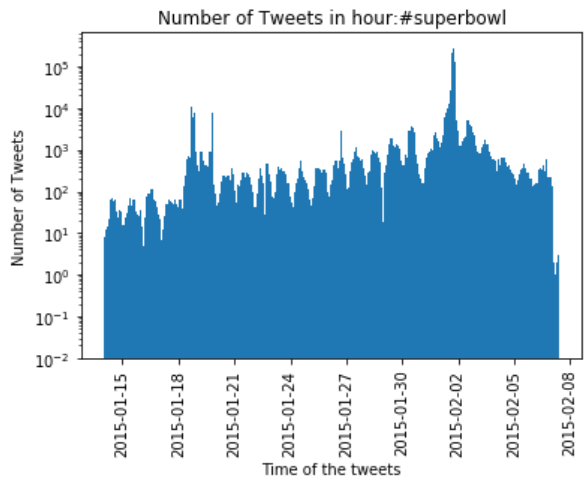
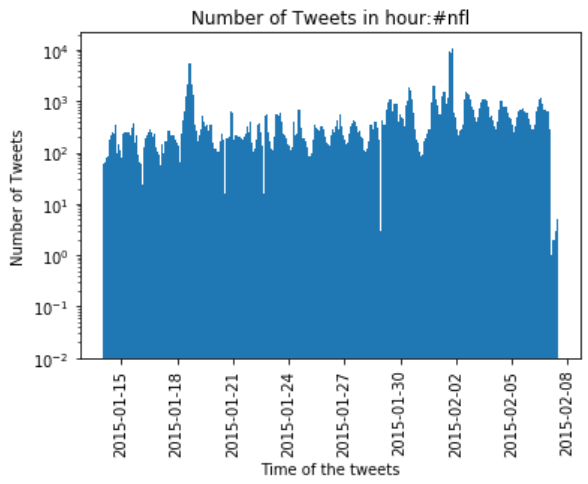
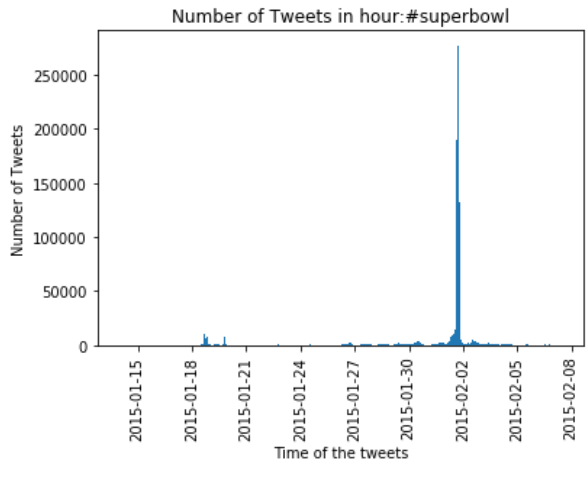
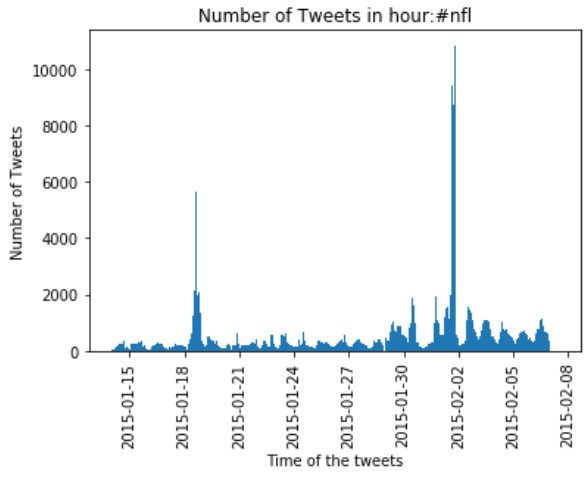
Implementation: From the constructed dataframe, we calculated the bucket using
`datetime.datetime.fromtimestamp(int(timestamp/3600)*3600, pst_tz)`

Number of tweets in hour" over time for #SuperBowl and #NFL

Implementation:

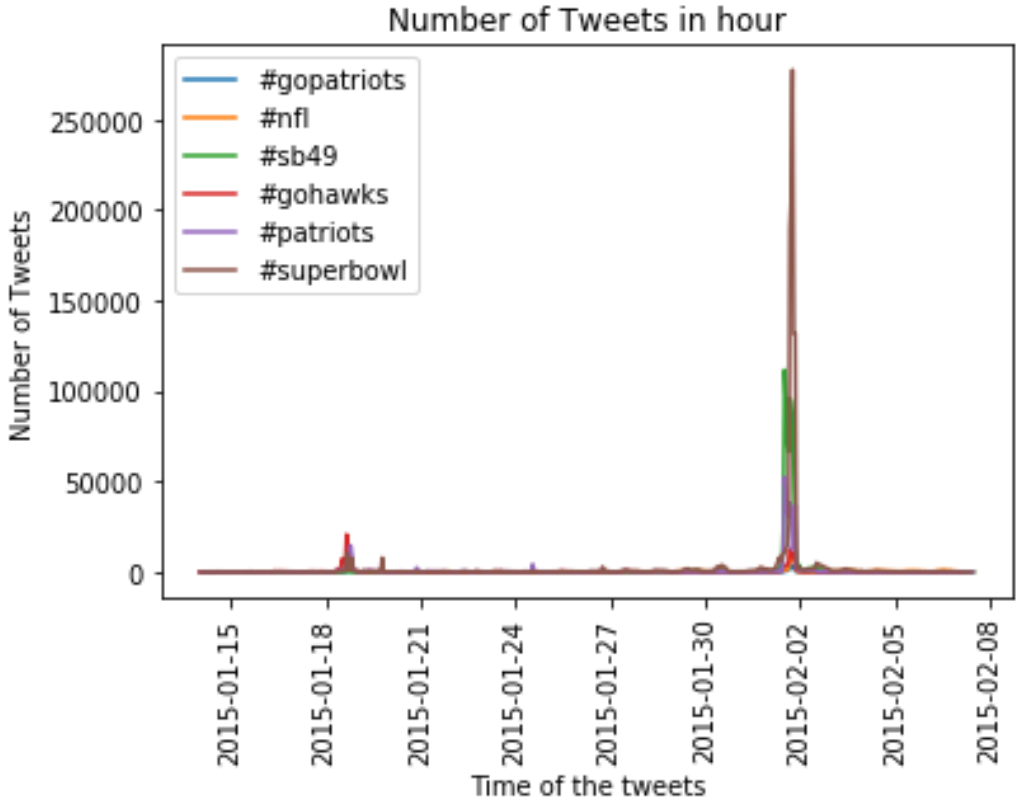
After calculating the hour bucket, we used the `groupby` operation on the pandas dataframe and then calculated the count of the tweets.

Note: Please note that the yscale is made logarithmic in order to observe the values clearly

Logarithmic scale	
	
Linear Scale	
	

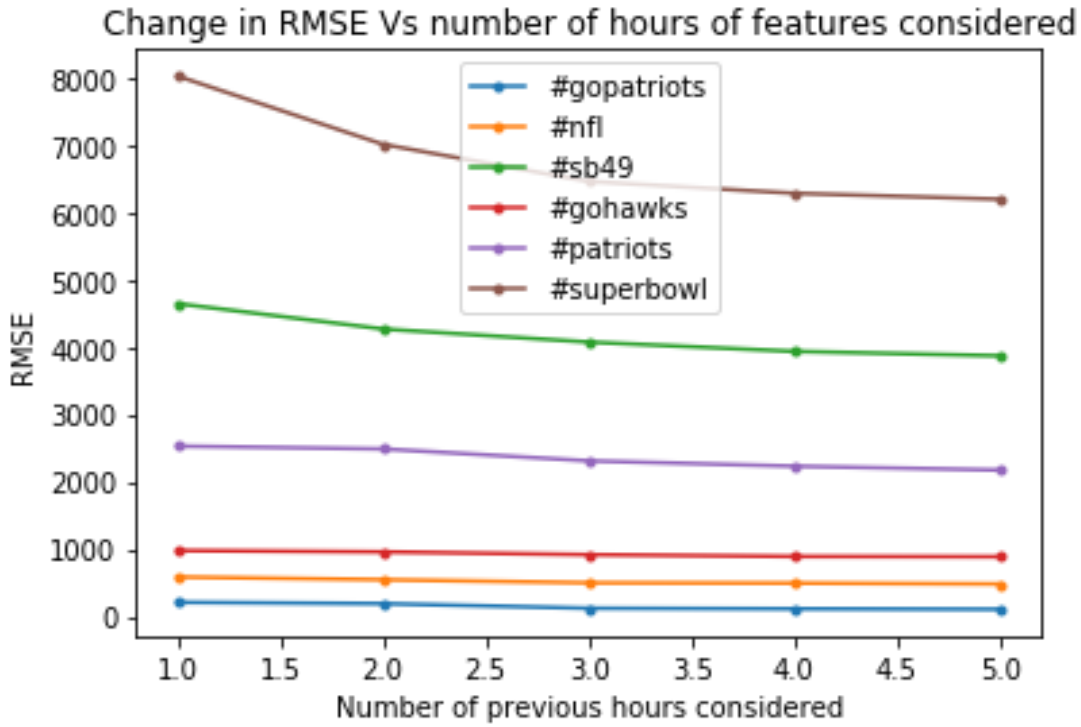
Analysis:

- In these plots, we see that there are 2 spikes. One is on the night of 18 January and the other on the night of February 1. 18 January is the day of semifinal of the NFL tournament and hence there is a major spike.
- Superbowl is nothing but the final of the NFL tournament. Even though it appears to be a very small spike on 18 Jan for #superbowl plot, it actually peaks to a value of ~8000 which is almost similar to that of #nfl. The explosion in the number of tweets in February 1 for #superbowl is due to the extreme popularity of the name of the game as Superbowl instead of NFL finals.
- We also see similar trends of peaks in the plots of the number of tweets in hour for the remaining other hashtags.



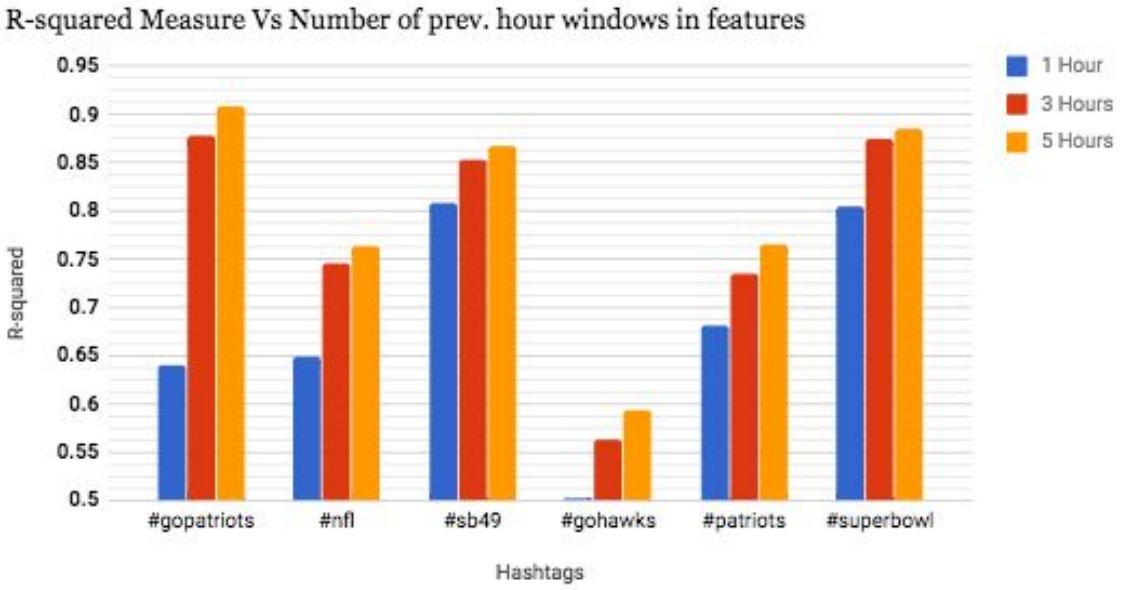
1.2 Linear Regression

RMSE Variation with the number of hours in features



The figure above shows the RMSE of all the hashtags when considering 1,2,3,4 and 5 hour features. We can clearly see the RMSE values decreasing with increasing window size.

R-squared Measure Variation with the number of hours in feature

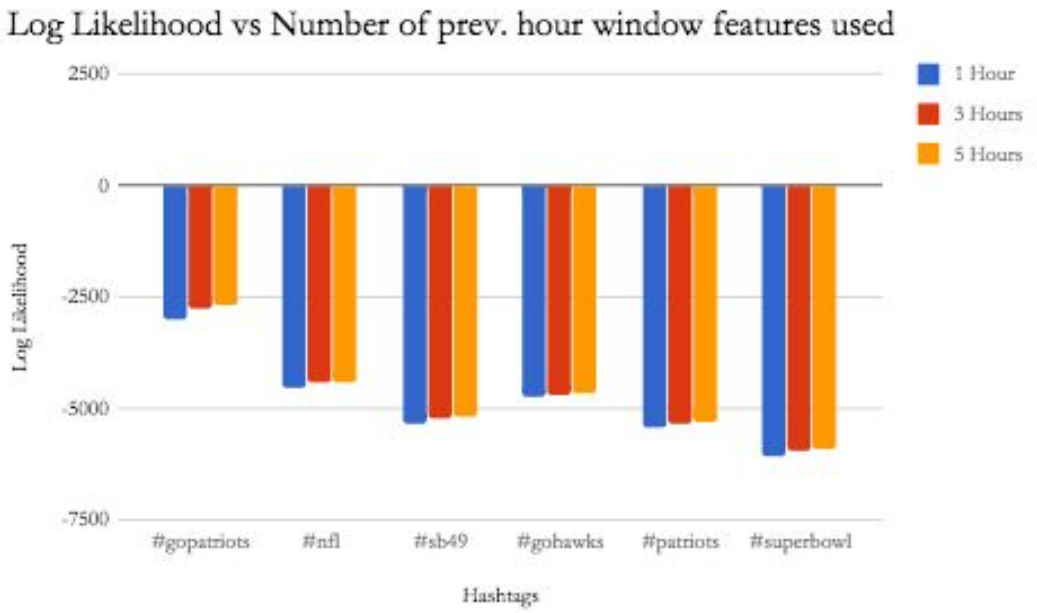


The above figure shows the R-squared measure for each hashtag for 1,3 and 5 hour window features.

Analysis

- R-squared is a statistical measure of how close the data are to the fitted regression line.
- The R-squared measure is observed to be higher in case of the hashtags with larger number of tweets.

Log Likelihood Variation with the number of hours in feature



Analysis:

- The log likelihood plot above shows the log likelihood of OLS regression model for each hashtag by taking 1,3, and 5 hour window features. We see that the log likelihood is always higher and R-squared is always lower when we are using features from higher window size here. It shows that the tweet count for the current window can be best explained by considering the past 5 hour trend compared to past 1 hour.
- Also the likelihood is relatively higher and the R-squared is relatively lower for the hashtags with less number of tweets. It is because, since the number of tweets is a continuous variable and when we are using linear regression model, the higher number of data entries adds up to the R-squared in this case.

Feature Coefficients of Linear Regression for hashtags with features from previous window

	Coefficients of each hashtag					
Feature name	**#gopatriots**	**#nfl**	**#sb49**	**#gohawks**	**#patriots**	**#superbowl**
Sum of followers	0.0003	8.11E-05	1.87E-05	-0.0002	-1.17E-06	-0.0001
Maximum Followers	-0.0004	-7.78E-05	9.99E-05	1.05E-05	2.00E-04	0.0008
Tweet count	-0.075	0.7297	1.1889	1.23	0.9225	2.3003
Retweets sum	0.4944	-0.18	-0.2152	-0.1307	-0.0883	-0.2904
hour_of_day	0.8059	8.3117	-3.271	9.8854	5.0149	-35.9739

P-values of Linear Regression for hashtags with features from previous 1 hour window

The values are rounded to 3 decimal digits

	P-Values of each hashtag					
Feature name	**#gopatriots**	**#nfl**	**#sb49**	**#gohawks**	**#patriots**	**#superbowl**
Sum of followers	0.259	2.00E-03	2.00E-01	0.044	9.64E-01	0
Maximum Followers	0.086	3.20E-02	4.50E-02	9.48E-01	8.60E-02	0
Tweet count	0.796	0	0	0	0	0
Retweets sum	0.052	0.005	0.018	0.003	0.135	0
hour_of_day	0.299	0	0.843	0.003	0.571	0.232

Analysis

- P-values of the features indicate the significance of the variables.
- A p-value is the probability that the results from the sample data occurred by chance.
- A small p-value (typically ≤ 0.05) indicates strong evidence, a large p-value (> 0.05) indicates weak evidence and p-values very close to the cutoff (0.05) are considered to be marginal as evidence to reject the null hypothesis.
- The idea of null hypothesis is that a claim is assumed valid if its counter-claim is improbable.
- So a lower P-value justifies the strength of the hypothesis. In the above table, we have highlighted the features which are significant for each hashtag

For #gopatriots, we donot find any significant feature but **tweet_count** is significant in #nfl, #gohawks , #patriots. We find followers max is a significant feature too in #sb49. All the features except #hour_of_day are significant in #superbowl. However we can see that in the t-values below, the tweet_count has more t-value. So if we use the rejection region approach, the hypothesis for tweet_count would be rejected with more probability making it the most significant.

ttest -values of Linear Regression for hashtags with features from previous 1 hour window

	t-test results for each feature					
Feature name	**#gopatriots**	**#nfl**	**#sb49**	**#gohawks**	**#patriots**	**#superbowl**
Sum of followers	1.13	3.08E+00	1.28E+00	-2.018	-4.50E-02	-6.987
Maximum Followers	-1.718	-2.15E+00	2.01E+00	6.50E-02	1.72E+00	5.381
Tweet count	-0.259	5.471	12.033	7.213	12.887	28.894
Retweets sum	1.946	-2.808	-2.363	-2.948	-1.495	-8.06
hour_of_day	1.039	3.712	-0.198	2.944	0.568	-1.196

Values are rounded to 3 decimal digits

Analysis:

- The t-test is a measure on the means or average of distributions. It is also a null hypothesis test. It is calculated as below

$$t = \frac{\bar{x} - \mu_0}{\frac{s}{\sqrt{n}}}$$

- The top portion is simply the difference between the means or averages of the two samples. The lower half of the ratio is a measurement of the dispersion, of the scores.
- The bottom part of this ratio is known as the standard error of the difference. To compute this part of the ratio, the variance for each sample is determined and is then divided by the number of individuals the compose the sample, or group. These two values are then added together, and a square root is taken of the result.
- When we perform t-test, $\bar{x}$ is the mean of the data we have and \Mu is the mean of the hypothesis sample. A large t-score tells you that the groups are different. A small t-score tells you that the groups are similar.

- A t score of 3 means that the samples are three times as different from each other than they are similar to each other.
- So the bigger the t-value, the more likely it is that the results are repeatable.
- We see both positive and negative t- scores. Positive values occur when sample mean is larger than the hypothesized mean, negative values reflect a sample mean smaller than than hypothesized mean.

1.3 Feature Extraction

In this part we will be designing a linear regression model using additional features to the ones covered in the part 1.2. We aim to find the best features to fit our data of each hashtag. Better features help in inferring more direct relationships between the feature values and predicted quantity. Accordingly, we will be looking at the following additional features falling in the categories of user features, content features, time series features and meme features.

1. # Tweets: Tweet count
2. # Authors: Number of unique authors of tweet
3. Average # Followers: Average number of followers
4. Retweets: Total number of retweets
5. Title length: Length of the tweet
6. Firstpost: Time of first tweet in the group
7. Acceleration: Total acceleration of all tweets in the group
8. Peak: Maximum peak time for activity
9. User mentions: Total number of user mentions
10. Impressions: Total number of times the tweet occur in the feed
11. Average retweet count: Total retweet count averaged by number of tweets
12. Average ranking score: Average ranking score for tweets in group
13. Original followers: Total original followers count for that group
14. Favorite count: Total favorite count for the group
15. Time - Hour: Hour of day when tweeted

Each of these features are extracted by first grouping the data into time intervals of 1 hour each. Based on these features, we trained linear and non-linear models using a 1-hour time window on the Twitter data for each of the hashtags to predict the number of tweets in the next hour. In this section we have added the scatter plots of the top 3 best features according to our model vs the number of tweets in the next hour.

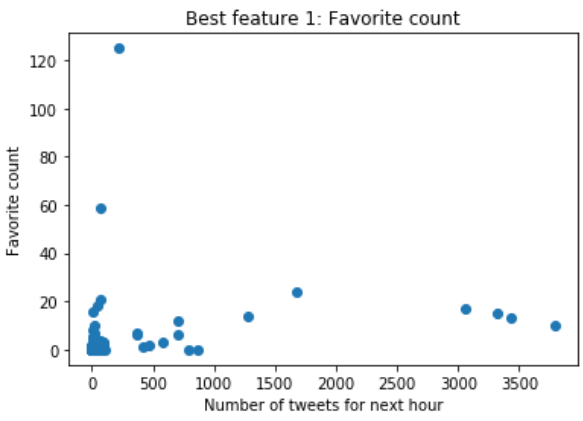
Scatter plots of predictant (number of tweets for next hour) versus value of that feature

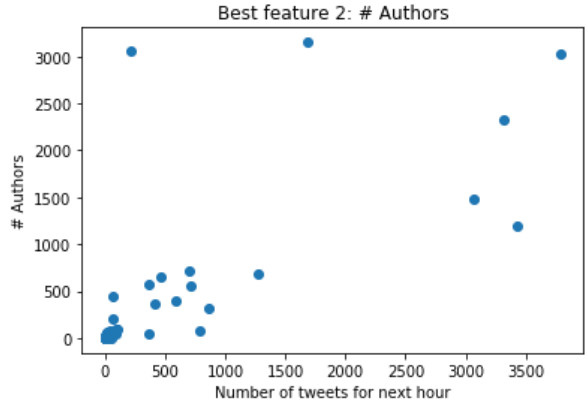
In this section, our model for each of the hashtags is the Ordinary Least Squares Linear Regression model from the statsmodels API. To select the best 3 features for our hashtag, we are using the P-values of the features as they indicate the significance of the variables. A small p-value (typically ≤ 0.05) indicates strong evidence, a large p-value (> 0.05) indicates weak evidence and p-values very close to the cutoff (0.05) are considered to be marginal as evidence to reject the null hypothesis. In addition to the p-values, we are also reporting the coefficients of the top 3 best features for our model. We are also reporting the Root Mean Square Error and Mean Absolute Error for each of the models on every hashtag in order to report the fitting accuracy or fit of the model on our dataset.

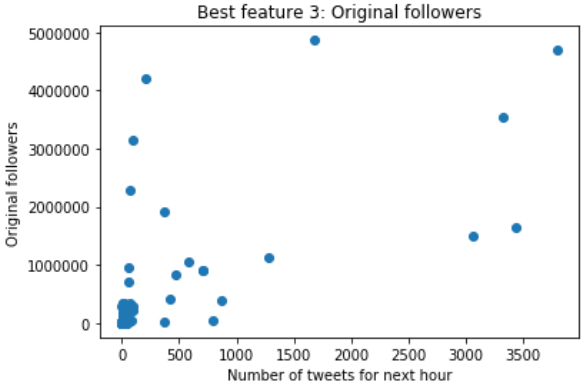
Our list of features is as follows -

feature_names = ['# Tweets', '# Authors', 'Average # Followers', 'Retweets', 'Title length', 'Firstpost', 'Acceleration', 'Peak', 'User mentions', 'Impressions', 'Average retweet count', 'Average ranking score', 'Original followers', 'Favorite count', 'Time - Hour']

#gopatriots

#gopatriots	



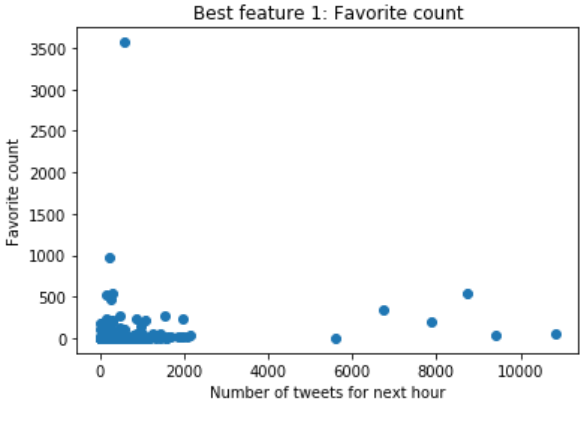


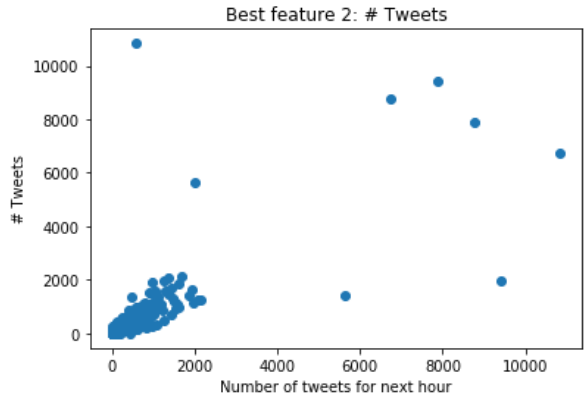
p-values	0.15, 0, 0.671, 0.001, 0.775, 0.127, 0.064, 0.032, 0.15, 0, 0, 0.143, 0, 0, 0.903
coefficients	-0.3296, 3.4694, -0.0005, -1.0855, 0.1153, -5.042E-08, 0.1758, 3.362E-08, -0.3296, -0.0006, 166.1267, 17.4787, 0.0007, -31.8699, 0.1195
RMSE	155.601173453
MAE	42.4286078462

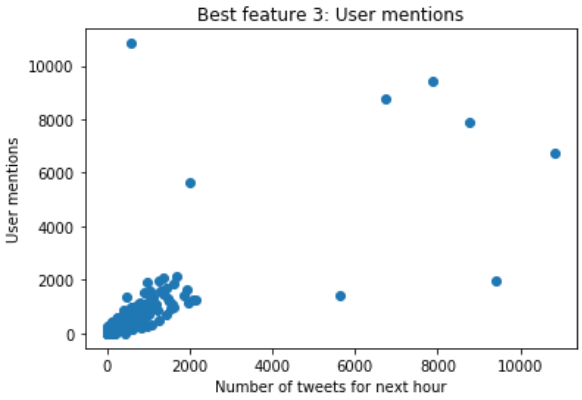
From the figures we can see that the features with the highest significance for the linear model trained on data for '#gopatriots' are favourite count, authors, original followers. We can infer from these that the number of authors probably indicate the number of supporters of the team Patriots and so the number of tweets in the next hour is linearly dependent on them as well as the number of original followers. Favourite count of tweets is a generally important predictor seen in many of the hashtag significance p-values.

#nfl

#nfl



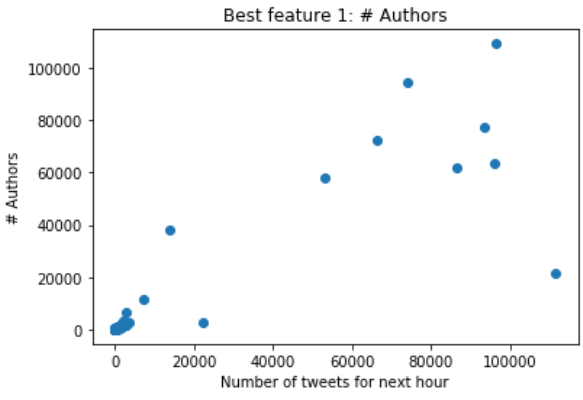


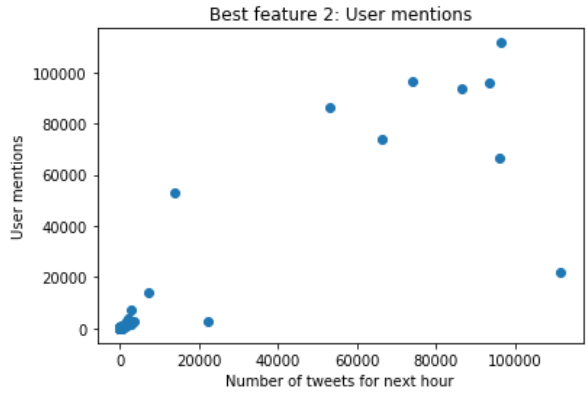


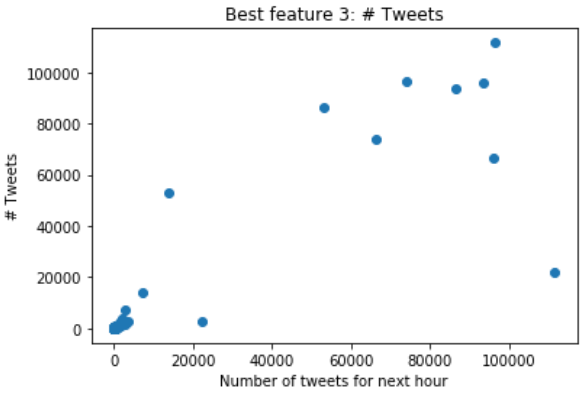
p-values	0, 0.007, 0.572, 0.08, 0.498, 0.915, 0.286, 0.323, 0, 0.734, 0.015, 0.253, 0.866, 0, 0.418
coefficients	0.9572, -0.95, 0.0026, -0.1177, -4.038, -6.498E-08, -0.0916, 5.382E-08, 0.9572, -7.445E-06, 91.9036, 148.3594, 3.377E-06, -2.1084, -2.6503
RMSE	483.24554671
MAE	178.328983433

From the figures we can see that the features with the highest significance for the linear model trained on data for '#nfl' are favourite count, tweets and user mentions. We can see a very clear linear trend for the number of tweets and user mentions with respect to the number of tweets in the next hour. The trend is very pronounced at the start near the origin.

#sb49

#sb49	



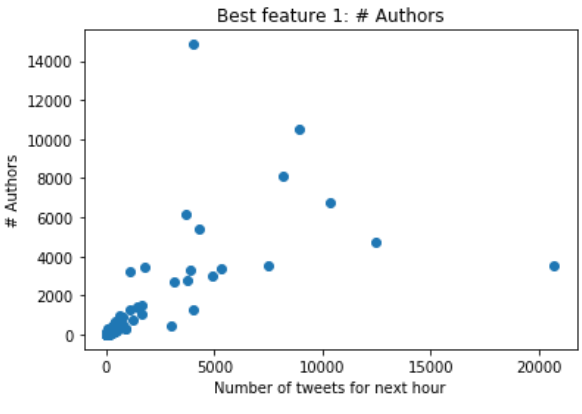


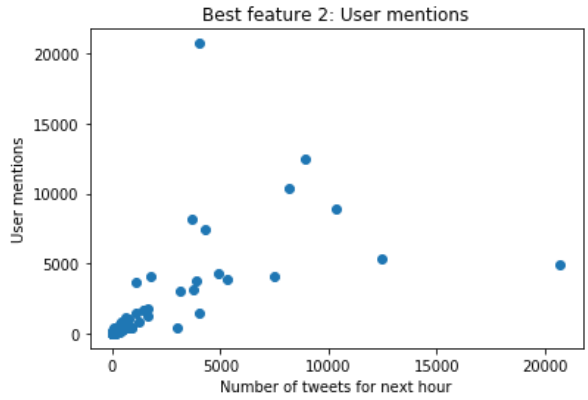
p-values	0, 0, 0.481, 0.006, 0.001, 0.008, 0.201, 0.46, 0, 0.49, 0.96, 0.179, 0.727, 0.003,0.356
coefficients	-1.3112, 3.0657, -0.0028, 0.3945, -50.3459, 3.519E-06, -0.3368, 2.981E-07, -1.3112, 4.245E-05, -1.8608, 498.6816, -2.256E-05, -0.3766, -25.5906
RMSE	4307.00614908
MAE	913.573445327

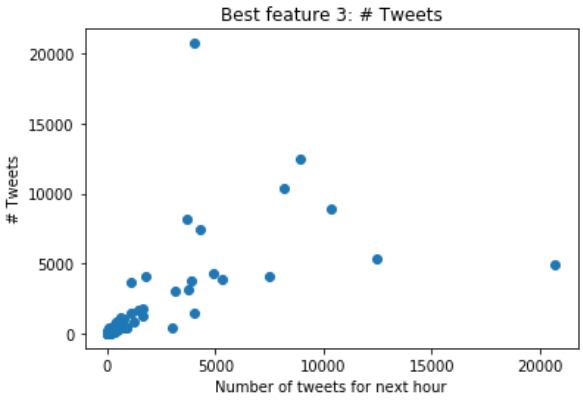
From the figures we can see that the features with the highest significance for the linear model trained on data for '#sb49' are authors, user mentions and tweets. From the graphs we do see the beginnings of a linear trend near the origin however it is not as pronounced as it was for '#nfl' probably due to lack of popularity of this hashtag in comparison to other similar hashtags.

#gohawks

#gohawks	



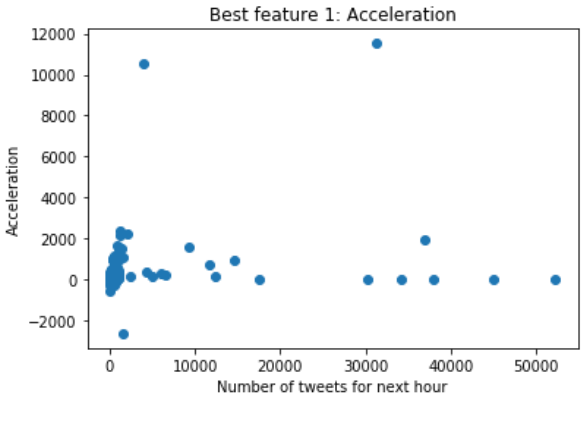


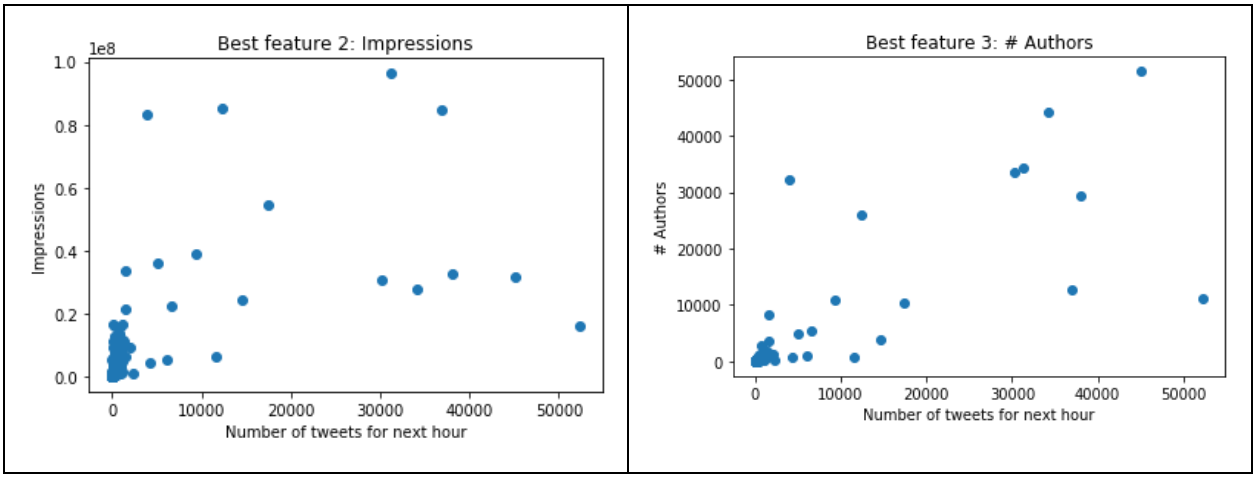


p-values	0, 0, 0.852, 0, 0.793, 0.464, 0, 0.768, 0, 0.081, 0.969, 0.505, 0.41, 0.17, 0.597
coefficients	-1.6156, 5.8488, 0.0019, -0.2376, 1.1733, -3.342E-07, -0.7599, 2.25E-08, -1.6156, -9.789E-05, 2.2996, 85.3817, 5.216E-05, 0.068, -3.301
RMSE	875.606295582
MAE	190.096985723

From the figures we can see that the features with the highest significance for the linear model trained on data for '#gohawks' are authors, user mentions and tweets. The trend seen for '#gohawks' is also pretty pronounced and linear. By looking at the features with the highest p-values we can guess that this hashtags predictive power depends on the supporters of the team and is driven by their interaction in the form of user mentions and tweets.

#patriots

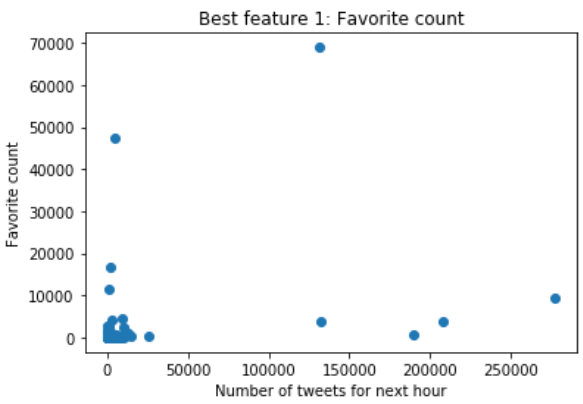
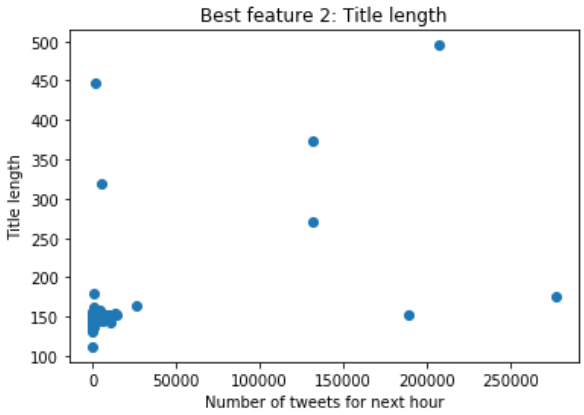
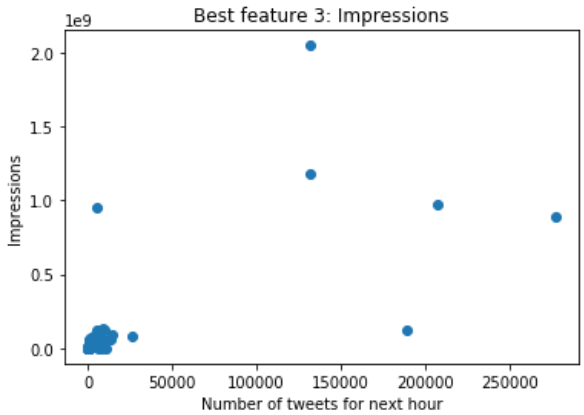
#patriots	



p-values	0, 0, 0.001, 0.552, 0.011, 0.262, 0, 0.881, 0, 0, 0.639, 0.534, 0.002, 0.039, 0.951
coefficients	-1.5522, 3.7309, -0.0244, 0.0857, 31.6237, -2.163E-06, -2.099, -3.497E-08, -1.5522, 0.0006, -15.0852, -270.7629, -0.0003, 0.578, -0.9615
RMSE	2290.26959227
MAE	646.613431354

From the figures we can see that the features with the highest significance for the linear model trained on data for '#patriots' are acceleration, impressions and authors. Since patriots is a team playing in the Superbowl, we would expect to see a significant feature authors as is observed above indicating the dependence of this hashtag on user engagement presumably of supporters of the team Patriots. This feature also has the most linear trend with number of tweets in the next hour as compared to the other 3 significant features.

#superbowl

#superbowl	
	

p-values	0.011, 0, 0.845, 0.001, 0, 0.01, 0.09, 0.109, 0.011, 0, 0.022, 0.615, 0, 0, 0.552
coefficients	0.5343, 2.114, 0.0038, -0.6159, -157.204, 1.262E-05, -0.4955, 1.289E-06, 0.5343, -0.0001, 674.327, 741.3124, 0.0001, -2.416, 28.4728
RMSE	6835.49788155
MAE	1678.31962093

From the figures we can see that the features with the highest significance for the linear model trained on data for '#super bowl' are favourite count, title length and impressions. This is the only hashtag where we found title length to be a significant indicator of number of tweets in the next hour. The third top feature is impressions which indicates the number of times the tweet appears in other users feeds. This makes sense for '#superbowl' because it is more likely to be retweeted if it has a higher impressions value. Another important thing to notice is that user mentions is not one of the top 3 significant features for this hashtag. User mentions and authors were very popular features for the hashtags of teams as seen previously indicating that the

number of tweets for those hashtags was more strongly linked to user interaction than it is for the general Superbowl related hashtags.

From all the RMSE and MAE values we can see that the hashtag '#gopatriots' has the lowest RMSE amongst all other hashtags. This is because that dataset was the smallest as compared to the dataset for other hashtags.

1.4 Cross Validation

For this section, we will divide the data into the following 3 time periods -

1. Before Feb. 1, 8:00 a.m. : Before the hashtags have become very active

2. Between Feb. 1, 8:00 a.m. and 8:00 p.m. 3. After Feb. 1, 8:00 p.m. : Active period of hashtags

3. After Feb. 1, 8:00 p.m. : After they have passed their active period

For each of the hashtags, we perform 10 - fold cross validation to perform a 90-10% split and report the Root Mean Square Error and Mean Absolute Error for each of the intervals. For this purpose we are using a 1 hour time window where we train on data of the past hour to predict the number of tweets in the next hour. In addition to reporting the RMSE and MAE for the individual hashtags, we are also reporting the RMSE and MAE of the aggregated data of all hashtags over the 3 time intervals. For this section, we are using an Ordinary Least Squares Linear model, Random Forest model with 25 estimators and max_depth=30 and a Neural Network with hidden layer size=(5X2), alpha=10^-5 and an lbfgs solver (for the neural network model we have also preprocessed the data by scaling it using StandardScaler).

Based on the time intervals, we can guess that the values of the errors for our models would be maximum in the interval of maximal activity. Additionally, training a model in the 12 hour interval with a 1 hour time window is not sufficient for it to predict the large values of number of tweets in the next hour. The large error values are also because the volume of tweets increase tremendously for the active hashtags during the second interval.

Average cross validation errors for linear model in 3 intervals for each hashtag and all hashtags combined

Model	Hashtag	Interval	RMSE	MAE
Linear Model (Ordinary Least Squares)	#gopatriots	Before	52.5310860402	12.5929775302
		During	823280.112266	275909.773526
		After	2713.6613878	379.077266144

	#nfl	Before	317.886945884	129.11324387
		During	81594.949865	52827.952056
		After	200.620692469	133.62460429
	#sb49	Before	119.439203136	48.216434199
		During	644422.28516	476509.935655
		After	269.948301295	138.287370787
	#gohawks	Before	1102.32554126	286.483729937
		During	59382.07751	31531.0478979
		After	13424.486559	1435.67732912
	#patriots	Before	1119.89571797	277.659893242
		During	680535.747532	439411.1252
		After	753.145505553	165.192043396
	#superbowl	Before	956.344295158	325.783049383
		During	47265092.6379	21397097.8066
		After	134892.762664	35788.6804973
	All combined	Before	2494.95734252	785.636454937
		During	36345163.3668	15282461.9215
		After	364237.623491	56288.2962309

From the MAE and RMSE results for each of the hashtags in the 3 time intervals we can conclude that the linear model does significantly worse in the time interval during the Super bowl. This was expected as we are training our model on an interval of 12 hours in 1 hour windows leading to very few training points. The large RMSE values are also explained by the fact that during the super bowl each of our hashtags was in its active period leading to greater number of tweets. The hashtag '#superbowl' has the highest RMSE and MAE in the during interval than any other hashtag presumably because of its popularity amongst users. We can see that the hashtag '#gopatriots' and '#gowaks' have a higher RMSE in the during and after interval indicating increased use of the hashtag by supporters since both these hashtags are related to teams that played in the Super bowl as opposed to the general hashtags '#nfl' and '#sb49'. The linear model is seen to perform better in the before interval than in any other

interval. We can see from the results for all combined hashtags that the linear model is not very good and will try other models further.

Average cross validation errors for Random Forest model in 3 intervals for each hashtag and all hashtags combined

Model	Hashtag	Interval	RMSE	MAE
Random Forest	#gopatriots	Before	69.7824642473	11.8599486171
		During	872.159937764	759.278
		After	10.6771744988	3.98211825397
	#nfl	Before	302.916018596	119.578890063
		During	2750.37422776	1683.958
		After	237.030289474	171.059802198
	#sb49	Before	133.676932749	42.7925641817
		During	37580.2960387	31513.044
		After	291.010782441	121.239340659
	#gohawks	Before	1098.68710708	171.230684989
		During	3629.87331884	2589.498
		After	78.4174022149	22.5203123932
	#patriots	Before	998.896832438	231.456285412
		During	16105.5043329	12117.372
		After	262.332657497	106.73643956
	#superbowl	Before	765.273485897	258.423904863
		During	86117.1155418	64281.688
		After	817.731587673	412.430153846
	All combined	Before	2624.83348255	654.281566596
		During	117980.496128	99383.77
		After	718.999576668	429.560153846

Similar to our linear model even the Random Forest model has very high values during the Super bowl. This was expected as we are training our model on an interval of 12 hours in 1 hour windows leading to very few training points. The large RMSE values are also explained by the fact that during the super bowl each of our hashtags was in its active period leading to greater number of tweets. Contrasting the above results with our results for the linear model, we can see that the random forest model performs much better. It has significantly lower values of RMSE and MAE even for the during Superbowl interval. From the results above we can also see that the random forest model does a much better job on the hashtags '#gohawks' and '#gopatriots' in all 3 intervals. For the before and after intervals of hashtags '#sb49' and '#nfl' we are getting similar results with random forest as we were with linear model but the during interval shows a significantly tighter or lower RMSE than the RMSE for the linear model. Similarly even the most active hashtag '#superbowl' shows lower RMSE values with the random forest model.

Average cross validation errors for Neural Network model in 3 intervals for each hashtag and all hashtags combined

Model	Hashtag	Interval	RMSE	MAE
Neural Network	#gopatriots	Before	70.3420238344	12.0322410148
		During	1650.95455722	1257.85
		After	11.4376514982	4.19807692308
	#nfl	Before	420.183606676	194.624630021
		During	4930.1295774	3475.05
		After	477.751582651	369.203296703
	#sb49	Before	260.05184818	93.3890063425
		During	52958.9486149	39619.9
		After	845.463109976	443.401098901
	#gohawks	Before	1092.60986893	207.389640592
		During	4793.26231809	3859.45
		After	135.015580772	41.4121794872
	#patriots	Before	1182.18336758	315.735412262

		During	20401.5877691	17019.9
		After	325.752305112	142.697252747
	#superbowl	Before	1144.8494253	441.399524313
		During	94543.7472491	61695.8
		After	917.973701711	581.204945055
	All combined	Before	2319.52856974	845.894186047
		During	189411.582479	157523.4
		After	1857.11172062	1168.38571429

Similar to our linear and Random Forest model even the Neural Network model has very high values during the Super bowl. This was expected as we are training our model on an interval of 12 hours in 1 hour windows leading to very few training points. The large RMSE values are also explained by the fact that during the super bowl each of our hashtags was in its active period leading to greater number of tweets. In this model we can observe similar trends with the RMSEs for the different hashtags as the previous models. The neural network performs better than the linear model but not as good as the random forest model.

Best Model

From the results above we noticed lower values of RMSE and MAE across most of the hashtags for each of the intervals for the random forest model. Hence we concluded that the best model was the random forest model with the following parameters -

RandomForestRegressor(n_estimators=25, max_depth=30, max_features=n_features, bootstrap=True, oob_score=True, random_state=42)

We will be using this classifier in the further section 1.5 and will train it on a concatenated feature vector containing 75 features from a 5 hour time window.

1.5 Testing best model on test data

For this section, we will be using the best model as determined from the previous results. Our best model is Random Forest with the following parameters -

RandomForestRegressor(n_estimators=25, max_depth=30, max_features=n_features, bootstrap=True, oob_score=True, random_state=42)

The table below shows the time intervals in each of the test files according to the 'firstpost' feature.

Test File	Minimum Time	Maximum Time
sample1_period1.txt	2015-01-29 11:00:05-07:00	2015-01-29 16:59:05-07:00
sample2_period2.txt	2015-02-01 12:00:00-07:00	2015-02-01 17:59:59-07:00
sample3_period3.txt	2015-02-02 04:00:02-07:00	2015-02-02 09:59:59-07:00
sample4_period1.txt	2015-01-25 15:00:04-07:00	2015-01-25 20:59:41-07:00
sample5_period1.txt	2015-01-27 18:00:20-07:00	2015-01-27 23:59:49-07:00
sample6_period2.txt	2015-02-01 10:00:01-07:00	2015-02-01 15:59:59-07:00
sample7_period3.txt	2015-02-02 23:00:03-07:00	2015-02-03 04:57:12-07:00
sample8_period1.txt	2015-01-28 17:00:05-07:00	2015-01-28 21:55:51-07:00
sample9_period2.txt	2015-02-01 11:00:00-07:00	2015-02-01 16:59:59-07:00
sample10_period3.txt	2015-02-05 13:00:43-07:00	2015-02-05 18:59:18-07:00

From this table we can see that all the test files except sample8_period1 have a time window of 6 hours. For sample8_period1, the time window is only 5 hours. Considering this time window, we will be training our model on feature vector of the concatenated data of the last 5 hours for our data consisting of all hashtags over all intervals (before Superbowl, during and after Superbowl) and predicting on the concatenated data of the last 5 hours in each test file to get the prediction of the number of tweets in the next hour.

In section 1.3, we extracted 15 features. In this part, after creating a concatenated data frame for the last 5 hours data, we will get 15*5=75 features. Our model is trained on these features and predicts the number of tweets for the next hour for a test feature also containing 75 features.

We will also be reporting scatter plots of the top 5 features with the highest feature importance attribute in the random forest model against the number of tweets in the next hour.

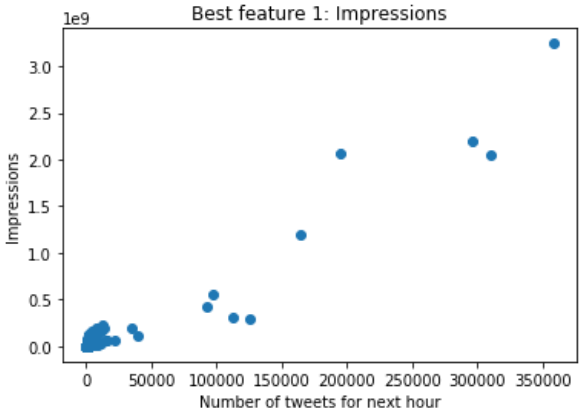
Predictions for the next hour

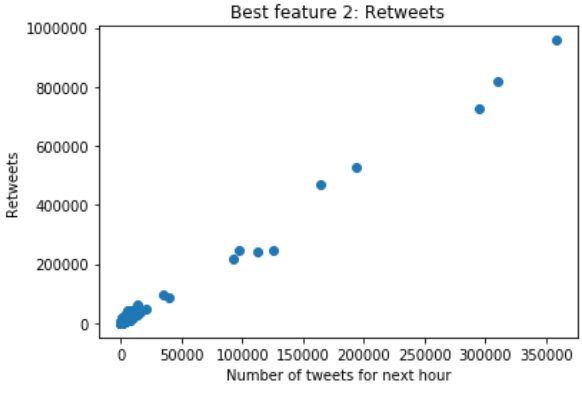
Test File	Prediction for next hour
sample1_period1.txt	218
sample2_period2.txt	88831

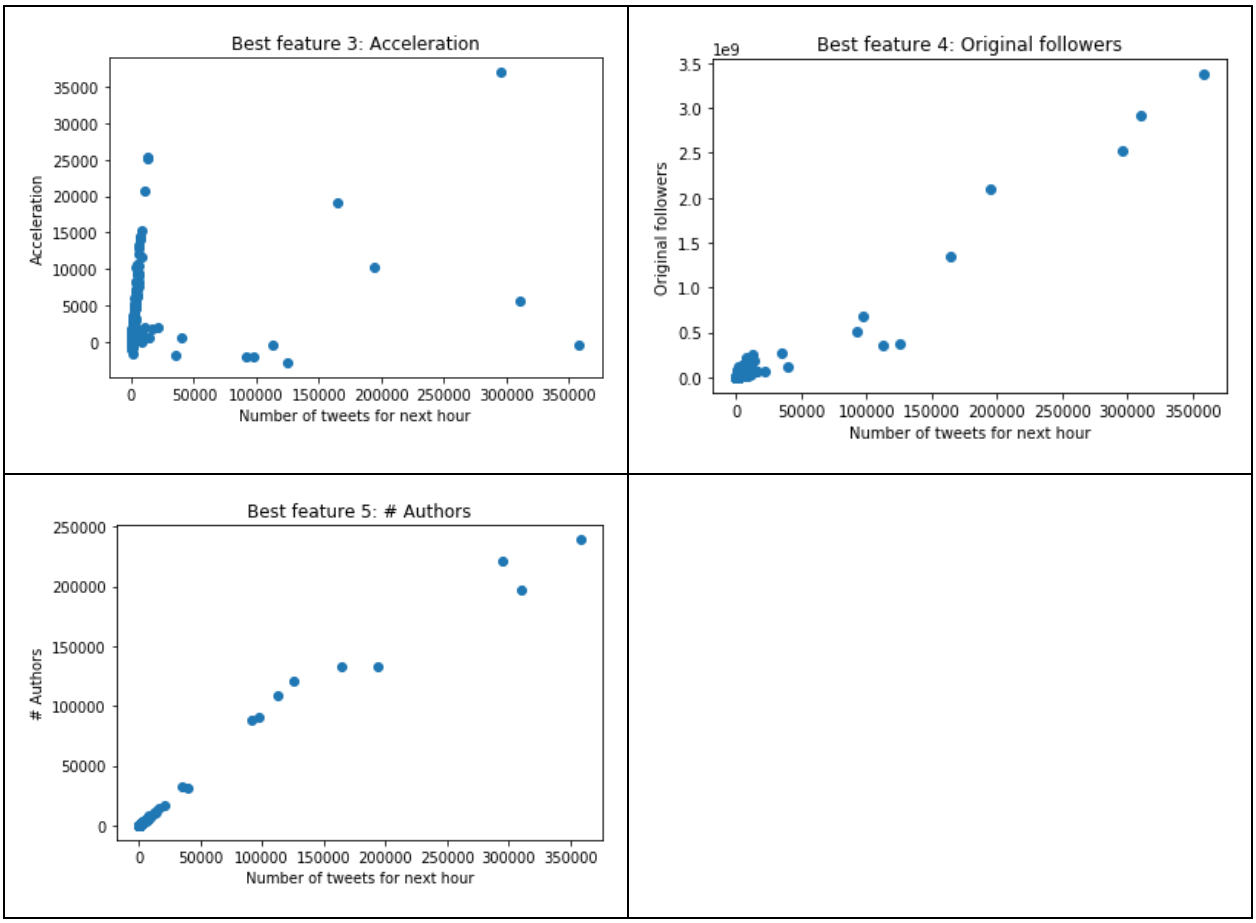
sample3_period3.txt	991
sample4_period1.txt	278
sample5_period1.txt	309
sample6_period2.txt	54481
sample7_period3.txt	81
sample8_period1.txt	73
sample9_period2.txt	1747
sample10_period3.txt	89

From these predictions for the next hour we can see that the number of tweets is significantly higher for the period2 test files. Thus we can guess from the prior performance of our models that period2 probably corresponds to the during Superbowl period. Correspondingly, period1 will be before the Superbowl and period3 after Superbowl.

Scatter plots of predictant (number of tweets for next hour) versus value of top 5 best features







The above figures are scatter plots of the top 5 features with the highest feature importance attribute in the random forest model against the number of tweets in the next hour. From these plots we can see that there is a linear trend in the value of the best feature and the number of tweets in the next hour. In section 1.3 we plotted similar scatter plots for each of the hashtags for all the 3 models we implemented. Contrasting these graphs with the graphs in section 1.3 we can clearly see that the improved performance of our best model on the aggregated data. We can see that the top 5 most important features are impressions, retweets, acceleration, original followers and number of authors. This result is understandable as the impressions denote how many times the tweet appeared in the feed of other users. Retweets, original followers and number of unique authors are also features that would intuitively affect the number of tweets in that particular hour thus affecting the trend for tweets in the next hour as they indicate popularity of the tweet. The graph for acceleration is a straight line concentrated in the early part of the time series. We can conclude that acceleration is probably a very good feature to predict the number of tweets at the beginning since it is in our top 5 features but it probably won't do a good job predicting the number of tweets towards the later half of our time series.

Part 2 : Fan Base Prediction

Building the Dataset

In this part of the project, we aim to classify the user's location based on the textual content of the tweet. We consider only the tweets included in the file containing the #superbowl hashtag. Out of all the tweets in this file, we filter out the ones with user's location in either Washington or Massachusetts.

The location field of the tweet is not very structured and getting the right location can be difficult. In the beginning, we tried to get the right tweets using regular expressions by matching the location with either 'WA' or 'Wash' for Washington and 'MA' or 'Mass' for Massachusetts. But this lead to a lot of tweets that weren't from users in Washington or Massachusetts, for example, tweets from users with the location 'Washington, D.C.' were also considered to be in Washington.

So we used another approach and split the tweet into words and matched those words with a list of correct locations and removed the tweets including words with incorrect locations like 'D.C' matched from an incorrect location list.

After getting the required tweets, we generated the target values as '0' for Washington and '1' for Massachusetts.

Total Number of Tweets Extracted : 40905 (19161 - Washington, 21744 - Massachusetts)

Feature Extraction and Selection

After the dataset was built, we split it into training and testing set with 90% training data and 10% testing data.

We then performed tokenization on the tweets and converted them to TF-IDF vector representations. In the process of tokenization, we also disregarded special symbols like ',.-:/()?{}*$#&' as well as any non-ascii characters such as emoticons.

We also performed Latent Semantic Indexing (LSI) and Non-Negative Matrix Factorization (NMF) on the TF-IDF vectors generated. We used two settings for the minimum document frequency parameter while building the TF-IDF matrix with min_df = 2 and min_df = 5.

So finally, we had four different types of representations of the tweets as follows :

1. LSI with min_df = 2
2. LSI with min_df = 5
3. NMF with min_df = 2
4. NMF with min_df = 5

In each of the dimensionality reduction techniques, we considered n_components = 50 to convert it to a 50-dimensional vector.

The shape of the matrices obtained were as follows :

Min df	Before Dimensionality Reduction		After Dimensionality Reduction	
	Training Set	Testing Set	Training Set	Testing Set
2	(36814, 10473)	(4091, 10473)	(36814, 50)	(4091, 50)
5	(36539, 4729)	(4060, 4729)	(36814, 50)	(4091, 50)

Classification Algorithms

Linear SVM

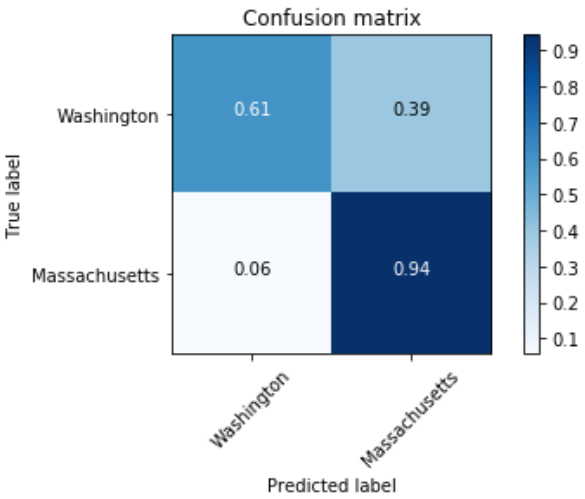
The first classifier we tried was the Linear Support Vector Machine Classifier. In each of the four representations of the tweets, we optimized the penalty parameter 'C' which gave the best results. We used GridSearchCV() with 5-fold cross-validation to obtain the optimal 'C' value. The different C values we considered were [0.0001,0.001,0.01,0.1,1,10,100,1000], ranging from soft margin SVM to hard margin SVM.

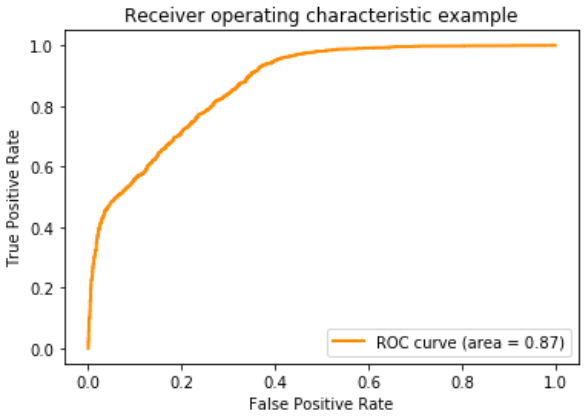
We then fit the model on the training set with the best parameter value obtained and predicted the target values of the testing set. We also calculated the precision, recall and accuracy of each model and plotted the confusion matrix and ROC curve.

LSI with min df = 2

Best Parameter	C = 100
Precision	0.733120113717

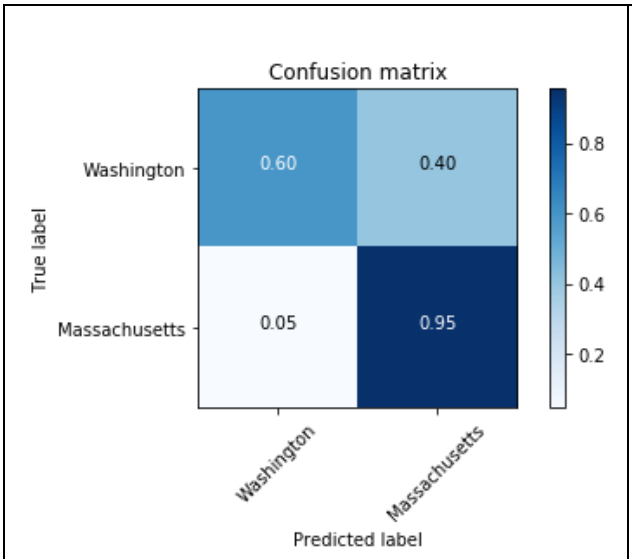
Recall	0.943301326017
Accuracy	0.786115864092

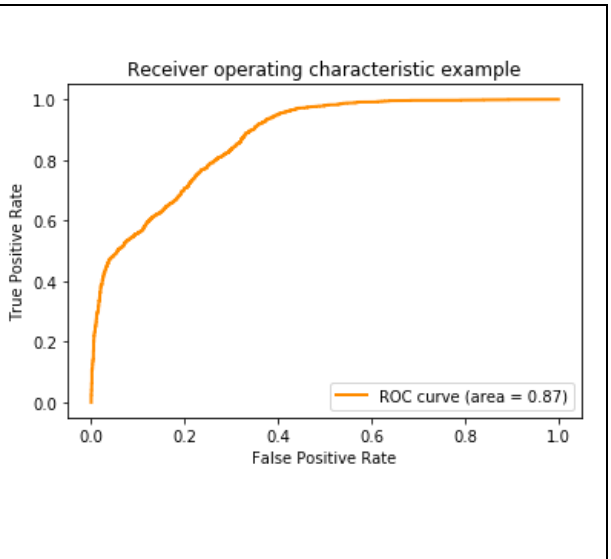




LSI with min df = 5

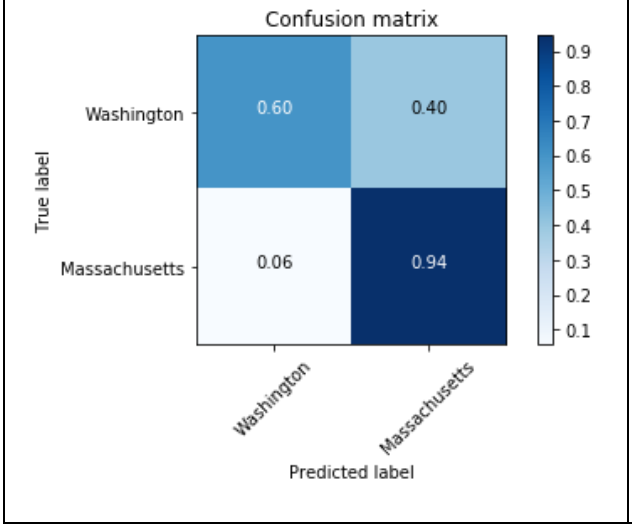
Best Parameter	C = 1
Precision	0.732389380531
Recall	0.946044810242
Accuracy	0.786360303104

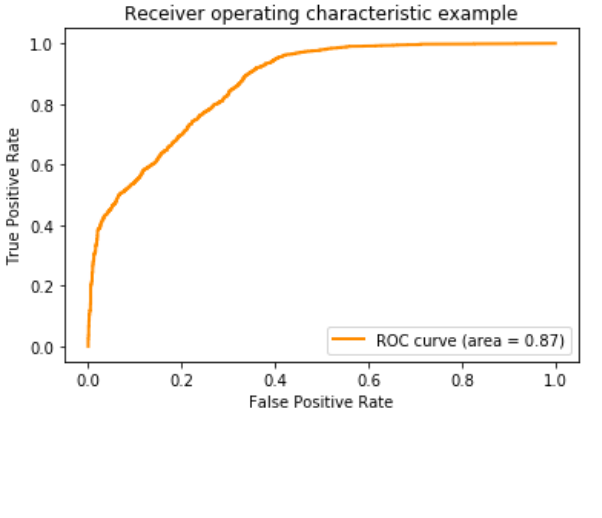




NMF with min df = 2

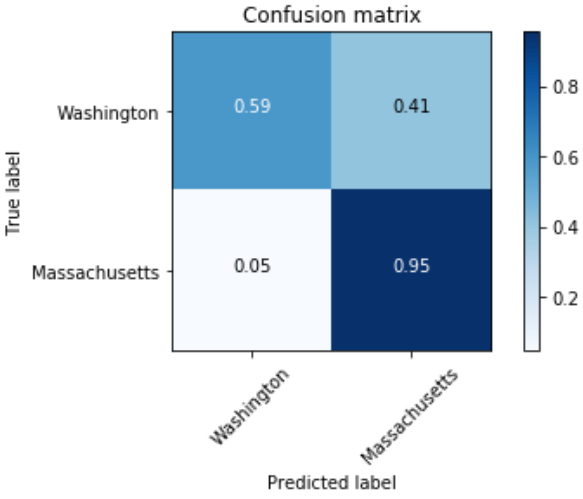
Best Parameter	C = 100
Precision	0.732009925558
Recall	0.944215820759
Accuracy	0.785382547055

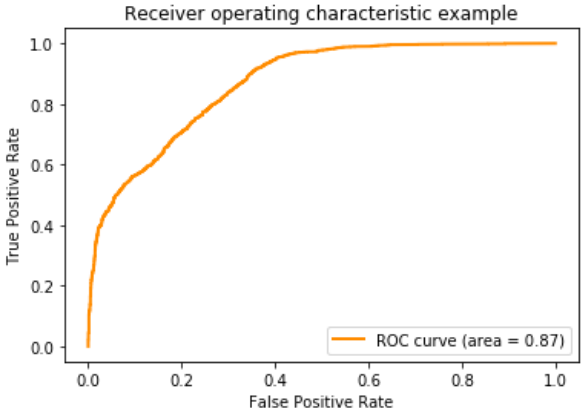




NMF with min df = 5

Best Parameter	C = 0.1
Precision	0.728706624606
Recall	0.950617283951
Accuracy	0.784404791005





Logistic Regression

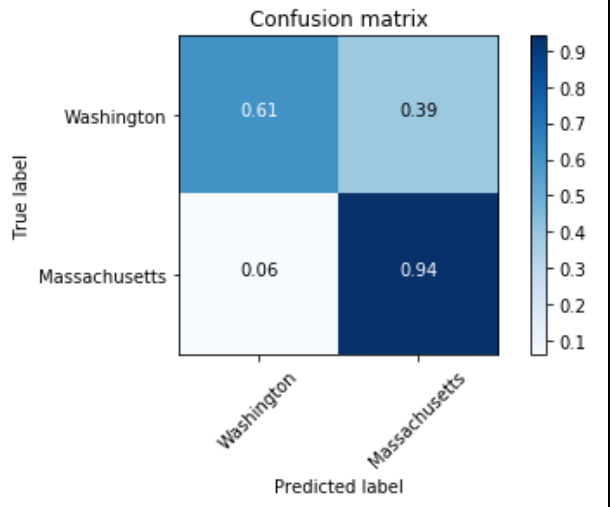
We then performed Logistic Regression to classify the tweets using both L1 and L2 regularization. In each type of regularization, we also optimized over the value of parameter 'C' (inverse of regularization strength) using GridSearchCV(), for each type of data representation. The parameter 'C' is optimized over values [0.001,0.01,0.1,1,10,100,1000].

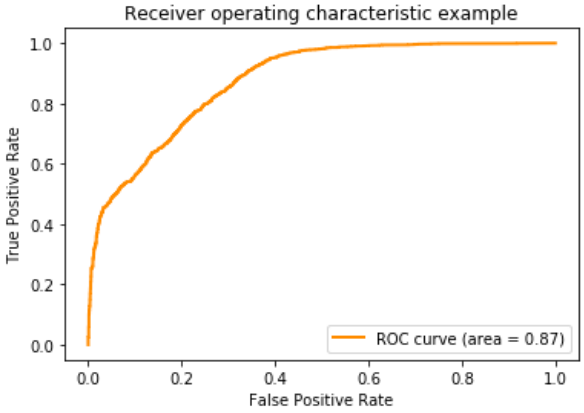
We then fit the model on the training set with the best parameter value obtained and predicted the target values of the testing set. We also calculated the precision, recall and accuracy of each model and plotted the confusion matrix and ROC curve.

L1 Regularization

LSI with min df = 2

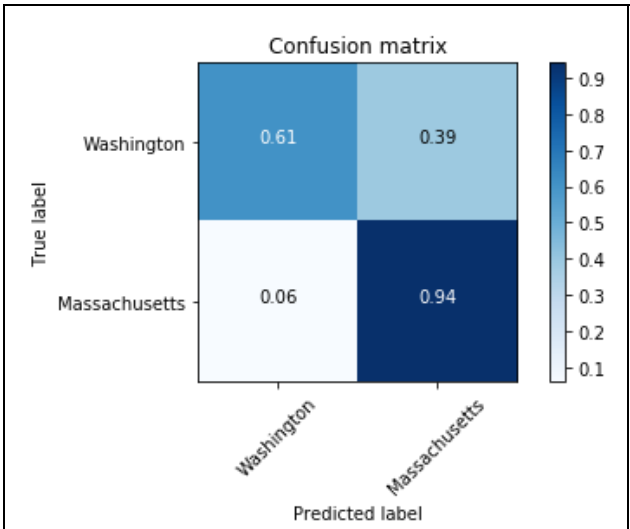
Best Parameter	C = 1
Precision	0.733689839572
Recall	0.941015089163
Accuracy	0.785871425079

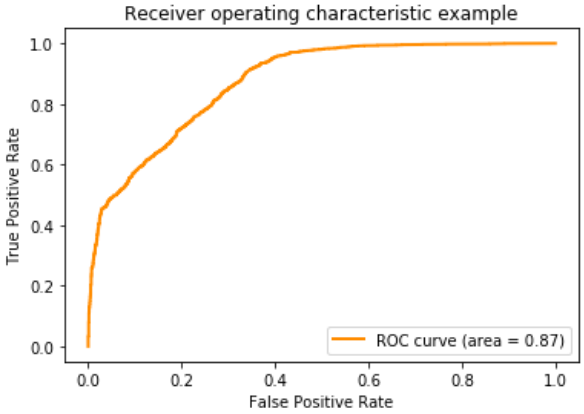




LSI with min df = 5

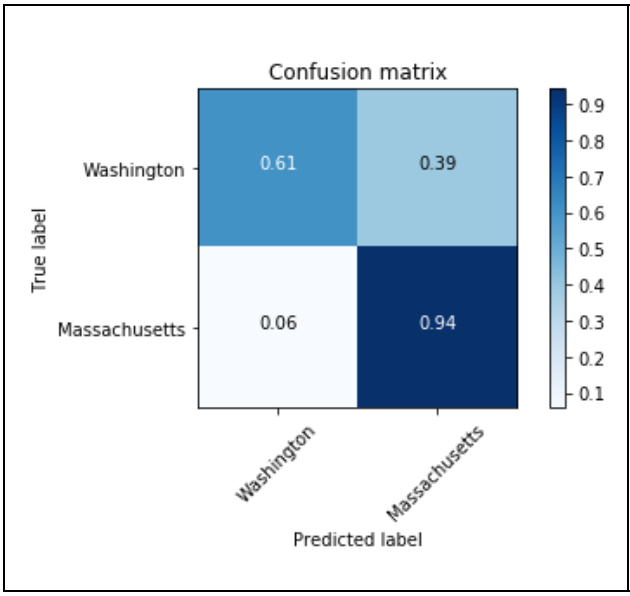
Best Parameter	C = 100
Precision	0.73483226267
Recall	0.941472336534
Accuracy	0.787093620142

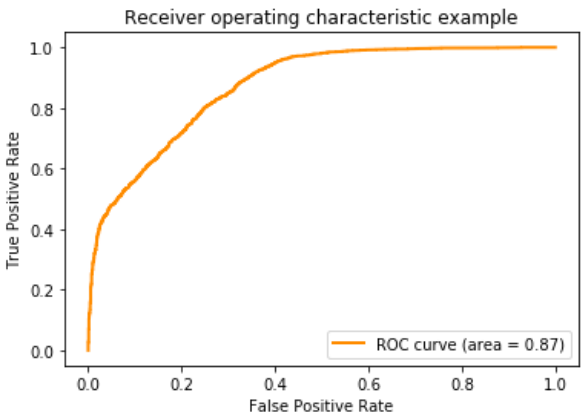




NMF with min df = 2

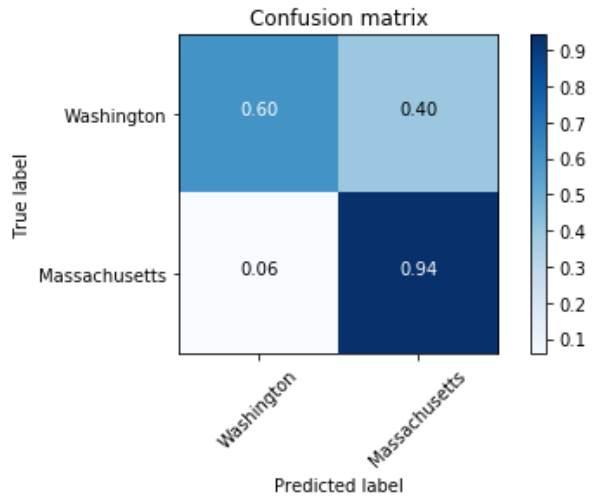
Best Parameter	C = 1000
Precision	0.733879586747
Recall	0.941929583905
Accuracy	0.786360303104

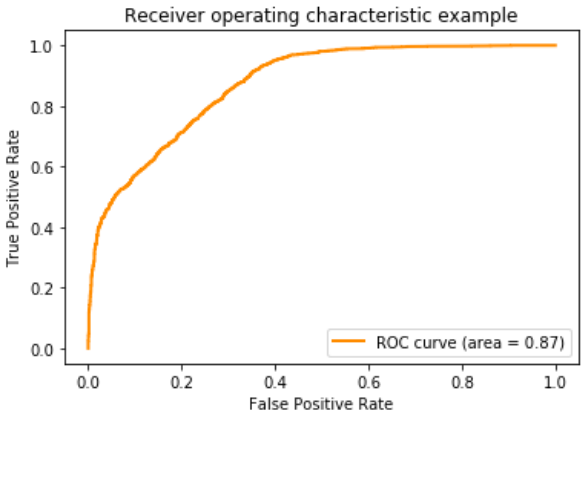




NMF with min df = 5

Best Parameter	C = 1
Precision	0.732314255244
Recall	0.941929583905
Accuracy	0.78489366903

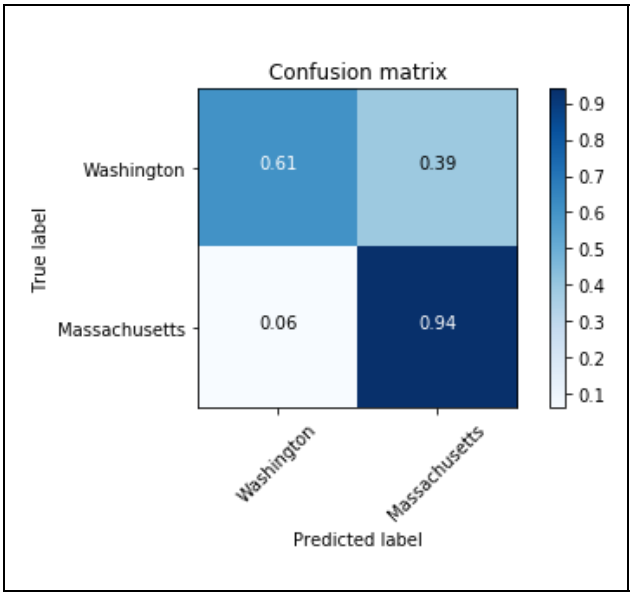


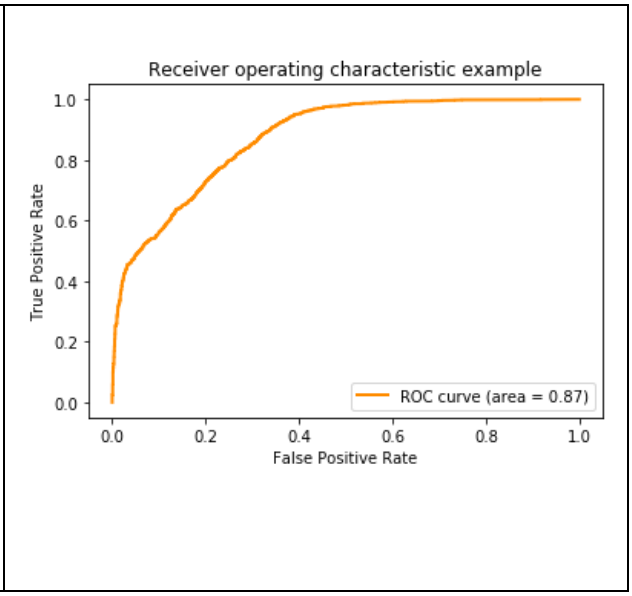


L2 Regularization

LSI with min df = 2

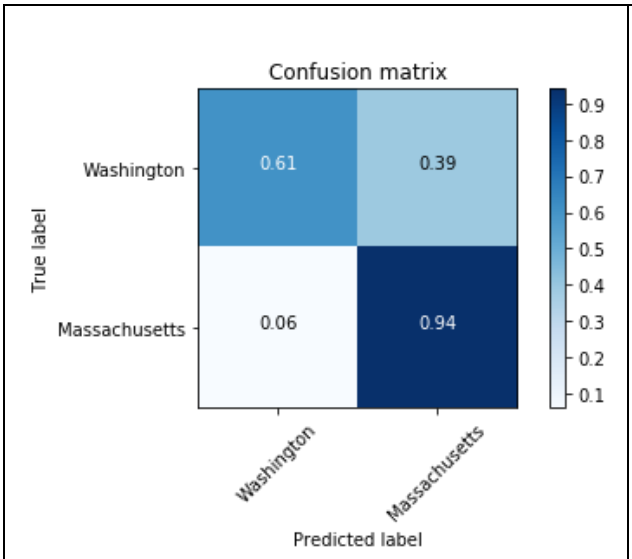
Best Parameter	C = 100
Precision	0.734190782422
Recall	0.939643347051
Accuracy	0.785871425079

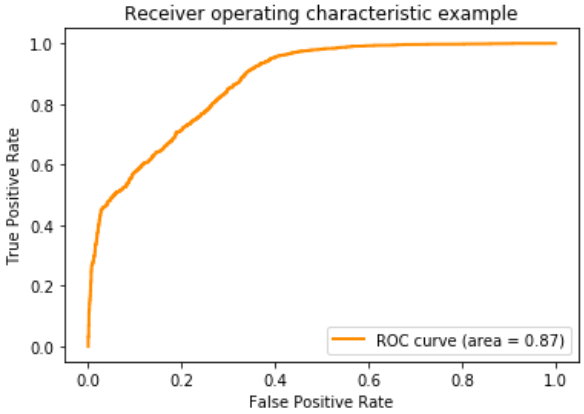




LSI with min df = 5

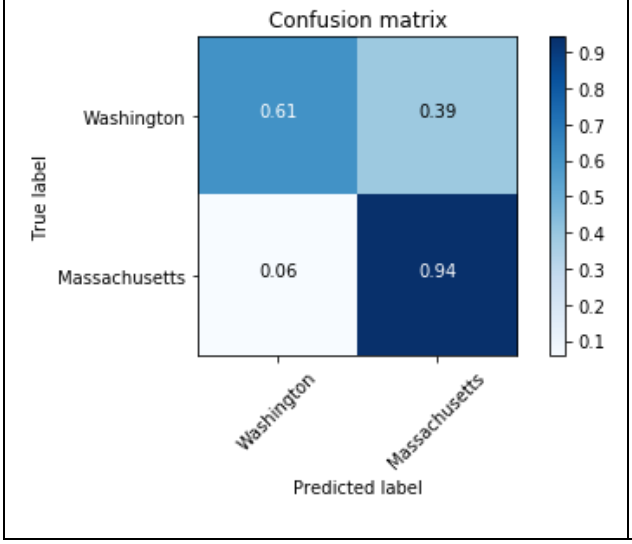
Best Parameter	C = 1000
Precision	0.734737593717
Recall	0.941015089163
Accuracy	0.786849181129

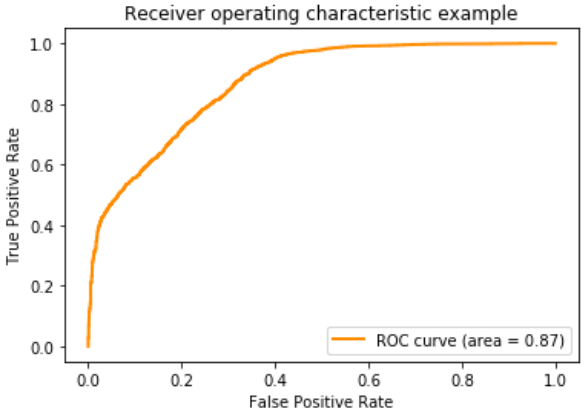




NMF with min df = 2

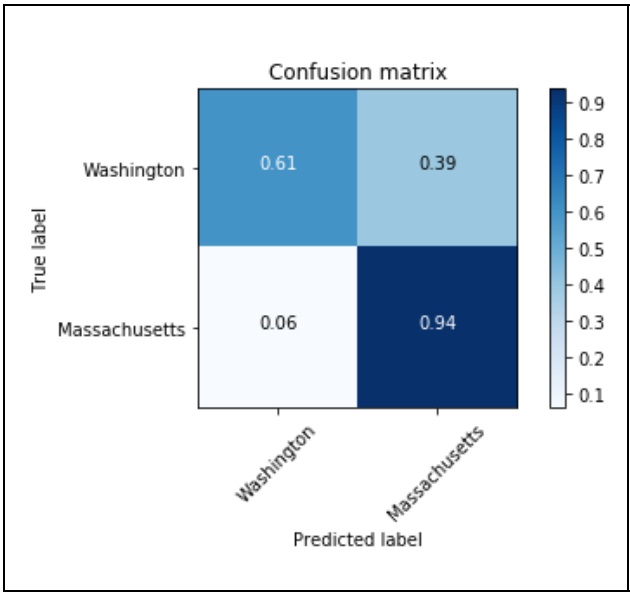
Best Parameter	C = 1000
Precision	0.733879586747
Recall	0.941929583905
Accuracy	0.786360303104

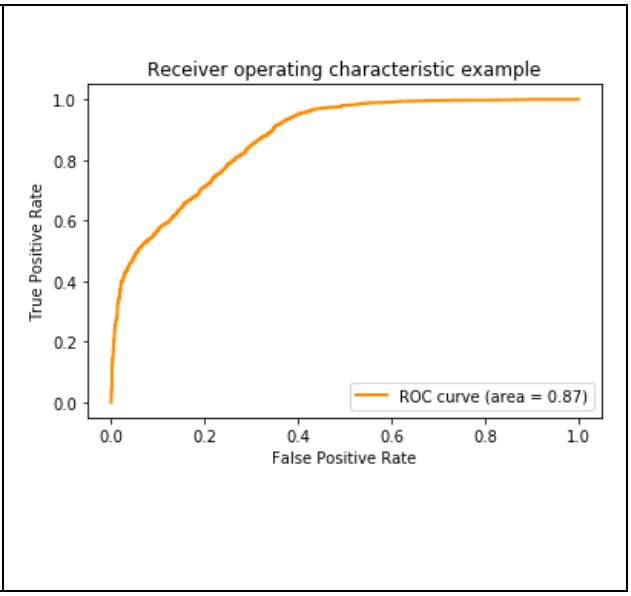




NMF with min df = 5

Best Parameter	C = 10
Precision	0.732334047109
Recall	0.938271604938
Accuracy	0.783671473967





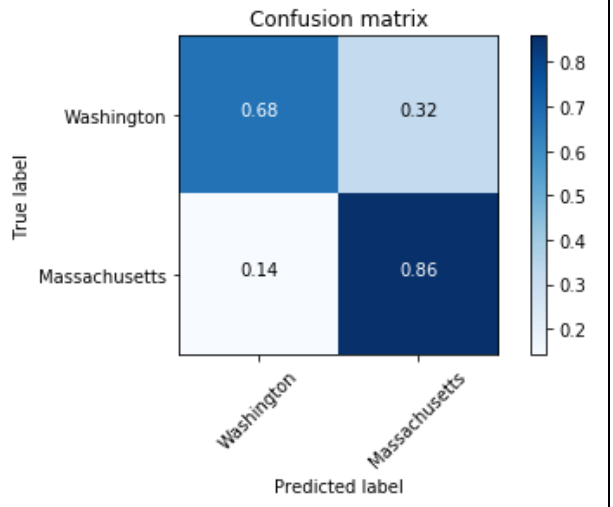
Random Forest Classifier

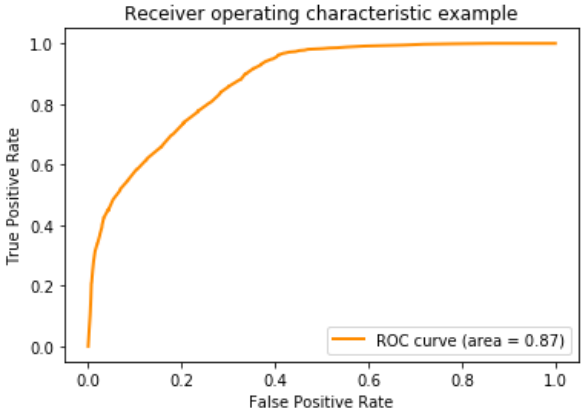
We then performed Random Forest Classification to classify the tweets. We also optimized over the value of parameter 'n_estimators' (number of trees in the forest) using GridSearchCV(), for each type of data representation. The parameter 'n_estimators' is optimized over values [10,20,30,40,50].

We then fit the model on the training set with the best parameter value obtained and predicted the target values of the testing set. We also calculated the precision, recall and accuracy of each model and plotted the confusion matrix and ROC curve.

LSI with min df = 2

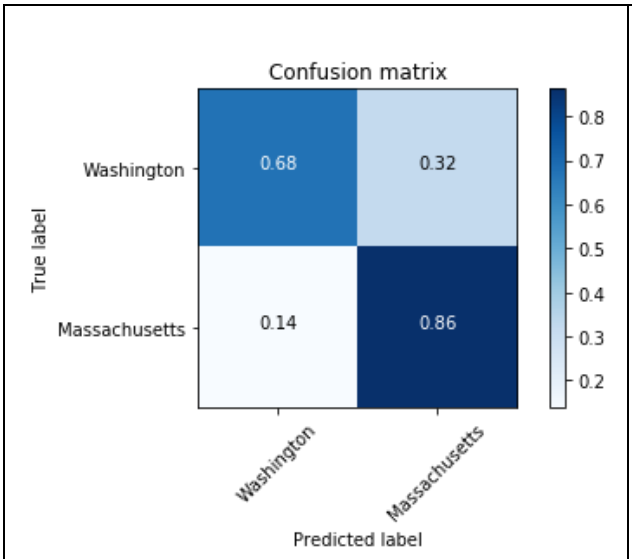
Best Parameter	50
Precision	0.753007217322
Recall	0.858710562414
Accuracy	0.773893913469

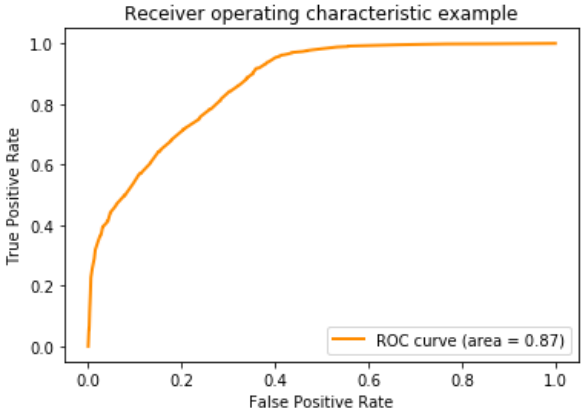




LSI with min df = 5

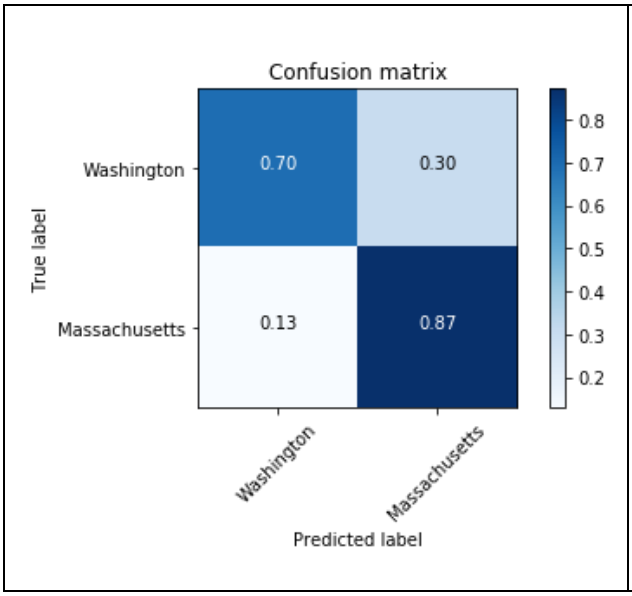
Best Parameter	50
Precision	0.753594249201
Recall	0.862825788752
Accuracy	0.775849425568

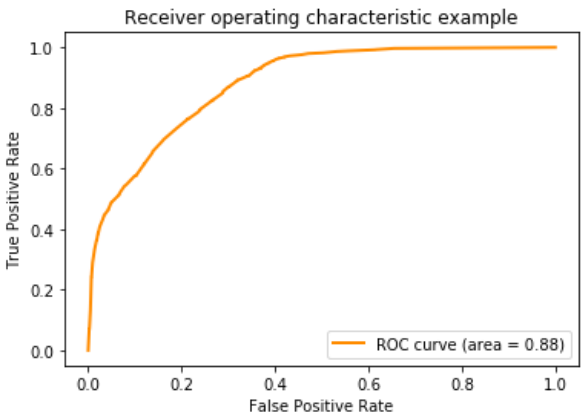




NMF with min df = 2

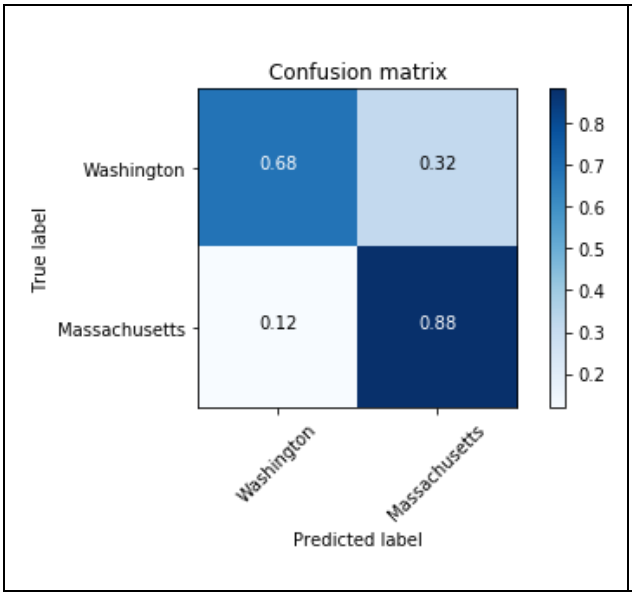
Best Parameter	50
Precision	0.767497988737
Recall	0.872427983539
Accuracy	0.790515766316

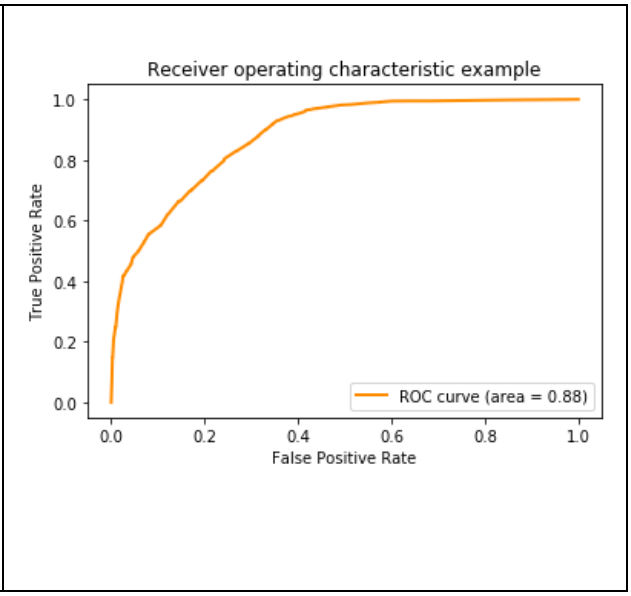




NMF with min df = 5

Best Parameter	50
Precision	0.762149348084
Recall	0.882030178326
Accuracy	0.789782449279





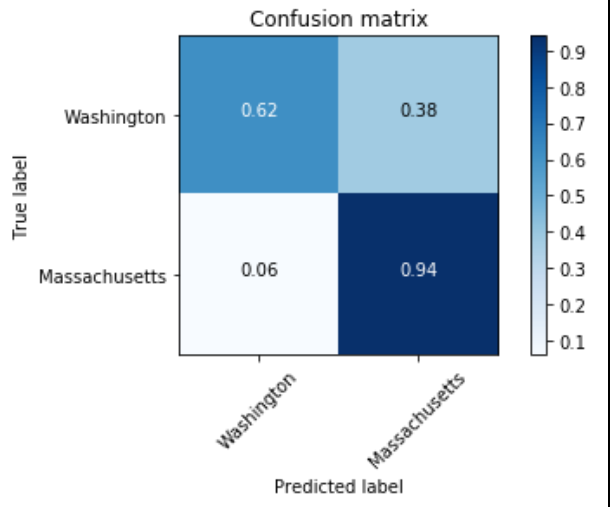
Neural Network Classifier

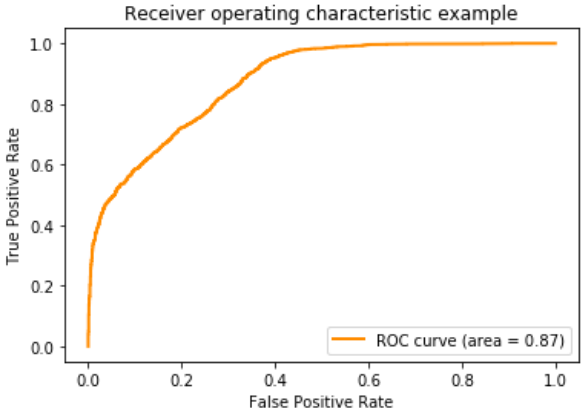
We then performed Neural Network Classification using MLPClassifier() from sklearn to classify the tweets. We also optimized over the type of activation function using GridSearchCV(), for each type of data representation. The parameter is optimized over values ['identity', 'logistic', 'tanh', 'relu'].

We then fit the model on the training set with the best parameter value obtained and predicted the target values of the testing set. We also calculated the precision, recall and accuracy of each model and plotted the confusion matrix and ROC curve.

LSI with min df = 2

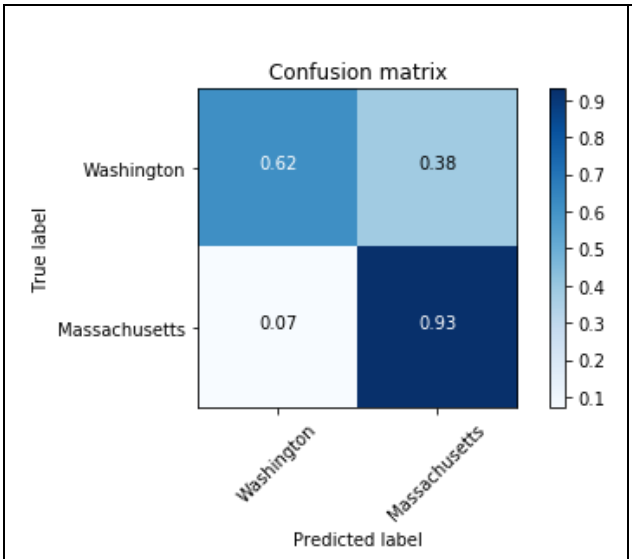
Best Parameter	relu
Precision	0.738599640934
Recall	0.940557841792
Accuracy	0.790271327304

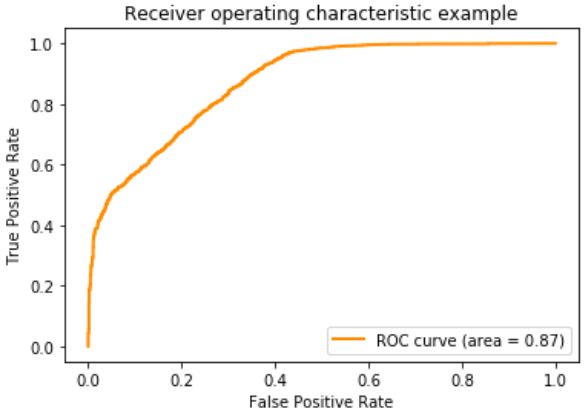




LSI with min df = 5

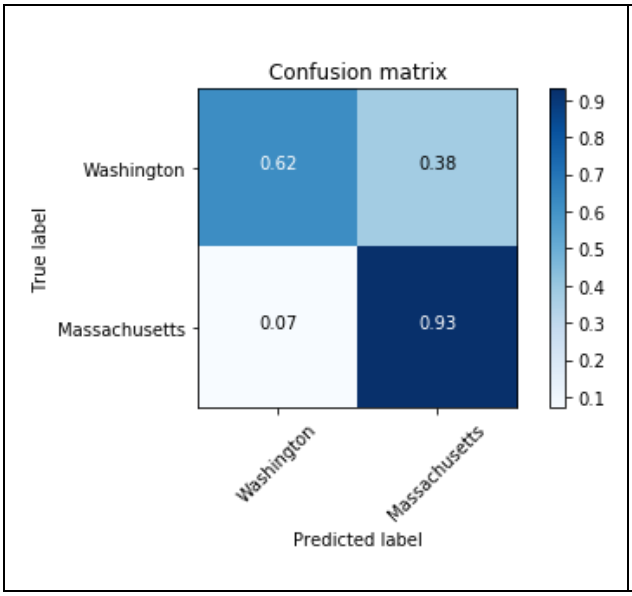
Best Parameter	relu
Precision	0.736156351792
Recall	0.930041152263
Accuracy	0.784404791005

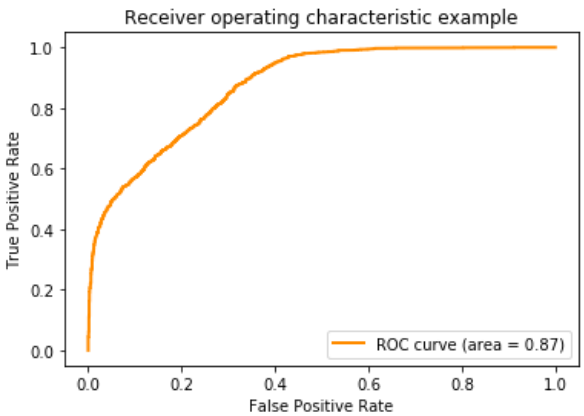




NMF with min df = 2

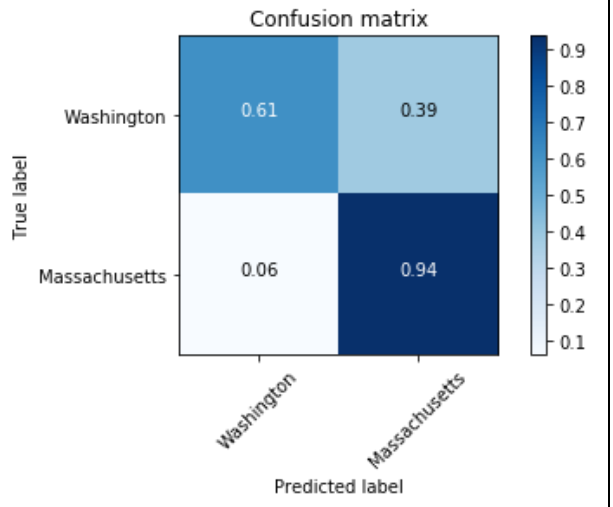
Best Parameter	relu
Precision	0.73892519971
Recall	0.930498399634
Accuracy	0.787093620142

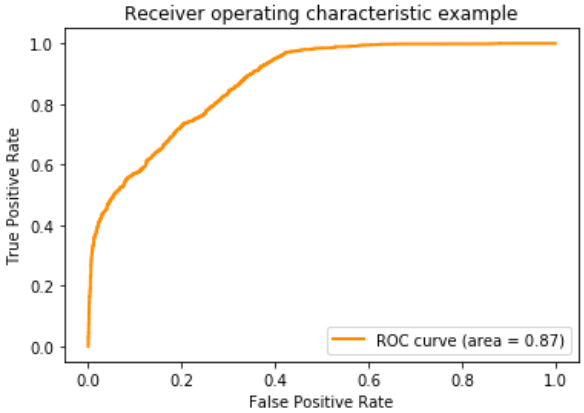




NMF with min df = 5

Best Parameter	relu
Precision	0.735652797704
Recall	0.937814357567
Accuracy	0.786604742117





Summary

The table below lists the best model obtained for each type of classifier based on the accuracy metric.

Classifier	Best Model	Accuracy
Linear SVM	LSI with min_df = 5 ; C = 1	0.786360303104
Logistic Regression	L1 Norm; LSI with min_df = 5; C = 100	0.787093620142
Random Forest	NMF with min_df = 2; n_estimators = 50	0.790515766316
Neural Network	LSI with min_df = 2; activation = relu	0.790271327304

- In all of the different classifiers we tried, there was not much difference in the accuracy of the models. The best model was the Random Forest Classifier and NMF with min_df = 2 which gave an accuracy of 0.7905.
- Most of the models gave better results with LSI rather than NMF, except Random Forest which performed better on NMF and in fact gave the best results.
- Logistic regression performed better with weaker regularization with C = 100, while Linear SVM performed better with C = 1. But, both of them did not perform well with stronger regularization with C < 1.
- Neural Network gave results pretty close to Random Forest, with the best scores achieved using the relu activation.
- All the classifiers had a harder time correctly predicting the location as Washington, as is evident from the confusion matrix, while the tweets from Massachusetts were correctly classified to a much greater extent.

Part 3

Problem Statement:

In this part, we analyze the sentiment for each of the hashtags for each non overlapping and continuous 6/9/12/16 hour windows. We then try to understand the sentiments of the tweets in each hour bin in correlation to the popular news/announcements that may have affected the sentiments.

Implementation:

We are using Python's Textblob module to determine the sentiment of the tweet. We are using 2 measures for each tweet to estimate the sentiments

1. **Polarity**
2. **Subjectivity**

We are then determining our sentiment measure

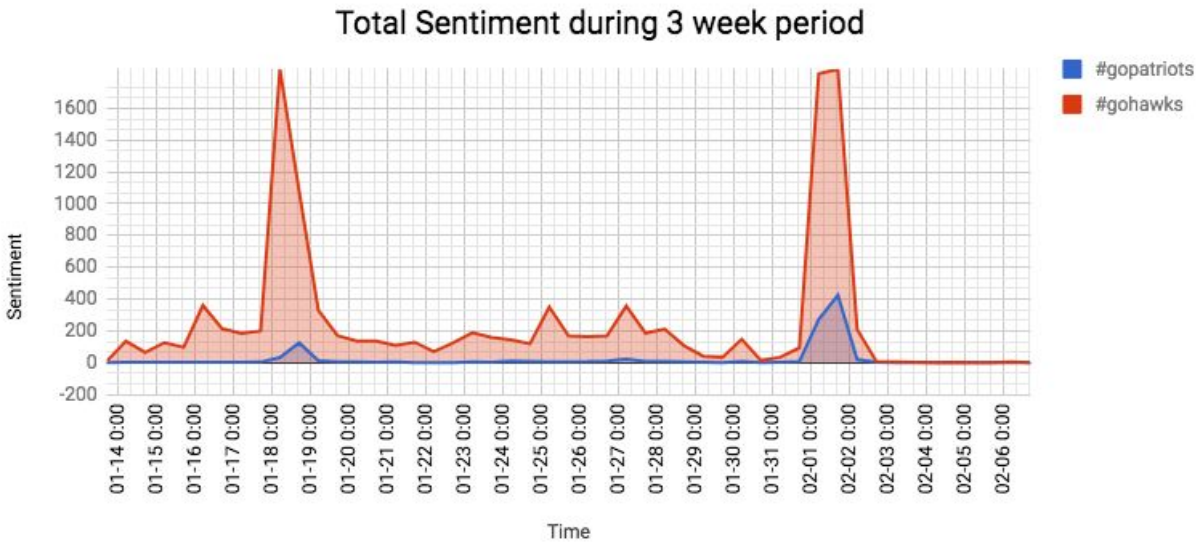
$$m = [\text{polarity} * \text{subjectivity}]$$

After that, for each hashtag, we plot the M Vs the hour window in chronological order. Here M is the sum of m of all tweets in that window. We also plotted the average sentiment for each window.

Analysis: Sentiments Over Three Weeks

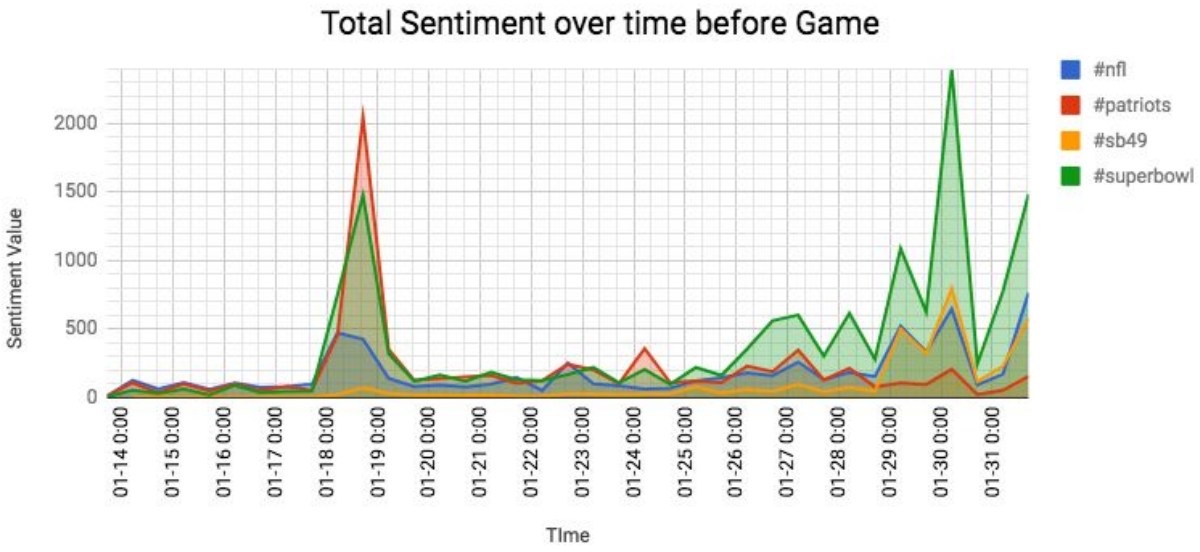
Total Sentiment

We first plotted the total sentiment values obtained by taking six hour windows. The graphs below show the different total sentiment values for each of the six hashtags.



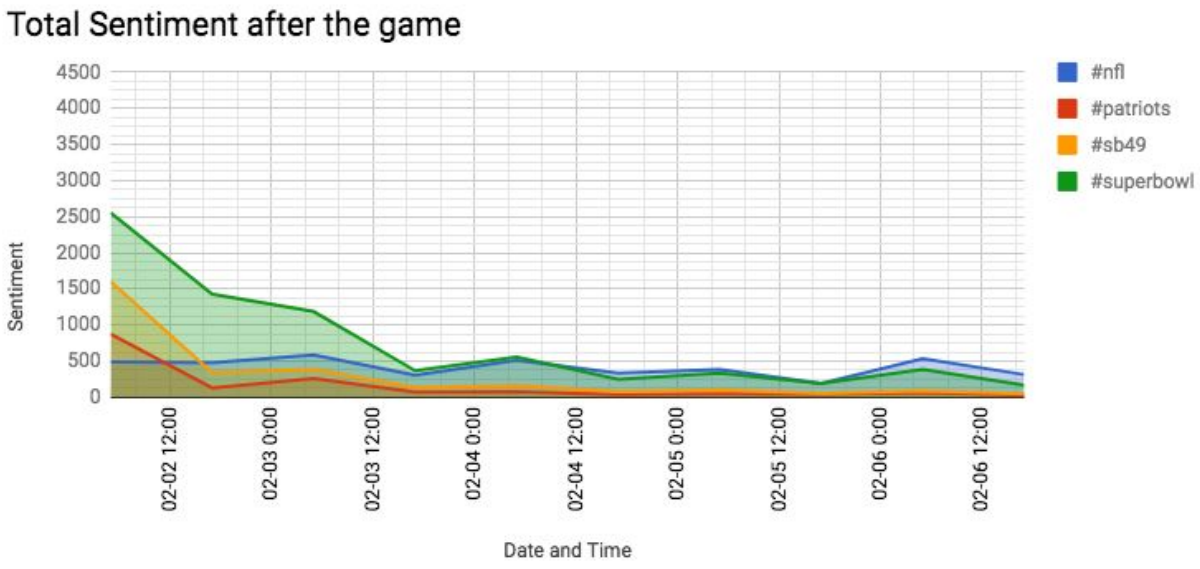
Analysis:

- In the graphs, we can see two peaks one at 18th Jan 2015 and the second one at 1st Feb 2015. This makes sense as these are the dates of the semifinal and final rounds of the superbowl.
- In the graphs of hashtags #gopatriots and #gohawks, the peaks on 18th Jan are much higher as compared to others on the same date, as these teams obviously won the semifinals with led to fans tweeting more about them.
- The sentiments for the hashtag #gohawks seems to be much higher as compared to #gopatriots, but his can be attributed to the fact that the number of tweets with #gopatriots are much smaller in comparison to #gohawks.
- We can also see increases in the sentiment values on the days leading up to the game.



Analysis:

- Similar to the graph above, here too we can observe huge spikes in the sentiment values during the days of the semifinals and the final game.
- For #nfl, #superbowl and #sb49, we can see higher sentiment values towards the end, which shows that more and more people started tweeting positively regarding the NFL in the days leading up to it.

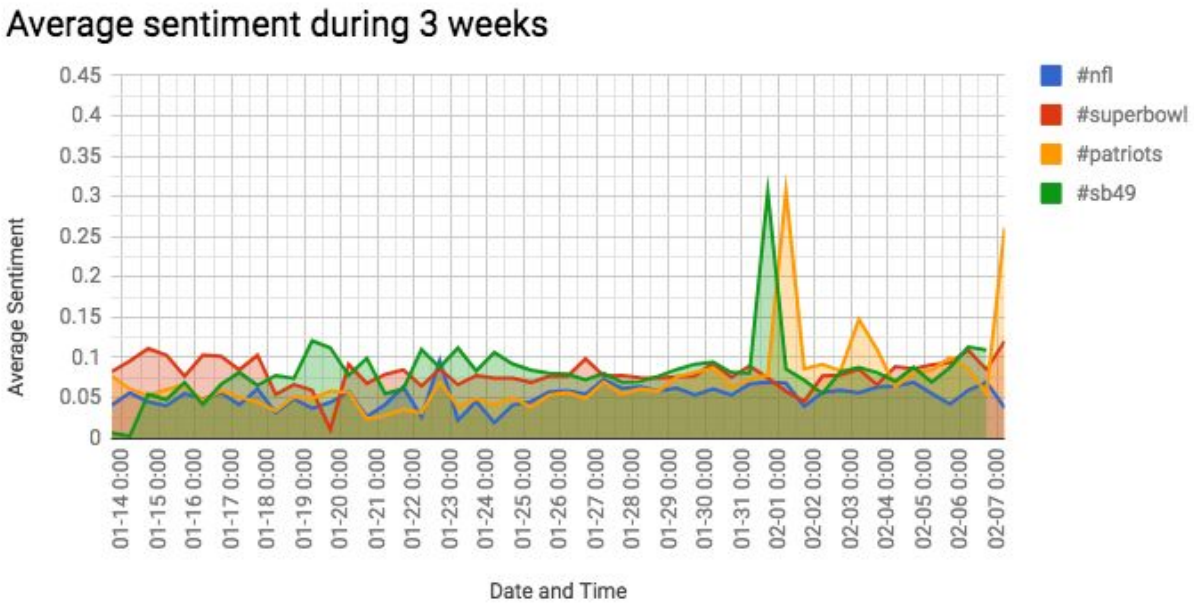


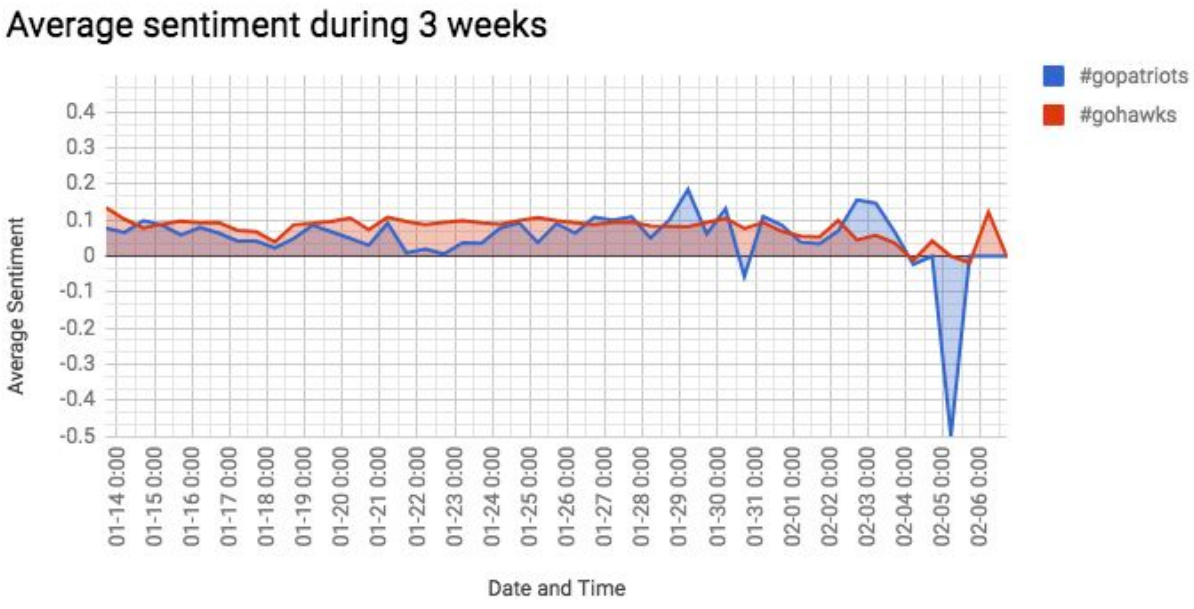
Analysis:

- The total sentiment values after the games gradually decreased. This makes sense as after the game, less and less people would tweet regarding it.
- The sentiments of #superbowl decrease slowly as compared to the rest, which shows that people may have used that hashtag to talk about the game even a few days after it was already over.

Average Sentiment

We then plotted the average sentiment values for the same six-hour windows as the previous set of graphs. The graphs we obtained for the different hashtags are as follows :

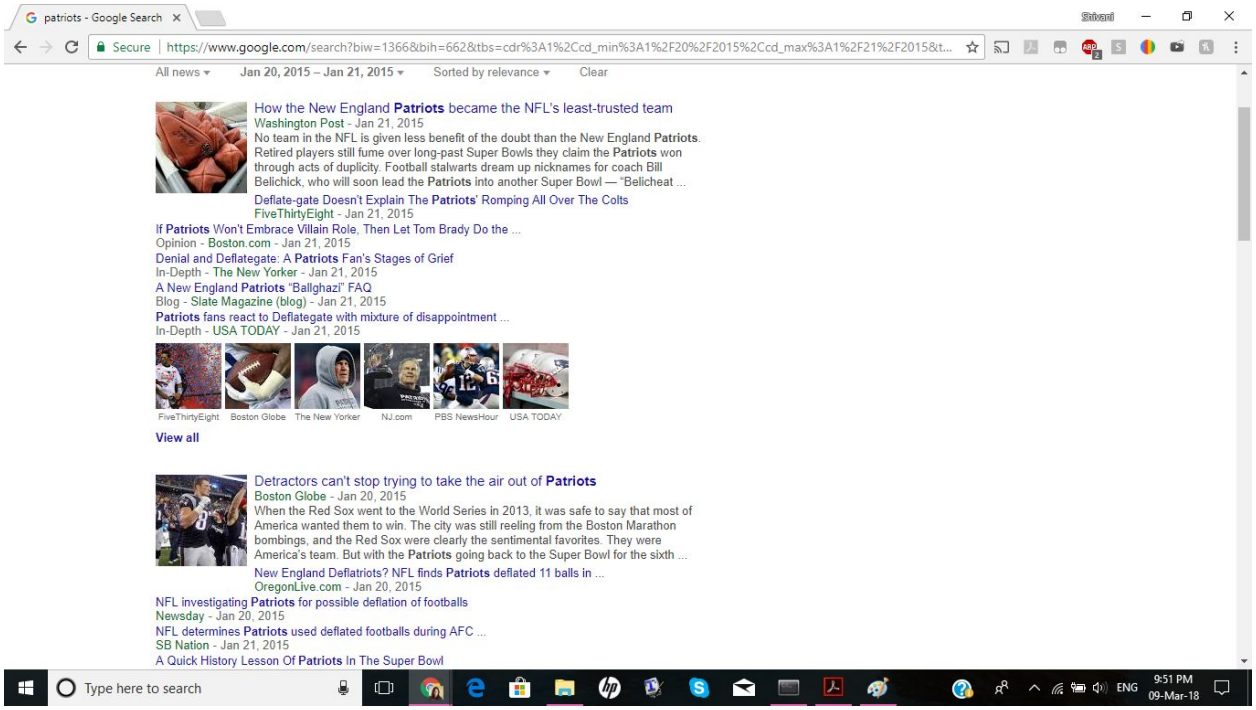




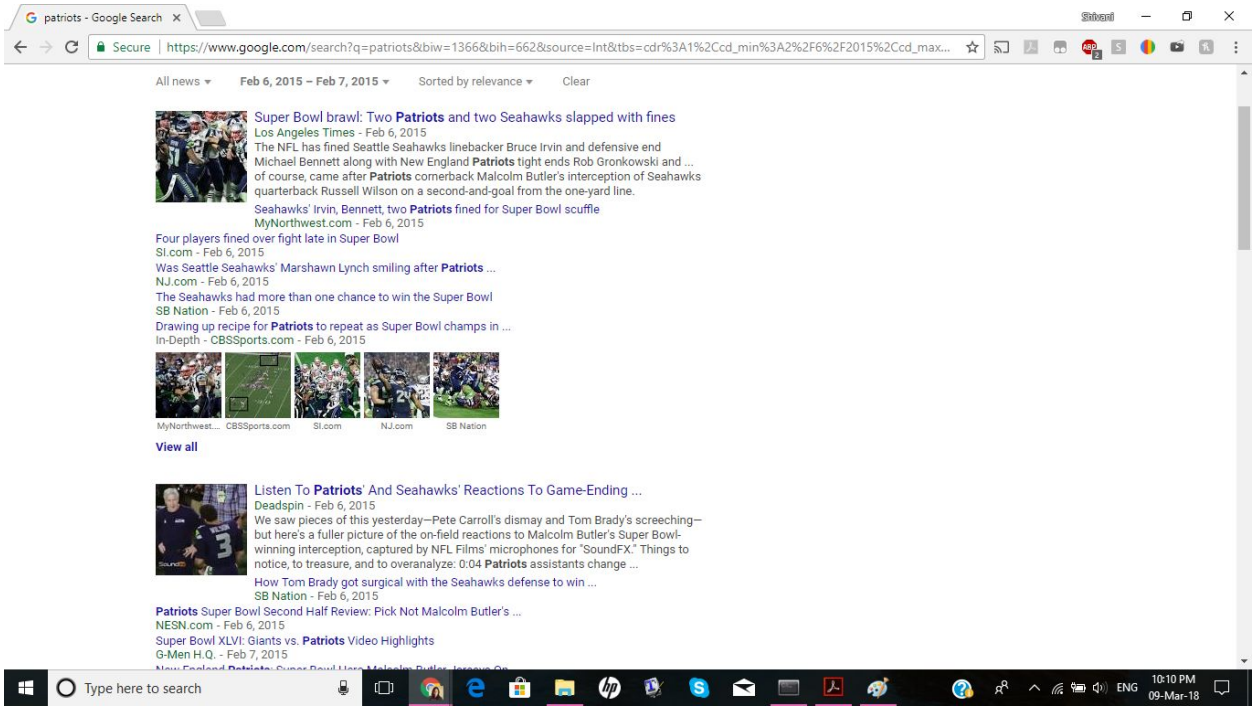
- The hashtag #patriots clearly shows the high average positive sentiment for the day of the superbowl, as the patriots turned out to be the winners. The hashtag #gopatriots

does not show a lot of variation in the average sentiment, this may be attributed to the fact that the number of tweets with #gopatriots are much smaller in number, and fans may have majorly used the hashtag #patriots to show their support.

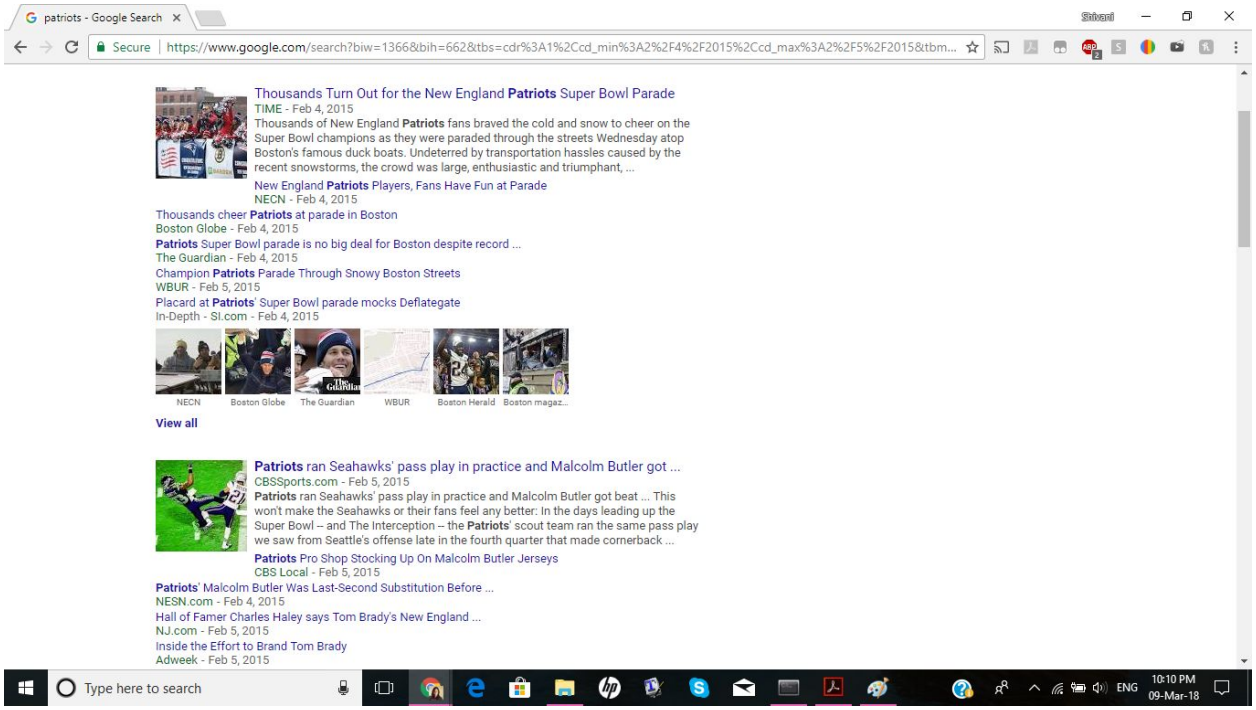
- The hashtag #gohawks does show a slight increase in sentiment value during the superbowl, but the value is much smaller as compared to the #patriots value. This is evident in the fact that the Seahawks lost the superbowl to the Patriots.
- Also, in the days after the superbowl, the average sentiment of the tweets with #gohawks decreases a lot compared to the #patriots tweets.
- Both hashtags #sb49 and #nfl seem to have a gradual increase in sentiment value over the days leading up to the superbowl for the most part.
- Tweets with hashtags #nfl and #superbowl showed slightly decreasing sentiment values over the period of 19th Jan to 22nd Jan. The news of the Patriots using deflated footballs during the game on 18th Jan was released during this time and this news may have caused negative reactions of the fans.
- The screenshot below shows the news :



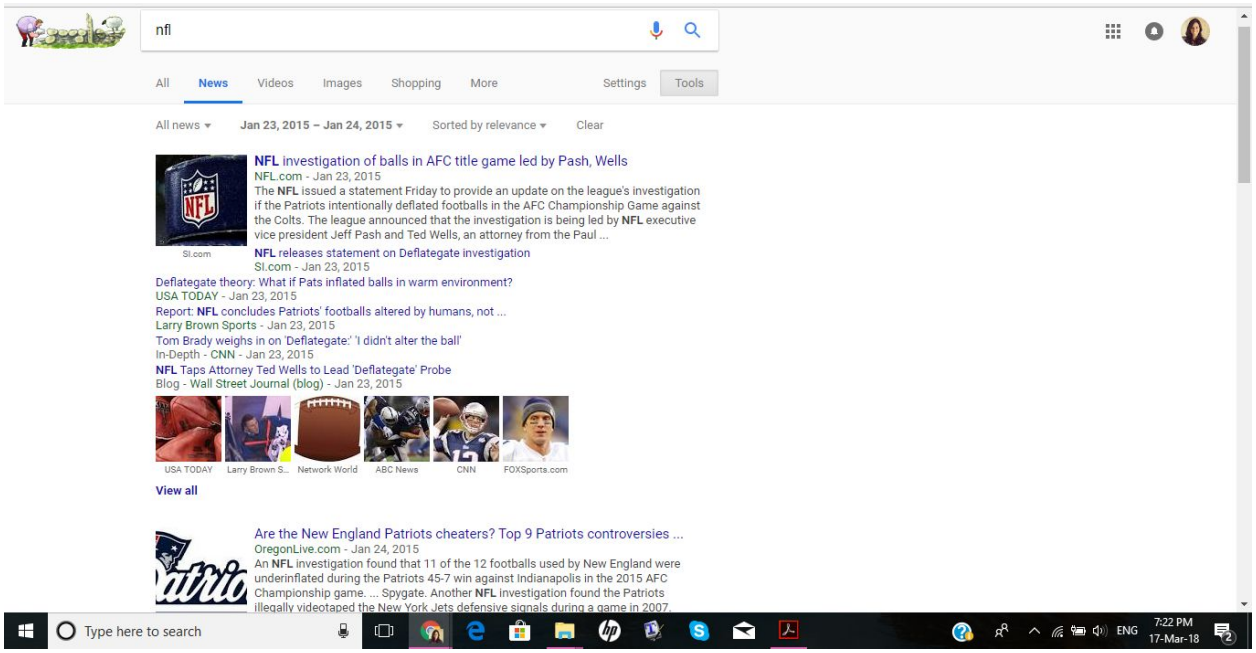
- We can also see a sharp decrease in sentiment for #gopatriots as well as decrease in #nfl and #gohawks during 5th Feb 2015 and 6th Feb 2016. This may be because of the news that two Patriots and two Seahawks players were fined for a Super Bowl scuffle. The screenshot below shows the news :



- We can also see increase in sentiment for #patriots before 4th Feb 2015. This could be attributed to their Super Bowl Parade as the news clipping below shows:



- We can see decrease in sentiment values for #nfl during the period 23rd Jan to 24th Jan. Because of the scandal involving the Patriots using deflated footballs, the NFL released a statement regarding it during this time period as the news shown below shows :

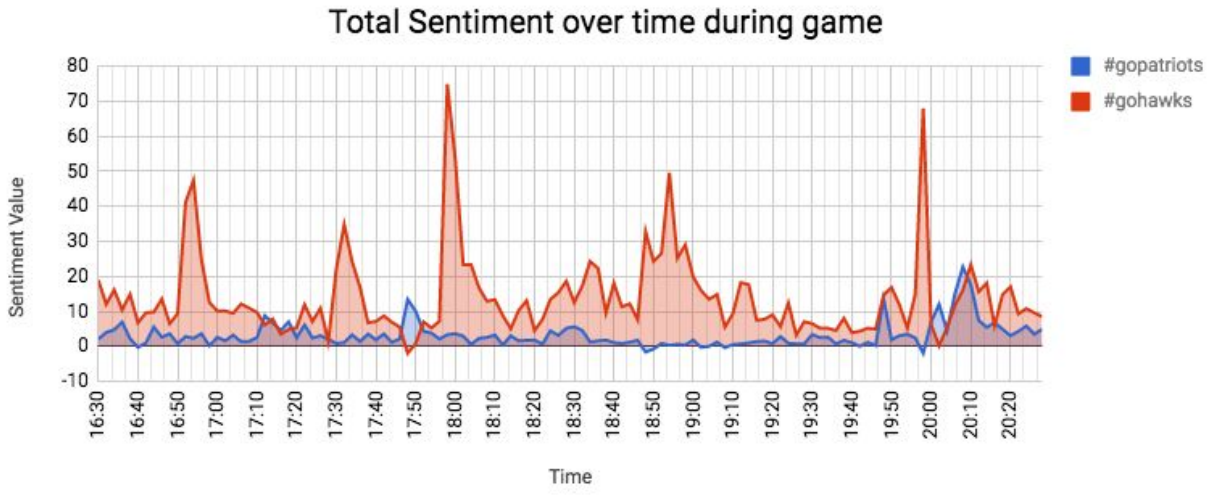


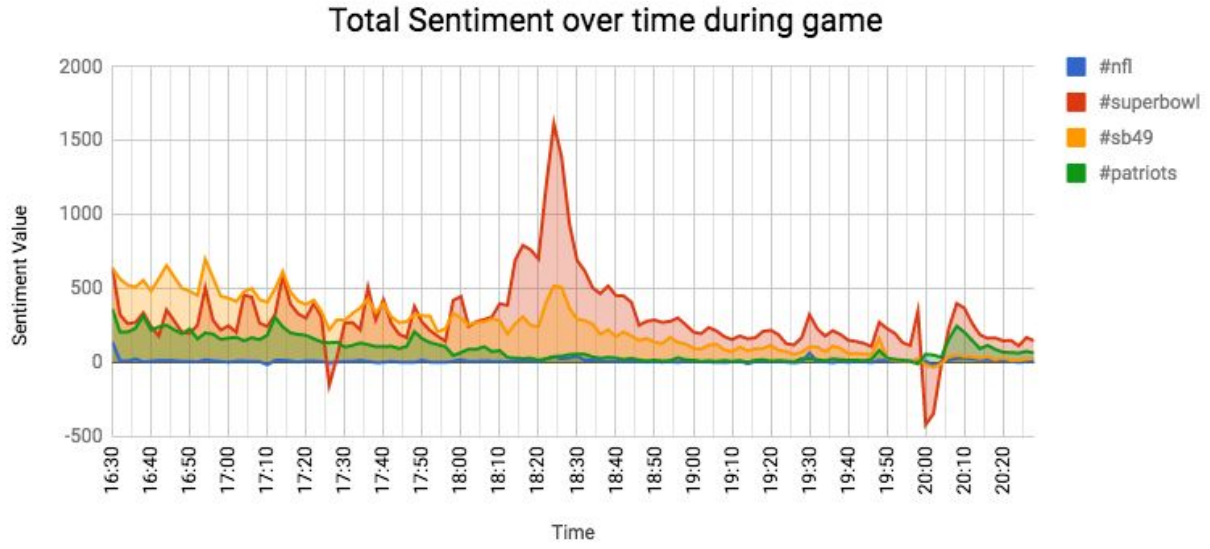
Analysis: Sentiments During the Game

We also wanted to observe the change in sentiment values over the duration of the game. We calculated sentiment values in windows of 2 minutes for this purpose as a lot can happen in a game in just 2 minutes.

Total Sentiment

We first plotted the total sentiment values for each of the six hashtags versus the time at that moment.





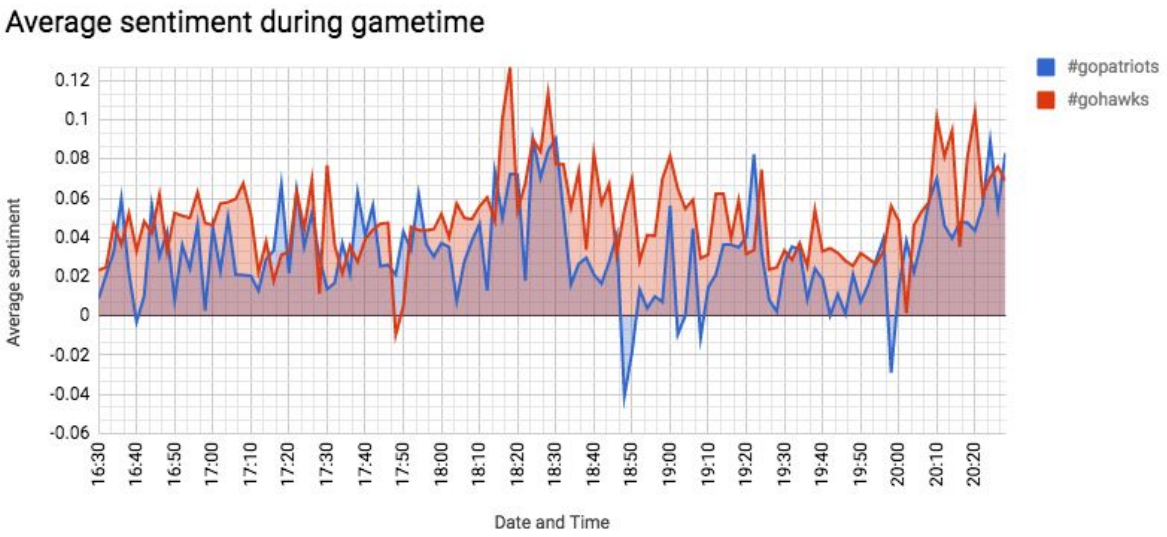
- In order to analyze these graphs, we looked at the scores of the superbowl and how they progressed. The following table shows the time and scores of the two teams at that time during the game.

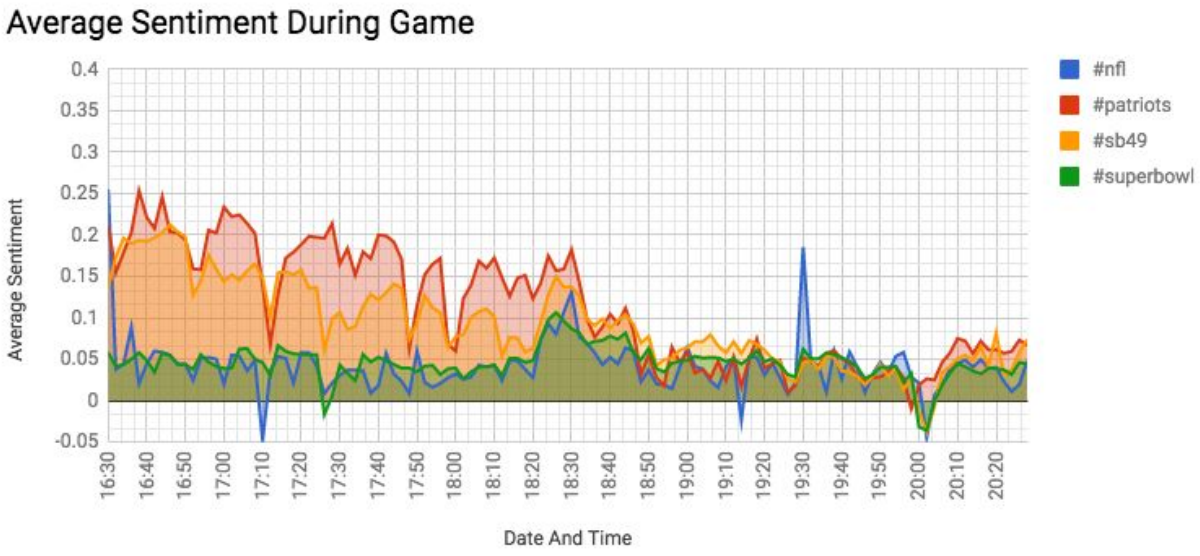
Time	Seahawks Score	Patriots Score
17:12	0	7

17:33	7	7
17:47	7	14
17:58	14	14
18:39	17	14
18:53	24	14
19:27	24	21
19:47	24	28

- Using the information from this table, we can clearly analyse the sentiments from the graphs. For the tweets with hashtags #gopatriots or #patriots, we can see increase in sentiment values around the times the patriots scored like 17:12, 17:47, 19:27 and 19:47. We can also see high sentiments around and after 20:00, after which the patriots won the superbowl.
- And we can observe a similar pattern for the tweets with the hashtag #gohawks, for when the Seahawks scored, around the times 17:33, 17:58 and 18:53. We can also see a huge spike in the sentiments around the end of the game, which may be explained by the fans making hopeful tweets for the Seahawks to score.
- We can also observe a decrease in sentiment value for #gohawks around 17:50, as at the time the Patriots were leading with a score of 14-7.
- For the graph with #superbowl, we can see a huge spike around the time 18:26, which could relate to the Super Bowl Halftime Show in which Katy Perry performed.

Average Sentiment





- Similar to the previous section, we also plotted the average sentiment values every 2 minutes during the game.
- Although these graphs are not as easy to ascertain as the ones in the previous section, we can still make a few observations.

- We can see lower sentiment values for #patriots and #gopatriots at times when the Seahawks scored. For example, we see a huge decrease in sentiment value in #gopatriots at around 18:53, when the Seahawks were leading with a score of 24-14.
- Similarly for #gohawks, we can see higher sentiment values slightly before, during and slightly after the halftime show, as the Seahawks were in the lead. We also see a decrease in sentiment value around 17:47, when the Patriots scored.
- Like in the last section, we observe a spike in the #superbowl graph, around the halftime show, which could be people tweeting related to Katy Perry's performance.
- At just after 20:00, we an see the sentiment values for every hashtag going negative. This may have been because of the scuffle and fight that broke out on the field just before the end of the game. Seattle linebacker Bruce Irwin seemed to have started the fight and was ejected from the game.