UCLA ECE219 - Project 5

March 22nd, 2019

Authors

Ning An - 205034447
Zaurbek Tsorojev - 805029443
Endi Xu - 005030030
Minya Yao - 704161217

Introduction

The goal of this project was to analyze the tweet activity for given hashtags from Twitter in order to predict its future popularity. We train a regression model with the given data, and use it to make predictions for other hashtags. The dataset that we are using was obtained by querying popular Twitter hashtags related to the 2015 Super Bowl spanning a period starting from 2 weeks before the game to a week after the game.

Question 1

From our datasets, #superbowl is the most tweeted hashtag, #sb49 is tweeted by users with the most followers, and all 6 hashtags have an approximately an average of 2 retweets per tweet.

Hashtag	**Average number of tweets per hour**	**Average number of followers of users posting the tweets per tweet**	**Average number of retweets per tweet**
#gohawks	340.97	2217.92	2.01
#gopatriots	41.47	1427.25	1.41
#nfl	461.43	4662.38	1.53
#patriots	759.69	3280.46	1.79
#sb49	1275.56	10374.16	2.53
#superbowl	2343.27	8814.97	2.39

Question 2

The plots of the "number of tweets in hour" over time for #SuperBowl and #NFL are shown below.

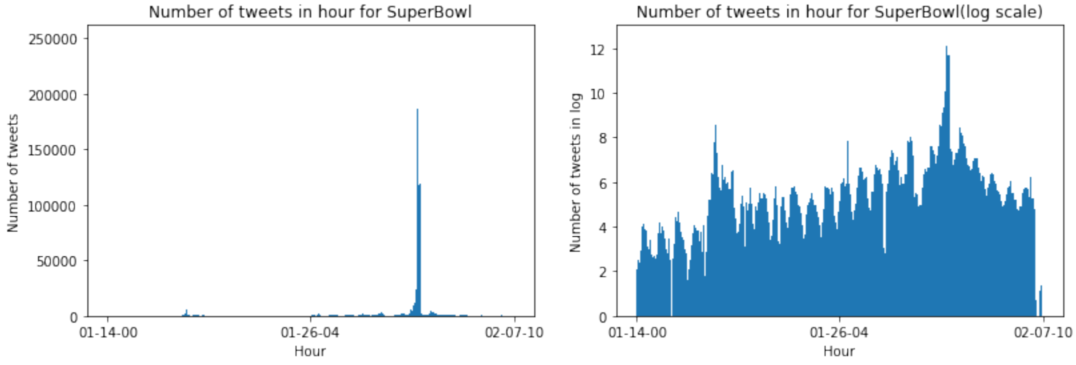


Figure 1 – Number of tweets in hour over time for #SuperBowl

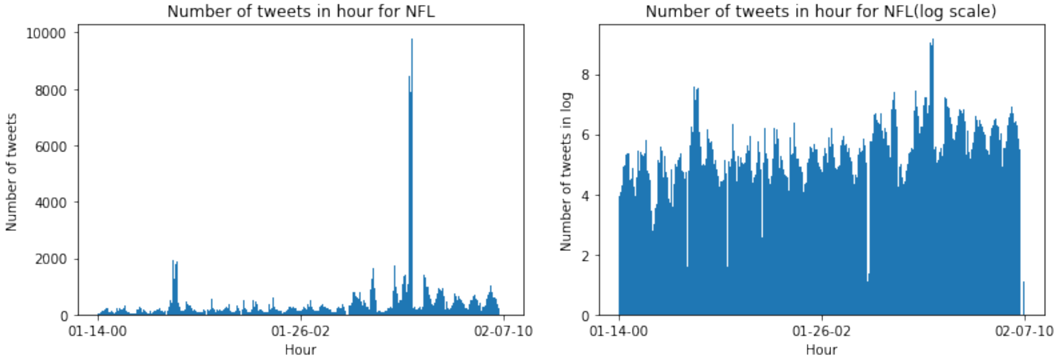


FIGURE 2 – Number of tweets in hour over time for #NFL

Question 3

For each of our models, we report the model's Mean Squared Error (MSE) and R-squared measure in the table below. R-squared is a statistical measure of how close the data are to the fitted regression line. A R-squared value close to 1 means that our model fits the data well.

Hashtag	MSE	R-squared
#gohawks	716101.6908	0.5289
#gopatriots	30590.4907	0.6058
#nfl	274001.9984	0.6518
#patriot	4579216.2447	0.7147
#sb49	13171504.3999	0.8416
#superbowl	34209180.3254	0.8699

We can see that as the size of the dataset increases, R-squared increase which means the model becomes more and more accurate.

Now, let's analyze the significance of each feature using the t-test and p-value. The table below shows these values for each metric and each hashtag.

Hashtag	#tweets		#retweets		$\sum$ of #followers		Max #followers		Time of day	
	t-test	p-value	t-test	p-value	t-test	p-value	t-test	p-value	t-test	p-value
#gohawks	9.529	4.421e-20	-4.447	1.044e-05	-3.840	1.367e-04	2.146	3.229e-02	2.202	2.814e-02
#gopatriots	-0.479	0.632	3.007	0.003	0.812	0.417	-1.429	0.154	0.964	0.335
#nfl	4.713	3.05e-06	-2.663	7.962e-03	4.083	5.048e-05	-3.035	2.507e-03	4.016	6.699e-05
#patriots	15.412	3.473e-45	-5.012	7.134e-07	1.575	1.157e-01	0.701	4.835e-01	1.392	0.164
#sb49	13.026	3.463e-34	-2.227	2.629e-02	0.916	3.597e-01	4.427	1.138e-05	-1.189	0.234
#superbowl	24.006	5.911e-89	-4.843	1.635e-06	-20.059	2.2112e-68	10.984	1.242e-25	-2.264	2.396e-02

We know that in general, a low p-value and large t-value indicates that the predictor is an important contribution to the model. From there, we see that the most valuable features in order are : # tweets, $\sum$ of

#followers, max #followers, #retweets. These are the features that have the lowest p-values. Intuitively, this makes sense, because if a lot of popular people start to tweet about something and get many retweets, it's likely that it will affect the number of tweets in the next hour.

Question 4

In this question, we consider 14 features. These features are :
[Number of tweets, Total number of retweets, Sum of the number of followers, Maximum number of followers, Time of the day, Total number of impressions, Total number of momentum, Total number of favorite count, Total number of ranking score, Total number of acceleration, Total number of replies, Total number of unique users, Total number of unique authors, Total number of user mentions]

For each hashtag, we report the MSE and R-squared values. We see that we obtain much lower MSEs and bigger R-squared, which is what we wanted. This shows that the features we added were meaningful contributions.

Hashtag	**MSE**	**R-squared**
#gohawks	414743.1429	0.7271
#gopatriots	11176.6426	0.8559
#nfl	166528.5934	0.7883
#patriot	3568809.45876	0.7776
#sb49	5532649.0557	0.9334
#superbowl	17383904.7948	0.9338

For each feature (in the same order as stated at the beginning of the question), we get the following p-values and t-values :

- **#gohawks :**
  - **p-values** :
    [4.32658e-04 6.09384e-05 1.34927e-01 2.63677e-01 7.83523e-03 1.73215e-03 1.30150e-16
    2.57739e-09 2.55563e-06 4.85694e-01 9.24758e-09 3.26926e-02 1.76026e-01 1.01607e-04]
  - **t-test** :
    [-3.54031072 -4.03969286 -1.49709824 -1.11885666 -2.66863321 3.1477661 -8.53490438
    6.05414237 4.7522881 0.69763197 5.83166609 -2.14111638 1.3547922 3.91464638]

- **#gopatriots :**
  - **p-values** :
    [5.77234e-01 3.23316e-13 2.10858e-40 5.45714e-20 1.10254e-01 9.85475e-24 9.15501e-01
    7.21365e-04 7.49629e-01 6.58658e-01 9.24197e-01 6.07378e-19 8.21335e-20 5.10791e-44]
  - **t-test** :
    [0.55775476 -7.46378187 14.4389285 -9.51251785 -1.59958749 -10.51779325 0.10615028
    -3.40019053 -0.31928486 -0.44200184 -0.09519046 -9.21819072 9.46303665 15.21744533]

- **#nfl :**
  - **p-values** :

    ```
    [3.01377e-01 1.27090e-02 2.33048e-01 4.14838e-03 5.37258e-01 1.23184e-01 2.15657e-01
     4.64225e-29 7.73308e-01 5.23172e-01 1.23648e-01 4.43465e-02 2.97779e-02 6.97922e-10]
    ```

  - **t-test** :

    ```
    [1.0344198 -2.49969416 -1.19380764 2.87824472 -0.61733593 -1.54382177 1.23953033
     -11.83515329 -0.2881858 -0.63885552 -1.54191096 2.01524242 -2.17847925 6.27329779]
    ```

- **#patriots :**
  - **p-values** :

    ```
    [1.73738e-01 1.65083e-02 9.24242e-04 3.43381e-09 7.77782e-01 9.16049e-01 1.91389e-03
     3.35497e-01 1.65606e-01 6.96285e-01 5.80794e-01 1.78965e-04 9.35407e-05 5.41098e-11]
    ```

  - **t-test** :

    ```
    [-1.3619847 -2.40456943 3.33004486 -6.00342856 0.28234358 -0.10545 -3.11775 0.96391
     1.38823198 -0.3905357 0.55254014 -3.77177149 3.9347083 6.6874702 ]
    ```

- **#sb49 :**
  - **p-values** :

    ```
    [1.63125e-02 2.16129e-01 1.63012e-05 9.35502e-04 2.45265e-01 4.16080e-02 7.05345e-03
     4.46572e-03 1.47996e-01 3.85667e-04 2.91569e-13 1.61150e-12 4.34859e-46 6.78630e-02]
    ```

  - **t-test** :

    ```
    [-2.40902177 -1.23826589 4.34775564 3.32672861 -1.1631303 -2.04204825 2.70413776
     -2.85465135 1.44861737 3.57106599 -7.47614165 -7.22589689 15.63446821 -1.82940451]
    ```

- **#superbow :**
  - **p-values** :

    ```
    [1.08823e-04 1.59193e-26 1.49322e-03 4.42400e-02 6.49641e-01 1.31890e-01 1.52971e-19
     3.29515e-13 1.99682e-02 5.90614e-18 3.44271e-02 5.81459e-06 1.35386e-03 1.10551e-01]
    ```

  - **t-test** :

    ```
    [-3.89726698 11.2165766 3.19148154 -2.01626455 0.45450068 -1.50884841 -9.38056793
     -7.45750738 2.33349687 -8.92812647 2.12011358 4.57614376 -3.22018971 -1.59819858]
    ```

Question 5

In this question, for each hashtag, we pick the top 3 features and draw a scatter plot of number of tweets for next hour versus values of each picked feature respectively.

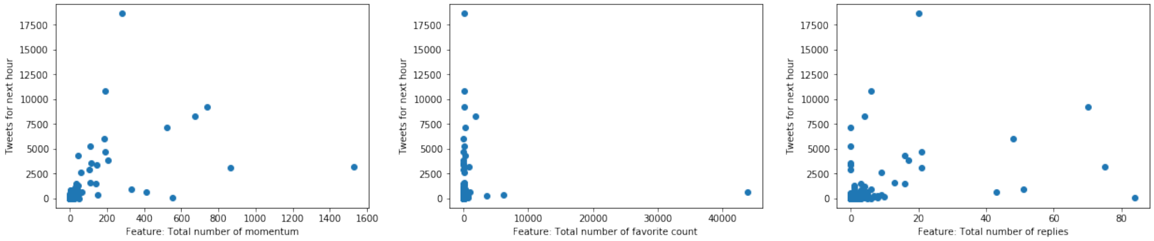


FIGURE 3 – Top 3 features vs #tweets in next hour for #gohawks

The coefficients of the top 3 features are -15.911, 0.1868, 78.1246. All coefficients do not agree with the trends.

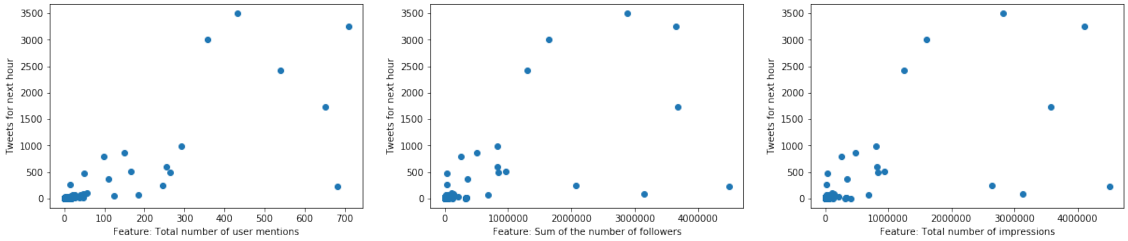


FIGURE 4 – Top 3 features vs #tweets in next hour for #gopatriots

The coefficients of the top 3 features are 6.2258, 0.0046, -0.0025. The first two regression coefficients agree with their trends and the last one does not agree with its trend.

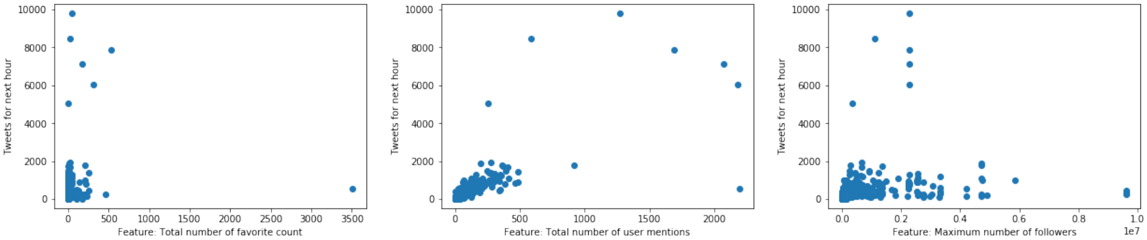


FIGURE 5 – Top 3 features vs #tweets in next hour for #NFL

The coefficients of the top 3 features are -2.678, 3.6286, 8.62e-5. All coefficients agree with the trends.

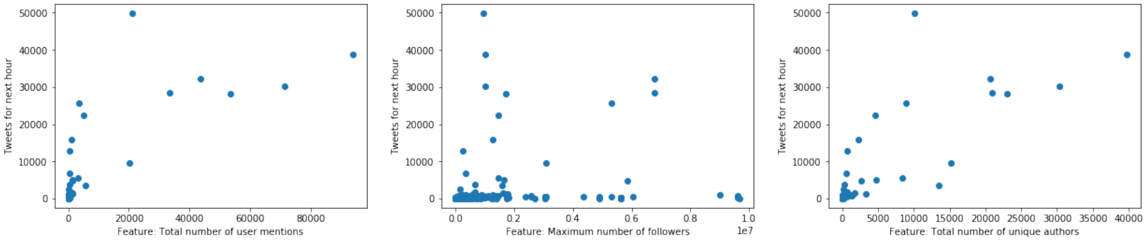


FIGURE 6 – Top 3 features vs #tweets in next hour for #patriots

The coefficients of the top 3 features are 1.5608, -0.0007, 10.7170. All coefficients agree with the trends.

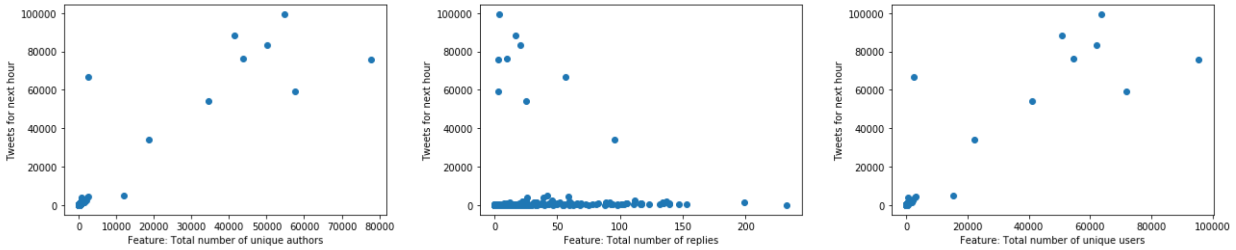


FIGURE 7 – Top 3 features vs #tweets in next hour for #sb49

The coefficients of the top 3 features are 14.7813, -44.5946, -6.5058. The first two regression coefficient agree with their trends and the last one do not agree with its trend.

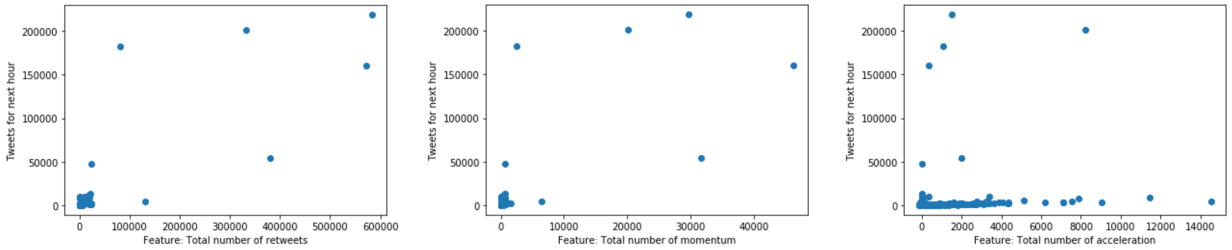


FIGURE 8 – Top 3 features vs #tweets in next hour for #superbowl

The coefficients of the top 3 features are -14.6017, -8.8013, -2.7068. The first two regression coefficients do not agree with their trends and the last one agrees with its trend.

Through comparing the regression coefficients with their corresponding trends, we can found that not all the coefficients agree with their trends. This is because some features are not linearly increasing/decreasing and the scatter points are not evenly distributed, which will confuse the linear regressor. At the same time, the order of magnitude of each feature are different. This also affects the values of the coefficients.

Question 6

In this question, we split the data into the following three parts based on the topic active rate :

- **Period 1 : before Feb. 1, 8 :00 a.m.** : The period before active time. We use 1-hour window
- **Period 2 : between Feb. 1, 8 :00 a.m. and 8 :00 p.m.** : The period during active time. We use 5-minute window
- **Period 3 : after Feb. 1, 8 :00 p.m.** : The period after active time. We use 1-hour window

The regression model we used for the three period is linear regression and we do 5-fold cross-validation for each period. The following is the averaged MSE, averaged RMSE and averaged R-squared score for each hashtag :

- **#gohawks** :

	Averaged MSE	Averaged RMSE	Averaged R-squared scores
Period 1	1006267.26	660.98	0.47
Period 2	84986.38	228.86	0.77
period 3	11190.96	49.61	0.81

- **#gopatriots** :

	Averaged MSE	Averaged RMSE	Averaged R-squared scores
Period 1	2780.11	38.14	0.79
Period 2	26019.52	138.67	0.66
period 3	10299.17	46.6	0.66

- **#nfl** :

	Averaged MSE	Averaged RMSE	Averaged R-squared scores
Period 1	77103.01	210.76	0.69
Period 2	28043.27	141.56	0.91
period 3	19657.62	133.06	0.94

- **#patriots** :

	Averaged MSE	Averaged RMSE	Averaged R-squared scores
Period 1	652456.67	605.97	0.56
Period 2	779186.76	794.93	0.89
period 3	385044.75	307.99	0.88

- **#sb49** :

	Averaged MSE	Averaged RMSE	Averaged R-squared scores
Period 1	13172.69	70.15	0.82
Period 2	1667395.53	1191.98	0.95
period 3	161576.36	261.91	0.87

- **#superbowl** :

	Averaged MSE	Averaged RMSE	Averaged R-squared scores
Period 1	864221.27	633.01	0.49
Period 2	24664198.85	3264.62	0.94
period 3	196274.42	327.92	0.91

Question 7

In this question, we aggregate the data of all hastags and train three models respectively for each of the time period as discussed in question 6.

The regression model we used for the three period is linear regression and we also do 5-fold cross-validation for each period. The following table shows the averaged MSE, averaged RMSE and R-squared score for the three time periods :

	Averaged MSE	Averaged RMSE	Averaged R-squared scores
Period 1	5465048	1557.09	0.576
Period 2	40241671.6	4906.23	0.94
period 3	1406628.4	801.6	0.94

According to the table above, the performance of the models on aggregated data become much worse compared to performance of the models on individual hashtags, which is expected, because when each hashtags has its own characteristics. When we combined all the hashtags together, some of these characteristics will be lost.

Question 8

In this question, we try ensemble methods on aggregated data to see whether the prediction results improve or not. We use random forest regressor and gradient boosting regressor as two examples and do grid search to find the best parameter set of the two respectively.

The following table shows the best parameter sets of the two regressors :

	max depth	max features	min samples leaf	min samples split	n estimators
Random Forest	80	auto	1	5	200
Gradient Boosting	40	sqrt	1	2	600

The following tables show the test errors for the two regressors with the best parameters :

	Mean cross-validated score of the best_estimator
Random Forest	7.27e8
Gradient Boosting	7.31e8

According to the table shown above, the test errors is very large. This is because the aggregated data set contains both active periods and inactive period, which is hard for the regressor to predict. At the same time, there may be over-fitting problems occurred.

Question 9

In this question, we compare the random forest regressor and gradient boosting regressor with the best parameters we found in question 8 with OLS on the entire data.

The following table show the MSE and RMSE for random forest regressor, Gradient Boosting regressor and OSL :

	MSE	RMSE
Random Forest	75025513.43	8661.72
Gradient Boosting	9.96e-8	0.0003
OSL	140986465.23	11873.77

According to the results shown above, both random forest and gradient boosting regressor improve the result and gradient boosting regressor returns the best result. However, gradient boosting regressor has overfitting issues.

Question 10

In this question, we still do the same grid search for gradient boosting regressor but we divide the aggregated dataset into three periods as discussed in question 6 and find the best parameters for each period instead.

The following table shows the best parameter sets of the regressor for each period :

	max depth	max features	min samples leaf	min samples split	n estimators
period 1	40	sqrt	2	5	200
period 2	200	sqrt	1	10	400
period 3	80	auto	4	10	200

The following table shows the cross-validation test errors for each period :

	Mean cross-validated score of the best_estimator
Period 1	4.42e6
Period 2	3.07e7
period 3	1.26e6

According to the table shown above, although the test errors are still large, the test error become much smaller than what did in question 8. The best parameters also changed for each period, which is expected. This is because the number of tweets changed greatly between active period and inactive periods. The best parameters will also change based on that.

Question 11

The MSE obtained from various neural network architectures that we tested are summarized below, with hidden layer sizes listed, grouped by the number of hidden layers. For each architecture, we test on both logistic and tanh activation functions. We did not use relu, because the resulting MSEs were very high.

1. **One hidden layer** : [2,5,10,50,100,200,300,400,500]
   - **logistic** :
     [28754.204105085813, 28753.855694454414, 28788.59475232106, 28740.597246997637, 28711.898292, 28719.499150106578, 28641.6375867408, 28649.84564102266, 28540.37562984672]
     Minimum MSE = 28540.37562984672

- **tanh** :
  [28775.593376809276, 28759.156362876012, 28788.4528120922, 28753.442763458068, 28706.560488, 28753.052086402895, 28750.588593469223, 28752.683491968422, 28749.134405383822]
  Minimum MSE = 28706.560488627467

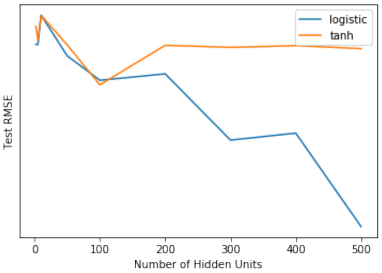


Figure 9 – RMSE vs # Hidden Units (for one hidden layer)

2. **Two hidden layers** : [(10,10), (20, 10), (20,20), (50,50), (100,100), (200,200)]
   - **logistic** :
     [28443.981591592787, 28444.09875841569, 28380.38820623823, 28378.480785872398, 28417.989732970007, 28384.012590969585]
     Minimum MSE = 28378.480785872398
   - **tanh** :
     [28788.103795861836, 28443.812738672033, 28380.381475064314, 28575.403698808364, 28457.392518130186, 28367.00906982105]
     Minimum MSE = 28367.00906982105

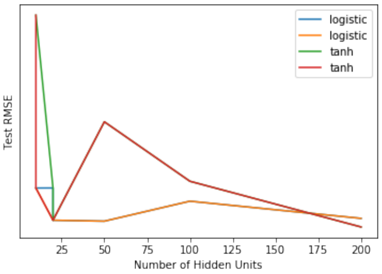


Figure 10 – RMSE vs # Hidden Units (for two hidden layer)

We see that the architecture that gives the lowest MSE is using tanh activation function, with 2 hidden layer of size (200, 200).

Question 12

We use StandardScaler to scale the data before feeding it to MLPRegressor with tanh activation function and two hidden layers with sizes (200, 200). The resulting MSE is 13684.990, which is much less than the

MSE obtained without preprocessing the input by scaling it. To understand this, we know that the idea behind StandardScaler is that it transforms the data so that the distribution will have zero mean and unit variance. This helps to train the model better in the sense that scaling avoids the situation when one or several features dominate others in magnitude.

Question 13

We use gridSearchCV to perform grid search on MLPRegressor to find the parameters of the architecture that yields the best performance, for each window length described in Question 6. We use StandardScaler to scale the data before feeding it to the grid search function. The parameters we fine-tuned are shown in the table below :

Parameter	**Range**
hidden layer sizes	(100,100,100), (20,20,20), (200,200), (100,100), (20,20), 20, 50, 100, 200, 500
activation functions	”logistic”, ”relu”, ”tanh”
alpha	0.01, 0.001, 0.0001
maximum number of iterations	1000, 2000, 3000, 5000, 10000

The optimal parameters for MLP Regressor for each period and the performance of the optimized MLP Regressor models are shown below :

Period	**Optimal Parameters**	**MSE**
1 (before Feb 1, 8 AM, 1-hour window)	Optimal parameters for MLP Regressor : {’activation’ : ’tanh’, ’alpha’ : 0.0001, ’hidden_layer_sizes’ : 100, ’max_iter’ : 1000, ’random_state’ : 42}	2.89e6
2 (between Feb 1, 8 AM and Feb 1, 8 PM, 5-min window)	Optimal parameters for MLP Regressor : {’activation’ : ’tanh’, ’alpha’ : 0.0001, ’hidden_layer_sizes’ : (20, 20), ’max_iter’ : 1000, ’random_state’ : 42}	1.74e7
3 (after Feb 1, 8 PM, 1-hour window)	Optimal parameters for MLP Regressor : {’activation’ : ’tanh’, ’alpha’ : 0.01, ’hidden_layer_sizes’ : (100, 100), ’max_iter’ : 1000, ’random_state’ : 42}	4.13e5

Question 14

In this question, we use the Random Forest Regressor to predict the number of tweets on the 6-th window, using the data from the previous 5 windows to train the model. The time span of the dataset for the all samples across different periods are shown below.

Period 1

- Sample 0 start and end date :
  PST dataset start date : 2015-01-31 05 :00 :37-08 :00
  PST dataset end date : 2015-01-31 10 :30 :38-08 :00
- Sample 1 start and end date :
  PST dataset start date : 2015-02-01 01 :00 :10-08 :00
  PST dataset end date : 2015-02-01 05 :40 :26-08 :00
- Sample 2 start and end date :
  PST dataset start date : 2015-01-26 17 :05 :59-08 :00
  PST dataset end date : 2015-01-26 20 :49 :01-08 :00

Period 2

- Sample 0 start and end date :
  PST dataset start date : 2015-02-01 19 :00 :05-08 :00
  PST dataset end date : 2015-02-01 19 :29 :48-08 :00
- Sample 1 start and end date :
  PST dataset start date : 2015-02-01 12 :30 :18-08 :00
  PST dataset end date : 2015-02-01 12 :59 :51-08 :00
- Sample 2 start and end date :
  PST dataset start date : 2015-02-01 08 :30 :12-08 :00
  PST dataset end date : 2015-02-01 08 :59 :15-08 :00

Period 3

- Sample 0 start and end date :
  PST dataset start date : 2015-02-04 00 :00 :16-08 :00
  PST dataset end date : 2015-02-04 05 :46 :07-08 :00
- Sample 1 start and end date :
  PST dataset start date : 2015-02-05 20 :04 :00-08 :00
  PST dataset end date : 2015-02-06 01 :50 :04-08 :00
- Sample 2 start and end date :
  PST dataset start date : 2015-02-05 17 :00 :04-08 :00
  PST dataset end date : 2015-02-05 22 :35 :12-08 :00

We see that the time span for each sample is not identical to each other, and not exactly 6 hours for periods 1 and 3, and not exactly 30 minutes for period 2. Therefore, we would expect to see that the predicted value deviates from the true value.

We yield the following results :

Period #	Period 1 (1-hour window)				Period 2 (5-min window)				Period 3 (1-hour window)			
Sample #	0	1	2	Mean	0	1	2	Mean	0	1	2	Mean
Prediction	182.1	486.2	199.4	289.2	3846.6	3725.6	967.3	2846.5	3387.7	3387.7	3387.7	3387.7
Normalized Prediction	3.0	8.1	3.3	4.8	3846.6	3725.6	967.3	2846.5	56.5	56.5	56.5	56.5

In the last row, we normalized our predictions to a rate of number of tweets per 5 min. We observe from the results that for the period where the event of Super Bowl is included, the model predicts a significantly higher number of tweets in the 5-minute window, compared to the 1-hour windows for the other periods. This is reasonable, since people tend to discuss about the news, and it is likely that more people tweet about the Super Bowl.

Question 15

In this question, we use the textual content of the tweet to predict the location of the author of a tweet (Washington or Massachusetts), by training a binary classifier. We consider all the tweets including `#superbowl`.

Point 1

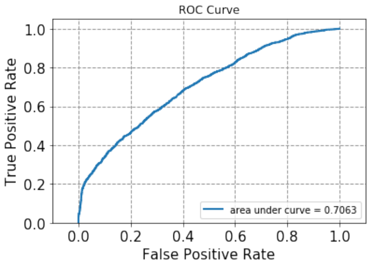
In order to only select tweets whose location is in Washington or Massachusetts, we got through all the tweets and check if their location is any of the cities of these states. If it is, we assign the tweet a label of ”0” for Massachusetts and ”1” for Washington. Processing this way, we collected 55,355 tweets.

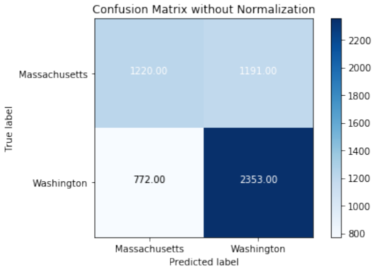
Point 2

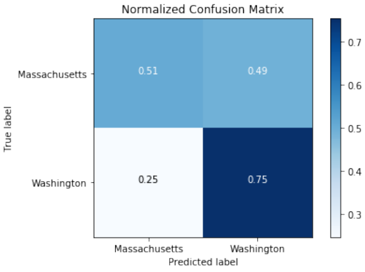
In this part, we will train 4 binary classifiers to predict the location of the author of a tweet.

a. Linear SVM Classifier

We start with a linear SVM classifier. The ROC curve and confusion matrix are reported below.





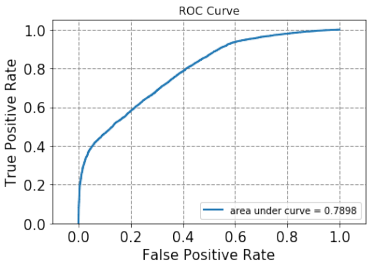


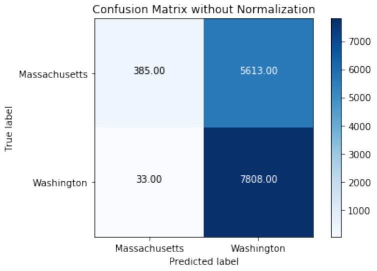
We get the following metric values :

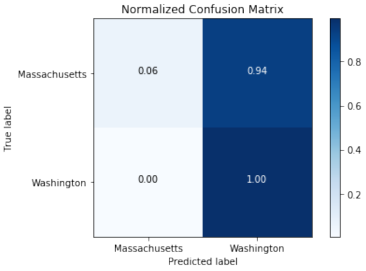
Accuracy	0.64541
Recall	0.75296
Precision	0.66394

b. MultinomialNB

Next, we train a Naïve Bayes classifier with the MultinomialNB model.



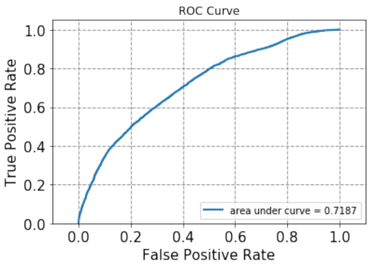


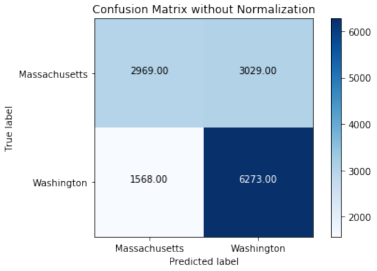


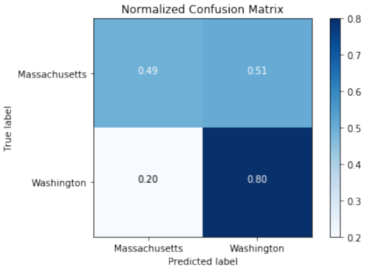
Accuracy	0.59202
Recall	0.99579
Precision	0.58177

c. GaussianNB

We try a Naïve Bayes classifier again, but this time with the GaussianNB model.



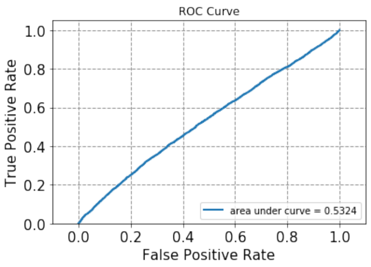


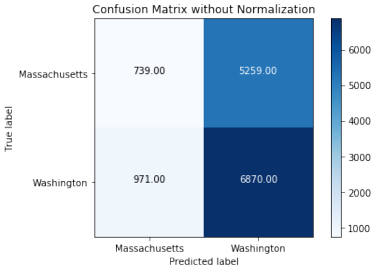


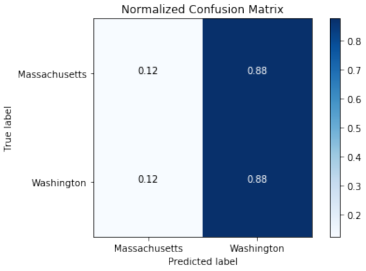
Accuracy	0.66782
Recall	0.80003
Precision	0.67437

d. Logistic Regression

Finally, we try a logistic classifier.







Accuracy	0.54982
Recall	0.87616
Precision	0.56641

Conclusion

From these 4 classifiers, the best results were achieved with the Linear SVM and GaussianNB classifiers. All of the classifiers were able to predict the location of Washington with good accuracy, however they all predicted Massachusetts with bad accuracy. A reason for this could be a lack of data or not having the perfectly right labels assigned when selecting the cities in the two states.

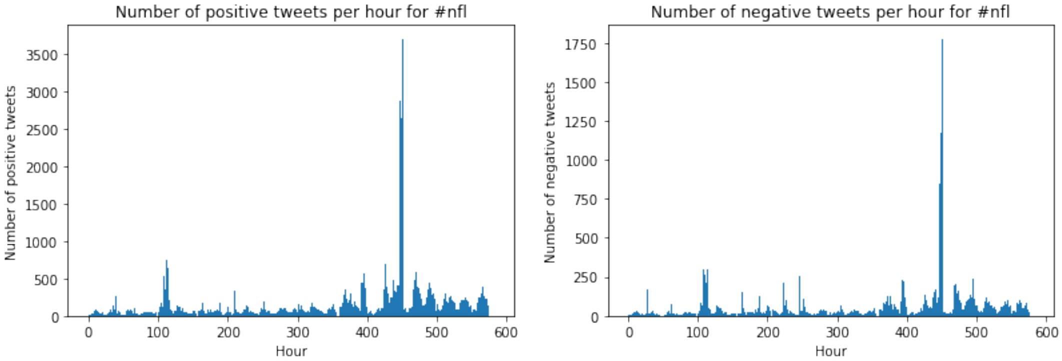
Question 16

In this last part, we will analyze the tweet sentiments for several hashtags in order to understand how people are feeling (positve or negative). The tweet sentiments are obtained using a Python library called Textblob, which allows us to apply NLP techniques.

We chose 4 different hashtags : `#nfl, #superbowl, #gopatriots, #gohawks`
For each hashtag, we will plot the number of positive and negative tweets over the two weeks of data that we have.

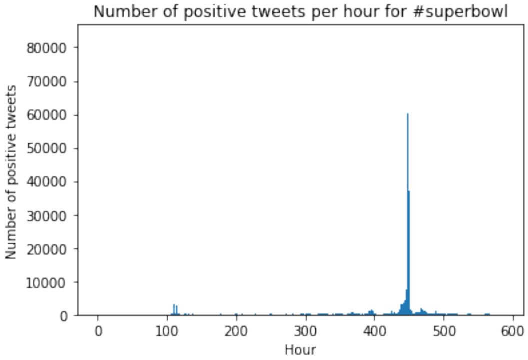
Hashtag #nfl

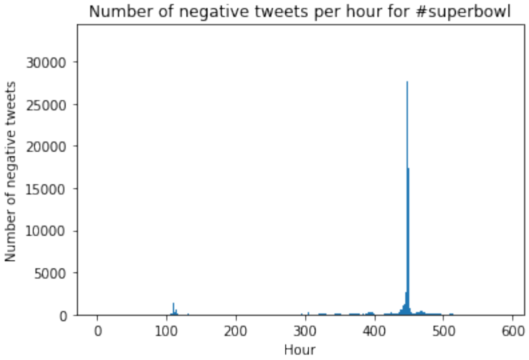
We see that we have two main spikes for both plots. The last spike is the biggest and it corresponds to the final of the Super Bowl. There is double number of positive tweets than negative tweets, which shows that the majority of people had a positive feeling about the event.



Hashtag #superbowl

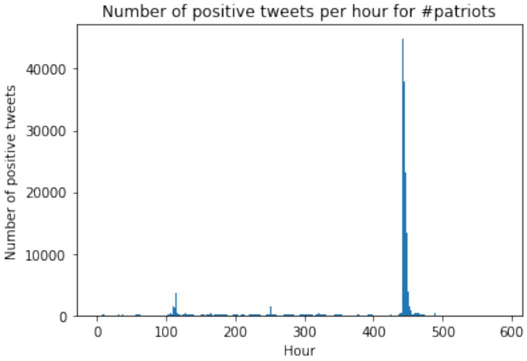
Here we have one main spike. At that moment in time, there are about 85,000 positive tweets and 35,000 negative tweets. The moment corresponds to the final of the Superbowl. The positive tweets are probably all the people happy that their team won, and the negative ones are those who are disappointed.

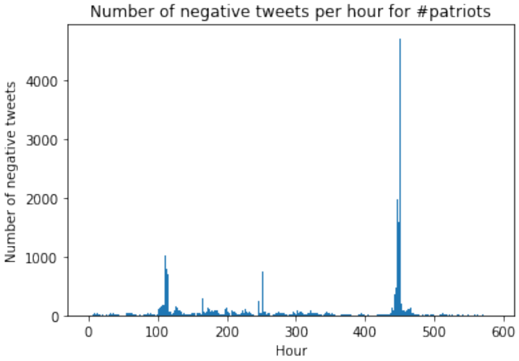




Hashtag #patriots

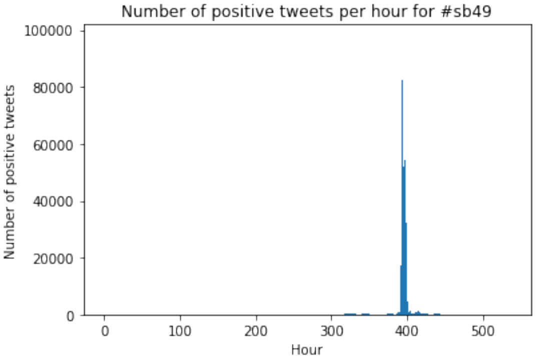
We know that the patriots won the Superbowl 2015, however it was very intense matchup and surprising win as everything changed in the last 2 minutes of the game. This explains the huge spike of tweets that we see. The spike is big both in positives and negatives, which shows that it was a controversial win.

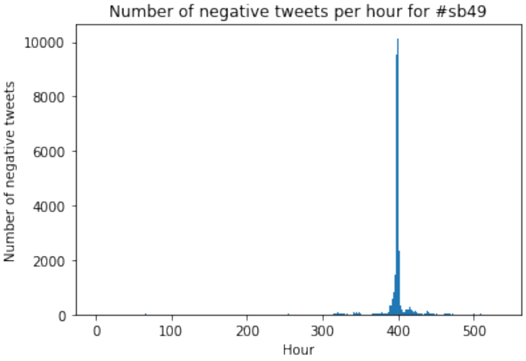




Hashtag #sb49

Finally, with the hashtag `#sb49`, we see an overwhelming amount of positive reactions during the finals. There are 10 times more positive tweets than negative ones. This shows us that overall people enjoyed the event and had fun watching it.


Number of positive tweets per hour for #sb49
Number of positive tweets
Hour


Number of negative tweets per hour for #sb49
Number of negative tweets
Hour