EE219 LARGE SCALE DATA MINING

Popularity Prediction on Twitter

Project-5

Devanshi Patel	504945601	devanshipatel@cs.ucla.edu
Ekta Malkan	504945210	emalkan@cs.ucla.edu
Pratiksha Kap	704944610	pratikshakap@cs.ucla.edu
Sneha Shankar	404946026	snehashankar@cs.ucla.edu

INTRODUCTION

Social Network services such as Twitter have become an integral part of daily lives of millions of users. People communicate and share their ideas with friends,family and pretty much everyone they follow or everyone who follows them on the platform. Tweets also provide a rich collection of data , which is being successfully used in recommendation engine algorithms to know a user's likes or dislikes. Sentiment Analysis is another successful application of machine learning on tweet data. Apart from that, tweets are also being used to know a user's alcoholic behaviour, users' favourite sports team, etc.

The paper "On the Real-time Prediction Problems of Bursting Hashtags in Twitter" provides insightful statistics on how some popular events become bursting topics. This paper also explains how the bursting hashtag popularity can be predicted in real-time. This includes calculating the probability of whether a hashtag will burst, if yes then when will a hashtag burst, and how long will the burst be active, or how soon will the burst fade away. This paper also studies the challenges of real-time prediction of bursting hashtags. We take the ideas present in this paper as a basis for our modelling and study the life-cycle of hashtag bursts during the famous Superbowl 2015. If we can predict whether a topic will burst before the actual burst, it can be useful in many ways.

In Part 1 of our project, Linear Regression is performed on the dataset using features like the number of tweets, the number of retweets for a given tweet, the number of followers of the author of the tweet, maximum number of followers of a user, the time of tweet ,etc. Designing and choosing good features is an important part for modelling. Hence we experiment with different features and choose the best features for each model. We also train a linear regression model for those chosen good features from the dataset and test it. We have performed period based analysis, dividing the time into 3 intervals of before, during and after the Superbowl 2015 event, and use cross validation for model evaluation. The linear regression model will fit a line through the observed values of features, and predicts output for new unseen samples.

In Part 2, we classify a users location as Washington or Massachusetts, purely based on their tweet. Different classification models are used and performance is compared for each model.

In Part 3, we have designed a new problem on our dataset. Using the rich data of a tweet, we have tried to predict whether a team will win or lose, purely based on sentiments of the fans during the event.

Dataset

We perform Popularity Prediction on 2015 Super Bowl tweet data. A football game is supposedly the main attraction of the Super Bowl, but the fanfare surrounding the game seems to subsume it at times. 2015 Super Bowl was played between Seattle seahawks and New England Patriots on February 1, 2015 at University of Phoenix Stadium in Glendale, Arizona. In this game New England Patriots defeated the Seattle Seahawks. The tweet data we use in our analysis is a rich collection of popular hashtags spanning from 2 weeks before the game upto a week after the game.

Hashtags, user-specified strings starting with a # symbol, have been commonly used as identities of topics in Twitter [1].

The data set contains tweet data for following hashtags for SuperBowl event:

1. #gopatriots
2. #gohawks
3. #nfl
4. #patriots
5. #sb49
6. #superbowl

Part 1 Data Analysis

In this part, we try to understand our dataset better. For this, we calculate 3 statistics for each hashtag in our dataset namely Average number of tweets per hour, average number of followers of authors, average number of retweets.

Average number of tweets = Total Number of Tweets/ Total Number of Hours

Average number of followers of users = Total number of followers / Total number of users

Average number of retweets = Total number of retweets/ Total number of tweets

Problem 1.1:

Hashtag	Avg. tweets per hour	Avg. followers of users tweeting	Avg. retweets
#gohawks	328.91	2203.93	2.0146
#gopatriots	58.68	1401.90	1.4001
#nfl	444.29	4653.25	1.5385
#patriots	834.26	3309.98	1.7828
#sb49	1528.56	10267.32	2.5111
#superbowl	2297.72	8858.97	2.3883

Table 1 : Statistics for Average number of Tweets, Followers and Retweets

The higher average number of tweets for #superbowl, #sb49 and #nfl indicate how famous the SuperBowl event was and that many people all over the world were interested in the match. We observe that #patriots was more tweeted than #gohawks. This means that New England Patriots share a larger fan base than Seattle Seahawks. This could also mean higher victory chances for New England Patriots.

The average number of followers of users tweeting gives us an idea of how popular the users are and how many times were their tweets retweeted. We observe that higher the number of followers for a user, higher is the corresponding average retweet count for that tweet.

But "Average" may not be the best indicator for popularity of a hashtag. Also an average tweet count of 400-2000 per hour seems too low for an event as popular as SuperBowl 49 with millions of users following, tweeting and retweeting about it. Average counts get regularized by range of the data. Hence, to further confirm on our derivations, we now plot the number of tweets for a hashtag against the hour of the data and analyze.

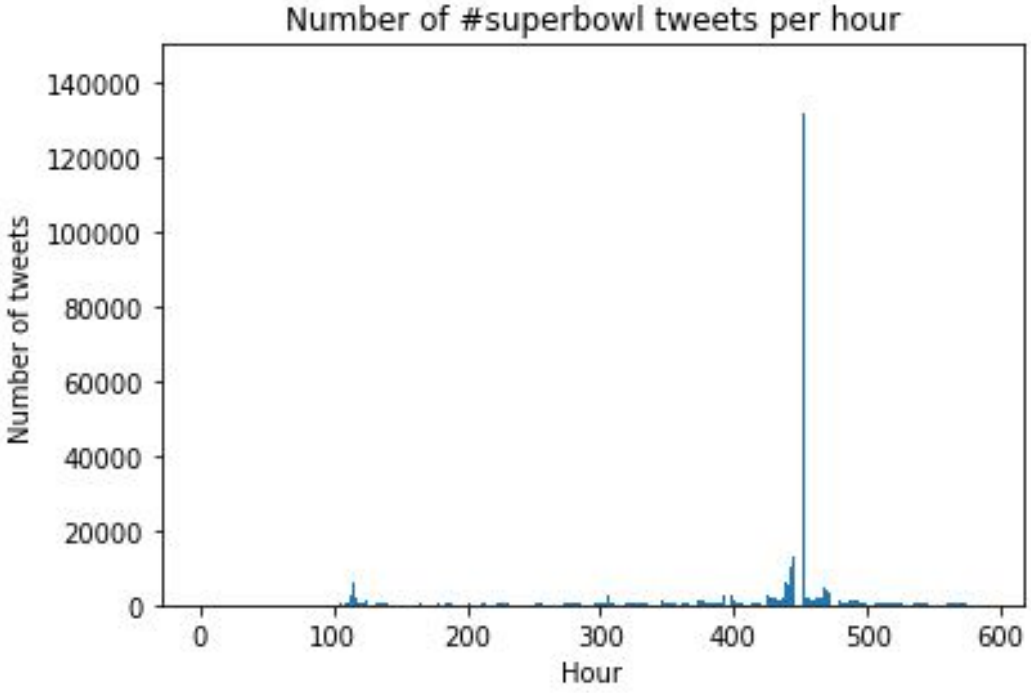
Plot "number of tweets in hour" over time for #SuperBowl and #NFL (a histogram with 1-hour bins). The tweets are stored in separate files for different hashtags and files are named as tweet_[#hashtag].txt.

Solution:

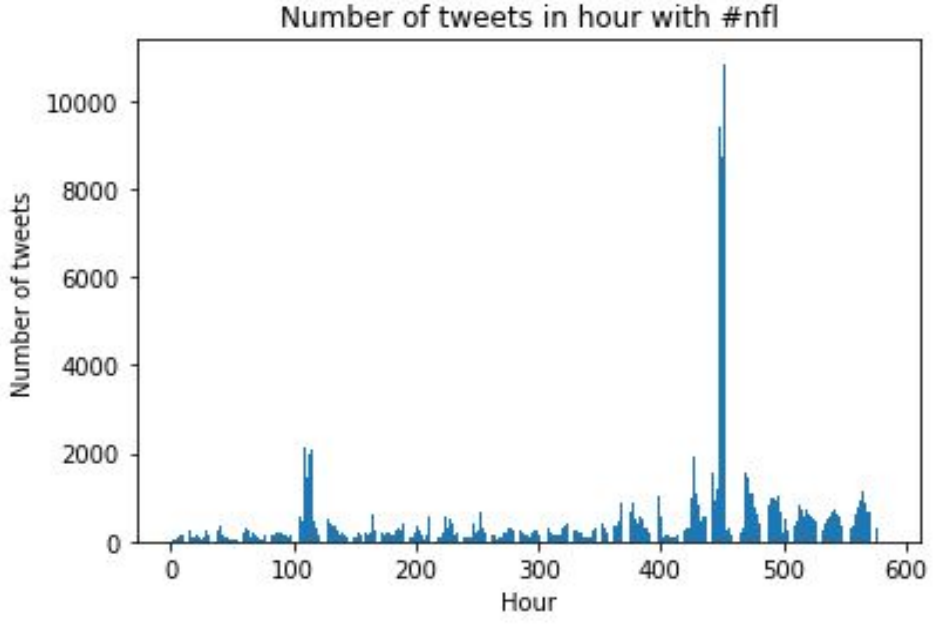
On plotting the number of tweets per hour for each hashtag, we noticed that all the hashtags have **burst** around 450th hour.

"Burstiness" is a phenomenon of Twitter Hashtag, in which a topic of discussion suddenly gains considerable popularity and then quickly fades away [1]. Twitter maintains and provides a list of "Trending Topics". These topics provide users with fresh discoveries and timely updates of important events.

i. Number of **#superbowl** tweets per hour



ii. Number of **#nfl** tweets per hour



We observe that the number of tweets in both #superbowl and #nfl have shooted up i.e. bursted in the **450th hour** (approximately).

While the average number of tweets for #superbowl and #nfl was 2297 per hour and 444 per hour respectively, we observe from the graph above that the maximum count of number of tweets is more than 15000. On verification , we found that for #superbowl maximum number of tweets was 277351 which occurs for a very small duration of few minutes in 450th hour.. Whereas most of the time before and after the burst, the average count is 270 -300, as follows:

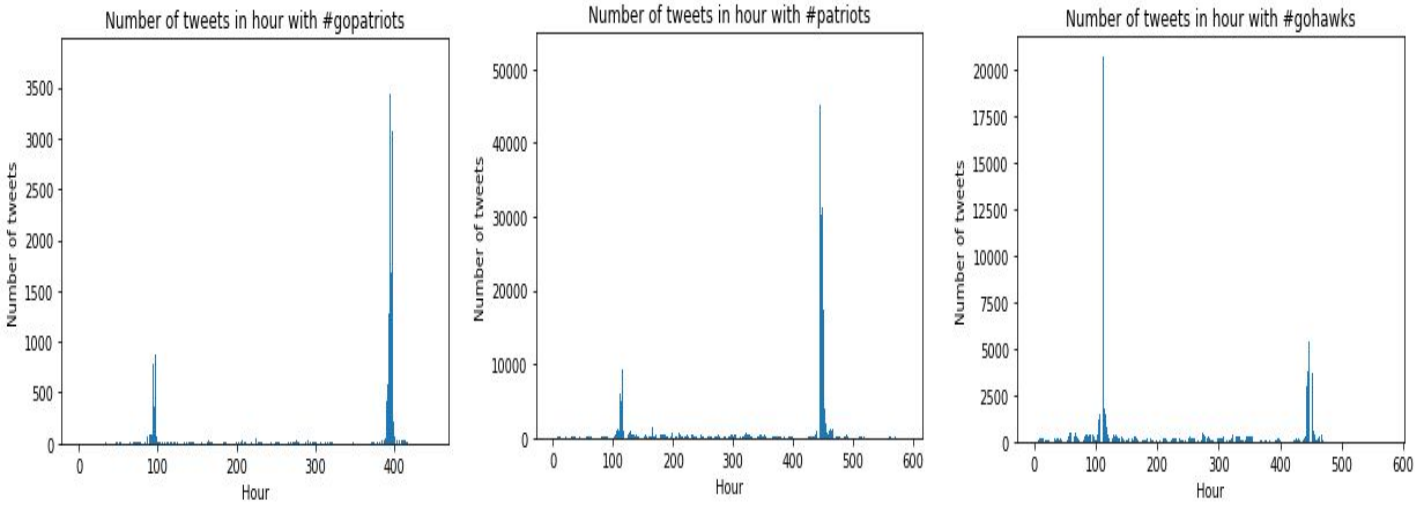
```
#Here we have divided the count of tweets per hour into 3 intervals, and analyze the mean per interval
print np.mean(x[:350])
print np.mean(x[350:500])
print np.mean(x[500:600])
#Thus we observe that the mean was highest for the burst interval i.e. 350-500
```

```
312.52
8103.08666667
274.965517241
```

Thus we can see from the result above, that the pre-burst period was 0-350 hour. Then, slowly the values began to rise. The values shooted up at around 450th hour, i.e. the burst time. The hashtag was off-burst from 450-600 hours. We do not have the data for the death of the hashtag hour.

As Superbowl is an extremely popular event, during the event more fans tweet their real time sentiments about who will win or lose, or who played well etc. Fans attending the event tweet about their experiences during the event, Fans of sportsmen tweet about how well did their favourite players play, etc.

Apart from #superbowl and #nfl, lets us also plot and analyze for other hashtags and see whether the statistics calculated in earlier Task 1.1 A are justified.



As we can see from the above plot, the popularity of tweets with #gopatriots and #patriots was low in the initial hours (i.e. weeks before the event), and increased considerably during the superbowl event. Also, we can see that number of tweets with #gohawks was extremely high at the beginning and reduced considerably during the event.

Two bursts are observed in each of the three plots. Smaller burst occurred during the 100th hour, and larger burst during the 450th hour for #patriots and #gopatriots. For #gohawks, the largest burst occurred in 100th hour, and the smaller burst occurred in the 450th hour.

This is justified by the results of the event, i.e. the fan base and popularity of the winning team i.e. Patriots increased and the popularity of the losing team i.e. Seattle Seahawks decreased as more and more fans saw them losing.

Problem 1.2

In this part, we use a linear regression model to predict the number of tweets in the next hour based on the features collected from the tweets in previous hour. We extract the following five features:

1. Number of tweets (x1)
2. Total number of retweets (x2)
3. Sum of number of followers of users posting the hashtag (x3)
4. Maximum number of followers of users posting the hashtag (x4)
5. Time of the day (x5)

We first generate the training set for the given tweet data. We extract each of the above five features for tweets belonging to the same hour. For each hour, we set the values of features to 0 initially. After grouping the data by citation date, we compute features for each group. Finally, we convert all nan and inf values to zero and infinite respectively. We can now use data from previous hour as the training data and the tweets from current hour as the true value. We fit the data in linear regression model using Ordinary Least Squares (OLS) method provided by StatsModels python API.

A well-fitting regression model results in predicted values close to the observed data values. Three statistics are used in Ordinary Least Squares (OLS) regression to evaluate model fit: R-squared, the overall F-test, and the Root Mean Square Error (RMSE). These three statistics are based on two metrics : Sum of Squares Total (SST) and Sum of Squared Error (SSE). SST measures the variance w.r.t. the mean, and SSE measures the difference between actual and predicted values. Different combinations of these two values provide different information about how the regression model compares to the mean model.

The OLS method returns "**R-squared**" as the measure to evaluate the training accuracy of the model. The value of R-squared indicates the proportion of variance in the dependent variable that can be explained using the independent variables.

R-squared= $(SST - SSE)/SST$

R-squared is the proportional improvement in prediction from the regression model, compared to the mean model. It indicates the goodness of fit of the model.

The scale of R-squared is intuitive. Its value lies in between 0 and 1, with zero meaning no improvement of the regression model over the mean model, and one meaning perfect prediction.

Generally higher values of R-squared are better for the accuracy of the model. Improvements in regression model result in proportional increases in R-squared. However, as we keep adding features, its value keeps increasing even if the features are not related to the label. Thus, a model with highest R-squared value is not necessarily the best model as it might cause overfitting and fail to generalize. Hence, adjusted-R squared metric can be used.

Adjusted R-squared, incorporates the model's degrees of freedom. Adjusted R-squared will decrease as more features are added if the increase in model fit does not make up for the loss of degrees of freedom. Likewise, it will increase as more features are added if the increase in model fit is worthwhile.

RMSE is the square root of variance of the residuals. It indicates absolute fit of the model to the data, i.e. how close are the observed data values to the predicted values. RMSE depicts absolute fit of the model, as compared to R-squared which depicts relative fit. For RMSE, the lower value is better.

We report RMSE value and detailed summary for each hashtag below. In the results presented below, p-value is the probability of observing the test statistic at least as large as the calculated value assuming that the null hypothesis holds. The t value measures the size of the difference in values relative to the variation in sample data. Larger values indicate more evidence against null hypothesis. Hence, we prefer features that have **low p-values and positive t values**.

- **#superbowl**

  RMSE: **8003.56**

OLS Regression Results

Dep. Variable:	y	R-squared:	0.802
Model:	OLS	Adj. R-squared:	0.800
Method:	Least Squares	F-statistic:	470.4
Date:	Sun, 11 Mar 2018	Prob (F-statistic):	2.21e-201
Time:	13:49:24	Log-Likelihood:	-6098.3
No. Observations:	586	AIC:	1.221e+04
Df Residuals:	580	BIC:	1.223e+04
Df Model:	5		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
const	-257.0211	669.627	-0.384	0.701	-1572.210	1058.168
x1	2.3029	0.080	28.925	0.000	2.146	2.459
x2	-0.2895	0.036	-8.039	0.000	-0.360	-0.219
x3	-0.0001	1.88e-05	-7.019	0.000	-0.000	-9.51e-05
x4	0.0008	0.000	5.343	0.000	0.000	0.001
x5	-24.3794	48.219	-0.506	0.613	-119.084	70.325

Omnibus:	1011.856	Durbin-Watson:	2.316
Prob(Omnibus):	0.000	Jarque-Bera (JB):	1834486.434
Skew:	10.106	Prob(JB):	0.00
Kurtosis:	276.357	Cond. No.	2.46e+08

Observations:

The RMSE value for this hashtag #superbowl is the **highest** among all the other hashtags (8003.56). This does not necessarily indicate a bad model because the RMSE might be huge also due to the large number of tweets present for this hashtag. Hence, even a small deviation in predicted value from actual value generates a huge RMSE.
The R-squared and adjusted R-squared values indicate a very good fit to the model for **#superbowl**. From the t-test and p-value, we can say that **x1: Number of tweets** and **x4: Maximum number of followers** are the most significant features for this hashtag.

- **#NFL**

  RMSE : **581.42**

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:	Sun, 11 Mar 2018	Prob (F-statistic):	3.60e-102
Time:	13:49:30	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	0.6843	0.134	5.103	0.000	0.421	0.948
x2	-0.1663	0.064	-2.606	0.009	-0.292	-0.041
x3	8.651e-05	2.63e-05	3.289	0.001	3.48e-05	0.000
x4	-9.073e-05	3.64e-05	-2.494	0.013	-0.000	-1.93e-05
x5	-0.0188	3.528	-0.005	0.996	-6.949	6.911

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

Observations:

The RMSE value for this hashtag is among the **lowest** compared to other hashtags, the value being 581.42. However, the R-squared value and adjusted R-squared is **low** which indicates that the linear regression model is **not a good fit** for **#NFL**. From the t-test and p-value, we can say that **x1: Number of tweets** and **x3: Sum of number of followers** are the most significant features for this hashtag.

- **#sb49**

  RMSE : **4470.45**

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:	Sun, 11 Mar 2018	Prob (F-statistic):	1.09e-201
Time:	13:49:47	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	1.1878	0.095	12.478	0.000	1.001	1.375
x2	-0.2139	0.088	-2.437	0.015	-0.386	-0.042
x3	1.856e-05	1.4e-05	1.323	0.186	-9e-06	4.61e-05
x4	9.581e-05	4.8e-05	1.997	0.046	1.57e-06	0.000
x5	-17.7694	27.150	-0.654	0.513	-71.094	35.556

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

Observations:

The RMSE value for this hashtag is the **higher** compared to other hashtags, the value being 4470.45. This is lower as compared to **#superbowl**. This does not necessarily indicate a bad model because the RMSE might be huge due to the large number of tweets present for this hashtag. Hence, even a small deviation in predicted value from actual value generates a huge RMSE.
The R-squared value and adjusted R-squared indicates a very good fit to the model for **#sb49**. From the t-test and p-value, we can say that **x1: Number of tweets** and **x4: Maximum number of followers** are the most significant features for this hashtag.

- **#patriots**

  RMSE: **2526.28**

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:	Sun, 11 Mar 2018	Prob (F-statistic):	6.42e-137
Time:	13:49:58	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	0.9208	0.072	12.867	0.000	0.780	1.061
x2	-0.0876	0.059	-1.484	0.138	-0.203	0.028
x3	-3.749e-07	2.62e-05	-0.014	0.989	-5.19e-05	5.12e-05
x4	0.0002	0.000	1.615	0.107	-3.59e-05	0.000
x5	-6.9399	15.278	-0.454	0.650	-36.946	23.067

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

Observations:

The RMSE value for this hashtag is slightly higher compared to other hashtags, but **lower** than #superbowl and #nfl. This RMSE value together with the R-squared and adjusted R-squared values indicate that the model is a decent fit for **#patriots**. From the t-test and p-value, we can say that **x1: Number of tweets** and **x4: Maximum number of followers** are the most significant features for this hashtag.

- **#gohawks**

RMSE: **972.50**

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:	Sun, 11 Mar 2018	Prob (F-statistic):	3.70e-77
Time:	13:50:02	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	1.2305	0.170	7.250	0.000	0.897	1.564
x2	-0.1272	0.044	-2.886	0.004	-0.214	-0.041
x3	-0.0002	8.51e-05	-2.064	0.039	-0.000	-8.54e-06
x4	1.807e-05	0.000	0.112	0.911	-0.000	0.000
x5	2.0470	5.951	0.344	0.731	-9.642	13.736

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

Observations:

The RMSE value for this hashtag is **low** compared to most other hashtags, which indicates a good absolute fit. However, the R-squared value is **low** which indicates that the model is a poor relative fit for **#gohawks**. From the t-test and p-value, we can say that **x1: Number of tweets** is the most significant feature for this hashtag.

- **#gopatriots**

  RMSE: **185.02**

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:	Sun, 11 Mar 2018	Prob (F-statistic):	9.76e-121
Time:	13:50:03	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	-0.0802	0.255	-0.314	0.754	-0.582	0.421
x2	0.5083	0.223	2.282	0.023	0.071	0.946
x3	0.0002	0.000	1.237	0.217	-0.000	0.001
x4	-0.0004	0.000	-1.908	0.057	-0.001	1.12e-05
x5	-0.1395	1.135	-0.123	0.902	-2.368	2.089

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

Observations:

The RMSE value for this hashtag is **lowest** among all other hashtags. Moreover, the R-squared value is in a good range which indicates that our regression model is a good fit for **#gopatriots**. From the t-test and p-value, we can say that **x2: Number of retweets** and **x3: Sum of number of followers** are the most significant features for this hashtag.

Problem 1.3

Till now, we have worked with 5 features. In this problem, we extract 4 more features from the dataset. This is done to obtain better accuracy than the previous 5-feature model. The following nine features were used to train a linear regression model :

1. **Hour of the day**: This refers to the hour when it was tweeted. (0-23 values)
2. **Retweet count**: This is the sum of the retweets in the current hour.
3. **Follower_count**: The total number of followers of the users.
4. **Influence Level**: This is the average of the influence level of the authors.

   item.get('author').get('influence_level')

5. **Replies**: The total number of replies to the tweet.

   item.get('metrics').get('citations').get('replies')

6. **Ranking Score**:The average ranking score of all the tweets.

   item['metrics']['ranking_score']

7. **Impressions**: The average of impressions of the tweets.

   item.get('metrics').get('impressions')

8. **Favorite count**: The total number of times a tweet was favorited.

   item['tweet']['favorite_count']

9. **Tweet count**: The total number of times a tweet in current hour.

These features were extracted for all the 6 hashtags and a different linear model was fitted for each of the hashtags.

The table below shows top three features of the ten for each hashtag :

	gopatriots	gohawks	patriots	nfl	sb49	superbowl
RMSE	133.6223	825.6494	2381.7392	479.8633	4482.2908	7150.3516
1st top feature	influence_level	influence_level	follower_count	favorite_count	favorite_count	favorite_count
2nd top feature	ranking_score	ranking_score	impressions	tweet_count	retweetcount	influence_level
3 top feature	tweet_count	tweet_count	tweet_count	influence_level	follower_count	tweet_count

We will now analyze the results for each hashtag in more detail.

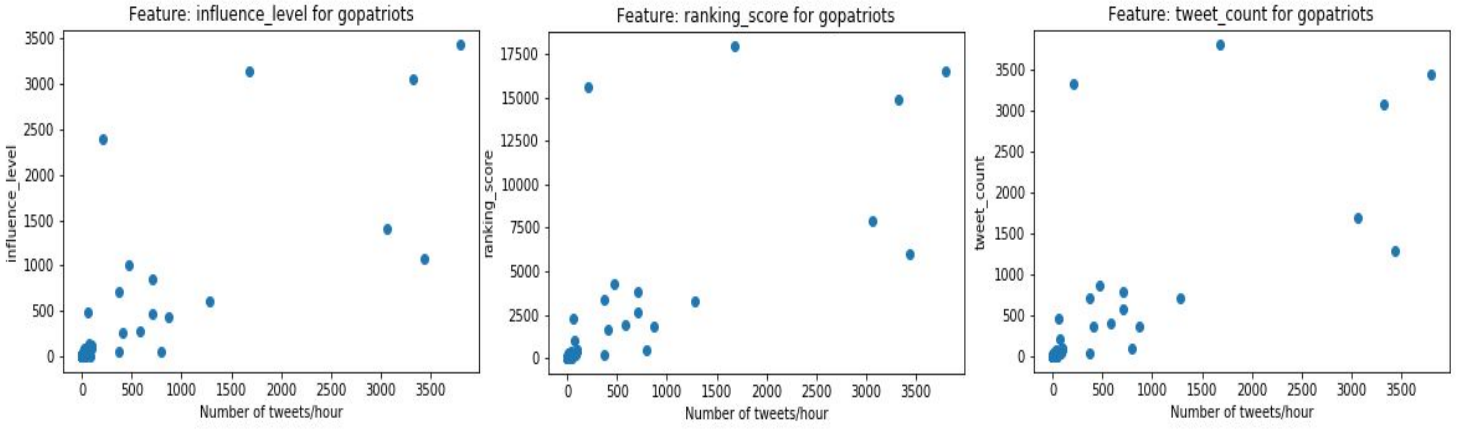
For #gopatriots

RMSE :133.6223
R-squared: 0.854

```
Result summary for  gopatriots
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.854
Model:                            OLS   Adj. R-squared:                  0.851
Method:                 Least Squares   F-statistic:                     284.2
Date:                Sat, 17 Mar 2018   Prob (F-statistic):          1.77e-176
Time:                        10:25:47   Log-Likelihood:                -2816.0
No. Observations:                 446   AIC:                             5650.
Df Residuals:                     437   BIC:                             5687.
Df Model:                           9
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
x1            -0.1022      0.499     -0.205      0.838      -1.083       0.878
x2            -1.6258      0.199     -8.172      0.000      -2.017      -1.235
x3             0.0007      0.000      2.623      0.009       0.000       0.001
x4             9.9517      0.487     20.451      0.000       8.995      10.908
x5             3.4319      2.697      1.272      0.204      -1.869       8.733
x6           -12.6795      0.680    -18.642      0.000     -14.016     -11.343
x7            -0.0005      0.000     -2.238      0.026      -0.001    -6.6e-05
x8             4.5700      1.689      2.706      0.007       1.251       7.889
x9            54.2660      2.979     18.214      0.000      48.410      60.122
==============================================================================
Omnibus:                      341.334   Durbin-Watson:                   1.759
Prob(Omnibus):                  0.000   Jarque-Bera (JB):            63306.013
Skew:                           2.294   Prob(JB):                         0.00
Kurtosis:                      61.185   Cond. No.                     3.31e+05
==============================================================================
```

Observations : The RMSE value for this hashtag is **lowest** among all other hashtags. Moreover, the R-squared value and adjusted R-squared value is in a good range i.e. 0.8 which is closer to 1. A R-squared value closer to 1 indicates perfect prediction, as we have learnt earlier. This means that that our regression model is a good fit for **#gopatriots**. From the p-value, we can say that **influence_level, ranking_score, tweet_count** are the most significant features for this hashtag.

This seems intuitive , as the **influence_level** of an author decides how famous the tweet will be in the next hour. Suppose a big celebrity or President tweets with an hashtag, there are a lot many chances of his tweets getting attention as compared to other users. **Ranking score** of tweet may give us an idea about how informative and relevant a tweet is. Hence it is another significant feature in deciding the total number of tweets in the next hour. Also, if the **tweet_count** for the current hour is high, the tweet count for the next hour would obviously be higher than the current hour. This follows the basic logic that in the next hour, more users would retweet the tweets in the current hour, as well as new tweets on fresh topics related to a hashtag would come up.



For further analysis, we have plotted scatterplots for each of these three significant features for predicted vs actual number of tweets for this hashtag for the next hour. We observe a slightly linear relationship in the plots.

#gohawks:

RMSE:825.6494

R-squared:0.646

Result summary for gohawks

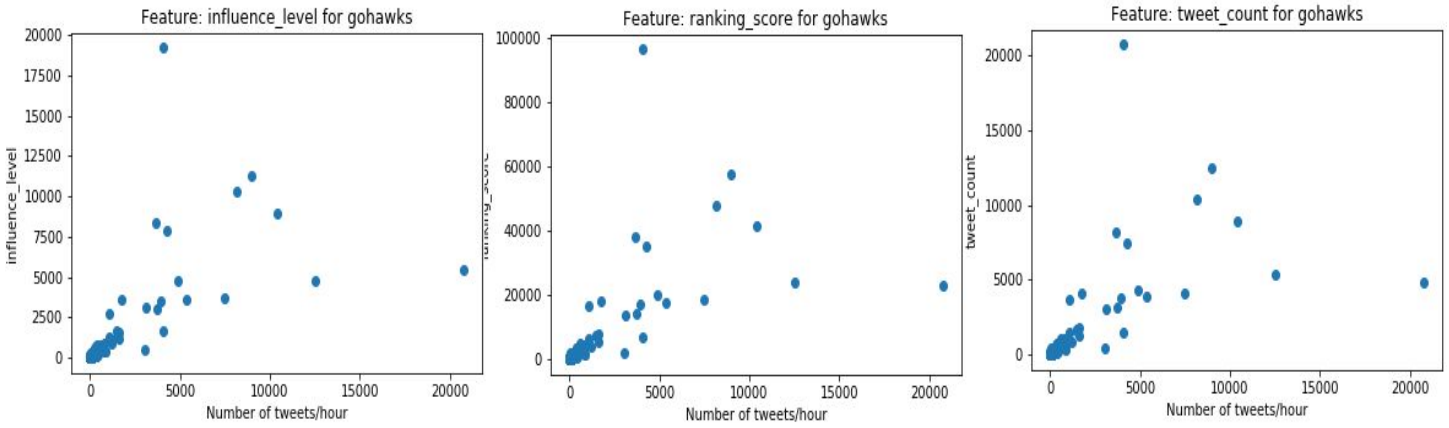
OLS Regression Results

Dep. Variable:	y	R-squared:	0.646
Model:	OLS	Adj. R-squared:	0.640
Method:	Least Squares	F-statistic:	113.7
Date:	Sat, 17 Mar 2018	Prob (F-statistic):	1.97e-120
Time:	10:30:50	Log-Likelihood:	-4645.1
No. Observations:	571	AIC:	9308.
Df Residuals:	562	BIC:	9347.
Df Model:	9		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
x1	-6.1129	3.000	-2.038	0.042	-12.005	-0.221
x2	-0.2205	0.054	-4.079	0.000	-0.327	-0.114
x3	-6.377e-05	0.000	-0.615	0.539	-0.000	0.000
x4	7.0705	0.497	14.224	0.000	6.094	8.047
x5	47.7063	8.658	5.510	0.000	30.699	64.713
x6	-6.0775	0.584	-10.412	0.000	-7.224	-4.931
x7	-0.0001	8.79e-05	-1.599	0.110	-0.000	3.21e-05
x8	0.0693	0.025	2.782	0.006	0.020	0.118
x9	22.6395	2.462	9.196	0.000	17.804	27.475

Omnibus:	958.838	Durbin-Watson:	2.094
Prob(Omnibus):	0.000	Jarque-Bera (JB):	662622.924
Skew:	9.938	Prob(JB):	0.00
Kurtosis:	168.699	Cond. No.	1.02e+06

Observations : The RMSE value for this hashtag is **second lowest** among all other hashtags. Moreover, the R-squared value and adjusted R-squared value is in a normal range i.e. 0.6 which is slightly closer to 1. As per RMSE, our regression model seems a good fit for **#gohawks**. However, as per R-squared, the model is not a very great fit for **#gohawks**. From the p-value, we can say that **influence_level, ranking_score, tweet_count** are the three most significant features for this hashtag.
As seen before, these three features are intuitively significant as compared to most other features used in the regression model. A **higher influence_level, a good ranking score and a extremely high tweet count** would surely result in **high number of tweets for the next hour**, and our model would predict this correctly. Also, if there are chances of tweet-bursting, this model would be able to predict it in advance for the next hour.



For further analysis, we have plot scatterplots for each of these three significant features for predicted vs actual number of tweets for this hashtag for the next hour. We observe **a very good linear relationship** in the plots.

#patriots:

RMSE: 2381.7392

R-squared: 0.717

Result summary for patriots

OLS Regression Results

Dep. Variable:	y	R-squared:	0.717
Model:	OLS	Adj. R-squared:	0.712
Method:	Least Squares	F-statistic:	162.1
Date:	Sat, 17 Mar 2018	Prob (F-statistic):	1.14e-151
Time:	10:32:04	Log-Likelihood:	-5388.0
No. Observations:	586	AIC:	1.079e+04
Df Residuals:	577	BIC:	1.083e+04
Df Model:	9		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
x1	2.8252	8.377	0.337	0.736	-13.628	19.279
x2	0.0298	0.080	0.375	0.708	-0.127	0.186
x3	0.0005	0.000	2.688	0.007	0.000	0.001
x4	-0.4118	0.869	-0.474	0.636	-2.118	1.295
x5	0.2012	6.083	0.033	0.974	-11.747	12.149
x6	-1.4438	1.278	-1.129	0.259	-3.955	1.067
x7	-0.0003	0.000	-1.360	0.174	-0.001	0.000
x8	-0.2547	0.219	-1.164	0.245	-0.685	0.175
x9	6.6941	5.012	1.336	0.182	-3.150	16.538

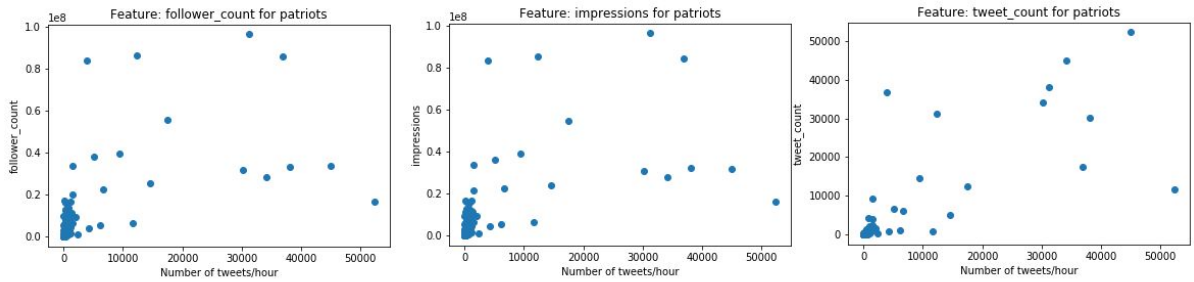
Omnibus:	1020.920	Durbin-Watson:	1.885
Prob(Omnibus):	0.000	Jarque-Bera (JB):	843025.636
Skew:	10.721	Prob(JB):	0.00
Kurtosis:	187.572	Cond. No.	1.12e+06

Observations : The RMSE value for this hashtag is **higher**. Moreover, the R-squared value and adjusted R-squared value is in a good range i.e. 0.71 which is closer to 1. This means that that our regression model is a good fit for **#patriots as per R-squared value**. From p-value, we can say that **follower_count, impressions , tweet_count** are the most significant features for this hashtag.

Follower_count tells us the total number of followers of the authors of the tweet. Higher follower_count would mean high chances of the tweet getting retweeted in the future. Hence it is a meaningful feature to predict the count of tweets in the next hour. **Impressions** of a tweet tell us the total number of views of the tweet. A higher impression would mean that more and more audience has viewed this tweet and/ or are probably going to share it in the next hour. Hence this is another meaningful feature for our regression model.

As already seen earlier , **tweet_count** of the current hour is an independent variable with a high regression coefficient. This measure carries a lot of significance, and more tweets in current hour would lead to more retweets of these tweets as well as new tweets in the next hour.

For further analysis, we have plot scatterplots for each of these three significant features for predicted vs actual number of tweets for this hashtag for the next hour.



As we can see from these scatter plots, we see almost identical scatterplots for the three features. Also, the points seems closer towards the origin, and more scattered away from the origin. The plot of tweet count looks slightly more linear than the other two plots.

#nfl:

RMSE 479.8633

R-squared: 0.765

Result summary for nfl

OLS Regression Results

Dep. Variable:	y	R-squared:	0.765
Model:	OLS	Adj. R-squared:	0.761
Method:	Least Squares	F-statistic:	206.7
Date:	Sat, 17 Mar 2018	Prob (F-statistic):	1.58e-173
Time:	10:33:17	Log-Likelihood:	-4418.8
No. Observations:	582	AIC:	8856.
Df Residuals:	573	BIC:	8895.
Df Model:	9		
Covariance Type:	nonrobust		

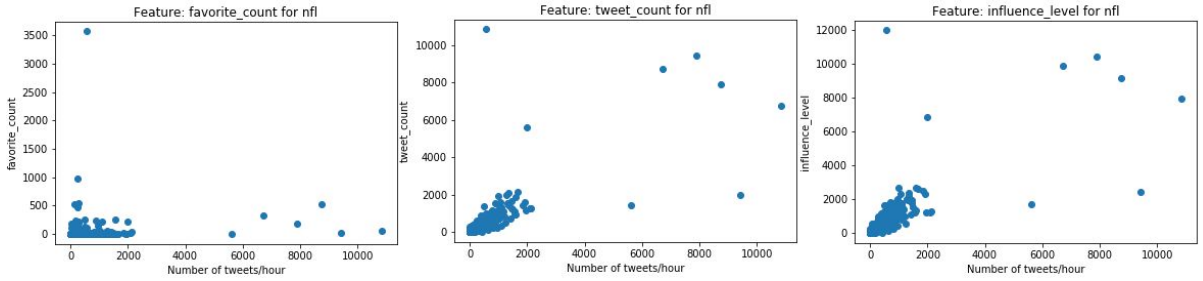
	coef	std err	t	P>|t|	[0.025	0.975]
x1	-2.4722	1.957	-1.263	0.207	-6.315	1.371
x2	-0.0862	0.053	-1.623	0.105	-0.190	0.018
x3	3.111e-05	2.96e-05	1.053	0.293	-2.69e-05	8.92e-05
x4	1.4503	0.258	5.612	0.000	0.943	1.958
x5	-3.9095	3.343	-1.169	0.243	-10.475	2.656
x6	-2.8255	0.525	-5.382	0.000	-3.857	-1.794
x7	-2.591e-05	2.72e-05	-0.954	0.340	-7.92e-05	2.74e-05
x8	-1.9315	0.169	-11.447	0.000	-2.263	-1.600
x9	12.1514	2.123	5.724	0.000	7.982	16.321

Omnibus:	877.314	Durbin-Watson:	2.312
Prob(Omnibus):	0.000	Jarque-Bera (JB):	275423.071
Skew:	8.299	Prob(JB):	0.00
Kurtosis:	108.272	Cond. No.	1.11e+06

Observations : The RMSE value for this hashtag is **low** . Moreover, the R-squared value and adjusted R-squared value is in a good range i.e. 0.76 which is closer to 1. This means that that our regression model is a **good fit** for **#nfl a**s per both RMSE and R-squared/adjusted R-squared. From the p-value, we can say that **favourite_count, tweet_count, influence_level** are the most significant features for this hashtag.

Favourite_count is a measure that tells us how many times a tweet has been favorited. This again influences the retweet probability of a tweet. If more and more users liked a particular tweet, there are higher chances of them retweeting this tweet in the next hour. Hence this feature can be a significant feature in some cases. For **tweet_count and influence_level** features , we are already aware of how they influence the number of tweets in the future, as seen earlier.

For further analysis, we have plot scatterplots for each of these three significant features for predicted vs actual number of tweets for this hashtag for the next hour. We observe a good linear relationship in the plots.



From the Scatter Plots, we can see **a very good linear relationship** between the features and the number of tweets in the next hour.

#sb49:

RMSE:4482.2908

R-squared:0.822

Result summary for sb49

OLS Regression Results

Dep. Variable:	y	R-squared:	0.822
Model:	OLS	Adj. R-squared:	0.818
Method:	Least Squares	F-statistic:	271.6
Date:	Sat, 17 Mar 2018	Prob (F-statistic):	2.72e-192
Time:	10:34:34	Log-Likelihood:	-5306.5
No. Observations:	540	AIC:	1.063e+04
Df Residuals:	531	BIC:	1.067e+04
Df Model:	9		
Covariance Type:	nonrobust		

	coef	std err	t	P>|t|	[0.025	0.975]
x1	7.4247	16.056	0.462	0.644	-24.117	38.966
x2	0.3639	0.136	2.678	0.008	0.097	0.631
x3	0.0002	8.47e-05	2.148	0.032	1.55e-05	0.000
x4	-0.4301	0.914	-0.471	0.638	-2.226	1.365
x5	-7.1552	7.458	-0.959	0.338	-21.806	7.495
x6	-2.0819	1.944	-1.071	0.285	-5.901	1.737
x7	-0.0001	8.42e-05	-1.617	0.106	-0.000	2.93e-05
x8	-0.3349	0.109	-3.084	0.002	-0.548	-0.122
x9	9.0414	7.761	1.165	0.245	-6.204	24.287

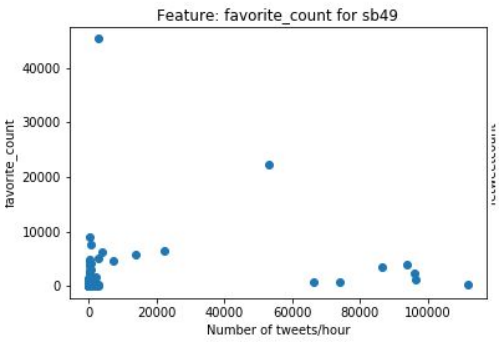
Omnibus:	1061.104	Durbin-Watson:	1.756
Prob(Omnibus):	0.000	Jarque-Bera (JB):	1773810.147
Skew:	13.457	Prob(JB):	0.00
Kurtosis:	282.485	Cond. No.	1.07e+07

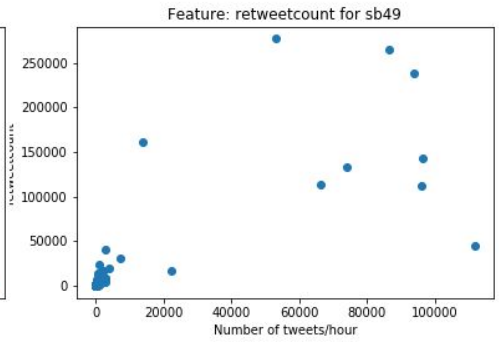
Observations : The R-squared value and adjusted R-squared value is in a very good range i.e. 0.8 which is closer to 1. This means that our regression model is a **good fit** for **#nfl a**s per R-squared/adjusted R-squared values only. From the p-value, we can say that **favourite_count, retweet_count, follower_count** are the most significant features for this hashtag.

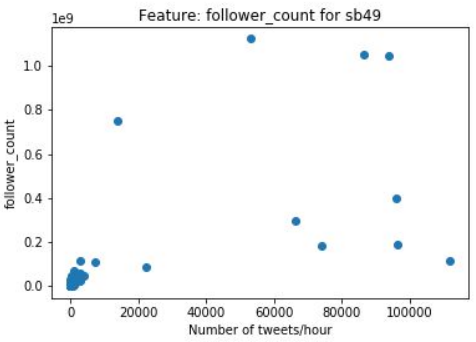
Favourite_count is a measure that tells us how many times a tweet has been favorited. This again influences the retweet probability of a tweet. If more and more users liked a particular tweet, there are higher chances of them retweeting this tweet in the next hour. Hence this feature can be a significant feature in some cases. For **retweet_count,** the number of times a current tweet has been retweeted is a measure of how famous the tweet is, and thus it has higher chances of being retweeted again in the future, thereby increasing the count of future tweets.

Follower_count tells us the total number of followers of the authors of the tweet. Higher follower_count would mean high chances of the tweet getting retweeted in the future. Hence it is a meaningful feature to predict the count of tweets in the next hour.

Thus the three significant features derived from our regression model are relevant. For further analysis, we have plot scatterplots for each of these three significant features for predicted vs actual number of tweets for this hashtag for the next hour.







We can see here almost identical scatter plots with more points near the origin and scattered away from the origin. the relationship looks decently linear.

#superbowl:

RMSE:7150.3516

R-squared:0.845

Result summary for superbowl

OLS Regression Results

Dep. Variable:	y	R-squared:	0.845
Model:	OLS	Adj. R-squared:	0.842
Method:	Least Squares	F-statistic:	348.6
Date:	Sat, 17 Mar 2018	Prob (F-statistic):	9.77e-227
Time:	10:35:29	Log-Likelihood:	-6032.2
No. Observations:	586	AIC:	1.208e+04
Df Residuals:	577	BIC:	1.212e+04
Df Model:	9		
Covariance Type:	nonrobust		

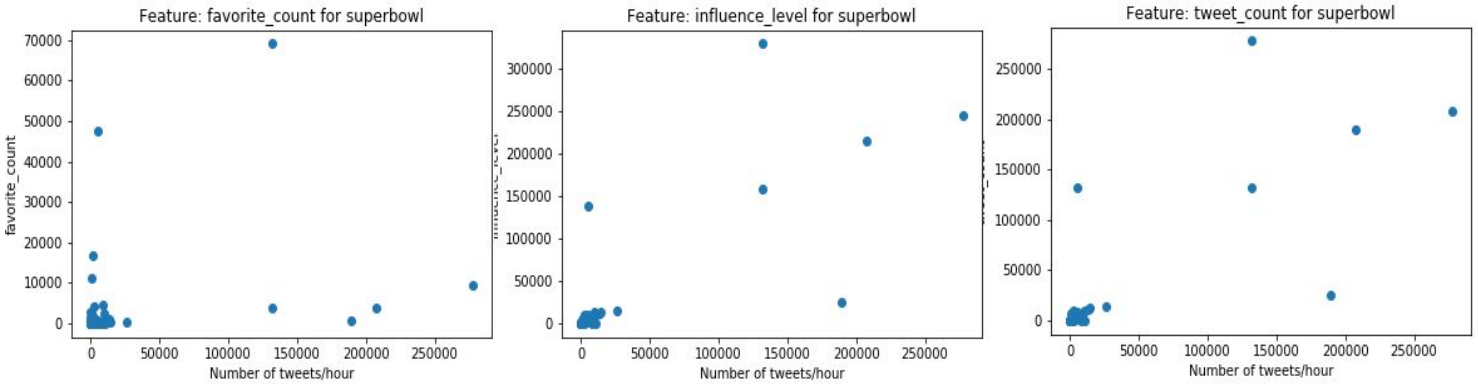
	coef	std err	t	P>|t|	[0.025	0.975]
x1	52.8758	24.326	2.174	0.030	5.098	100.653
x2	-0.0314	0.060	-0.518	0.604	-0.150	0.087
x3	0.0004	0.000	1.444	0.149	-0.000	0.001
x4	4.3515	1.125	3.868	0.000	2.142	6.561
x5	1.5347	20.812	0.074	0.941	-39.342	42.412
x6	-7.9380	2.625	-3.023	0.003	-13.095	-2.781
x7	-0.0005	0.000	-1.879	0.061	-0.001	2.13e-05
x8	-2.2049	0.263	-8.378	0.000	-2.722	-1.688
x9	33.1994	10.830	3.065	0.002	11.928	54.471

Omnibus:	1305.274	Durbin-Watson:	2.143
Prob(Omnibus):	0.000	Jarque-Bera (JB):	3746146.966
Skew:	18.048	Prob(JB):	0.00
Kurtosis:	393.029	Cond. No.	1.45e+07

Observations : The R-squared value and adjusted R-squared value is in a very good range i.e. 0.84 which is closer to 1. This means that that our regression model is a **good fit** for **#superbowl** as per R-squared/adjusted R-squared values only. From the p-value, we can say that **favourite_count, influence_level, tweet_count** are the most significant features for this hashtag.

As we have already seen before, these three features have an impact in predicting the tweet count.

To further analyze on the relationship, we have plot scatterplots for each of these three significant features for predicted vs actual number of tweets for this hashtag for the next hour.



The plots show clusters near the origin. However, there does not seem to be a clear linear relationship from the plots above for #superbowl. Too few points are scattered, and most of the points lie in the cluster.

Inference

From all the above results, we can see that almost all of the features have a good linear relation with the output. Within each hashtag, the patterns are very similar with respect to different attributes. It can seen from the plots that there is a clustering for low values of tweets/hour. This is because the initial hours have less number of tweets. **Tweet count** and **influence level** perform better than other features as they are more linear and less scattered compared to others. Also, compared to the previous question, the accuracy has improved upon adding more features as seen from the increase in R-squared value. This is not surprising as adding more features provides more information to the regression model so that it can provide a better fit.

Problem 1.4

In this part, we use the 9 features obtained from previous part to perform cross-validation on the dataset. Here, we split the data into 10 equal parts out of which 9 are used for fitting the model. We predict the number of tweets for the remaining part. We evaluate the performance of the model using average prediction error which defined below:

Average Prediction Error = $|N_{predicted} - N_{real}|$

We perform analysis of regression models for 3 different time periods during the SuperBowl. We extend this analysis to two other regression models: Support Vector Machines (SVM) and Random Forest apart from OLS. The time periods are as mentioned below:

1. Before Feb 1, 8:00 a.m. (when the hashtags haven't become very active)
2. Between Feb 1, 8:00 a.m and 8:00 p.m. (during the active period)
3. After Feb 1, 8:00 p.m. (after the high activity period)

During the implementation, we segregate each tweet based on its citation time and also split it into one hour windows. We then perform 10-fold cross validation on 3 different time periods for each of the 3 different models and calculate average prediction error by taking mean across 10 folds. The results for total 9 combinations is listed in the table below:

Hashtag	Time period	OLS (Ordinary Least Squares)	SVR (Support Vector Regressor)	RFR (Random Forest Regressor)
SuperBowl	Before active period	315.71	431.89	250.14
	During active period	965778.46	120504.2	54253.15
	After active period	427.87	589.59	263.61
NFL	Before active period	133.45	188.43	116.75
	During active period	24158.90	5236.9	1662.84
	After active period	116.63	293.44	151.46
Sb49	Before active period	47.53	103.45	46.33
	During active period	212232.40	31638.2	26057.60
	After active period	123.74	345.88	115.15
Patriots	Before active period	354.88	264.65	235.80
	During active period	54486.09	11972.5	16055.08
	After active period	338.42	148.17	108.77
GoHawks	Before active period	338.54	224.83	159.59

	During active period	31813.64	3374.15	2411.01
	After active period	758.02	36.45	21.27
Gopatriots	Before active period	13.01	13.13	11.55
	During active period	4561.31	1460.65	624.48
	After active period	9.11	5.28	3.83

Observations:

It is clearly observable that the error for the **second (active) period** is much **larger** compared to the other two periods. One reason for this might be the fact that the data is collected for only 10 hours during the active time period. Since the amount of data is small, it might not be enough to train the model properly; hence leading to a bad fit and huge error.

Another reason is that during the superbowl, suddenly there are a large number of users posting tweets and this unusual behavior is hard to predict. A simple linear model might not be able to fit this behavior related to a sudden burst of tweets.

It can be seen from the table that **SVM performs better than OLS** and reduces the error during the active period by a large margin. The error during the other two periods are not affected much. But that does not really matter as those values were small from the beginning.The **best performance** is achieved by the **Random Forest Regressor** as it succeeds in hugely bringing down the error during the active period.

Among the hashtags, **Superbowl** has the **largest** error while **GoPatriots** have the **least** error during second period.

Time period	MAE on aggregated data
Before active period	689.31
During active period	105345.22
After active period	401.65

While performing analysis on individual hashtags, we found that **Random Forest Regressor** with **70 trees** was the **best** model overall. We now use the data aggregated from all hashtags and train using this model for 3 different time periods.

Observations:

The average prediction error for first and third period is low compared to that for second period. Even after using the best model, the error value on aggregated data is larger compared to that for individual hashtags. This is because the aggregated model is not able to generalize as well as the individual hashtag model and this behavior is quite expected. The average error for the active period(105345.22) on aggregated data is comparable to the error(120504.2) obtained for #superbowl using SVM model.

Problem 1.5

For this part , we have used **Random Forest regression** as that was the best model in previous question. We have used 70 trees. Keeping it same as that of the previous question.

We have used a sliding window of 5 hours to predict the output of next hour. We have predicted the for the last hour in each test file and the result are as follows:

Total RMSE=584.6248

	Actual	Predicted
sample1_period1	1	84
sample2_period2	4	25
sample3_period3	523	1172
sample4_period1	201	184
sample5_period1	1	132
sample6_period2	14	32
sample7_period3	120	71
sample9_period2	1	1622
sample10_period3	61	38

We can observe that upon training just one single model of all the consolidated data, we don't get very good result. That is because one single model cannot capture the trend in the number of tweets really well as this trend is highly depended on the period when the tweets were posted. The trend that we observe is different for each time period, i.e, before superbowl, during superbowl and after superbowl. A generalized model cannot capture the trend. But using a five hours sliding window improves the accuracy compared to using a one hour window. That is because, a five hour window has more information about the past then the one-hour window.

Part 2: Fan Based Prediction

People belonging to a geographic location tend to support a sports team from that location. For example, in Super Bowl 2015, most people from Washington state would support Seattle Seahawks team, and most people from Massachusetts would support the New England team.

Analyzing tweets , we can get information about a users' support for a particular team, or whether the user has opposing views for that team. Hence, in this task we design a binary classifier, which would classify whether users location from either Washington or Massachusetts purely based on textual content of the tweet.

Question : Train a binary classifier to predict the location of the author of a tweet (Washington or Massachusetts), given only the textual content of the tweet (using the techniques you learnt in project 1). Try different classification algorithms (at least 3) in your submission. For each, plot ROC curve, report confusion matrix, and calculate accuracy, recall and precision.

For this task, we first combine data from all the hashtags. We have then filtered the data such that only users belonging to any city in the states of Washington or Massachusetts is present.

Now, we use the textual content of the tweet as a feature, and after doing preprocessing, pass it to our classification algorithms. The location of the author of the tweet is passed as target, and we compare the target and the predicted locations to analyze how well did our model classify the data.

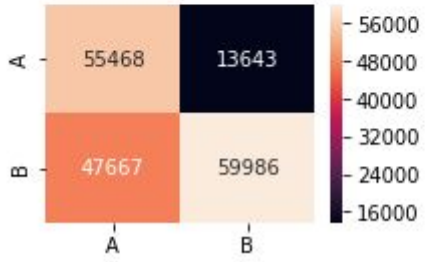
For tweet text preprocessing, following steps were performed:

1. Tokenize the data using our custom designed tokenizer.
2. Remove stop words and perform stemming using Snowball stemmer.
3. Perform TF-IDF on tweet data.
4. As the TF-IDf matrix is sparse, we perform SVD (Singular Value Decomposition) in order to retain only the top features.
5. We use this SVD matrix as our feature vector for classification.
6. We have run different classification algorithms and found accuracy, precision recall and other metrics for each. We perform cross validation for each classification, as well as use train-test split for 85-15 ratio.
7. **Finally, we design our own new ensembled classification algorithm, combining top 3 algorithms from the previous step that gave us the best results. We find whether our new algorithm gives better accuracy that the best out-of-the box classification algorithms.**

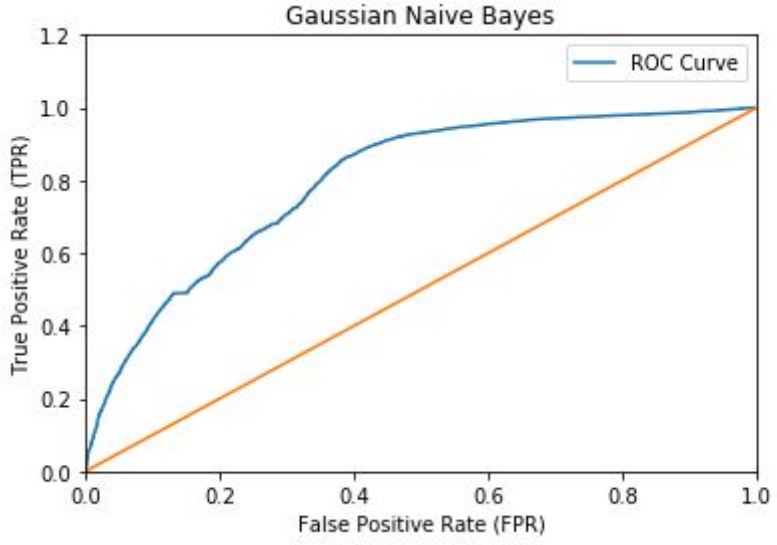
A comparison for different models tested is as follows:

Gaussian Naive Bayes Classifier:

```
Based on 10 Fold CV
Accuracy :  65.3153357018
Precision :  67.6262762473
Recall :  67.9904585642
AUC :  0.679904585642
Confusion Matrix :

[[55468 13643]
 [47667 59986]]
```



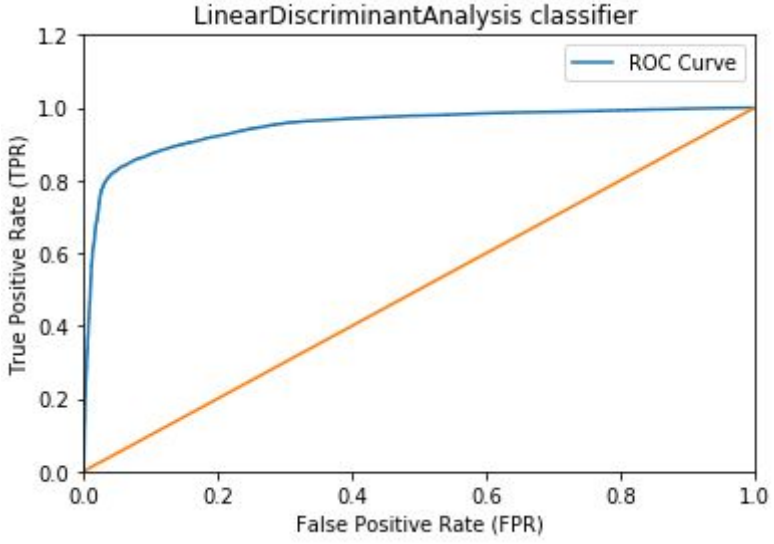
ROC based on 85-15 Train Test Split:



The ROC for Gaussian Naive Bayes is in good shape. Even the Accuracy and precision values are 65% and 66%, which are decent. Gaussian Naive Bayes is a naive algorithm, which means that it assumes that features are almost independent, or the dependence of features is spread equally amongst classes. This is true in our example, as the data after svd truncation has features which are dependent almost equally between the two classes, Washington and Massachusetts.

Linear Discriminant Analysis Classifier

```
Accuracy :  86.7003462243
Precision :  87.2147449799
Recall :  84.7057797046
AUC :  0.847057797046
Confusion Matrix :

[[ 52219  16892]
 [  6617 101036]]
```



As we can see here, the performance for LDA has improved over Gaussian NB to 86% from 65 %. LDA requires an assumption of equal variance-covariance matrices (between the input variables) of the classes. This assumption is important for classification stage of the analysis. As the performance by LDA is really good, this means that the assumption of equal variance of covariance matrizes is true for our data.

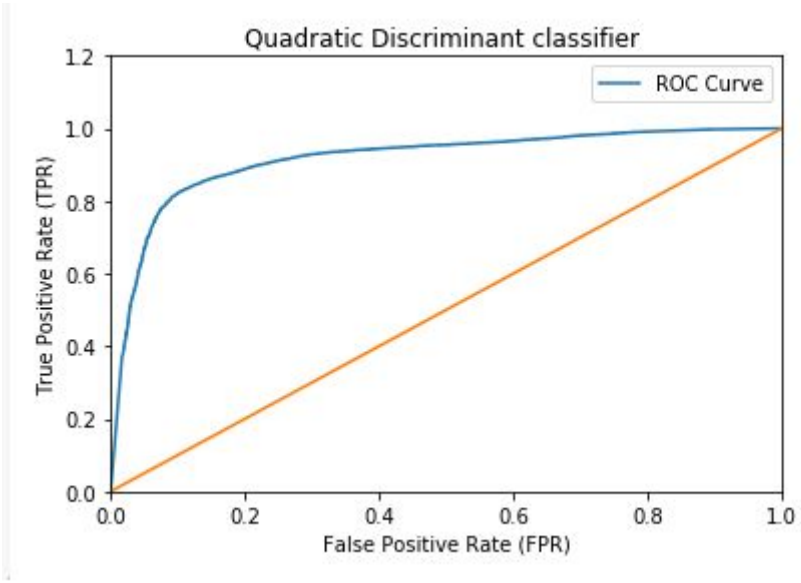
LDA is same as Fisher Discriminant Analysis, and uses Bayes Approach to classify objects.

Quadratic Discriminant Classifier

To verify whether our earlier assumption of equal variance covariance matrices is true, let us apply Quadratic Discriminant Analysis Classifier.

If the covariance matrices substantially differ in variance, observations will tend to be assigned to the class where variability is greater. To overcome the problem, **QDA** was invented. QDA is a modification of LDA which allows for the above heterogeneity of classes' covariance matrices.

```
***10 Fold CV********
Accuracy :  83.4479871467
Precision :  83.4471238917
Recall :  85.1166526732
AUC :  0.851166526732
Confusion Matrix :

[[64114  4997]
 [24261 83392]]
```

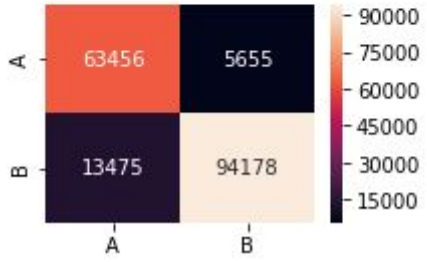


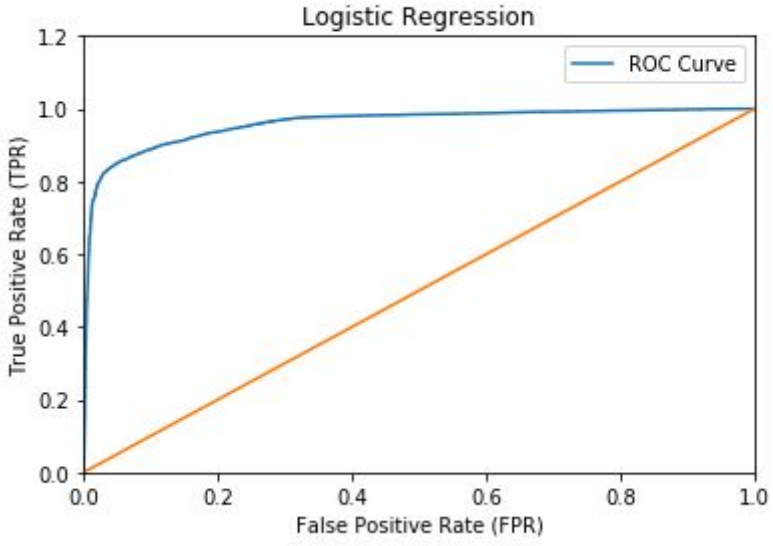
We have applied Quadratic Discriminant Classifier on our tweet data, and found accuracy and precision to be decrease to 78% from 86% of LDA. This algorithm performs better than Naive Bayes, however it does not meet our expectation standards and does not win over LDA.

This means that our assumption of covariance matrices having equal variance is true.

Logistic Regression L2 Normalization

```
Accuracy :  89.1776606096
Precision :  88.4099222345
Recall :  89.6502211153
AUC :  0.896502211153
Confusion Matrix :

[[63456  5655]
 [13475 94178]]
```





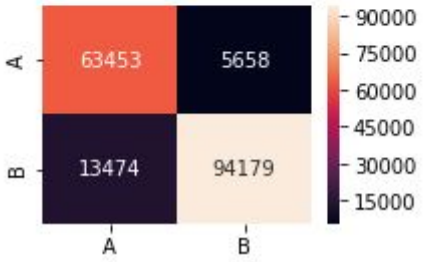
Logistic regression models the probability of the default class using a logistic function. The coefficients of logistic regression are estimated using maximum likelihood estimation. as we can see, our classification performance has increased considerably to **89%**. Logistic regression is a technique that is well suited for examining the relationship between a categorical response variable and one or more categorical or continuous predictor variables. In this case, we have our tweet text as continuous variables(after

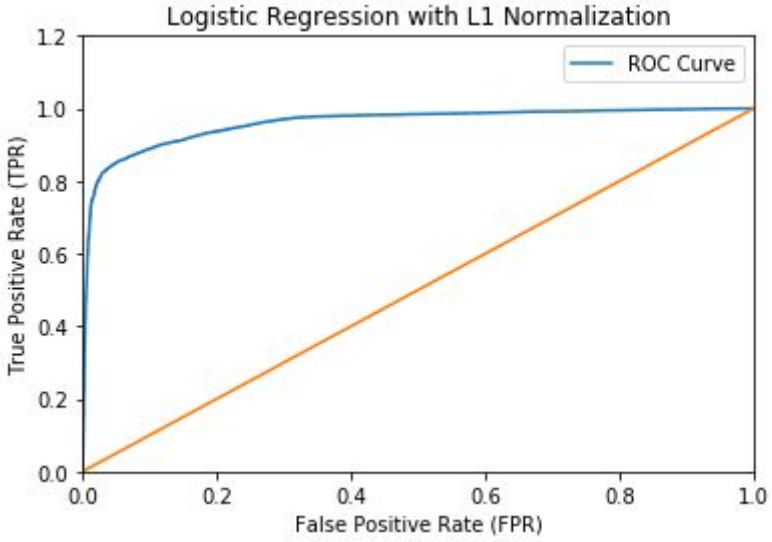
preprocessing and truncated svd), and our location test data has been assigned value 0 or 1, for classes Massachusetts and Washington respectively. **Hence, Logistic regression gives good performance for continuous feature variables and categorical target variables.**

The default Logistic Regression Classifier of sklearn uses "L2" as default penalty. Let us now experiment and see the results if "L1" penalty is used instead.

Logistic Regression L1 Normalization

```
Accuracy :  89.1765291575
Precision :  88.4087278417
Recall :  89.648515149
AUC :  0.89648515149
Confusion Matrix :

[[63453  5658]
 [13474 94179]]
```



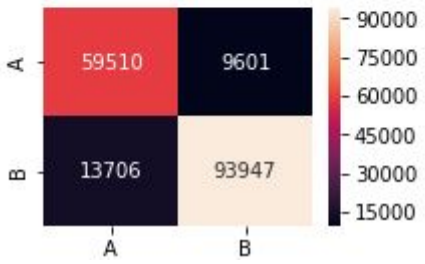


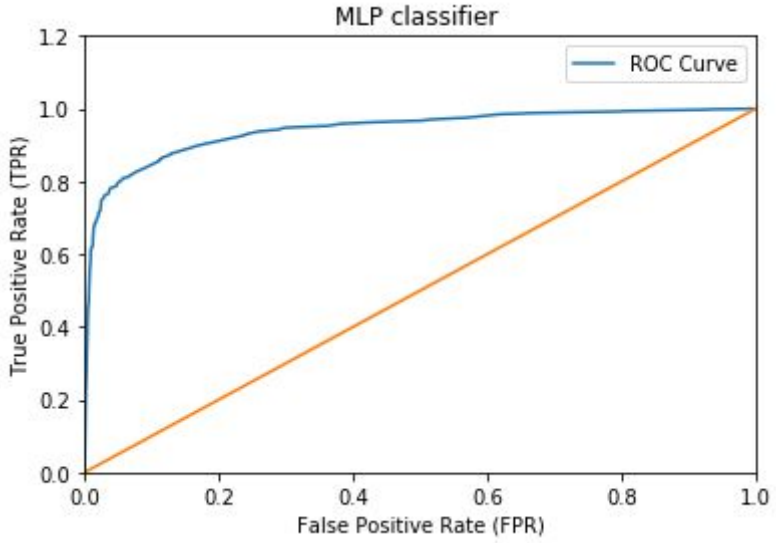
As we can see, the results for using Logistic Regression with L1 Penalty is almost the same as L2 penalty. L1 normalization is useful for sparse inputs as it results in feature-selection, which is the built-in speciality of this model. L2 normalization is computationally efficient than L1- normalization and results in a single solution. As we do not have sparse inputs for tweet data after applying SVD, L2 is giving us better results than L1.

MLP Classifiers (Multilevel Perceptrons)

MLP Classifier is a type of neural network classifier provided by python's sklearn library. It is different from logistic regression, in that between the input and the output layer, there can be one or more non-linear layers, called hidden layers. More precisely, it trains using some form of gradient descent and the gradients are calculated using Backpropagation. For classification, it minimizes the Cross-Entropy loss function, giving a vector of probability estimates

```
Accuracy :  86.814622887
Precision :  86.0040098228
Recall :  86.6881042138
AUC :  0.866881042138
Confusion Matrix :

[[59510  9601]
 [13706 93947]]
```



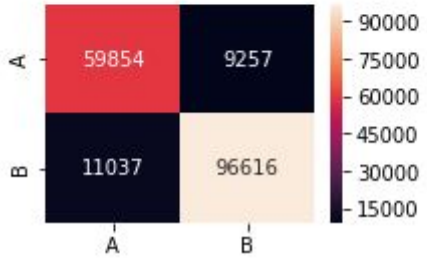


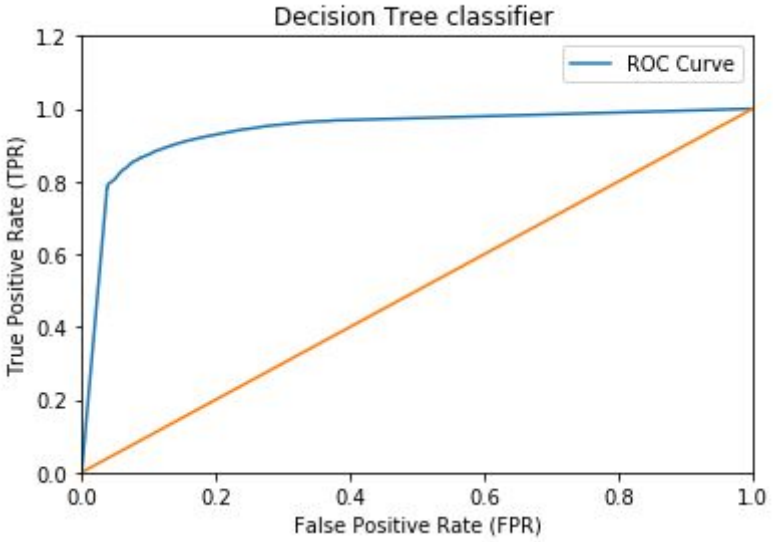
MLP Classifier needs parameter tuning, for the number of hidden layers, as well as the activation function to be used. Different parameters can yield completely different results. In our classification, we have experimented with 35/50 hidden layers and relu/tanh as activation functions. The results obtained were around **86%** for both these cases.

Decision Tree Classifier:

Decision Tree Classifiers match human intuition when making decisions. Also there is hardly any parameter tuning required. Hence, let us now use Decision tree Classifiers and see the performance on tweet data.

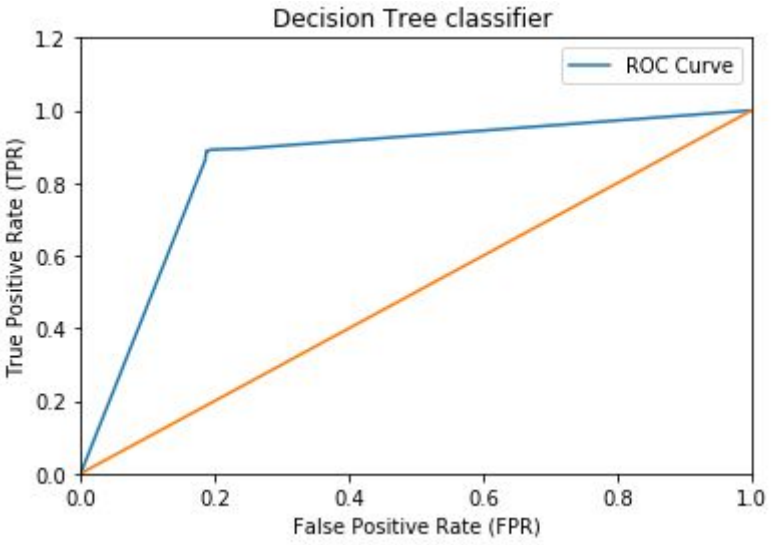
```
Accuracy :  88.5191554841
Precision :  87.8437666759
Recall :  88.1766102488
AUC :  0.881766102488
Confusion Matrix :

[[59854  9257]
 [11037 96616]]
```





As we can see, the ROC is in good shape, but with a sharp point at 0.8. This ROC was obtained after setting min_samples_leaf to 10, which is 1 by default.

When this parameter is not tuned, ROC curve obtained is as below, which has only three points and is shap. This indicates overfitting.



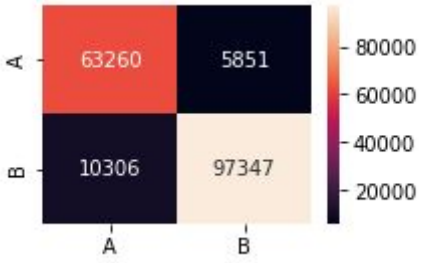
This means that although decision trees generally give good performance, they are prone to overfitting.

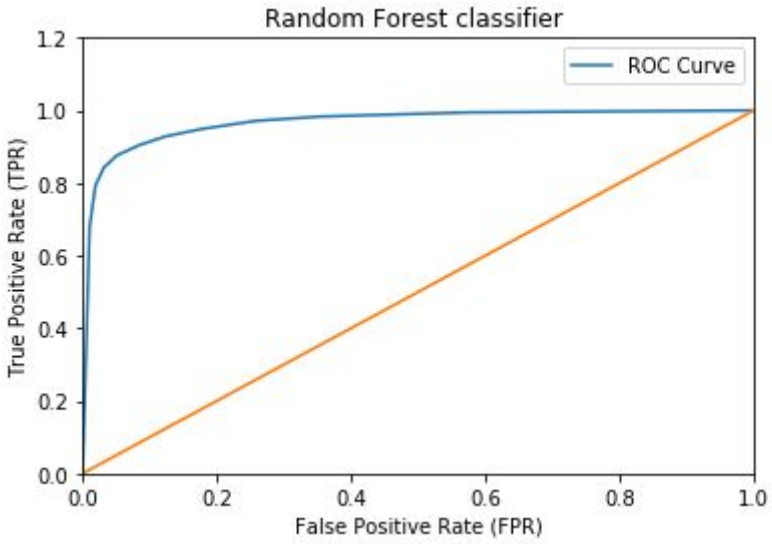
Now let us explore Random forest and Extra Tree Classifiers, which are a variant/extension of Decision Tree Classifier.

Random Forest Classifier

As we saw earlier that decision trees are prone to overfitting. Random forests help to solve this problem of Decision tree Classification. Random Forests are an ensembling technique that construct a series of decision trees , and use “mode” of the results during classification, and average of the results during regression.

```
Accuracy :  90.8595641646
Precision :  90.1605637257
Recall :  90.9802789012
AUC :  0.909802789012
Confusion Matrix :

[[63260  5851]
 [10306 97347]]
```



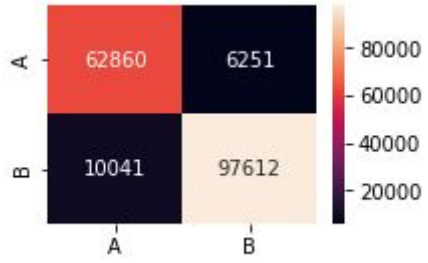


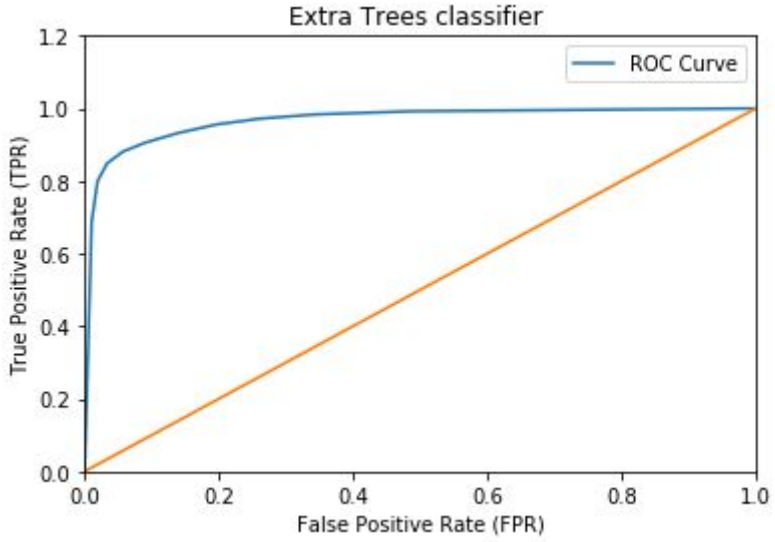
The ROC curve that the best shape among the classifiers that we have seen till now. It is smooth and covers maximum area under the curve. The accuracy,precision,recall values are around **90%** which is a very good performance.

In our classifier, we have set the number of trees to be 10.

Extra Tree Classifier

```
Accuracy :  90.7831911475
Precision :  90.1040106194
Recall :  90.8139700071
AUC :  0.908139700071
Confusion Matrix :

[[62860  6251]
 [10041 97612]]
```





Random forest classifier studied previously is an example of Bagging Algorithm which uses a basic algorithm (e.g. decision trees), and applies it multiple times on subsamples to achieve better results.

Another example of such Bagging Technique is Extra Tree Classifier. In this Classification technique, one further step of randomization yield extremely randomized trees, called extra trees. They differ from random forest in a sense that top-down splits in a random forest is deterministic based on gini impurity, whereas in case of extra trees, its random.

As we can see, the accuracy,precision and recall values for both the bagging methods are almost similar. Even the ROC curves look identical.

Classifier Ensembling

Sometimes, in order to achieve higher performance for classification, ensembling can be done. Ensembled Methods combine predictions from several base estimators, and learn from the mistakes of previous estimators so as to build a more generalized /robust classifier model.

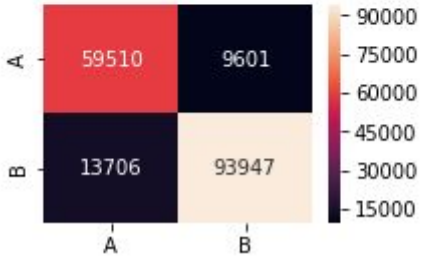
Classifier ensembling is of three types:

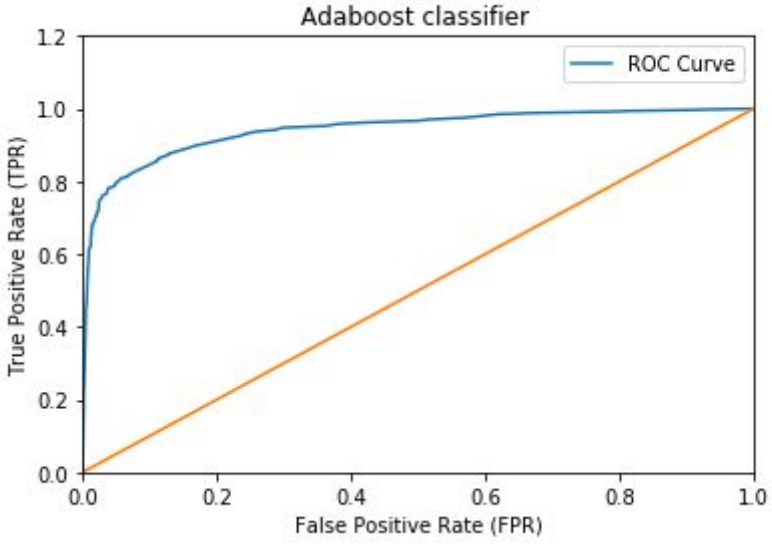
1. **Bagging Classifiers**
2. **Boosting Classifiers**
3. **Voting Classifiers**

Of these, we have already seen Random Forest and extra Tree Classification as examples of Bagging CLassifiers. Lets us now explore the different Boosting Classifiers and Voting classifiers.

Adaboost Classifier

```
Accuracy :  86.814622887
Precision :  86.0040098228
Recall :  86.6881042138
AUC :  0.866881042138
Confusion Matrix :

[[59510  9601]
 [13706 93947]]
```





Here, we have used Adaboost Classifier. The ROC curve is in good shape. The Accuracy obtained is around 86%, similar to MLP Classifier.

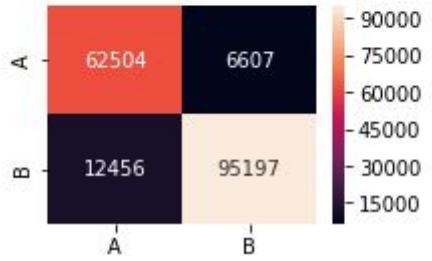
Adaboost Classifier works by classifying on the original dataset once, and then correcting the incorrectly identified classes in the subsequent runs.

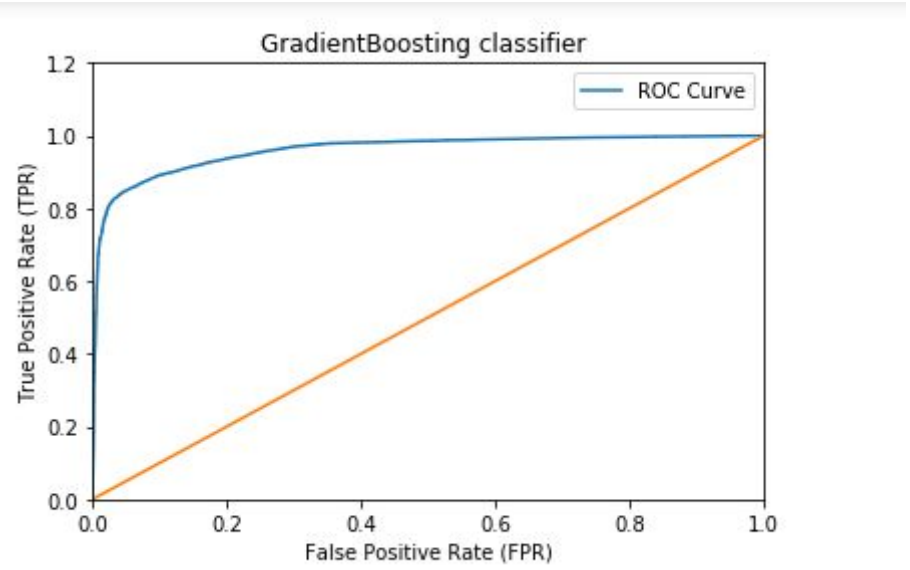
Gradient Boosting Classifier

Another Boosting Technique that we have explored is Gradient Boosting Classifier.

It provides boosting to arbitrary differentiable loss functions. This model is very good for Binary Classification, but for multi-class classification, random forests should be used. This is because at each stage, Gradient boosting classifier creates number of trees =n_classes * n_estimators.

```
Accuracy :  89.2155642552
Precision :  88.4466079314
Recall :  89.4347539173
AUC :  0.894347539173
Confusion Matrix :

[[62504  6607]
 [12456 95197]]
```





As we can see , the performance obtained by Gradient Boosting Classifier was very good i.e. **89%**. Even the ROC curve is well-shaped.

Voting Classifier

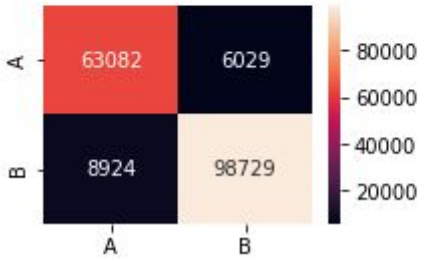
Voting Classifiers combine several different machine learning algorithms, and then use voting to decide the class label. It uses hard voting, to decide label based on majority, or soft voting, which returns the maximum of sum of predicted probabilities for each individual classifier.

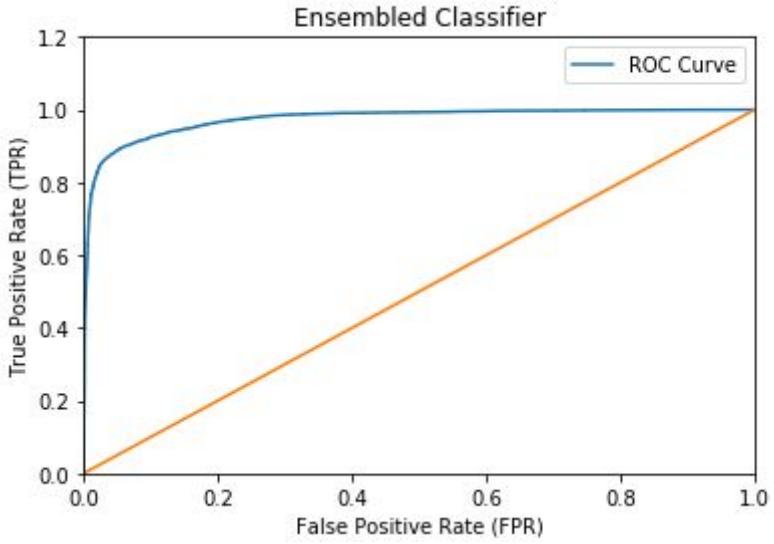
In this project, we implemented a soft Voting Classifier, combining the 3 classifiers that we have seen earlier, which gave highest accuracies.

The top 3 classifiers that gave us maximum accuracies are Random Forest Classifier, Extra Tree Classifier and Gradient Boosting Classifier. These themselves individually are also ensembled classifiers.

The performance obtained was as follows:

```
Accuracy :  91.5406983322
Precision :  90.9257096418
Recall :  91.4933777014
AUC :  0.914933777014
Confusion Matrix :

[[63082  6029]
 [ 8924 98729]]
```





The performance thus achieved using Voting Classifier is **91%**, which is slightly better than individual classifiers.

Thus we observe that, selecting any good machine learning classification algorithm depends on the data and the task at hand. Different algorithms may perform well or worst in different scenarios.

Part 3: Design your own project

Hundreds and thousands of tweets and thereby hashtags are observed every day on twitter. The amount of such tweets increases manifolds during any important events such as matches related to sports, elections, global news, etc. We have been given data of one such high-data generating event – Superbowl 2015. Though the data would be rich throughout the duration of the event, intuitively, we

can narrow down on the fact that data during exciting matches would be even richer and thus would be perfect for our analysis. The Superbowl 2015 final match played between the 'Seattle Seahawks' and 'New England Patriots' was one such super-exciting match which gripped the fans of both the teams to their seats.

Project Design:

We propose to design a project which will analyze the sentiments of fans of both the teams hidden in their tweets. By gauging these analysis, we can predict which team will win and then we can match our results with the actual results of Superbowl 2015.

Here, by fans, we mean the people who belong to a particular team's homeland. For instance, we analyze the tweets with *#gohawks* hashtags of only people residing at Seattle or Washington. They are considered as fans of the Seattle Seahawks. On the other hand, we consider people residing at Boston or Massachusetts with tweets bearing *#gopatriots* or *#patriots* hashtags as ardent fans of New England Patriots.

If there is an overall positive sentiment observed in the tweets of a certain fan-base, then that team is likely to win. On the other hand, a negative sentiment of fans shows that their team is likely to lose.

Also, we will restrict our data to only the second half of the final match held on 1st February 2015 so as to grab the sentiments towards the end of the match. These sentiments will help us rightly predict who would be winning and who would be losing.

We know that the finals was played between Seattle Seahawk and New England Patriots on 1st February 2015. It started at 6:30 pm Eastern Time. So, according to PST, this would be 3:30 pm. In order to get the most accurate results, we have taken data of the second half of the match i.e. from 6pm onwards (according to PST). This was done so as to catch hold of the sentiments of fans even after the match, which in our case might be recorded under 2nd February because UTC is ahead of ET by 4 hours. To avoid such problems, we used PST conversion and analysed the data from 6pm till midnight.

Also, we are aware of the fact that New England Patriots won the finals. So, according to the design of our project, there should be a positive sentiment arising from the tweets of the people of Boston/Massachusetts. Consecutively, there should be a negative sentiment in the tweets of the people from Seattle/Washington.

What is Sentiment Analysis?

Sentiment analysis is the process of 'computationally' determining whether a piece of writing is positive, negative or neutral. We can think it as 'opinion mining', deriving the opinion or attitude of a speaker. This is generally used to identify a person's attitude towards a brand or any other thing of importance. The tone of the person (tone of the text in this case), context in the text, emotions, etc. are used widely to understand the sentiment of a person. Sentiment analysis is generally used these days to gather public opinion of a brand, to analyze customer satisfaction or to simply gather critical feedback. Sentiment analysis could be used in the prediction of an event too. The example of such events are market fluctuations and thereby the prediction of stocks, prediction of electoral results of a particular region, prediction of match results, etc. In this project, we will harness this power of 'Sentiment Analysis' to predict the winners of Superbowl 2015 and verify them.

Performing Sentiment Analysis:

Today, there are many tools/packages available to perform sentiment analysis on data. However, we will restrict to the following:

- TextBlob : This is a python library for processing textual data.
- SentiWordNet: This is an NLTK interface. It is a lexical resource for sentiment analysis which means that it can be applied only on individual words and not sentences. SentiWordNet assigns to each synset of WordNet three sentiment scores: positivity, negativity, and objectivity. (The literature mentioned in the project document highly focusses on the use of SentiWordNet for sentiment analysis.) We first had to run the nltk.downloader() to get this interface from the nltk corpus.

Steps undertaken to perform sentiment analysis are as follows:

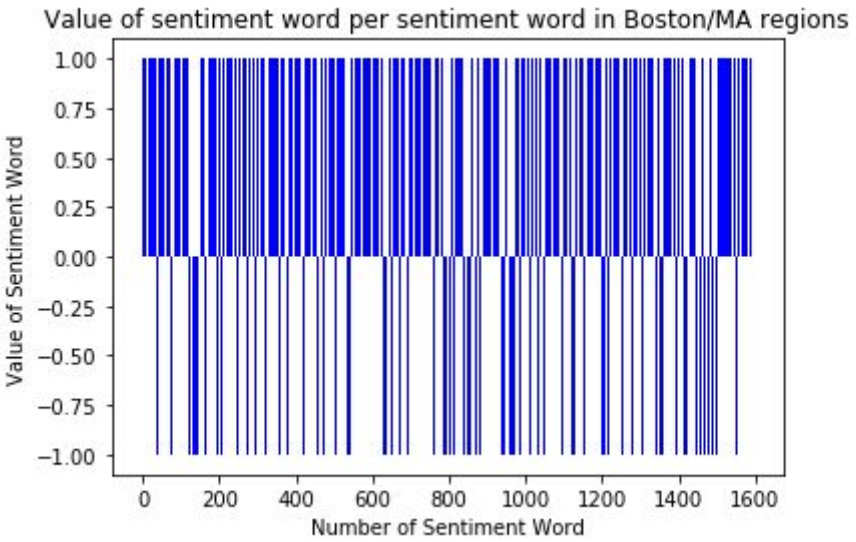
i. Convert the tweet data of gopatriots, patriots and gohawks to csv files. These csv files should essentially contain citation_date, firstpost_date, location of the user and the actual tweet data.
ii. Convert the datetime of citation_time to pst
iii. Extract the data of only 1st February 2015; 6pm onwards
iv. Filter the data specific to fans residing at the homeland.
v. Apply the following pre-processing techniques:
    a. Remove all hashtags from the tweets
    b. Remove all urls from the tweets
    c. Disregard all tweets having language other than English.
vi. Tokenize the tweets into words
vii. Remove the stop-words from the tweets if any
viii. Supply each word to the SentiWordNet interface. This analyzer will then predict the score of the word.
ix. Perform steps i-viii for #gopatriots and #patriots to analyze sentiments for New England Patriots. Then perform the same on #gohawks to analyze sentiments for Seattle Seahawks.

Observations:

1. **#gopatriots and #patriots**
    - Location Filter:
        - MA, Boston, Massachusetts, anything ending with MA
    - Time filter:
        - Starting from the 18th hour of 1st February 2015; i.e. 6pm PST onwards

Total Positive Sentiments	Total Negative Sentiments
1109	484

We also plotted a graph wherein we created a list. And for every positive sentiment, we added a +1 in the list and for every negative sentiment, we added a -1 in the list. We plotted this against the number of data collected.



Analysis of our results:

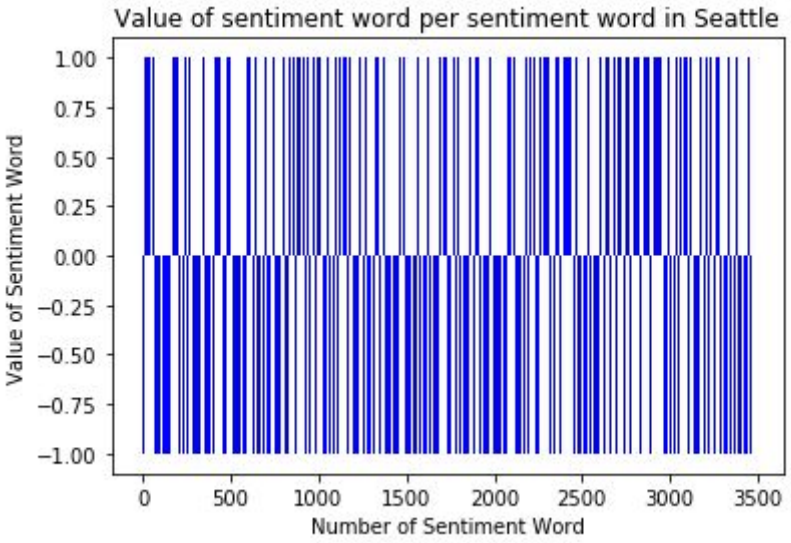
- Since we see that there are a lot of values above 0, the overall sentiment of the users during this time period has been positive
- This means that they have a strong positive opinion about New England Patriots and they are performing well.
- Since this data is collected towards the end of the match (6:30 pm PST onwards), we see a lot of positive sentiments between 200 and 750. This must be the time when the team won the match. We can verify from the statistics on the internet that Superbowl 2015 was actually won by New England Patriots.
- After this, we observe that the density of positive sentiments decrease slightly. This is because probably the excitement has been faded now and it's been a long time after the match. This region also verifies the above claim that the match must have been won when the tweet count of our data was 200 – 750.

2. #gohawks

- Location filter:
  - Redmond, WA, Seattle, Kirkland, and any other words ending with WA
- Time filter:
  - Starting from the 18th hour of 1st February 2015; i.e. 6pm PST onwards

Total Positive Sentiments	Total Negative Sentiments
1439	2031

The graph below captures the sentiments of the tweets. The portion above 0 shows positive sentiments whereas the portion below 0 shows negative sentiments. This is the graph for tweets of users residing at Seattle/other areas in Washington.



Analysis of our results:

- The fact that we see more negative values in the above graph depicts that the people in Seattle or Washington region had negative sentiments about the Seattle Seahawks.
- Naturally, this means that this team would have lost the match. And verifying from actual statistics, we know they actually did lose!
- Also, we see that in the same time interval (i.e. 6pm PST onwards), the number of tweets posted by fans of this team is very large as compared to the tweets of the fans of New England Patriots. This means that the Seahawks naturally had a huge fan base!
- We see the density of negative tweets was high between 1400 and 2100. We can conclude that this was when the team lost the match.
- We also note that this finishing region appears later in this graph as compared to the previous graph because of the high number of tweets for gohawks. This observation also supports our claim that the time when the tweet count of our data was 1400 to 2100 was indeed the one when the match was finished (i.e. Seahawks lost the match).
- In this graph, we also notice that soon after the finishing period, there is an increase in the positive sentiment in the tweets. This is quite natural of fans to continue supporting their teams even after they have lost the match.

The following image is the final result of Superbowl 2015 which verifies our predictions :

Source: https://en.wikipedia.org/wiki/Super_Bowl_XLIX

REFERENCES

1. Kong S., Mei Q., Feng L., Zhao Z. (2014) Real-Time Predicting Bursting Hashtags on Twitter. In: Li F., Li G., Hwang S., Yao B., Zhang Z. (eds) Web-Age Information Management. WAIM 2014. Lecture Notes in Computer Science, vol 8485. Springer, Cham
2. https://www.theanalysisfactor.com/assessing-the-fit-of-regression-models/