Application - Twitter data

University of California, Los Angeles
ECE 219 Project 5

Joshua Hannan (UID: 805221851)
James (Zach) Harris (UID: 105221897)
Dennis Wang (UID: 105229662)
Akash Deep Singh (UID: 405228294)

jhannan@g.ucla.edu
jzharris@g.ucla.edu
dmwang626@g.ucla.edu
akashdeepsingh@g.ucla.edu

Introduction

By using social network analysis, we will predict the future popularity of feeds corresponding to Super Bowl ILIX. Twitter, with its public discussion model, is chosen as the platform with which to perform this analysis. Our aim is to predict the future tweet activity of a particular hashtag knowing its current and previous tweet activity. More specifically, we aim to predict how popular the tweet activity of a particular hashtag is over the span of Super Bowl ILIX.

Part 1: Popularity Prediction

1. A first look at the data

The tweet data used in this project is separated into a train[1] and test[2] set. Each set contains six text files with information recorded by a particular hashtag specified by the file name. We take these files, extract the features we require for a particular section, and perform our analysis on these features. The twitter data was collected by querying hashtags realted to Super Bowl ILIX spanning a period starting from 2 weeks before the game to a week after the game.

Question 1 The training data was downloaded, parsed, and analyzed. Key features were extracted from every tweet object. Table 1 displays the average tweets per hour, followers per tweeter, and retweets per tweet for every hashtag. These statistics give us an idea of how to approach popularity prediction. The average tweets per hour varies per hashtag, and is the reason why we will be training a separate model for every hashtag.

Hashtag	Avg tweets per hour	Avg followers per tweet	Avg retweets per tweet
gohawks	292	1587	2.0
gopatriots	41	1306	1.4
nfl	397	4356	1.5
patriots	751	1697	1.8
sb49	1277	2343	2.5
superbowl	2072	3652	2.4

Table 1. Popularity statistics of hashtags

Question 2 The number of tweets per hour for the #superbowl hashtag is shown in Figure 1. The number of tweets per hour for the #nfl hashtag is shown in Figure 2. It is noticeable from Figure 2 that there are short bursts of activity when there are many tweets followed by brief points in time when few tweets are

[1] https://ucla.box.com/s/24oxnhsoj6kpxhl6gyvuck25i3s4426d
[2] https://ucla.box.com/s/nkgqr39embdt67lod4pc289xe07eg6wr

present. There is a spike towards the beginning of the time period, which can be taken as discussion of the upcoming game. The second spike occurs after the game has ended, and can be taken as discussion of the results of the game. We will use these properties in the data to help us predict the number of tweets in the hour following the one which we sampled.

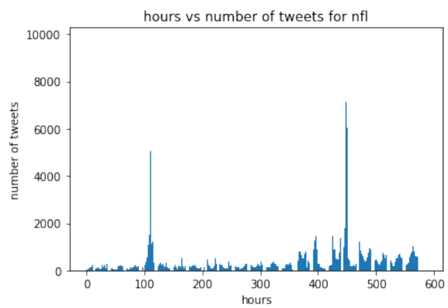

Fig. 1. tweets per hour for #superbowl

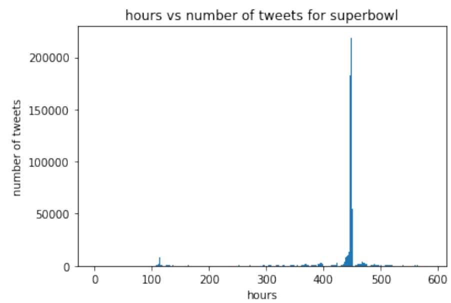

Fig. 2. tweets per hour for #nfl

2. Linear regression

Question 3 For this question, we divide our data into two sets, training and testing in the ratio of 9:1 i.e. 10% of all data is used as testing data. We report the MSE on both training and testing sets while the R-squared measure is reported on the testing set.

For the significance of features analysis, we use the **testing dataset** to fit our OLS model.

In the analysis, the features are numbered from x1 to x5, these correspond to:

1. **x1:** Total number of tweets
2. **x2:** Total number of retweets
3. **x3:** Total number of followers
4. **x4:** Maximum number of followers
5. **x5:** Time of Day

1. **#gohawks**
   (a) MSE training: 803087.59
   (b) MSE testing: 382082.60
   (c) R-squared measure: 0.8867

```
Hashtag: gohawks
X shape: (586, 5)
y shape: (586,)
MSE: train = 803087.5978721752
MSE: test = 382082.6092302024
R2:  0.8867637310828246
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.887
Model:                            OLS   Adj. R-squared:                  0.876
Method:                 Least Squares   F-statistic:                     84.58
Date:                Thu, 21 Mar 2019   Prob (F-statistic):           2.72e-24
Time:                        11:34:45   Log-Likelihood:                -340.32
No. Observations:                  59   AIC:                             690.6
Df Residuals:                      54   BIC:                             701.0
Df Model:                           5
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
x1            -0.0299      0.060     -0.498      0.620      -0.150       0.090
x2             0.0154      0.007      2.154      0.036       0.001       0.030
x3             0.0005   8.52e-05      5.384      0.000       0.000       0.001
x4            -0.0004   9.16e-05     -4.866      0.000      -0.001      -0.000
x5             3.2400      0.965      3.359      0.001       1.306       5.174
==============================================================================
Omnibus:                        8.613   Durbin-Watson:                   1.979
Prob(Omnibus):                  0.013   Jarque-Bera (JB):                8.369
Skew:                           0.910   Prob(JB):                       0.0152
Kurtosis:                       3.302   Cond. No.                     9.26e+04
==============================================================================
```

Fig. 3. Feature analysis for #gohawks

2. **#gopatriots**

   (a) MSE training: 18859.59

   (b) MSE testing: 153476.51

   (c) R-squared measure: 0.9637

```
-------------
Hashtag: gopatriots
X shape: (586, 5)
y shape: (586,)
MSE: train = 18859.599533908844
MSE: test = 153476.51743000178
R2:  0.9637555236015055
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.964
Model:                            OLS   Adj. R-squared:                  0.960
Method:                 Least Squares   F-statistic:                     287.2
Date:                Thu, 21 Mar 2019   Prob (F-statistic):           1.35e-37
Time:                        11:34:45   Log-Likelihood:                -184.05
No. Observations:                  59   AIC:                             378.1
Df Residuals:                      54   BIC:                             388.5
Df Model:                           5
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
x1             1.2462      0.313      3.986      0.000       0.619       1.873
x2             0.0432      0.021      2.090      0.041       0.002       0.085
x3            -0.0013      0.000     -4.786      0.000      -0.002      -0.001
x4             0.0019      0.000      6.037      0.000       0.001       0.002
x5             0.0074      0.115      0.064      0.949      -0.223       0.238
==============================================================================
Omnibus:                       23.859   Durbin-Watson:                   1.760
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               86.139
Skew:                           0.940   Prob(JB):                     1.97e-19
Kurtosis:                       8.613   Cond. No.                     2.31e+05
==============================================================================
```

Fig. 4. Feature analysis for #gopatriots

3. **#nfl**

   (a) MSE training: 187019.44

   (b) MSE testing: 1574527.91

   (c) R-squared measure: 0.7483

```
Hashtag: nfl
X shape: (586, 5)
y shape: (586,)
MSE: train = 187019.44258428575
MSE: test = 1574527.917336355
R2:  0.7482686765262225
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.748
Model:                            OLS   Adj. R-squared:                  0.725
Method:                 Least Squares   F-statistic:                     32.10
Date:                Thu, 21 Mar 2019   Prob (F-statistic):           4.98e-15
Time:                        11:34:45   Log-Likelihood:                -401.72
No. Observations:                  59   AIC:                             813.4
Df Residuals:                      54   BIC:                             823.8
Df Model:                           5
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
x1            -0.7349      0.292     -2.514      0.015      -1.321      -0.149
x2            -0.1926      0.165     -1.168      0.248      -0.523       0.138
x3             0.0003   5.18e-05      6.572      0.000       0.000       0.000
x4            -0.0003   5.86e-05     -5.738      0.000      -0.000      -0.000
x5            22.2922      2.508      8.887      0.000      17.263      27.321
==============================================================================
Omnibus:                        6.642   Durbin-Watson:                   2.474
Prob(Omnibus):                  0.036   Jarque-Bera (JB):                5.976
Skew:                           0.587   Prob(JB):                       0.0504
Kurtosis:                       4.026   Cond. No.                     4.57e+05
==============================================================================
```

Fig. 5. Feature analysis for #nfl

4. **#patriots**

   (a) MSE training: 4490876.82

   (b) MSE testing: 14134817.89

   (c) R-squared measure: 0.9216

```
Hashtag: patriots
X shape: (586, 5)
y shape: (586,)
MSE: train = 4490876.826579147
MSE: test = 14134817.891468015
R2:  0.921584767880181
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.922
Model:                            OLS   Adj. R-squared:                  0.914
Method:                 Least Squares   F-statistic:                     126.9
Date:                Thu, 21 Mar 2019   Prob (F-statistic):           1.41e-28
Time:                        11:34:45   Log-Likelihood:                -385.31
No. Observations:                  59   AIC:                             780.6
Df Residuals:                      54   BIC:                             791.0
Df Model:                           5
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
x1            -0.3302      0.053     -6.276      0.000      -0.436      -0.225
x2             0.2375      0.040      5.923      0.000       0.157       0.318
x3          4.659e-05   2.98e-05      1.562      0.124   -1.32e-05       0.000
x4          -2.53e-05   3.88e-05     -0.653      0.517      -0.000    5.24e-05
x5             8.3328      1.814      4.595      0.000       4.697      11.969
==============================================================================
Omnibus:                       27.602   Durbin-Watson:                   2.062
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               48.002
Skew:                           1.606   Prob(JB):                     3.77e-11
Kurtosis:                       6.034   Cond. No.                     8.44e+05
==============================================================================
```

Fig. 6. Feature analysis for #patriots

5. **#sb49**

   (a) MSE training: 17791079.68

   (b) MSE testing: 829389.08

   (c) R-squared measure: 0.9700

```
Hashtag: sb49
X shape: (586, 5)
y shape: (586,)
MSE: train = 17791079.6837765
MSE: test = 829389.0810928787
R2:  0.9700598249543628
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.970
Model:                            OLS   Adj. R-squared:                  0.967
Method:                 Least Squares   F-statistic:                     349.9
Date:                Thu, 21 Mar 2019   Prob (F-statistic):           7.81e-40
Time:                        11:34:45   Log-Likelihood:                -386.78
No. Observations:                  59   AIC:                             783.6
Df Residuals:                      54   BIC:                             794.0
Df Model:                           5
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
x1             0.2690      0.031      8.814      0.000       0.208       0.330
x2             0.0309      0.009      3.547      0.001       0.013       0.048
x3          4.099e-05   5.75e-06      7.129      0.000    2.95e-05    5.25e-05
x4         -7.411e-05   1.64e-05     -4.527      0.000      -0.000   -4.13e-05
x5             1.8975      2.002      0.948      0.347      -2.116       5.911
==============================================================================
Omnibus:                       60.565   Durbin-Watson:                   2.043
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              808.340
Skew:                           2.386   Prob(JB):                    2.96e-176
Kurtosis:                      20.494   Cond. No.                     1.34e+06
==============================================================================
```

Fig. 7. Feature analysis for #sb49

6. **#superbowl**

   (a) MSE training: 40603248.51

   (b) MSE testing: 187781139.51

   (c) R-squared measure: 0.4821

```
Hashtag: superbowl
X shape: (586, 5)
y shape: (586,)
MSE: train = 40603248.516695604
MSE: test = 187781139.5116839
R2:  0.4821481610512124
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.482
Model:                            OLS   Adj. R-squared:                  0.434
Method:                 Least Squares   F-statistic:                     10.06
Date:                Thu, 21 Mar 2019   Prob (F-statistic):           7.69e-07
Time:                        11:34:45   Log-Likelihood:                -489.15
No. Observations:                  59   AIC:                             988.3
Df Residuals:                      54   BIC:                             998.7
Df Model:                           5
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
x1             0.3440      0.562      0.611      0.543      -0.784       1.472
x2            -0.1972      0.193     -1.023      0.311      -0.584       0.189
x3          2.066e-05   3.67e-05      0.563      0.576   -5.29e-05    9.42e-05
x4             0.0001      0.000      0.946      0.348      -0.000       0.000
x5            18.1275     12.436      1.458      0.151      -6.806      43.061
==============================================================================
Omnibus:                      107.665   Durbin-Watson:                   2.016
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             3334.391
Skew:                           5.571   Prob(JB):                         0.00
Kurtosis:                      38.103   Cond. No.                     1.04e+07
==============================================================================
```

Fig. 8. Feature analysis for #superbowl

3. Feature analysis

Question 4 For this part, in addition to the previous 5 features, we have used 6 other features selected from the data. These include:

1. **Influence Score:** Users who have more influence can generate higher activity and hence more number of tweets
2. **Number of replies:** If a tweet has generated a large number of replies, it means that the topic is engaging and we can expect more tweets.
3. **Impressions:** Higher number of impressions can correlate to more tweet activity.
4. **Peak:** Peak of acceleration. refer to [XZJ+16]
5. **Acceleration:** The rate of change of the velocity of tweet stream where velocity is the rate of change of the volume of tweet stream. refer to [XZJ+16]
6. **User's status - verified or not:** Verified users have more followers on an average and any activity by them will generate a greater response.

Again, for this question, we divide our data into two sets, training and testing in the ratio of 9:1 i.e. 10% of all data is used as testing data. We report the MSE on both training and testing sets while the R-squared measure is reported on the testing set.

For the significance of features analysis, we use the **testing dataset** to fit our OLS model.

In the analysis, the features are numbered from x1 to x11, these correspond to:

1. **x1:** Total number of tweets
2. **x2:** Total number of retweets
3. **x3:** Total number of followers
4. **x4:** Maximum number of followers
5. **x5:** Influence Score
6. **x6:** Number of Replies
7. **x7:** Number of Impressions
8. **x8:** Peak
9. **x9:** Acceleration
10. **x10:** User's status: verified or not?
11. **x11:** Time of Day

1. **#gohawks**

   (a) MSE training: 689619.01

   (b) MSE testing: 366067.65

   (c) R-squared measure: 0.9427

```
Hashtag: gohawks
X shape: (586, 11)
y shape: (586,)
MSE: train = 689619.0155376416
MSE: test = 366067.64967003796
R2:  0.9427297968350369
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.943
Model:                            OLS   Adj. R-squared:                  0.930
Method:                 Least Squares   F-statistic:                     71.83
Date:                Thu, 21 Mar 2019   Prob (F-statistic):           6.05e-26
Time:                        11:49:02   Log-Likelihood:                -320.21
No. Observations:                  59   AIC:                             662.4
Df Residuals:                      48   BIC:                             685.3
Df Model:                          11
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
x1             0.5986      0.168      3.566      0.001       0.261       0.936
x2            -0.1142      0.049     -2.339      0.024      -0.212      -0.016
x3          8.142e-05      0.000      0.777      0.441      -0.000       0.000
x4            -0.0001      0.000     -1.043      0.302      -0.000       0.000
x5            11.5325      3.642      3.167      0.003       4.210      18.855
x6             8.1708      8.167      1.000      0.322      -8.251      24.592
x7          3.304e-05   2.91e-05      1.137      0.261   -2.54e-05    9.15e-05
x8         -3.711e-09   4.16e-09     -0.892      0.377   -1.21e-08    4.65e-09
x9            -0.0794      0.117     -0.680      0.500      -0.314       0.155
x10            9.5903      5.967      1.607      0.115      -2.407      21.587
x11            0.6043      0.863      0.700      0.487      -1.132       2.340
==============================================================================
Omnibus:                       10.353   Durbin-Watson:                   2.064
Prob(Omnibus):                  0.006   Jarque-Bera (JB):               23.253
Skew:                           0.311   Prob(JB):                     8.93e-06
Kurtosis:                       6.012   Cond. No.                     3.39e+10
==============================================================================
```

Fig. 9. Feature analysis for #gohawks

2. **#gopatriots**

   (a) MSE training: 12561.03

   (b) MSE testing: 108210.94

   (c) R-squared measure: 0.9452

```
Hashtag: gopatriots
X shape: (586, 11)
y shape: (586,)
MSE: train = 12561.034976186296
MSE: test = 108210.94481689823
R2:  0.945243695605184
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.945
Model:                            OLS   Adj. R-squared:                  0.933
Method:                 Least Squares   F-statistic:                     75.33
Date:                Thu, 21 Mar 2019   Prob (F-statistic):           2.08e-26
Time:                        11:49:02   Log-Likelihood:                -196.23
No. Observations:                  59   AIC:                             414.5
Df Residuals:                      48   BIC:                             437.3
Df Model:                          11
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
x1            -0.1803      0.415     -0.434      0.666      -1.015       0.655
x2             0.1664      0.155      1.073      0.289      -0.145       0.478
x3             0.0009      0.003      0.316      0.754      -0.005       0.007
x4             0.0006      0.001      0.515      0.609      -0.002       0.003
x5             2.2300      5.867      0.380      0.706      -9.566      14.026
x6            -1.4972      4.901     -0.305      0.761     -11.352       8.357
x7            -0.0014      0.003     -0.497      0.622      -0.007       0.004
x8          2.586e-09    2.1e-09      1.230      0.225   -1.64e-09    6.81e-09
x9            -0.0930      0.087     -1.073      0.289      -0.267       0.081
x10          -54.2876     69.056     -0.786      0.436    -193.135      84.560
x11            0.4449      0.130      3.411      0.001       0.183       0.707
==============================================================================
Omnibus:                       50.172   Durbin-Watson:                   1.587
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              251.437
Skew:                           2.304   Prob(JB):                     2.52e-55
Kurtosis:                      12.002   Cond. No.                     7.56e+11
==============================================================================
```

Fig. 10. Feature analysis for #gopatriots

3. **#nfl**

   (a) MSE training: 163832.03

   (b) MSE testing: 2707278.00

   (c) R-squared measure: 0.9223

```
Hashtag: nfl
X shape: (586, 11)
y shape: (586,)
MSE: train = 163832.03576963246
MSE: test = 2707278.003069478
R2:  0.9223838132661387
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.922
Model:                            OLS   Adj. R-squared:                  0.905
Method:                 Least Squares   F-statistic:                     51.86
Date:                Thu, 21 Mar 2019   Prob (F-statistic):           8.12e-23
Time:                        11:49:02   Log-Likelihood:                -367.01
No. Observations:                  59   AIC:                             756.0
Df Residuals:                      48   BIC:                             778.9
Df Model:                          11
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
x1             0.1378      0.195      0.706      0.484      -0.255       0.530
x2             0.2012      0.142      1.416      0.163      -0.084       0.487
x3             0.0004      0.000      2.849      0.006       0.000       0.001
x4            -0.0002   5.63e-05     -4.253      0.000      -0.000      -0.000
x5           -12.2182      7.493     -1.631      0.110     -27.283       2.847
x6            11.9740     10.741      1.115      0.270      -9.622      33.571
x7            -0.0002      0.000     -1.683      0.099      -0.000    3.88e-05
x8         -1.193e-08   4.97e-09     -2.399      0.020   -2.19e-08   -1.93e-09
x9            -0.1675      0.157     -1.066      0.292      -0.483       0.148
x10            9.3487      6.778      1.379      0.174      -4.279      22.977
x11            6.4120      2.129      3.012      0.004       2.132      10.692
==============================================================================
Omnibus:                        4.105   Durbin-Watson:                   1.950
Prob(Omnibus):                  0.128   Jarque-Bera (JB):                3.560
Skew:                           0.601   Prob(JB):                        0.169
Kurtosis:                       3.071   Cond. No.                     3.44e+10
==============================================================================
```

Fig. 11. Feature analysis for #nfl

4. **#patriots**

   (a) MSE training: 4018431.97

   (b) MSE testing: 9747408.08

   (c) R-squared measure: 0.9609

```
Hashtag: patriots
X shape: (586, 11)
y shape: (586,)
MSE: train = 4018431.979524465
MSE: test = 9747408.085532961
R2:  0.9608887502574825
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.961
Model:                            OLS   Adj. R-squared:                  0.952
Method:                 Least Squares   F-statistic:                     107.2
Date:                Thu, 21 Mar 2019   Prob (F-statistic):           6.96e-30
Time:                        11:49:02   Log-Likelihood:                -364.79
No. Observations:                  59   AIC:                             751.6
Df Residuals:                      48   BIC:                             774.4
Df Model:                          11
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
x1             0.2212      0.139      1.595      0.117      -0.058       0.500
x2            -0.0987      0.203     -0.486      0.629      -0.507       0.310
x3            -0.0002      0.000     -1.146      0.258      -0.000       0.000
x4          2.761e-05   3.12e-05      0.886      0.380   -3.51e-05    9.03e-05
x5             5.1592      6.676      0.773      0.443      -8.263      18.582
x6             1.9024      2.614      0.728      0.470      -3.353       7.158
x7             0.0001      0.000      1.027      0.310      -0.000       0.000
x8          7.691e-10   5.17e-09      0.149      0.882   -9.63e-09    1.12e-08
x9            -0.0933      0.167     -0.560      0.578      -0.428       0.242
x10           26.1065      4.182      6.243      0.000      17.699      34.514
x11            2.8370      1.613      1.759      0.085      -0.407       6.080
==============================================================================
Omnibus:                       52.837   Durbin-Watson:                   2.043
Prob(Omnibus):                  0.000   Jarque-Bera (JB):              272.941
Skew:                           2.460   Prob(JB):                     5.39e-60
Kurtosis:                      12.318   Cond. No.                     5.88e+10
==============================================================================
```

Fig. 12. Feature analysis for #patriots

5. **#sb49**

   (a) MSE training: 16620518.44

   (b) MSE testing: 2519209.44

   (c) R-squared measure: 0.9912

```
Hashtag: sb49
X shape: (586, 11)
y shape: (586,)
MSE: train = 16620518.44350101
MSE: test = 2519209.4423165093
R2:  0.9912782264602494
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.991
Model:                            OLS   Adj. R-squared:                  0.989
Method:                 Least Squares   F-statistic:                     496.0
Date:                Thu, 21 Mar 2019   Prob (F-statistic):           1.82e-45
Time:                        11:49:02   Log-Likelihood:                -350.40
No. Observations:                  59   AIC:                             722.8
Df Residuals:                      48   BIC:                             745.7
Df Model:                          11
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
x1             0.6637      0.178      3.738      0.000       0.307       1.021
x2             0.1009      0.021      4.774      0.000       0.058       0.143
x3          6.267e-05   2.29e-05      2.740      0.009    1.67e-05       0.000
x4         -5.394e-05   2.33e-05     -2.319      0.025      -0.000   -7.17e-06
x5            -9.9948      2.993     -3.340      0.002     -16.012      -3.977
x6            -3.6774      1.635     -2.249      0.029      -6.965      -0.390
x7         -1.128e-05   9.42e-06     -1.197      0.237   -3.02e-05    7.66e-06
x8         -6.352e-09   3.33e-09     -1.908      0.062    -1.3e-08    3.43e-10
x9            -0.0202      0.047     -0.425      0.673      -0.116       0.075
x10           11.1295      3.649      3.050      0.004       3.793      18.466
x11            1.6518      1.299      1.272      0.210      -0.959       4.263
==============================================================================
Omnibus:                       18.273   Durbin-Watson:                   1.934
Prob(Omnibus):                  0.000   Jarque-Bera (JB):               34.640
Skew:                           0.949   Prob(JB):                     3.01e-08
Kurtosis:                       6.239   Cond. No.                     2.55e+10
==============================================================================
```

Fig. 13. Feature analysis for #sb49

6. **#superbowl**

   (a) MSE training: 37809842.29

   (b) MSE testing: 143138928.97

   (c) R-squared measure: 0.6052

```
Hashtag: superbowl
X shape: (586, 11)
y shape: (586,)
MSE: train = 37809842.29122893
MSE: test = 143138928.97942558
R2:  0.6052682417738593
                            OLS Regression Results
==============================================================================
Dep. Variable:                      y   R-squared:                       0.605
Model:                            OLS   Adj. R-squared:                  0.515
Method:                 Least Squares   F-statistic:                     6.691
Date:                Thu, 21 Mar 2019   Prob (F-statistic):           1.21e-06
Time:                        11:49:02   Log-Likelihood:                -481.14
No. Observations:                  59   AIC:                             984.3
Df Residuals:                      48   BIC:                             1007.
Df Model:                          11
Covariance Type:            nonrobust
==============================================================================
                 coef    std err          t      P>|t|      [0.025      0.975]
------------------------------------------------------------------------------
x1             0.7341      0.801      0.917      0.364      -0.876       2.344
x2             1.2343      0.712      1.734      0.089      -0.197       2.666
x3            -0.0001      0.000     -0.498      0.621      -0.001       0.000
x4             0.0002      0.000      1.390      0.171    -8.2e-05       0.000
x5           -25.0769     29.488     -0.850      0.399     -84.367      34.213
x6            13.4032     37.945      0.353      0.725     -62.891      89.698
x7          6.719e-06      0.000      0.028      0.978      -0.000       0.000
x8         -5.195e-08   2.98e-08     -1.743      0.088   -1.12e-07    7.98e-09
x9             0.3160      0.592      0.534      0.596      -0.874       1.506
x10           50.2192     16.559      3.033      0.004      16.924      83.514
x11            4.2407     12.789      0.332      0.742     -21.473      29.955
==============================================================================
Omnibus:                      104.118   Durbin-Watson:                   1.836
Prob(Omnibus):                  0.000   Jarque-Bera (JB):             3077.834
Skew:                           5.265   Prob(JB):                         0.00
Kurtosis:                      36.781   Cond. No.                     2.42e+11
==============================================================================
```

Fig. 14. Feature analysis for #superbowl

Question 5 Top 3 features for each hashtag:
Top 3 features for #gohawks:

1. Total number of tweets
2. Total number of retweets
3. Influence Score

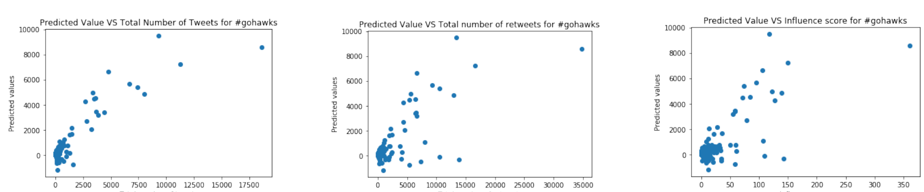

Fig. 15. Top 3 features for #gohawks

Top 3 features for #gopatriots:

1. Acceleration
2. Peak
3. Total number of retweets

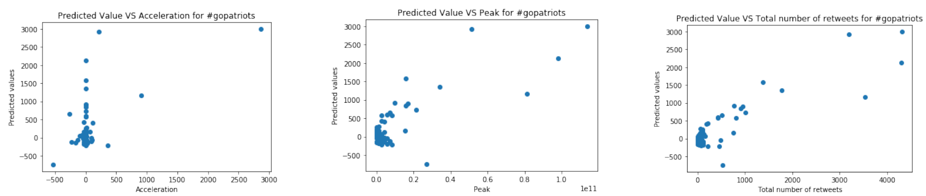

Fig. 16. Top 3 features for #gopatriots

Top 3 features for #nfl:

1. Total number of followers
2. Maximum number of followers
3. Imopressions

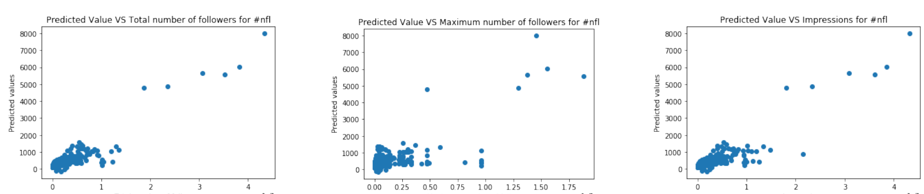

Fig. 17. Top 3 features for #nfl

Top 3 features for #patriots:

1. Total number of tweets
2. Total number of followers
3. Impressions

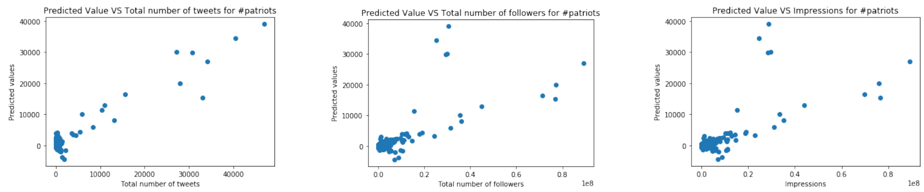

Fig. 18. Top 3 features for #patriots

Top 3 features for #sb49:

1. Total number of tweets
2. Total number of retweets
3. Influence Score

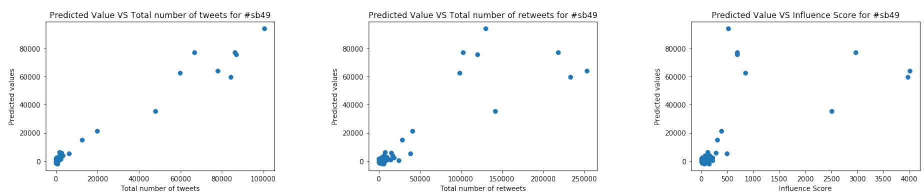

Fig. 19. Top 3 features for #sb49

Top 3 features for #superbowl:

1. Total number of retweets
2. Peak
3. Maximum number of followers

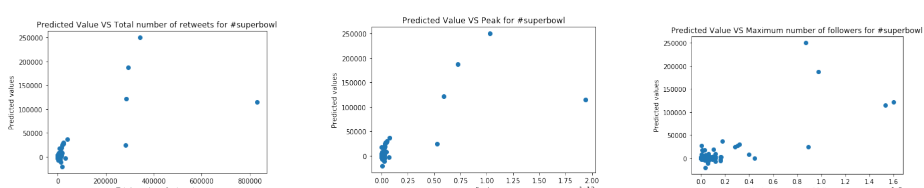

Fig. 20. Top 3 features for #superbowl

Analysis: The accuracy of the Linear Regression model increases if we use more features. Also, we can notice that some of the newly introduced features have a higher significance than the old features. In all of the plots above, we can observe a linear relationship between the predictant and the value of feature. This means that the features are a able to predict the number of tweets in the next hour very well.

The regression coefficients are basically the slope of the line. By looking at the plots above we can see that the **regressions coefficients agree with the plots.**

4. Piece-wise linear regression

In this section, instead of dividing our data into train and test sets, we perform a 10-fold cross validation. We report the MSE on both training and validation sets while the R-squared measure is reported on the validation set only.

Question 6

1. **#gohawks**
   (a) **Before Feb. 1, 8:00 a.m.: 1-hour window**
      i. MSE training: 713515.05
      ii. MSE validation: 971076.11
      iii. R-squared measure: 0.8981
   (b) **During Feb. 1, 8:00 a.m. and 8:00 p.m.: 5-minute window**
      i. MSE training: 71272.59
      ii. MSE validation: 88566.25
      iii. R-squared measure: 0.9717
   (c) **After Feb. 1, 8:00 p.m.: 1-hour window**
      i. MSE training: 1653.48
      ii. MSE validation: 10163.34
      iii. R-squared measure: 0.1904
2. **#gopatriots**
   (a) **Before Feb. 1, 8:00 a.m.: 1-hour window**
      i. MSE training: 1739.24
      ii. MSE validation: 1972.76
      iii. R-squared measure: 0.9296
   (b) **During Feb. 1, 8:00 a.m. and 8:00 p.m.: 5-minute window**
      i. MSE training: 13528.63
      ii. MSE validation: 21219.55
      iii. R-squared measure: 0.7610
   (c) **After Feb. 1, 8:00 p.m.: 1-hour window**
      i. MSE training: 40.81
      ii. MSE validation: 422.29
      iii. R-squared measure: 0.0
3. **#nfl**
   (a) **Before Feb. 1, 8:00 a.m.: 1-hour window**
      i. MSE training: 65822.01
      ii. MSE validation: 68272.52

iii. R-squared measure: 0.9067

(b) **During Feb. 1, 8:00 a.m. and 8:00 p.m.: 5-minute window**
   i. MSE training: 20897.52
   ii. MSE validation: 25105.72
   iii. R-squared measure: 0.9214

(c) **After Feb. 1, 8:00 p.m.: 1-hour window**
   i. MSE training: 16540.17
   ii. MSE validation: 18256.87
   iii. R-squared measure: 0.9986

4. **#patriots**
   (a) **Before Feb. 1, 8:00 a.m.: 1-hour window**
      i. MSE training: 335489.9
      ii. MSE validation: 480760.85
      iii. R-squared measure: 0.9306
   (b) **During Feb. 1, 8:00 a.m. and 8:00 p.m.: 5-minute window**
      i. MSE training: 665369.2
      ii. MSE validation: 768002.08
      iii. R-squared measure: 0.8908
   (c) **After Feb. 1, 8:00 p.m.: 1-hour window**
      i. MSE training: 10928.53
      ii. MSE validation: 556520.27
      iii. R-squared measure: 0.9044
5. **#sb49**
   (a) **Before Feb. 1, 8:00 a.m.: 1-hour window**
      i. MSE training: 6810.38
      ii. MSE validation: 7985.99
      iii. R-squared measure: 0.8963
   (b) **During Feb. 1, 8:00 a.m. and 8:00 p.m.: 5-minute window**
      i. MSE training: 1298290.43
      ii. MSE validation: 1841917.33
      iii. R-squared measure: 0.9695
   (c) **After Feb. 1, 8:00 p.m.: 1-hour window**
      i. MSE training: 72039.19
      ii. MSE validation: 129386.66
      iii. R-squared measure: 0.8758
6. **#superbowl**
   (a) **Before Feb. 1, 8:00 a.m.: 1-hour window**
      i. MSE training: 511103.87
      ii. MSE validation: 658401.39
      iii. R-squared measure: 0.9297
   (b) **During Feb. 1, 8:00 a.m. and 8:00 p.m.: 5-minute window**
      i. MSE training: 6763117.23
      ii. MSE validation: 20853554.36
      iii. R-squared measure: 0.9764
   (c) **After Feb. 1, 8:00 p.m.: 1-hour window**
      i. MSE training: 109427.01
      ii. MSE validation: 207211.56
      iii. R-squared measure: 0.9723

Question 7

1. **Before Feb. 1, 8:00 a.m.: 1-hour window**
   (a) MSE training: 4415470.09
   (b) MSE validation: 4734751.73
   (c) R-squared measure: 0.9185
2. **During Feb. 1, 8:00 a.m. and 8:00 p.m.: 5-minute window**
   (a) MSE training: 17177842.96
   (b) MSE validation: 32165266.55
   (c) R-squared measure: 0.9879
3. **After Feb. 1, 8:00 p.m.: 1-hour window**
   (a) MSE training: 435691.47
   (b) MSE validation: 1319725.92
   (c) R-squared measure: 0.8989

Comparison of combined model with individual models: From the statistics in Questions 6 and 7, we can conclude that the combined model performs better for the second time period, i.e. During Feb. 1, 8:00 a.m. and 8:00 p.m. based on the R-squared measure and MSE. However, for the time period After Feb. 1, 8:00 p.m., the combined model performs considerably worse. For the time period: Before Feb. 1, 8:00 a.m., both show similar performance.

Model	Before Feb. 1, 8:00 a.m.			During Feb. 1, 8:00 a.m. and 8:00 p.m.			After Feb. 1, 8:00 p.m.		
	MSE train	MSE valid	R^2	MSE train	MSE valid	R^2	MSE train	MSE valid	R^2
Combined	4415470.09	4734751.73	0.9185	17177842.96	32165266.55	0.9879	435691.47	1319725.92	0.8989
#gohawks	713515.05	971076.11	0.8981	71272.59	88566.25	0.9717	1653.48	10163.34	0.1904
#gopatriots	1739.24	1972.76	0.9296	13528.63	21219.55	0.7610	40.81	422.29	0.0
#nfl	65822.01	68272.52	0.9067	20897.52	25105.72	0.9214	16540.17	18256.87	0.9986
#patriots	335489.9	480760.85	0.9306	665369.2	768002.08	0.8908	10928.53	556520.27	0.9044
#sb49	6810.38	7985.99	0.8963	1298290.43	1841917.33	0.9695	72039.19	129386.66	0.8758
#superbowl	511103.87	658401.39	0.9297	6763117.23	20853554.36	0.9764	109427.01	207211.56	0.9723

Table 2. Comparison between the combined and individual models

5. Nonlinear regressions

Ensemble methods The Random Forest Regressor (RFR) and Gradient Boosting Regressor (GBR) ensemble regressors are used to predict the number of tweets in the next hour given features of the data in the present hour.

Question 8 A grid search was performed over the parameters shown in Table 3. 5-fold cross validation was used, and the 'neg_mean_squared_error' scoring function was used to minimize the MSE. The original five features from Question 3. Table 4 shows the top results from performing the grid search on the aggregated training set. The smallest test MSE that the grid search found was 2.06e+08, which is incredibly large. One explanation for this result is that the regression algorithms could not fit the large spikes in the data, such as those in Figures 1 and 2.

Procedure	**Options**
max depth	10, 20, 40, 60, 80, 100, 200, and None
max features	auto and sqrt
min samples leaf	1, 2, and 4
min samples split	2, 5, and 10
number of estimators	200, 400, 600, 800, 1000, 1200, 1400, 1600, 1800, and 2000

Table 3. Parameters for the ensemble methods

It is worth noting the difference between the test and train MSEs. The train MSE is always at least 3 orders of magnitude lower than the test MSE. The models are clearly over-fitting on the training examples they are given, and are unable to generalize the dataset. **This is the case for every grid search example in Question 10 as well.** There are models in the grid search with extremely low train MSEs on the order of e-08, which support this theory. Fitting the models on tweets which occurred before, during, and after the game may improve these results.

Table 5 shows the best parameters for the Random Forest grid search. The best test MSEs are higher than the test MSEs of the Gradient Boosting Regressor, however their training MSEs are closer to original test score. This suggests that the Random Forest Regressor performs less overfitting on the dataset. The Random Forest Regressor is therefore the more stable classifier.

	mean_test_score	mean_train_score	type	depth	feature	leaves	splits	estimators
2143	-2.061977e+08	-56849.752986	GBR	60.0	sqrt	4	5	800
2866	-2.065202e+08	-2931.518875	GBR	NaN	sqrt	4	5	1400
2695	-2.066646e+08	-8750.317487	GBR	200.0	sqrt	4	10	1200
2139	-2.074178e+08	-153.364510	GBR	60.0	sqrt	4	2	2000
1609	-2.079053e+08	-130.429872	GBR	10.0	sqrt	4	5	2000
1598	-2.088445e+08	-436.409037	GBR	10.0	sqrt	4	2	1800
1954	-2.088784e+08	-18063.586477	GBR	40.0	sqrt	4	2	1000
2864	-2.091391e+08	-20336.807186	GBR	NaN	sqrt	4	5	1000
1955	-2.092728e+08	-7293.322956	GBR	40.0	sqrt	4	2	1200
1596	-2.095378e+08	-2680.138808	GBR	10.0	sqrt	4	2	1400

Table 4. Tuned parameters from Gradient Boosted Regressor grid search

	mean_test_score	mean_train_score	type	depth	feature	leaves	splits	estimators
760	-2.374766e+08	-9.993287e+07	RFR	80.0	auto	2	5	200
30	-2.389683e+08	-8.405991e+07	RFR	10.0	auto	2	2	200
754	-2.390084e+08	-8.493333e+07	RFR	80.0	auto	2	2	1000
1292	-2.390505e+08	-8.770154e+07	RFR	NaN	auto	2	2	600
579	-2.391917e+08	-8.487251e+07	RFR	60.0	auto	2	2	2000
752	-2.392440e+08	-8.438472e+07	RFR	80.0	auto	2	2	600
935	-2.394018e+08	-8.413976e+07	RFR	100.0	auto	2	2	1200
934	-2.394504e+08	-8.489432e+07	RFR	100.0	auto	2	2	1000
758	-2.396859e+08	-8.333865e+07	RFR	80.0	auto	2	2	1800
933	-2.396979e+08	-8.538250e+07	RFR	100.0	auto	2	2	800

Table 5. Tuned parameters from Random Forest Regressor grid search

Question 9 The best estimator from Question 8 scored a mean test MSE of 2.06e+08. OLS over the entire dataset yields a cross-validation test MSE of **2.16e+08**. The ensemble methods were able to perform better than the Ordinary Least Squares method, which is to be expected since they are able to learn a more complex distribution on the training data. However it was expected that the nonlinear regressors would perform significantly better than OLS. Further fine-tuning is required to determine if there is a combination of hyperparameters for the RF and GB Regressors which gets a lower test MSE for the aggregated training data.

Question 10 The cross-validation error improves when the grid search is performed again but on time periods corresponding to before, during, and after game day. In each of the three time periods, the best parameters of the estimator changed with respect to the original best estimator. This fact alone suggests that the behavior of tweeters changed during these three time periods.

Table 6 shows the best parameters from a grid search performed on the time period corresponding to before game day. The best scoring GBR estimator scored an average MSE of **5.7e+06**, which is two orders of magnitude lower than when the model was fitted on the entire period. The max depth, number of leaves, number of splits, and number of estimators is lower than the previous estimator. This suggests that predicting the number of tweets before game day is an easier task than predicting the number of tweets across the entire period.

	mean_test_score	mean_train_score	type	depth	feature	leaves	splits	estimators
141	-5.689844e+06	-5.711139e-01	GBR	10.0	sqrt	2	10	400
321	-5.695530e+06	-7.716574e-02	GBR	20.0	sqrt	2	10	400
685	-5.742796e+06	-9.941796e-08	GBR	60.0	sqrt	2	10	1200
140	-5.775355e+06	-2.062176e+02	GBR	10.0	sqrt	2	10	200
1223	-5.813588e+06	-5.018112e-07	GBR	200.0	sqrt	2	10	800
1351	-5.815243e+06	-9.846533e-08	GBR	NaN	sqrt	1	2	400
688	-5.863930e+06	-9.950413e-08	GBR	60.0	sqrt	2	10	1800
502	-5.866801e+06	-5.576346e-04	GBR	40.0	sqrt	2	10	600
142	-5.870778e+06	-2.062107e-03	GBR	10.0	sqrt	2	10	600
1225	-5.883174e+06	-9.916614e-08	GBR	200.0	sqrt	2	10	1200

Table 6. Tuned parameters from grid search, before game

Table 7 shows the best parameters from a grid search performed on the time period corresponding to during game day. The best scoring GBR estimator scored an average MSE of **2.8e+07**, which is yet again lower than when the model was fitted on the entire period. The mean train scores are much lower than the mean train scores for before game day, which suggest that the model is likely over-fitting on this portion of the dataset. The lesser number of samples for this period may be playing a role in why the models are over-fitting.

	mean_test_score	mean_train_score	type	depth	feature	leaves	splits	estimators
270	-2.848297e+07	-9.900099e-08	GBR	20.0	sqrt	1	2	200
92	-2.908657e+07	-9.872871e-08	GBR	10.0	sqrt	1	2	600
1375	-2.913824e+07	-9.808062e-08	GBR	NaN	sqrt	1	10	1200
1357	-2.946787e+07	-9.836797e-08	GBR	NaN	sqrt	1	2	1600
520	-2.952291e+07	-1.194109e+05	GBR	40.0	sqrt	4	5	200
845	-2.961121e+07	-9.886103e-08	GBR	80.0	sqrt	2	2	1200
279	-2.986481e+07	-9.895955e-08	GBR	20.0	sqrt	1	2	2000
1174	-2.990555e+07	-9.868195e-08	GBR	200.0	sqrt	1	2	1000
678	-2.991034e+07	-9.950403e-08	GBR	60.0	sqrt	2	5	1800
994	-3.014752e+07	-9.901552e-08	GBR	100.0	sqrt	1	2	1000

Table 7. Tuned parameters from grid search, during game

Table 8 shows the best parameters from a grid search performed on the time period corresponding to after game day. The best scoring GBR estimator scored an average MSE of **3.3e+05**, which is the lowest of all periods. This could be due to the fact that the tweets become less sporadic after game day. During and before game day, there are outside factors that are affecting when people tweet, such as a sudden change in the game, or speculation beforehand.

	mean_test_score	mean_train_score	type	depth	feature	leaves	splits	estimators
169	-331572.339233	-1.110052e-07	GBR	10.0	sqrt	4	5	2000
1236	-336085.514476	-8.297246e-07	GBR	200.0	sqrt	4	2	1400
700	-337121.354505	-4.634730e+02	GBR	60.0	sqrt	4	5	200
344	-338091.737329	-3.207575e-04	GBR	20.0	sqrt	4	5	1000
525	-338362.920420	-1.345420e-05	GBR	40.0	sqrt	4	5	1200
1410	-338972.416711	-4.568849e+02	GBR	NaN	sqrt	4	2	200
890	-340831.686698	-6.154387e+02	GBR	80.0	sqrt	4	10	200
888	-343294.875047	-1.255015e-07	GBR	80.0	sqrt	4	5	1800
521	-343313.807463	-7.297854e+00	GBR	40.0	sqrt	4	5	400
1237	-344354.934503	-2.197247e-07	GBR	200.0	sqrt	4	2	1600

Table 8. Tuned parameters from grid search, after game

Neural network

Question 11 We fit a neural network of varying sizes to the entire data set with 1-hour windows. From table 9, we see that the best size is 10*(50,), or 10 layers with 50 neurons each. After that, we notice that there is severe overfitting, as 10*(100,) has very low training MSE but high validation MSE.

Layer sizes	Validation MSE	Train MSE
(50,50)	59745874037	9108725423
2*(100,)	3508811621	281766647889
3*(100,)	10658749042	30772545538
4*(100,)	8591186972	8205157949
10*(50,)	692571491	8027511120
10*(100,)	1038210307	2809171097

Table 9. Validation and Training MSEs for different neural networks

Question 12 By scaling the data, we see in Table 10 that the average MSEs are significantly lower for the suboptimal architectures. However, for the 10*(50,) architecture, we notice that the performance is not significantly better (only about a 3% improvement).

Layer sizes	Validation MSE	Train MSE
(50,50)	780638931	644161652
2*(100,)	662703464	376914873
3*(100,)	261528210	231483313
4*(100,)	319585942	189917573
10*(50,)	677650603	109547125
10*(100,)	697102371	70324031

Table 10. Validation and Training MSES on scaled data, neural network

Question 13 Using a grid search, we found different optimal architectures for each time period. These results are found in Table 11. These results make sense because there are the most tweets between 8am and 8pm, so having a deeper and wider architecture is better as it has a higher capacity to learn without overfitting. In contrast, the periods before and after the Super Bowl have fewer amount of tweets, so we would want a shallower or narrower architecture.

Time period	Architecture	Val MSE
Before 2/1, 8am	2*(200,)	4680478
Between 8am, 8pm	5*(250,)	37522544
After 2/1, 8pm	5*(150,)	900373

Table 11. Grid search results for different time periods, neural network

6. Using 6x window to predict

In this section, we wish to predict the number of tweets in a 6x window for each time period. To do so, there are several methods we could use:

1. Predict window-by-window, so we get 6 numbers for each 6x window, and add all of them together.
2. Combine 6 windows together by adding / maxing in the train set (so we still have 5 features), then predict after fitting to this new train set.
3. Combine 6 windows by concatenating features (so we now have 30 features), then predict after fitting to this new train set.

The first two methods are very naïve, as we lose a lot of information. Thus, we chose to use the 3rd method.

Question 14 We tried multiple models, but it turned out that the vanilla linear regression performed the best.

Test file name	True # of tweets	Predicted # of tweets
`sample0_period1`	568	568
`sample0_period2`	13377	13407.92
`sample0_period3`	410	436.2
`sample1_period1`	2280	2280
`sample1_period2`	5549	5570.54
`sample1_period3`	305	432.34
`sample2_period1`	953	953
`sample2_period2`	152	187.62
`sample2_period3`	399	473.28

Table 12. Predicting # of tweets in test set

We also trained a linear regression to predict how many tweets would appear in the next time window using only the previous hour. The results are shown in Tables 13 - 21. As we can see, the predictions are significantly worse than when we used 30 features to predict the total amount of tweets in the test set.

	1	2	3	4	5	6
True #	52	79	94	101	122	120
Linear Regression	486.95	462.29	504.15	532.9	519.66	446.83

Table 13. `sample0_period1`

	1	2	3	4	5	6
True #	3472	3834	2258	1455	1235	1123
Linear Regression	5402.06	5620.64	4021.69	3120.12	2993.65	2834.92

Table 14. `sample0_period2`

	1	2	3	4	5	6
True #	59	48	94	45	77	87
Linear Regression	514.26	344.25	355.09	302.99	314.56	290.17

Table 15. `sample0_period3`

	1	2	3	4	5	6
True #	203	180	202	294	555	846
Linear Regression	591.83	567.96	576.28	657.88	824.76	1038.6

Table 16. `sample1_period1`

	1	2	3	4	5	6
True #	960	995	870	960	861	903
Linear Regression	1734.13	1817.2	1674.23	1979.44	1640.55	1732.81

Table 17. `sample1_period2`

	1	2	3	4	5	6
True #	58	87	43	27	44	46
Linear Regression	338.21	37.53	-79.15	-108.7	363.08	343.25

Table 18. `sample1_period3`

	1	2	3	4	5	6
True #	401	141	102	144	104	61
Linear Regression	605.77	478.23	456.98	477.07	432.08	419.06

Table 19. `sample2_period1`

	1	2	3	4	5	6
True #	24	19	25	27	29	28
Linear Regression	317.29	312.71	323.97	339.72	327.36	370.48

Table 20. `sample2_period2`

	1	2	3	4	5	6
True #	81	90	40	58	87	43
Linear Regression	48.52	56.63	-16.71	338.21	37.53	-79.15

Table 21. sample2_period3

Part 2: Fan Base Prediction

In this section we examine another characteristic of Twitter data: the geographic origin of tweets. The Twitter data corresponds to Superbowl XLIX in which teams (the New England Patriots and the Seattle Seahawks) with fan bases in the states of Massachusetts and Washington played. We use the textual content of tweets posted by users in either Massachusetts or Washington to predict their location. We limit the scope of our analysis to considering only tweets that include # superbowl, however our analysis could easily be extended to cover the entire dataset.

Defining Locations

To determine whether location of the tweet is in Washington or Massachusetts, we first extract the location of every tweet using the following location field extraction code:

```
location = json_obj['tweet']['user']['location']

locations.append(location)
```

We determined that the **number of unique locations in the dataset is 179770** using the following code:

```
unique_locations = []

for location in locations:
if location in unique_locations: # Avoid duplicates
    continue

unique_locations.append(location)

print('number of unique locations is:
 ↪ {}'.format(len(unique_locations)))
```

We then determined every unique location that is in Massachusetts or Washington using the following techniques. Firstly, we defined that locations must contain one of the following sub-strings:

```
[' MA',' WA', 'Massachusetts', 'Washington', 'Boston', 'Seattle']
```

We looked for strings that have a space before 'MA' or 'WA' to avoid selecting locations such as 'IOWA.' We then removed all locations from Washington DC / D.C. The umber of unique locations in MA or WA is 5119. The number of unique locations in MA or WA after removing DC is 4650. Having defined all the locations in Massachusetts and Washington, we selected all of the tweets originate from these states and extracted their textual data using the following code:

```
# get tweets that are only in MA and WA (not DC)
location = json_obj['tweet']['user']['location']

if not any(MAWA in location for MAWA in all_MA_WA_no_DC):
    continue

if any (loc in location for loc in only_MA):
    # get textual data
    text = json_obj['tweet']['text']
    tweet_textual_data.append(text)

    # add location is in MA (0)
    tweet_location_labels.append(0)

    num_tweets_MA += 1

if any (loc in location for loc in only_WA):
    if any (loc in location for loc in DC): # Check if contains DC
        continue

    # get textual data
    text = json_obj['tweet']['text']
    tweet_textual_data.append(text)

    # add location is in WA (1)
    tweet_location_labels.append(1)

    num_tweets_WA += 1
```

The number of tweets from Massachusetts state is 19815 and the number of tweets from Washington state is 17243. Figure 21 is a plot of the number of tweets from Massachusetts vs Washington. We notice that the distribution is fairly equal between the two states.

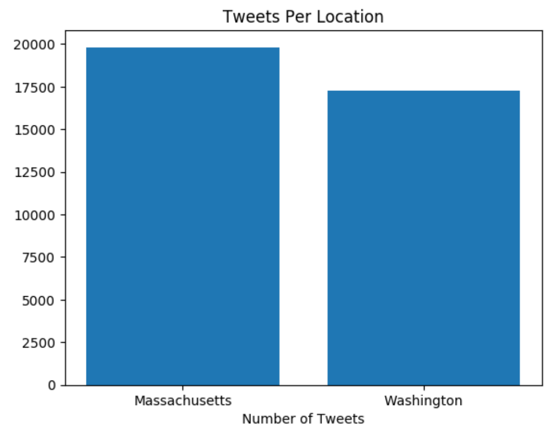

Fig. 21. Number of tweets from Massachusetts vs Washington

Training and Testing Datasets

We defined our training dataset by taking 90% of the randomly shuffled textual data and location labels and defined our testing dataset by taking 10% of the randomly shuffled textual data and location labels. We performed this using the following code:

```
from sklearn.model_selection import KFold
from sklearn.model_selection import train_test_split

X = np.asarray(tweet_textual_data)
y = tweet_location_labels

kf = KFold(n_splits=10, shuffle = True)

for trainset, testset in kf.split(X):
    X_train, X_test = X[trainset], X[testset]
    y_train, y_test = y[trainset], y[testset]
```

We also experimented with using 66%/33%, 75%/25%, and 80%/20% training / testing splits, but we found that the results were approximately the same (if not worse, in the case of using a 66%/33% training / testing split).

Feature Extraction

Classification of the extracted Twitter textual data cannot be easily performed without first pre-processing the data. The textual data is first passed through a lemmatizer. The process of lemmatization uses vocabulary and morphological analysis to deconstruct each word into their base constituent, called a lemma.

After each document is lemmatized, frequently occuring words are removed by a dictionary of words called "stopwords". The english stopwords collection consists of 318 words that are too common in the English language to be of any use for classification. This collection includes the words "no, to, or, not, very, rather" and many others. All 318 words are removed from the Tweets.

After each lemmatized document is filtered by stopwords, features can be extracted. The frequency of each word used in the document is analyzed using the "Term Frequency-Inverse Document Frequency (TF-IDF)" metric. This metric measures the number of times a word is used in a single document, normalized over a function of how many times the word appears in the entire corpus. The following specs were used during lemmatization:

- Use the "english" stopwords of the CountVectorizer
- Use min_df = 3 for the CountVectorizer
- Exclude terms that are numbers (e.g. "123", "-45", "6.7", "5e07" etc.)
- Perform lemmatization with nltk.wordnet.WordNetLemmatizer and pos_tag

The resulting dimensions of the TF-IDF matrices are shown in Table 22.

Subset	TF-IDF Matrix shape
train	(33353, 6676)
test	(3705, 6676)

Table 22. Dimensions of TF-IDF matrices

Dimensionality Reduction

With large data sets, the dimensionality of the representation vectors will be extremely high, but our learning algorithms often are not efficient and do not perform well in high dimensions. This is referred to as the curse of dimensionality. To mitigate this, we can take advantage of the fact that our TF-IDF matrix is relatively sparse, so we can take a significantly smaller subset of data points that give us the most "information" about the data set as a whole. We used Latent

Semantic Indexing (LSI) and Non-negative Matrix Factorization (NMF) on our textual data.

To compare how each of these dimensionality-reducing algorithms performed, we compare their objective values: $\|X - U_k \Sigma_k V_k^T\|_F^2$ for LSI and $\|X - WH\|_F^2$ for NMF. For the training data, we get:

$$\begin{aligned} \mathrm{LSI}_{\mathrm{train}} &= 25784.076295041883 \\ \mathrm{NMF}_{\mathrm{train}} &= 25678.749629641556 \end{aligned}$$

and for the testing data we get

$$\begin{aligned} \mathrm{LSI}_{\mathrm{test}} &= 2842.4320260810796 \\ \mathrm{NMF}_{\mathrm{test}} &= 2838.8071520811523 \end{aligned}$$

In both cases, we see that LSI performed better, as it is a closer approximation to X than NMF is. This may be the case because LSI does not have any non-negativity constraints, making it more flexible in the optimal $X_{reduced}$ found. In optimization terms, the LSI problem likely has a feasible set that contains a solution that is closer to the true X than the NMF problem, resulting in LSI being a better approximation. Furthermore, if there are a few ($\leq k$) terms that are significantly more important (i.e. have much higher singular values) than the others, then the SVD will give us a very accurate approximation to X because the information in X will be predominantly in the first k singular vectors and singular values. NMF will likely not be able to attain this level of accuracy because it is not directly computed using the SVD. **As a result, we used LSI for all of our classification algorithms.**

Classification Algorithms

In the following sections, we trained and tested different classifiers, namely **soft and hard margin support vector machines, unregularized and regularized logistic classifiers, and Naïve Bayes classifiers**, on our LSI dimension-reduced data set to predict the location of the author of a tweet. We treated this as a binary classification problem: users are either from Massachusetts or Washington. We then use various classification measures to determine how well each classifier performed.

Classification Measures To measure the performance of our classifiers, we look at the **confusion matrix**, **ROC curve**, **accuracy score**, **precision and recall scores**, and **F-1 score**.

1. SVM, Hard and Soft Margin Classifiers

A **Linear Support Vector Machine (SVM)** is a classifier that uses an optimal hyperplane to categorize data into separate classes.

In this report, we analyze the effects of a hard margin (when $\gamma \gg 1$) and a soft margin (when $\gamma \ll 1$). The performance scores of both the classifiers are plotted and compared in Tables 23 and 24. Fig. 22 and Fig. 23 show the Confusion matrices for hard margin linear SVM and soft margin linear SVM respectively. Fig. 24 and Fig. 25 depict the ROC curves for hard margin linear SVM and soft margin linear SVM respectively. **The area under curve for the hard margin SVM is 0.8731 whereas it is 0.8596 for the soft margin SVM classifier.**

Accuracy score:	0.7873
Precision score:	0.8982
Recall score:	0.6132
F-1 score:	0.7288

Table 23. Hard Margin Linear SVM

Accuracy score:	0.5430
Precision score:	0.6082
Recall score:	0.0853
F-1 score:	0.1497

Table 24. Soft Margin Linear SVM

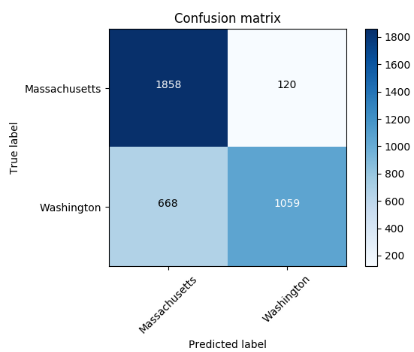

Fig. 22. Confusion matrix for linear SVM with hard margin ($\gamma = 1000$)

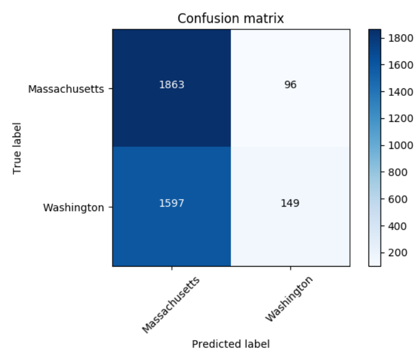

Fig. 23. Confusion matrix for linear SVM with soft margin ($\gamma = 0.0001$)

2. Logistic Regression without Regularization

We trained an **unregularized logistic classifier** on our training data set reduced by LSI. Fig. 26 and Fig. 27 show the confusion matrix and and ROC curve of our classifier. **The area under the curve is 0.8745. Unregularized logistic classification had the following scores:**

Accuracy score:	0.7892
Precision score:	0.8948
Recall score:	0.6207
F-1 score:	0.7330

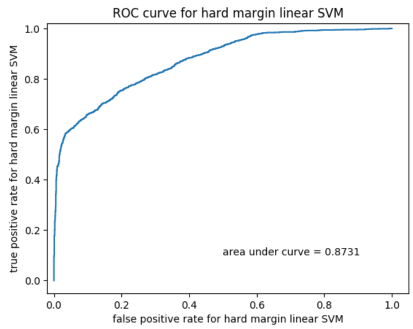

Fig. 24. ROC curve for linear SVM with hard margin ($\gamma = 1000$)

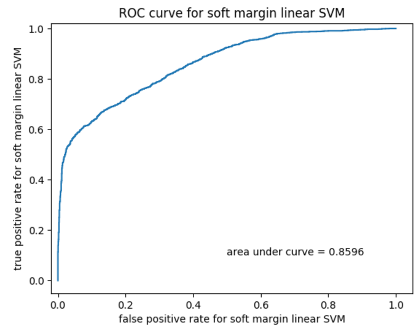

Fig. 25. ROC curve for linear SVM with soft margin ($\gamma = 0.0001$)

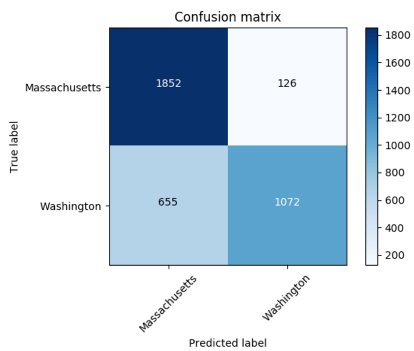

Fig. 26. Confusion matrix for unregularized Logistic Regression

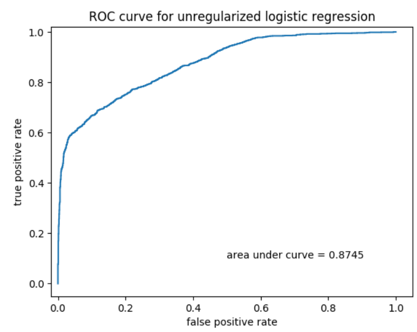

Fig. 27. ROC curve for unregularized Logistic Regression

3. Logistic Regression With L1 and L2 Regularization Classifier

We trained two **regularized logistic regressions** with **L1 and L2 regularization**. Using 5-fold cross validation on the dimension-reduced-by-SVD training data, we found the optimal regularization strength C in the range $\{10^k| - 3 \leq k \leq 3, k \in \mathbb{Z}\}$ for logistic regression with L1 and L2 regularization. We found that the best **regularization strength for logistic regression with L1 regularization is 0.1**. We found that the best **regularization strength for logistic regression with L2 regularization is 0.01**.

Fig. 28 and Fig. 29 show the confusion matrices of logistic regression with L1 and L2 regularization, respectively. Fig. 30 and Fig. 31 show the ROC curves of logistic regression with L1 and L2 regularization, respectively. **The area under the curve for logistic regression with L1 regularization is 0.8746. The**

area under the curve for logistic regression with L2 regularization is 0.8745. Logistic regression with L1 regularization had the following scores:

Accuracy score:	0.7889
Precision score:	0.8947
Recall score:	0.6202
F-1 score:	0.7326

Logistic regression with L2 regularization achieved:

Accuracy score:	0.7892
Precision score:	0.8948
Recall score:	0.6207
F-1 score:	0.7330

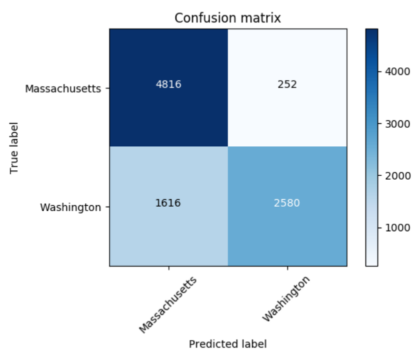

Fig. 28. Confusion matrix for logistic regression with L1 regularization

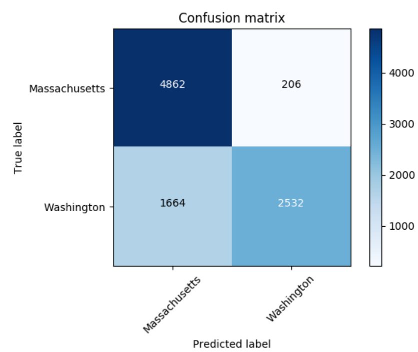

Fig. 29. Confusion matrix for logistic regression with L2 regularization

4. Naive Bayes Gaussian Classifier

We trained a Gaussian Naïve Bayes (NB) classifier on the train subset reduced using the LSI metric. Fig. 32 shows the confusion matrix after fitting. Fig. 33 shows the ROC of the classifier trained using the subset reduced by LSI. **The Gaussian NB classifier had an ROC auc of 0.7896. The scores which the Gaussian NB classifier achieved when fitted with the LSI-reduced subset are the following:**

Accuracy score:	0.7263
Precision score:	0.7437
Recall score:	0.6300
F-1 score:	0.6821

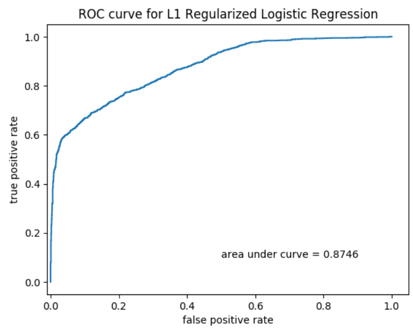

Fig. 30. ROC curve for L1 Regularized Logistic Regression

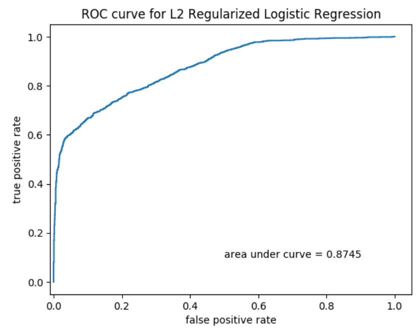

Fig. 31. ROC curve for L2 Regularized Logistic Regression

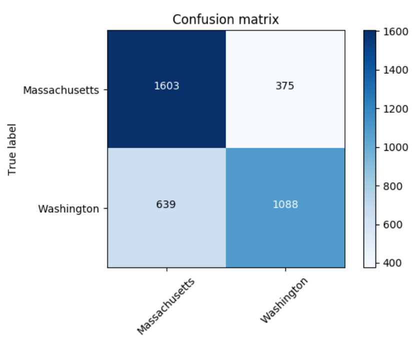

Fig. 32. Confusion matrix for Gaussian NB from LSI reduction.

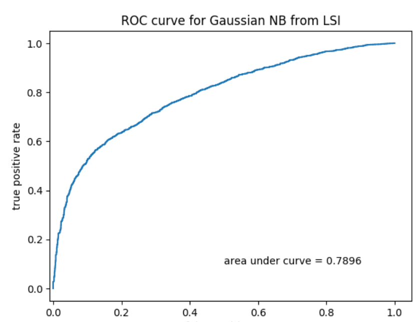

Fig. 33. ROC curve for Gaussian NB from LSI reduction.

Analysis of Fan Base Prediction Classification Methods

Of the three main classes of classification algorithms that we implemented (SVM, logistic regression, and Gaussian NB), logistic regression demonstrated marginally better performance in general. The performance of the Gaussian NB is not as good as Logistic Regression or SVM, however. One reason why this might be the case is that the Gaussian NB assumes the data distribution to be fairly separable and to have a Gaussian probability distribution. Although these assumptions may be true in many real-world cases, we have not proven that they hold true for our corpus of textual data.

Part 3: Predicting ball possession during gameday using sentiment analysis

We analyzed the polarity and subjectivity of tweets before, after, and during game day using TextBlob. Figures 34 and 35 show the sentiment from the #gopatriots and #gohawks hashtags during game day, respectively. It is evident that these sentiments may be related to specific events that are occuring throughout the game. For example, if the Patriots are in possession of the ball, the sentiment of patriots fans may raise in the #patriots and #gopatriots hashtags.

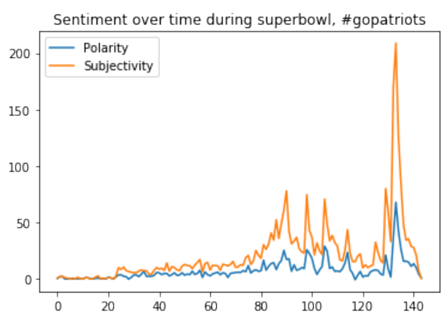

Fig. 34. sentiment for #gopatriots during gameday

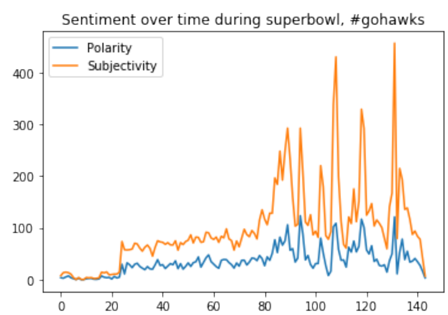

Fig. 35. sentiment for #gohawks during gameday

Therefore, we propose using the sentiment of tweets on game day to predict who is in possession of the ball. The during-game day tweets are separated into five-minute windows, giving 144 measurements over the day. There is no present dataset available which translates the events which occurred during Super Bowl ILIX to real-time, so we recorded these events manually. Table 25 shows the data which we recorded. A window of time is recorded for each time period that a particular team had possession of the ball. Any points the team scored during this time period are also recorded. The play-by-play times are first converted to PDT, then to PST, and finally to the particular bin the time window resides in.

These bins use the same scale as the sentiment analyses previous performed. We can now compare the change of sentiment with who had possession of the ball.

Q	play start	play end	vid start	vid end	event	**possession**	RT start	RT end	**bin**
1	15:00		00:00			Patriots	6:30 PM		90
1	15:00	11:44	00:00	05:31		Patriots	6:30 PM	6:35 PM	91
1	11:44	09:30	05:31	09:37		Seahawks	6:35 PM	6:39 PM	92
1	09:30	01:46	09:37	18:10		Patriots	6:39 PM	6:48 PM	94
2	01:46	14:01	18:10	23:02		Seahawks	6:48 PM	6:53 PM	95
2	14:01	09:47	23:02	30:10	TD+K	Patriots	6:53 PM	7:00 PM	96
2	09:47	08:10	30:10	32:33		Seahawks	7:00 PM	7:02 PM	97
2	08:10	07:13	32:33	35:31		Patriots	7:02 PM	7:05 PM	97
2	07:13	02:17	35:31	41:17	TD+K	Seahawks	7:05 PM	7:11 PM	98
2	02:17	00:31	41:17	50:48	TD+K	Patriots	7:11 PM	7:20 PM	100
2	00:31	00:02	50:48	59:27	TD+K	Seahawks	7:20 PM	7:29 PM	102
						Halftime			
3	15:00	11:09	1:02:45	1:07:26	exta K	Seahawks	8:05 PM	8:10 PM	110
3	11:09	08:09	1:07:26	1:12:16		Patriots	8:10 PM	8:14 PM	111
3	08:09	04:54	1:12:16	1:18:03	TD+K	Seahawks	8:14 PM	8:20 PM	112
3	04:54	03:19	1:18:03	1:23:49		Patriots	8:20 PM	8:26 PM	113
3	03:19	00:57	1:23:49	1:27:26		Seahawks	8:26 PM	8:30 PM	114
4	00:57	14:21	1:27:26	1:30:25		Patriots	8:30 PM	8:33 PM	115
4	14:21	12:14	1:30:25	1:33:22		Seahawks	8:33 PM	8:35 PM	115
4	12:14	07:55	1:33:22	1:40:29	TD+K	Patriots	8:35 PM	8:43 PM	117
4	07:55	07:00	1:40:29	1:44:19		Seahawks	8:43 PM	8:46 PM	117
4	07:00	02:02	1:44:19	1:53:56	TD+K	Patriots	8:46 PM	8:56 PM	119
4	02:02	00:20	1:53:56	2:04:22		Seahawks	8:56 PM	9:06 PM	121
4	00:20	00:18	2:04:22	2:08:23		Patriots	9:06 PM	9:10 PM	122

Table 25. Events during game recorded in real-time

Linear SGD Grid Search

A grid search is performed using Linear SGD as the classification model. This model is capable of applying many different linear classifiers to the problem including Linear SVM and Regression, depending on the loss. The grid search was used to step through these various classifiers and regressors, and to optimize the parameters of the best one. Table 26 lists the parameters which were used in this grid search.

Procedure	**Options**
loss	hinge, log (categorical cross-entropy), modified huber, squared hinge, and perceptron
penalty	l1, l2, and elasticnet
alpha	0.0001, 0.001, and 0.01
l1 ratio	0.15, 0.05, 0.25
learning rate	optimal
tol	1e-3, 1e-4, and 1e-5
max iters	3000, 5000, and 10000

Table 26. Parameters for the Linear SGD classifier

The grid search was performed for each hashtag. Table 27 shows the best classification accuracy of the grid search per hashtag. It is clear that none of the models tried were able to generalize well on the data. Since there are only two possibilities for who has possession (Patriots or Seahawks), a random guess would incur 50% accuracy. A smarter algorithm may simply choose to pick one option or the other, which would result in a 56.25% accuracy. The best accuracy out of the hashtags is **75.0%**, which is marginally better than a smart guess.

Hashtag	Best cross-validation score
gohawks	65.62%
gopatriots	62.50%
nfl	56.25%
patriots	59.38%
sb49	75.0%
superbowl	43.75%

Table 27. Best SGD classifier per hashtag

Table 28 shows the particular grid search for hashtag #sb49. Although the test accuracy score is high, the train accuracy score is low and both do not change as the parameters of the classifier are modified. It is evident that although the model is able to make some conclusion about which team had possession throughout the game, it is likely in part due to chance and not due to generalization of the training data.

	mean_test_score	mean_train_score	alpha	l1_ratio	lr	loss	max_iter	penalty	tol
422	0.75	0.454	0.001	0.15	opt	hinge	5000	elast	0.00001
693	0.75	0.454	0.001	0.25	opt	hinge	10000	l2	0.00100
803	0.75	0.454	0.001	0.25	opt	percept	10000	l2	0.00001
802	0.75	0.454	0.001	0.25	opt	percept	10000	l2	0.00010
801	0.75	0.454	0.001	0.25	opt	percept	10000	l2	0.00100
794	0.75	0.454	0.001	0.25	opt	percept	5000	l2	0.00001
793	0.75	0.454	0.001	0.25	opt	percept	5000	l2	0.00010
792	0.75	0.454	0.001	0.25	opt	percept	5000	l2	0.00100
785	0.75	0.454	0.001	0.25	opt	percept	3000	l2	0.00001
784	0.75	0.454	0.001	0.25	opt	percept	3000	l2	0.00010

Table 28. Tuned parameters from grid search, #sb49

Recurrent Neural Network

The SGD Classifier is capable of taking in the current sentiment and classifying possession. However, since this is time series data, a classifier may be able to use the sentiment of the previous bins to improve classification. We implemented a Long Short-term Memory (LSTM) Neural Network to take advantage of this time component. Figure 36 shows the structure of the LSTM.

```
_________________________________________________________________
Layer (type)                 Output Shape              Param #
=================================================================
lstm (LSTM)                  (None, 20)                2080
_________________________________________________________________
dense (Dense)                (None, 50)                1050
_________________________________________________________________
dropout (Dropout)            (None, 50)                0
_________________________________________________________________
dense_1 (Dense)              (None, 2)                 102
=================================================================
Total params: 3,232
Trainable params: 3,232
Non-trainable params: 0
```

Fig. 36. LSTM structure

Table 29 shows the accuracy of the LSTM with different window sizes. The larger the window, the farther back in time the LSTM will look in order to make a classification. The LSTM was trained for 500 epochs using each hashtag sentiment window as a separate training sample. The Adam optimizer with learning rate 1e-5 was used for an optimizer, and categorical crossentropy was used for loss. The LSTM accuracy improves as the window is increased to 4.

After this, the accuracy degrades, suggesting that looking too far back confuses the network.

Window size	LSTM classfication acc
1	54.69%
2	55.38%
3	57.70%
4	58.60%
5	56.00%

Table 29. LSTM performance per window

Future work

Overall, the performance of the LSTM and SGD Classifier are very low. This could be due to multiple factors.

The events of the game occurred one after another in very quick succession. An algorithm may see these as random events if not given enough information as context. The information could therefore be sub-sampled further to measure the sentiment every minute during the game.

The times we measured for the game were approximated, and may not be the actual times when events occurred. Increasing the accuracy of these approximations would enable both classifiers to make the most use out of the sentiment data.

Finally, the limited amount of information contained in the span of one game may not be enough for the LSTM neural network to make a prediction. Multiple games could be measured in real time, for different games of the NFL throughout the year to provide a better foundation for the LSTM to make a prediction.

Bibliography

[Roy19] Roychowdhury, Vwani: Project 5: Application - Twitter data, University of California, Los Angeles, 2019

[XZJ+16] Xie, Wei ; Zhu, Feida ; Jiang, Jing ; Lim, Ee-Peng ; Wang, Ke: Topicsketch: Real-time bursty topic detection from twitter. In: *IEEE Transactions on Knowledge and Data Engineering* 28 (2016), Nr. 8, S. 2216–2229