EE 219 Project 5: Popularity Prediction on Twitter

Winter 2017

March 22th, 2017

Arunav Singh (304760844)
John Moon (204774912)
Jiawen Yu (204753330)

Part 1: Calculate the average number of tweets per hour, average number of followers per tweet, and average number of retweets per tweet. Plot number of tweets in hour over time for #superbowl and #nfl.

The output of the code for each hashtag is detailed in *Figures 1-8*. Excel files of the original histogram calculations have been included.

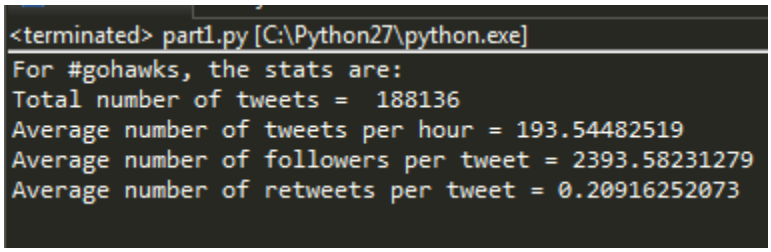

Figure 1: Statistics for #gohawks

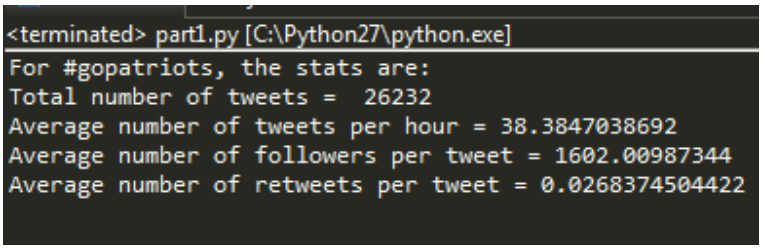

Figure 2: Statistics for #gopatriots

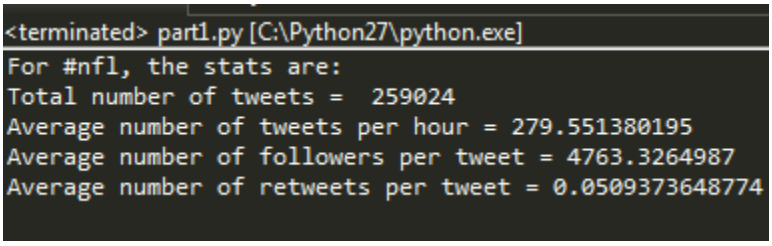

Figure 3: Statistics for #nfl

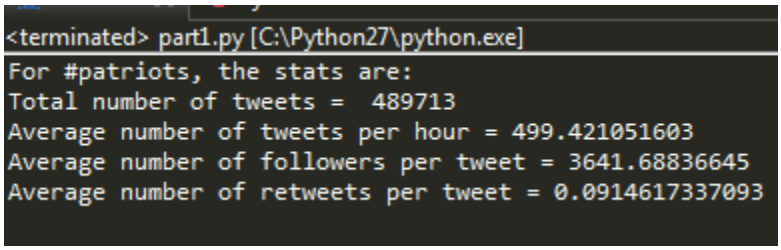

Figure 4: Statistics for #patriots

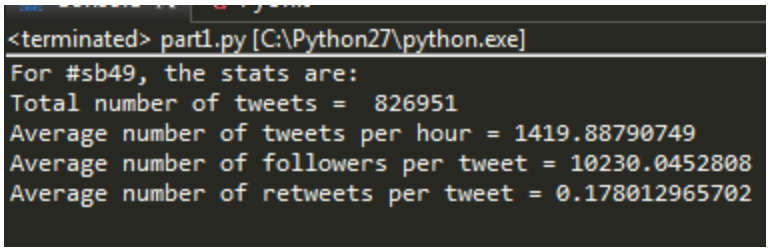

Figure 5: Statistics for #sb49

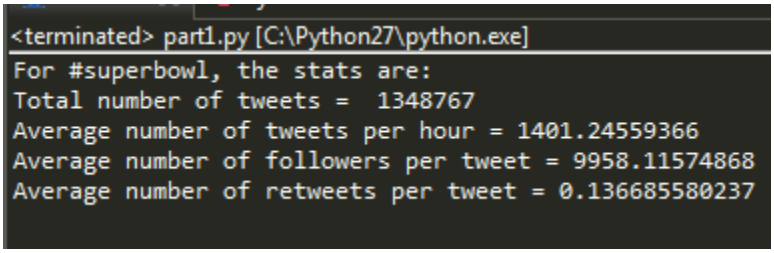

Figure 6: Statistics for #superbowl

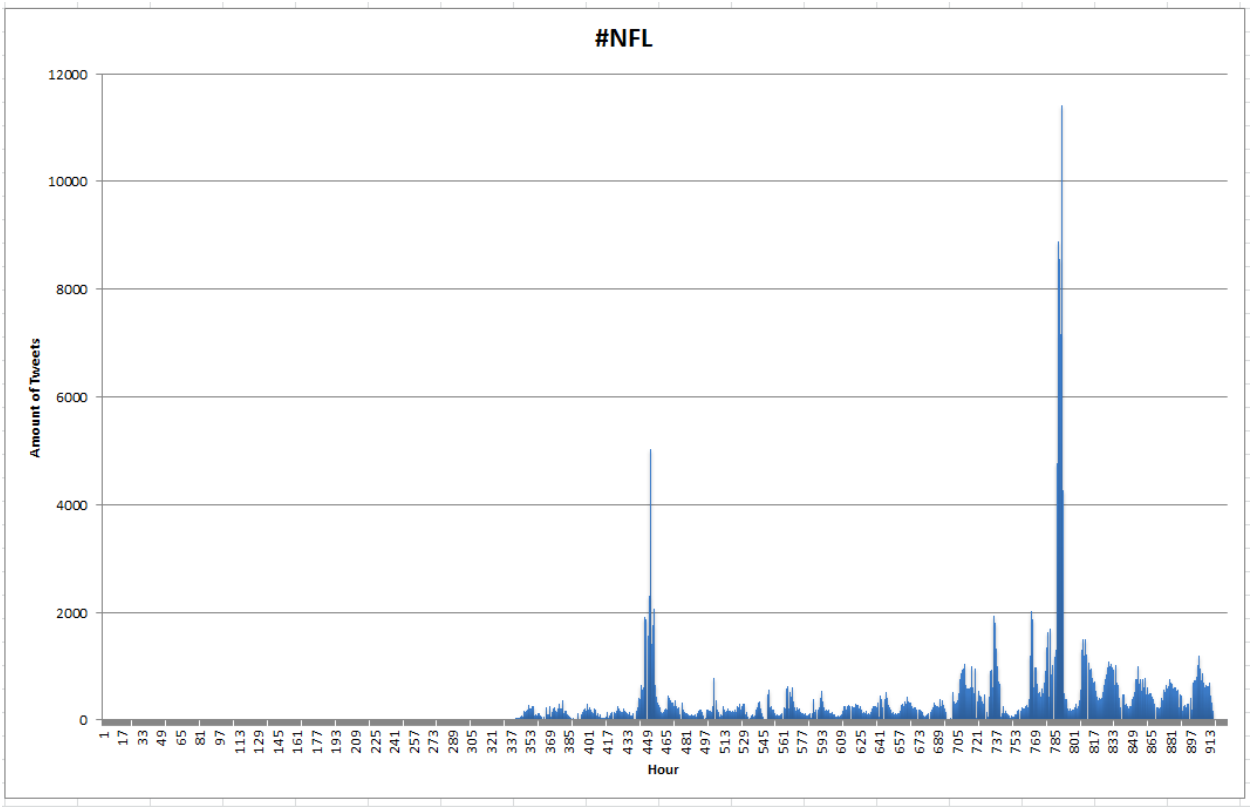

Figure 7: Histogram for #nfl

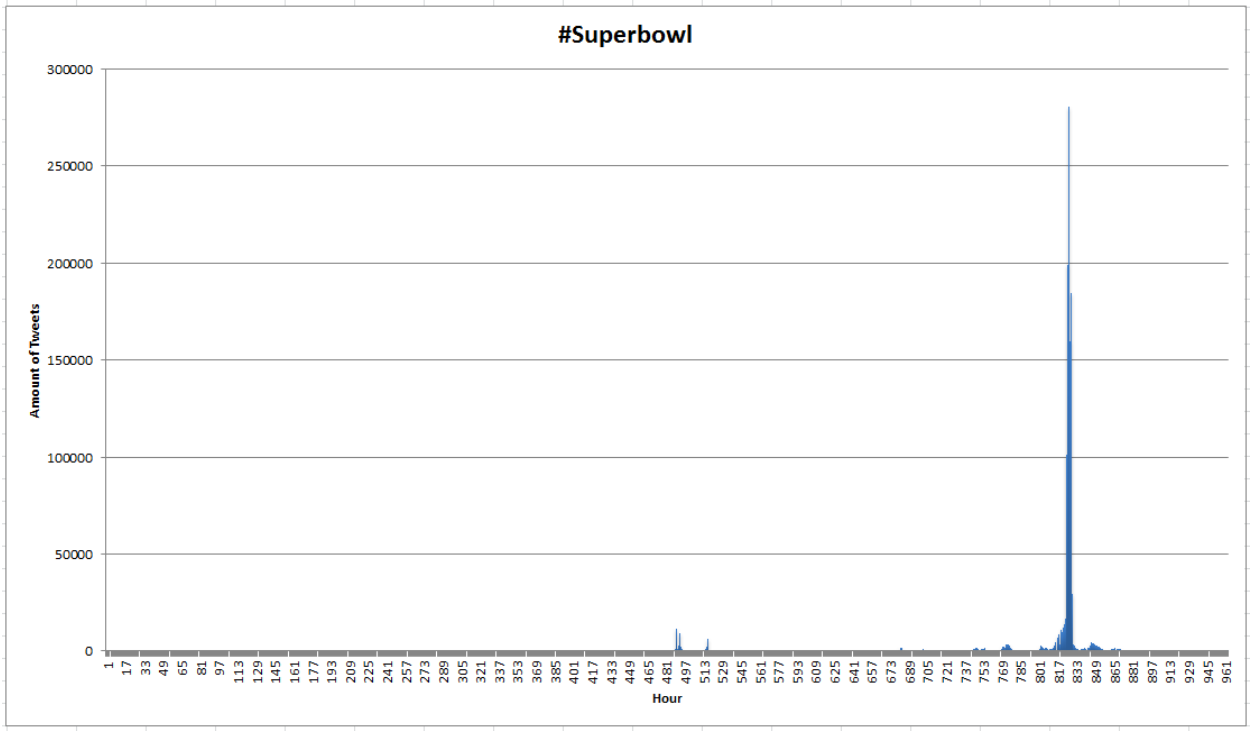

Figure 8: Histogram for #superbowl

Part 2: Fit a linear regression model using 5 features to predict tweets in the next hour. Explain your model's training accuracy and the significance of each feature using the t-test and p-value results.

First, each hashtag was fitted with a linear regression model and the RMSE value for each hashtag is calculated and the results are detailed in *Figures 9-14.* The time of day feature is assigned a value from 1-24, where 1 is 12 AM and 24 is 11 PM. The earliest tweet is found in the 'gohawks' hashtag thus is used as the first hour mark.

Then, a linear regression model that uses all tweets across all hashtags was fitted and the RMSE of the total was calculated. That RMSE is detailed in *Figure 15.*

```
<terminated> part2.py [C:\Python27\python.exe]
tweets_#gohawks.txt
The RMSE of #gohawks is : 617.656679619
```

Figure 9: RMSE for #gohawks

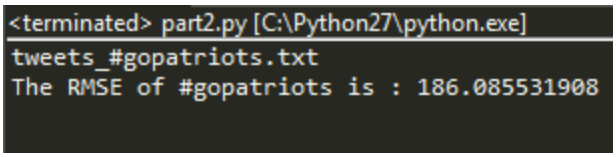

Figure 10: RMSE for #gopatriots

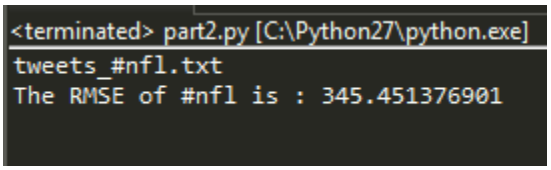

Figure 11: RMSE for #nfl

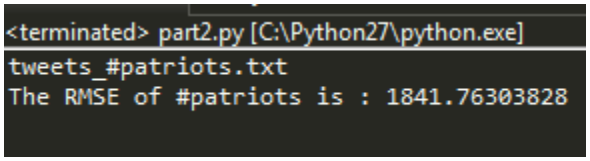

Figure 12: RMSE for #patriots

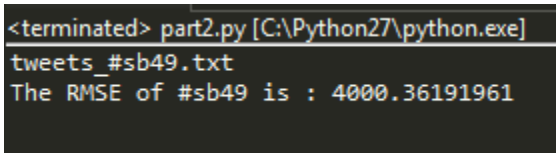

Figure 13: RMSE for #sb49

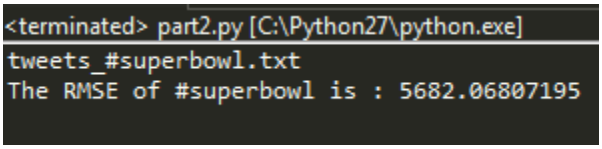

Figure 14: RMSE for #superbowl

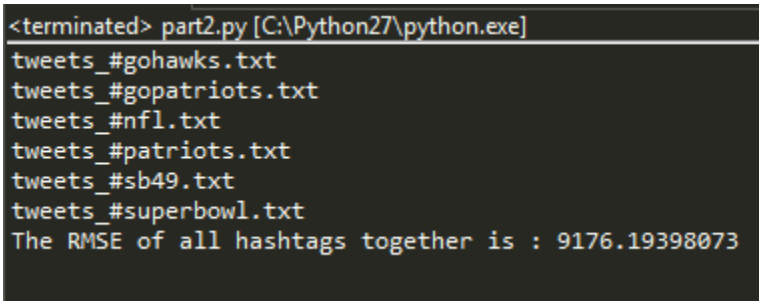

Figure 15: RMSE for all hashtags together

We can see in *Figures 9-15* that as the dataset gets larger, our RMSE increases. This is an intuitive notion as we are using linear regression. Given that the amounts of tweets in each file range from thousands to millions, our RMSE values

suggest that our regression is performing well. Also, the fact that most of the tweets in the next hour are 0 (due to many periods of inactivity) followed by a burst, our model is trying to fit the smaller values leading to big errors when the tweet numbers in the next hour get large.

Next, we analyze the significance of each feature using t-test and p-value results from the linear regression model. The results for each hashtag are detailed below in *Figures 16-22*.

#nfl	*t Stat*	*P-value*
numTweets	9.027651027	1.00642E-18
numRetweets	-14.25245363	8.57632E-42
sumFollowers	7.38334584	3.45444E-13
maxFollowers	-5.433345973	7.0835E-08
timeOfDay	-1.509276834	0.131571392

Figure 16: Statistics for #nfl

Most significant features: numTweets, numRetweets, sumFollowers

#Superbowl	*t Stat*	*P-value*
numTweets	6.62314953	5.85868E-11
numRetweets	-28.15464558	1.6815E-127
sumFollowers	1.904019042	0.057207712
maxFollowers	3.784085603	0.000163873
timeOfDay	0.113775564	0.909439598

Figure 17: Statistics for #Superbowl

Most significant features: numTweets, numRetweets, maxFollowers

#Patriots	*t Stat*	*P-value*
numTweets	34.91086836	7.9329E-174
numRetweets	-2.318546537	0.020625905
sumFollowers	-7.278488197	6.94589E-13
maxFollowers	4.723561013	2.65777E-06
timeOfDay	-0.587512168	0.556995953

Figure 18: Statistics for #patriots

Most significant features: numTweets, sumFollowers, maxFollowers

#SB49	t Stat	P-value
numTweets	33.17946737	9.0751E-136
numRetweets	-3.566868565	0.000391322
sumFollowers	-3.351312678	0.000856986
maxFollowers	5.589135341	3.52525E-08
timeOfDay	-1.047878955	0.295133872

Figure 19: Statistics for #sb49

Most significant features: numTweets, numRetweets/sumFollowers, maxFollowers

#GoPatriots	t Stat	P-value
numTweets	12.1113149	1.06554E-30
numRetweets	-17.6063546	2.07452E-57
sumFollowers	-0.06435373	0.948707571
maxFollowers	0.175682273	0.860596097
timeOfDay	0.010696548	0.991468704

Figure 20: Statistics for #GoPatriots

Most significant features: numTweets, numRetweets

#gohawks	t Stat	P-value
numTweets	8.269298258	4.42877E-16
numRetweets	0.127504055	0.898568001
sumFollowers	0.896022625	0.370463904
maxFollowers	-2.923422952	0.003542853
timeOfDay	-0.504345069	0.614133926

Figure 21: Statistics for #gohawks

Most significant features: numTweets, maxFollowers

AllHashtags	t Stat	P-value
numTweets	23.69775657	2.17889E-98
numRetweets	-13.08282266	3.78552E-36
sumFollowers	-1.652421655	0.098770916
maxFollowers	7.389976643	3.15623E-13
timeOfDay	-0.956941973	0.338834001

Figure 22: Statistics for All Hashtags

Most significant features: numTweets, numRetweets, maxFollowers

Looking at the p-value and t-stat for each of the hashtags, we can identify which features were most significant. Features that may be significant for one hashtag

may not be as relatively significant in others as seen in the figures above; different hashtags exhibit different behaviors.

Intuitively, we know that the number of tweets (numTweets) is most likely to have a low p-value and high t-stat across all hashtags seeing as our target is derived from this feature. The number of retweets behaves similarly except in the one case for #patriots as seen above. These are reasonable; it is not hard to imagine how the number of tweets in the current hour, combined with the number of retweets relates to a topic's popularity and as a result number of tweets in the next hour. Other features fluctuate in importance depending on the hashtag, but the general trend of these two features being important is apparent.

Part 3: Use new features from the metadata and fit a linear regression model on the data; report fitting accuracy and significance of variables. For the top 3 features in your measurements, draw a scatterplot of number of tweets in the next hour versus feature value and analyze it.

Fitting accuracy (RMSE) for all the models (individual hashtags then all hashtags together) was calculated. The results are detailed in *Figures 23-29*.

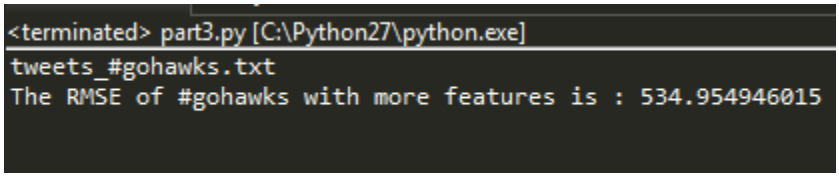

Figure 23: RMSE for #gohawks with 15 features

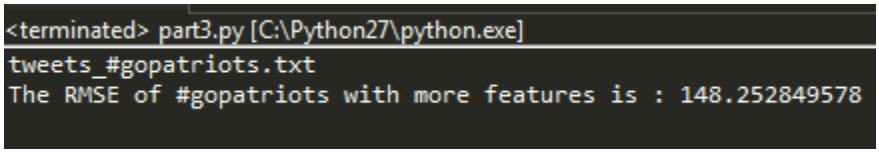

Figure 24: RMSE for #gopatriots with 15 features

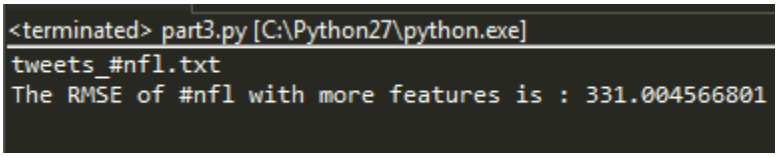

Figure 25: RMSE for #nfl with 15 features

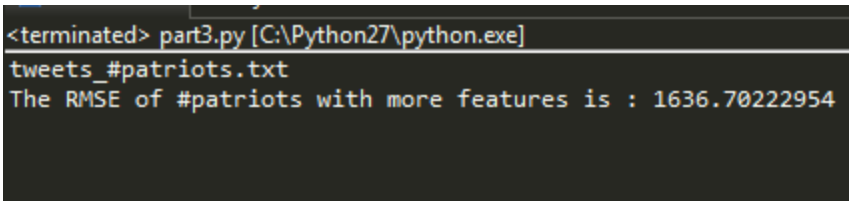

Figure 26: RMSE for #patriots with 15 features

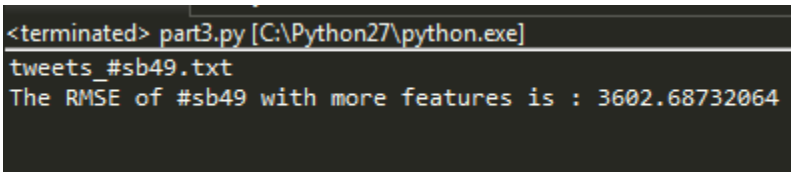

Figure 27: RMSE for #sb49 with 15 features

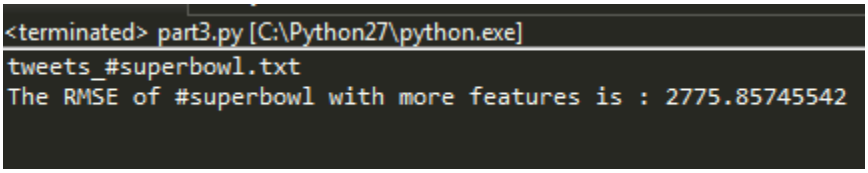

Figure 28: RMSE for #superbowl with 15 features

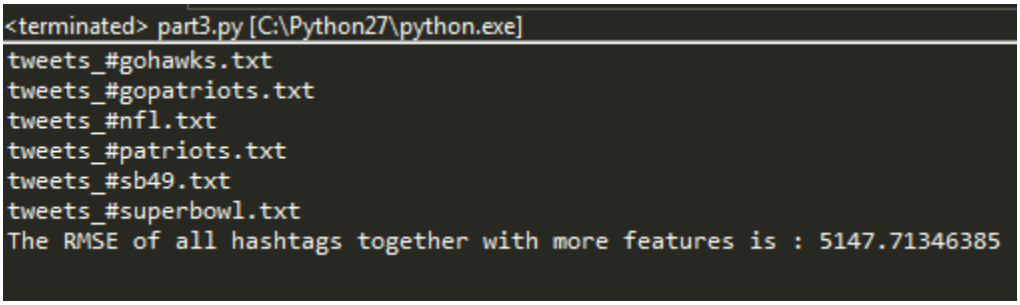

Figure 29: RMSE for all hashtags together with 15 features

Making comparisons from Part 2, all RMSE values have decreased for every hashtag, leading to the conclusion that the inclusion of these features have helped develop a more powerful regression algorithm.

Here is a list of all the new features included in these new models:

1. Sum of the favorites of each tweet
2. Sum/average of the ranking scores of each tweet
3. Sum/average of the citations of each tweet
4. Sum/average of the impressions per tweet
5. Sum/average of the momentums of each tweet
6. Number of unique users posting tweets within the hour

This gives us 10 new features, combined with the 5 existing features giving us 15 features in total. Next, a description of each feature and why we used said feature will be detailed.

Favorites: The bigger the number of favorites, the more popular a tweet becomes; if there is a trend of many favorites within an hour, that may tell us something about a potential spike in popularity for a given hashtag.

Ranking score: The ranking score is an evaluation of the rank of a tweet vs all tweets at that time. A higher ranking score means that more people see that tweet compared to other tweets. This is clearly correlated with popularity trends; the higher the ranking score, the more people will see the tweet and be compelled to tweet about it themselves. The average and sum are taken as two measures of density and variance.

Citations: The bigger the number of citations, the more popular a tweet becomes; if there is a trend of many citations within an hour that may tell us something about a potential spike in popularity for a given hashtag. This is very similar to the concept behind favorites and retweets.

Impressions: The bigger the number of impressions, the more popular a tweet becomes; if there is a trend of many impressions within an hour that may tell us something about a potential spike in popularity for a given hashtag. Again, this is very similar to the concept behind favorites and retweets.

Momentum: This measures the speed of a tweet being promoted; higher momentum means it will be promoted to users faster. Like the definition of momentum in physics implies, the higher the momentum, the more ramping speed a hashtag or topic is developing, and thus is useful in predicting popularity trends.

Number of unique users: The more unique users among the tweets, the more exposure a topic or hashtag gets, and thus correlates with popularity trends. Fairly intuitive; the more unique users we have in an hour, the wider of an audience is shown the topic or hashtag.

In order to look at the big picture and treat the entire Super Bowl as a topic to predict popularity trends of, we consider the significance of variables of all the relevant hashtags when analyzing the significance of variables. Their p-values and t-stat values are described in *Figure 30*.

AllHashtagsNew	*t Stat*	*P-value*
Intercept	0.131695654	0.895252504
numTweets	-1.978450545	0.048162221
numRetweets	-12.31888655	1.72642E-32
sumFollowers	23.93471546	9.7561E-100
maxFollowers	-3.15590825	0.001649626
timeOfDay	-0.275822615	0.782743428
sumFavorites	6.029088321	2.34591E-09
sumRankScore	-3.905918795	0.000100414
avgRankScore	0.134073294	0.893372602
sumCitations	11.61376998	2.81385E-29
avgCitatitions	-2.210273092	0.027320466
sumImpressions	-16.85433701	4.65442E-56
avgImpressions	-0.281905043	0.778076829
sumMomentum	-11.71410958	1.00153E-29
avgMomentum	2.504721485	0.01241889
uniqueUsers	14.453889	5.26092E-43

Figure 30: Significance of variables for new model

Most significant features: numRetweets, sumFollowers, sumFavorites, sumCitations, sumImpressions, sumMomentum, uniqueUsers

As we can see, all of our new features have relatively low p-values and high t-stats. While the averages are not as significant as the sums, they are still relatively low in magnitude in p-values and thus we kept them in out model moving forward.

Our top 3 features in terms of the p-value metric are sumFollowers, sumImpressions, and uniqueUsers. Those scatterplots of these features versus number of tweets next hour are detailed in *Figures 31-34.*

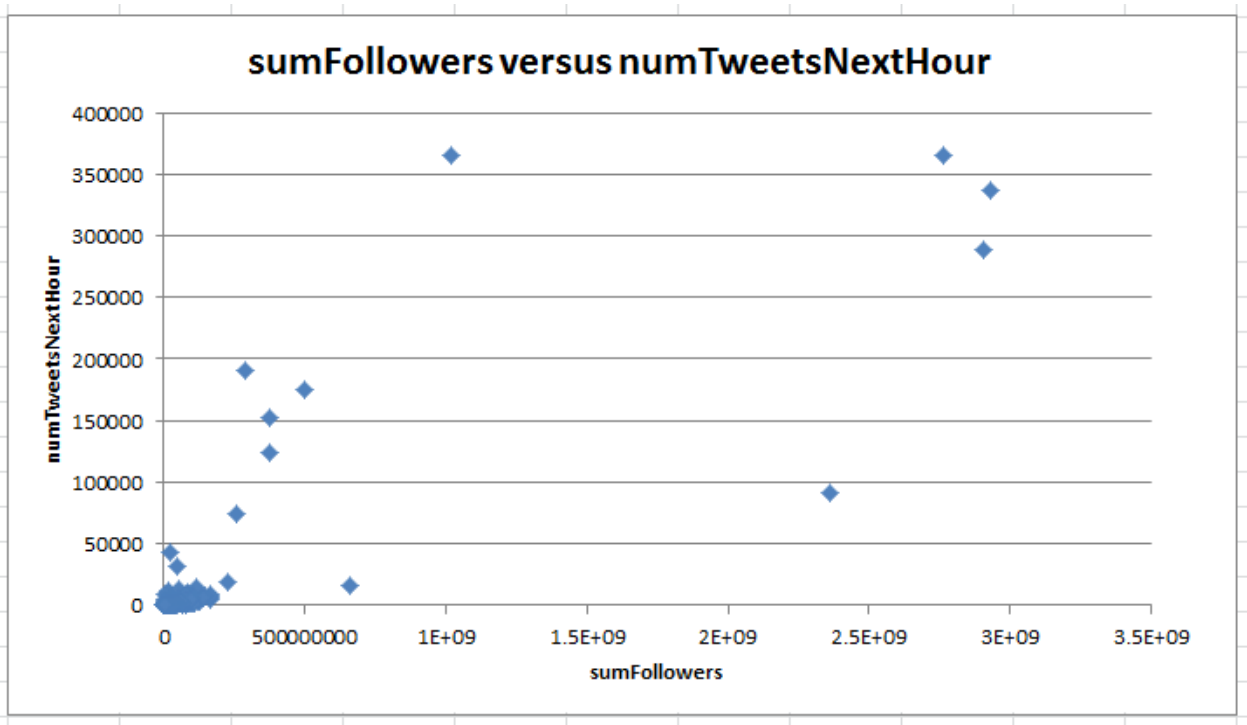

Figure 31: Scatterplot of sumFollowers vs numTweetsNextHour

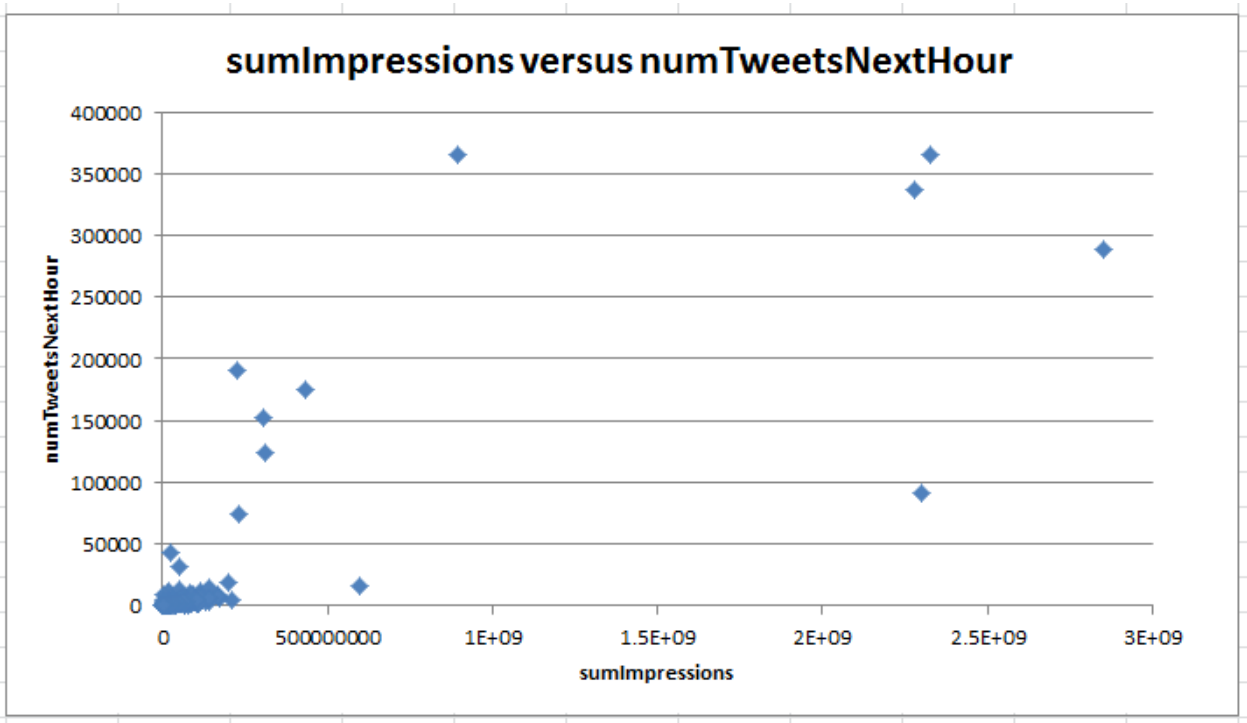

Figure 32: Scatterplot of sumImpressions vs numTweetsNextHour

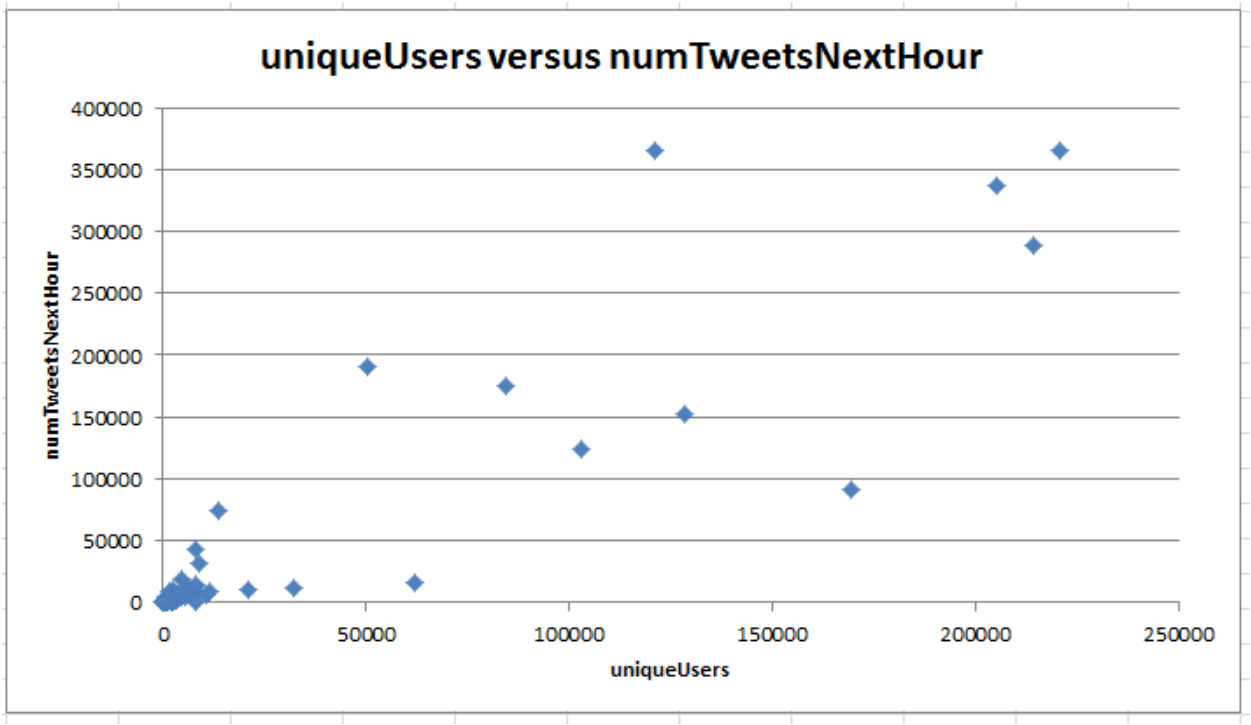

Figure 33: Scatterplot of uniqueUsers vs numTweetsNextHour

As expected, with a smaller number of followers/impressions/unique users, we expect a small number of tweets next hour. Given that most hours don't have many tweets this is to be expected. We do see when the variables spike up, there is a general trend of the number of tweets next hour also spiking up. Thus, we can say that these variables are revealing some patterns and trends in popularity for the topic of Super Bowl discussion.

Part 4: Report average prediction error for each test, then the average over all tests (using 10 fold cross validation). Train 3 regression models for 3 different time periods and report the average prediction error for each test, then the average over all tests for each period (using 10 fold cross validation).

Moving forward, all models will use the 15 features discussed in Part 3. 10 fold cross validation is performed for all the models (individual hashtags then all hashtags together) and the results are detailed in *Figures 34-40*.

```
<terminated> part4a.py [C:\Python27\python.exe]
For #gohawks:
The average prediction error for each test is:
20.8214815088
27.3410543507
69.7371404779
98.2517498433
102.677838498
551.860678686
160.170857643
134.636622426
1189.6313035
109.737789753
The average prediction error over all tests is : 246.486651669
```

Figure 34: CV errors for #gohawks

```
<terminated> part4a.py [C:\Python27\python.exe]
For #gopatriots:
The average prediction error for each test is:
8.85274490935
3.24527533623
15.1215394779
137.796667332
20.5299027725
27.6398726476
21.224940375
44.2805665485
308.15677052
2.2513179976
The average prediction error over all tests is : 58.9099597917
```

Figure 35: CV errors for #gopatriots

```
<terminated> part4a.py [C:\Python27\python.exe]
For #nfl:
The average prediction error for each test is:
13.0616920647
13.9186066523
41.8251092809
45.7095901202
208.005582179
96.2505500078
84.6198255036
188.497553749
444.739302696
139.402795035
The average prediction error over all tests is : 127.603060729
```

Figure 36: CV errors for #nfl

```
<terminated> part4a.py [C:\Python27\python.exe]
For #patriots:
The average prediction error for each test is:
45.823316049
43.9200918255
85.7784226845
194.024443331
206.107997538
1032.58284124
559.797107141
425.590025431
6224.60079996
335.807079259
The average prediction error over all tests is : 915.403212446
```

Figure 37: CV errors for #patriots

```
<terminated> part4a.py [C:\Python27\python.exe]
For #sb49:
The average prediction error for each test is:
368.681439758
215.254767756
258.812843649
218.328327427
270.879282444
304.44922056
2852.16974156
5028.81123818
813.108814
948.292188488
The average prediction error over all tests is : 1127.87878638
```

Figure 38: CV errors for #sb49

```
<terminated> part4a.py [C:\Python27\python.exe]
For #superbowl:
The average prediction error for each test is:
123.456538098
133.189799159
123.163670544
479.460037384
256.468258137
1483.51896226
496.03033832
1228.40372066
8533.1572082
1263.48814409
The average prediction error over all tests is : 1412.03366768
```

Figure 39: CV errors for #superbowl

```
<terminated> part4a.py [C:\Python27\python.exe]
For all hashtags together:
The average prediction error for each test is:
116.005021153
149.84114435
483.482743096
796.070589508
316.80842368
3115.26401995
1102.85476604
2416.92869272
29391.9056853
3359.03185992
The average prediction error over all tests is : 4124.81929458
```

Figure 40: CV errors for all hashtags together

Again, we can see that the bigger the dataset is, the higher the average prediction error is, as shown by the testing when using the tweets over all hashtags. This trend was first noticed and described in Part 2.

Now, the datasets are split into 3 periods as detailed in the project. The results are detailed below in *Figures 41-47.*

```
<terminated> part4b.py [C:\Python27\python.exe]
For #gohawks:
Period 1 :
The average prediction error for each test is:
38.8412538051
33.734083579
40.860672276
67.9734105971
82.9517036998
480.223251107
364.611186241
162.778236426
155.833037512
276.541426932
The average prediction error over all tests is : 170.434826217


Period 2 :
The average prediction error for each test is:
3356.50566986
4758.93061127
1348.12647399
4631.12639583
20229.1038518
9088.1121458
6034.92400625
4992.7686735
3677.80298379
208257.155488
The average prediction error over all tests is : 26637.45563


Period 3 :
The average prediction error for each test is:
634.166138014
1577.31789746
18.9057698679
4.47032687861
15.2334800208
5.89439388868
14.8870291441
4.47942520262
16.8812245518
4.0926491849
The average prediction error over all tests is : 229.632833421
```

Figure 41: CV errors for the 3 periods for #gohawks

```
<terminated> part4b.py [C:\Python27\python.exe]
For #gopatriots:
Period 1 :
The average prediction error for each test is:
7.43035349556
4.28746685994
7.43533224762
55.4765438397
46.3884739034
23.3749114879
28.3914166237
17.7073008616
9.5542863449
24.8154931581
The average prediction error over all tests is : 22.4861578822


Period 2 :
The average prediction error for each test is:
3160.3173187
1890.6608973
1550.1386261
19523.6431054
26876.8539198
7661.68079498
5516.07182141
5386.4107588
16064.7433061
731.412795895
The average prediction error over all tests is : 8836.19333445


Period 3 :
The average prediction error for each test is:
43.0828500208
8.19952183094
0.639474824895
0.707244599642
2.30964068271
0.617800354612
0.383063080101
0.884858893345
0.382029889647
0.284409592048
The average prediction error over all tests is : 5.74908937687
```

Figure 42: CV errors for the 3 periods for #gopatriots

```
<terminated> part4b.py [C:\Python27\python.exe]
For #nfl:
Period 1 :
The average prediction error for each test is:
11.4008008928
11.4083077794
14.0784663326
46.0178412849
44.4646345686
188.162479103
72.0317601716
72.6240886138
76.5284128982
376.747766342
The average prediction error over all tests is : 91.3464557987


Period 2 :
The average prediction error for each test is:
62750.6903435
5703.98257162
645.894982474
1152.6549772
1646.81433321
14106.9083289
2061.67333257
117.228238452
20333.7353447
7275.19524895
The average prediction error over all tests is : 11579.4777702


Period 3 :
The average prediction error for each test is:
170.005079242
223.804242331
93.0545784344
123.437351114
146.587096246
50.753589234
76.317753865
116.748394546
172.213772165
588.261282062
The average prediction error over all tests is : 176.118313924
```

Figure 43: CV errors for the 3 periods for #nfl

```
<terminated> part4b.py [C:\Python27\python.exe]
For #patriots:
Period 1 :
The average prediction error for each test is:
12.251110551
13.5875987697
25.1798059768
45.6040144723
89.9383484809
150.314879709
507.523950728
409.489791169
265.843879158
302.005552117
The average prediction error over all tests is : 182.173893113


Period 2 :
The average prediction error for each test is:
24721.0973554
22855.4705659
21497.2545564
12202.8068267
20268.7673327
9013.5358484
9035.80548914
11231.7374732
43785.4977839
54909.2289475
The average prediction error over all tests is : 22952.1202179


Period 3 :
The average prediction error for each test is:
593.694335949
87.0025883427
70.8488979004
18.8998483997
44.0629204701
26.7766246871
14.1797617479
20.7804494798
33.9685421464
43.1590173513
The average prediction error over all tests is : 95.3372986474
```

Figure 44: CV errors for the 3 periods for #patriots

```
<terminated> part4b.py [C:\Python27\python.exe]
For #sb49:
Period 1 :
The average prediction error for each test is:
32.777901698
12.9814985844
30.2516759544
19.5001736925
16.9305239322
16.3916195818
34.3355748433
33.5279315283
162.503042286
217.560330777
The average prediction error over all tests is : 57.6760272878


Period 2 :
The average prediction error for each test is:
253116.6762
51391.5121734
2298.43526565
497914.806294
117.279022676
5388.21167348
894.481862441
8546.59102443
16604.2114335
258548.195716
The average prediction error over all tests is : 109482.040067


Period 3 :
The average prediction error for each test is:
366.568628862
153.699979714
75.4577624426
161.935180262
51.8766767871
64.9945105439
37.8754106863
47.9423634431
55.5489941036
73.5724312641
The average prediction error over all tests is : 108.947193811
```

Figure 45: CV errors for the 3 periods for #sb49

```
<terminated> part4b.py [C:\Python27\python.exe]
For #superbowl:
Period 1 :
The average prediction error for each test is:
16.0664958088
18.6777722901
17.730766023
20.33754066
84.9710403575
443.541719005
229.290635363
124.838035911
161.910604443
979.100381766
The average prediction error over all tests is : 209.646499163


Period 2 :
The average prediction error for each test is:
60685.7568183
156287.3545
49113.1927497
73544.7506303
2327440.73562
2441194.40696
2998850.57594
1429377.28943
1658881.04734
1181172.15122
The average prediction error over all tests is : 1237654.72612


Period 3 :
The average prediction error for each test is:
758.908542403
354.814634799
302.786107761
177.762645528
121.077275275
43.3321476003
62.7149156928
96.7256800757
104.229686197
540.076767771
The average prediction error over all tests is : 256.24284031
```

Figure 46: CV errors for the 3 periods for #superbowl

```
<terminated> part4b.py [C:\Python27\python.exe]
For all hashtags together:
Period 1 :
The average prediction error for each test is:
87.1799479572
102.524657581
131.695300491
238.484346815
357.98370423
436.985607712
1735.02027412
530.471311026
453.291306676
6002.47978882
The average prediction error over all tests is : 1007.61162454


Period 2 :
The average prediction error for each test is:
401565.48132
115261.915481
190833.015903
168636.130156
150224.609293
126945.754054
219897.111278
609683.052275
21920.4797646
764224.71402
The average prediction error over all tests is : 276919.226355


Period 3 :
The average prediction error for each test is:
1331.91838335
560.195560468
402.869680635
252.026423953
211.610978239
151.772164452
166.743733175
279.075514558
326.546700722
608.480723644
The average prediction error over all tests is : 429.123986319
```

Figure 47: CV errors for the 3 periods using the dataset including all tweets across all hashtags

Twitter activity during the Super Bowl is bound to be very volatile and spiky; not to mention the dataset of 12 hours versus the long durations of period 1 and 3 will lead to less accurate predictions for data in period 2. As shown in *Figures 41-47*, our average prediction error during times of relative inactivity is fairly low;

however during the superbowl (period 2), we get large prediction errors due to the lack of data (only 12 points) and the volatility of tweet statistics during the event itself.

Part 5: Run your model and make predictions for the next hour in each test file using a 6 hour window.

Since the test data contains a hashtag's tweets for a 6 hour window and Part 2, 3 and 4 were done using a 1 hour window, the feature vectors have been modified to use 6 hour windows in lieu of 1 hour windows for this part (this was done in Excel). For the sake of simplicity and brevity, only the model that considered all the hashtags has been used as a more representative model of general popularity trends for the Super Bowl.

Furthermore, the models used for each test data are derived only from the data within the specified period. For example, test data from period 1 will be predicted using a model derived from data in period 1 of the training data. The predictions for each file are detailed below in *Figure 48-57.*

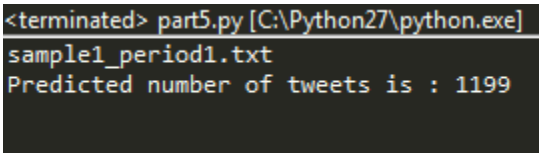

Figure 48: Prediction for Sample 1

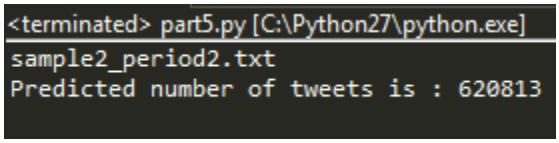

Figure 49: Prediction for Sample 2

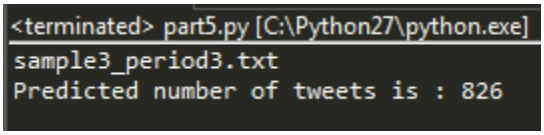

Figure 50: Prediction for Sample 3

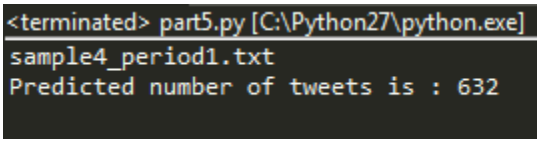

Figure 51: Prediction for Sample 4

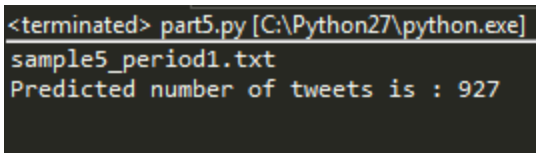

Figure 52: Prediction for Sample 5

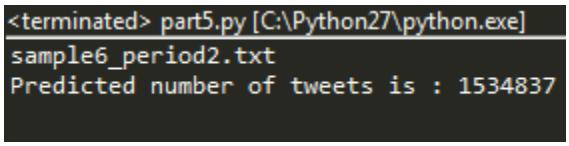

Figure 53: Prediction for Sample 6

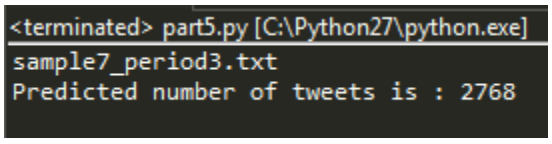

Figure 54: Prediction for Sample 7

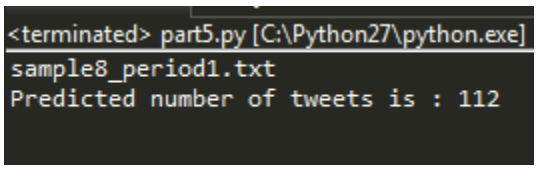

Figure 55: Prediction for Sample 8

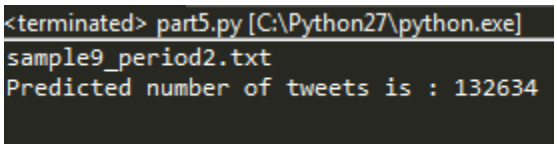

Figure 56: Prediction for Sample 9

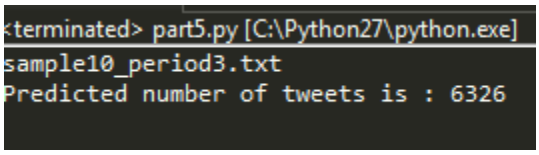

Figure 57: Prediction for Sample 10

Next, we table the time relative to the Super Bowl of these sample tweet datasets and the predicted number of tweets.

Sample 1: 3 days before SB -> 1199
Sample 2: 11 AM to 5 PM -> 620,813
Sample 3: 1 day after SB -> 826
Sample 4: 6 days before SB -> 632
Sample 5: 5 days before SB -> 927

Sample 6: 9 AM to 3 PM -> 1,534,837
Sample 7: 1 day after SB -> 2768
Sample 8: 4 days before SB -> 112
Sample 9: 10 AM to 4 PM -> 132,634
Sample 10: 4 days after SB -> 6326

We can see that for samples before and after SB, the predicted numbers are relatively small, while samples during the SB, the predicted number are relatively big. Thus, our model is predicting popularity fairly well.

Part 6: Train a classifier to predict the location of the author given only the textual context; report the ROC curve, confusion matrix, accuracy, recall, and precision of your classifier.

For this problem, we first decided to sort through the #superbowl data set by searching for specific strings that would be given in the location field of the dataset for users tweeting from either Massachusetts or Washington. The keywords for each included the name of the top 5 cities from each state as well as the abbreviations for each of the states, MA and WA.

Upon filtering we were able to retrieve a set of 9800 data points to work with. We then used a similar approach to Project 2 in which we used stemmed TF-IDF vectors to represent each word in each tweet. Here, our corpus was essentially our collection of each tweet with the text in each tweet representing a document. We then used LSI to reduce the dimensionality of the matrix to an optimal dimension and trained an SVM in order to classify each tweet.

We found that the SVM outperformed other methods of classification such as k-nearest neighbor. We improved on the result we got by then transitioning to soft-margin SVM and found the optimal gamma variable that yielded us the results outlined below in *Figures 58 and 59*.

n = 1964	Predicted: CT	Predicted: RA	
Actual: CT	202	358	
Actual: RA	164	1239	

	CT	RA	Average
Precision	0.551912568	0.775829681	0.663871124
Recall	0.360714286	0.883107627	0.621910956
Accuracy	0.734080489		0.673287523

Figure 58: Confusion matrix, accuracy, precision, and recall

We were able to achieve an accuracy of 73% and as shown in the ROC curve below in *Figure 59*, our classifier shows improvement on the random guess classifier.

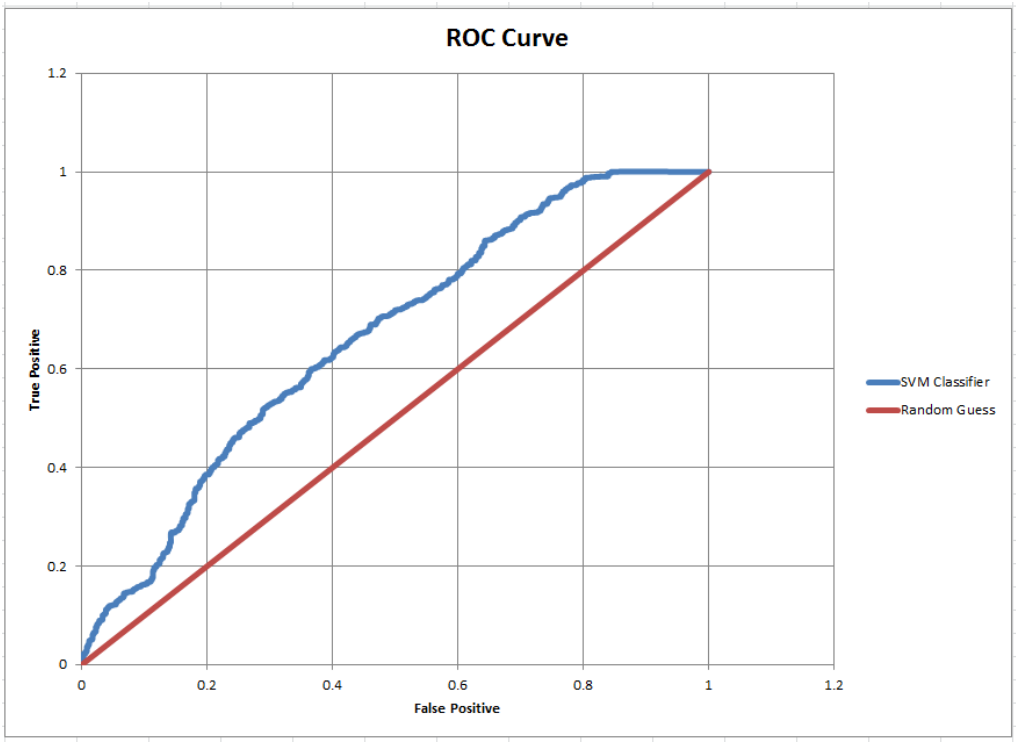

Figure 59: ROC Curve for Soft Margin SVM Classifier

Considering this was based solely on text data with each tweet being no more than 140 characters, we believe we achieved sufficient results with our classifier.

Part 7: Come up with your own project!

While it is interesting to calculate statistics and popularity of Twitter data, something that could be very useful is predicting the general emotion of users posting tweets pertaining to a certain topic over time.

For example, tweets that have a lot of capital letters, question marks and exclamation points, and other various features lead us to believe that the user who posted the tweet is emotionally riled up in some way; they could have been upset over a loss, excited about a play, or confused about the call of a referee.

We attempt to predict excitement level for tweets in the next hour based on the excitement level for tweets in the current hour along with other features that will be described next. In order to keep it simple, we decided to use minimal features that we thought would help determine emotional trends in the tweets.

Here is our procedure, step by step:

1) Include data from all tweets across all hashtags; this will represent the data representing opinions on the Super Bowl. For each tweet, we calculate the following:

I. Number of exclamation marks used in the tweet
II. Number of question marks used in the tweet
III. Percentage of letters that are capitalized in the tweet

To ensure that only the content of the tweet is captured, and not the extraneous parts, the hashtags, mentions, and links have been taken out to remove any bias in the percentage of capital letters. For example, links have many capital letters that have nothing to do with the content of the tweet itself.

2) Now, based on the quantities for each tweet in 1, we assign an excitement flag to the tweet, 0 being not excited and 1 being excited. The criteria we used were this:

1. If there was an exclamation mark present.
2. If there were 2 or more question marks present.
3. If the percentage of capital letters exceeded 20% of the entire tweet.

If any of these were satisfied, the tweet was marked with a 1 to indicate excitement.

For the exclamation mark, if even one is present, we can assume the user is excited/mad/happy about something and thus is showing emotion.

For the question marks, one question mark could be a simple question, but often when one shows emotion, one will ask a lot of questions (for example, questioning someone's performance).

For the capital letters, usually a couple of words will be capitalized to show interest/excitement over a topic or word, and is a good indicator of emotion.

3) For every hour, the percentage of tweets that were "excited" was calculated. Along with this metric, the average number of retweets, followers, and favorites were included for every hour. Thus, we have 4 features that attempt to predict the excitement level for the next hour of tweets.

The excitement level is measured as a percentage (number between 0 and 1). If the level is 0, that means no one is excited versus a percentage of 1 where everyone is excited and showing emotion in their tweets.

Based on the current excitement level, retweets, followers, and favorites, we attempt to predict the excitement level of the following hour. To accomplish this, we use a linear regression model, similar to how we tried to predict popularity, and report the relevant metrics to show that this works in *Figures 60-61*.

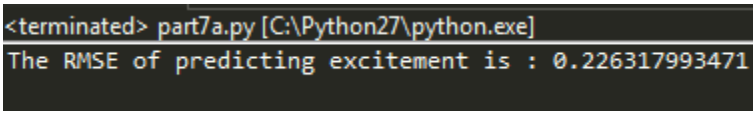

Figure 60: RMSE value of predicting excitement

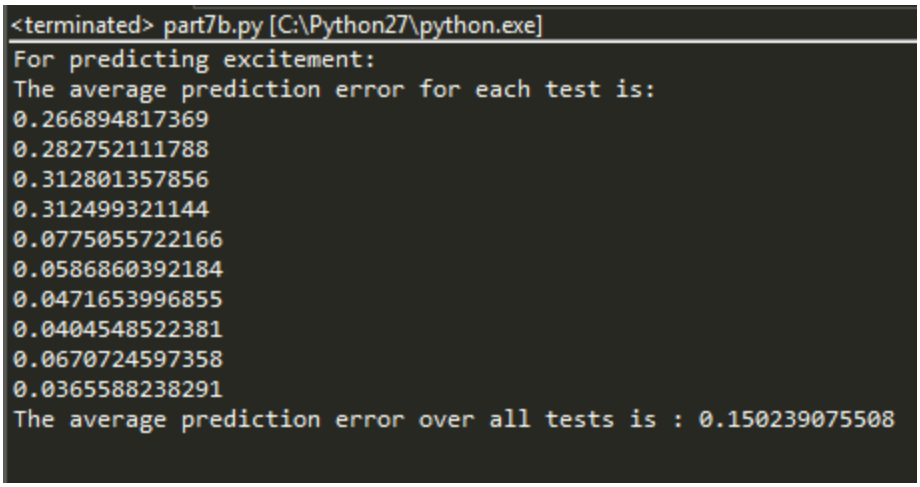

Figure 61: 10 fold CV on predicting excitement

As we can see in the figures above, our algorithm is doing a pretty good job in predicting excitement levels. In *Figure Y* especially, we see that on average we are only 15% off the actual percentage of excited tweets in the next hour, with a maximum of 31% off the actual percentage.

Further work for this project would include developing more features, grouping excitement levels and quantizing them (perhaps a binary classification of 0 being an excitement level of 50% and below and 1 being above) for a more general classifier. However, even with only 4 features, our algorithm is doing a decent job of predicting emotional levels for different hours based on Twitter metadata.

Of course, this isn't to say that our definition of "excitement" truly reflects the emotional state of the users posting the tweets. The criteria for an emotional tweet was fairly arbitrarily decided based on intuition; perhaps a deeper look into the human psyche could tell us more decisive criteria in determining emotion.