Popularity Prediction on Twitter

Large Scale Data Mining

Bhargav Parsi (804945591)

Kelly Bielaski (405023971)

Shreyas Lakhe (105026650)

Konark J S Kumar(204759469)

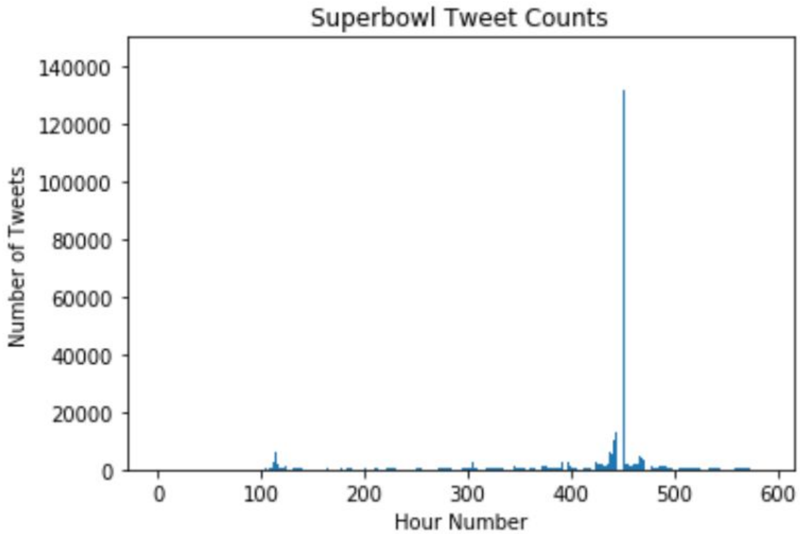
Part 1: Popularity Prediction

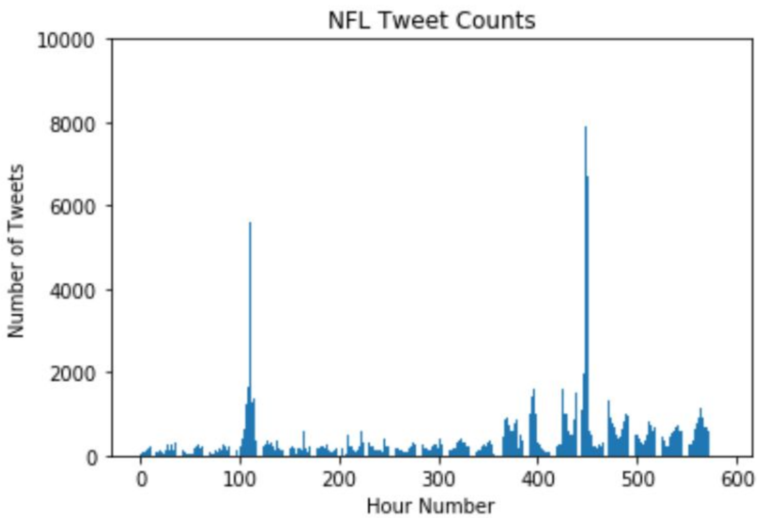
Problem 1.1

To complete this portion of the assignment, we downloaded the training tweet data and read it in line by line. As each line was read in, we saved the necessary features, which was the time, the number of followers, and the number of retweets, in a pandas dataframe. We then converted each row in the dataframe to be a seperate hour so that we could get the average number of tweets per hour. Initially we faced a problem to read the large text files, but we allocated the necessary space in the pandas dataframe before hand and kept only the necessary information. This increased the speed of processing upto **300** times.

	Average number of tweets per hour	Average number of followers of users posting tweets	Average number of retweets
#gohawks	328.909	2203.93	2.014
#gopatriots	58.684	1401.89	1.400
#nfl	444.295	4653.25	1.538
#patriots	834.264	3309.97	1.782
#sb49	1528.560	10267.31	2.511
#superbowl	22997.729	8858.97	2.388

For the superbowl and nfl hashtags, we got the number of tweets for each hour and plotted those in the following graphs.





Problem 1.2

We combined all the features for each hour to get a training feature matrix for the model. Then we trained the model using statsmodels.regression.linera_model.OLS. We use this linear model instead of the linear model in the sklearn linear model because it returns an object of type statsmodels which we can use to evaluate the model. Using the trained model, we predicted the number of tweets in the following hour for each of the hours in the training sets. We compared the predictions to the actual values to get the following results about the model. The values that we are comparing below are the RMSE, R squared measure, p-values, and t-values. The RMSE is the root mean squared error which means, it is the square root of the average of all the prediction's difference between the prediction and target values. The r-squared measure demonstrates how close the data is to the fitted regression line. It is determined by the ratio of the explained variation over the total variation. The p-values test the null hypothesis. If the p-value is small, it indicates that you can reject the null hypothesis and conclude the alternative hypothesis which is the model. For this reason, we are looking for small p-values that contribute to the alternative hypothesis. The t-values represent the same measure as the p-values but it uses the t-distribution as opposed to the normal distribution.

	RMSE	R squared Measure	P values	T values
#gohawks	974.	0.501	Num tweets = 1.3277 * 10^-12 Num retweets = 3.6613*10^-3 Num followers = 3.9971*10^-2 Max followers = 8.6064*10^-1 Hour = 8.0210 * 10^-3	Num tweets = 7.2531 Num retweets = -2.9179 Num followers = -2.0587 Max followers = 0.17563 Hour = 2.6604
#gopatriots	185	0.640	Num tweets = 0.75184 Num retweets = 0.021610 Num followers = 0.22563 Max followers = 0.060473 Hour = 0.35758	Num tweets = -0.31636 Num retweets = 2.3034 Num followers = 1.2130 Max followers = -1.8810 Hour = 0.92073
#nfl	585	0.647	Num tweets = 4.1428 * 10^-8 Num retweets = 5.4664 * 10^-3 Num followers = 2.8327 * 10^-3 Max followers = 4.4205 * 10^-2 Hour = 6.7856 * 10^-4	Num tweets = 5.5588 Num retweets = -2.7886 Num followers = 2.9981 Max followers = -2.0165 Hour = 3.4165
#patriots	2527	0.681	Num tweets = 1.4557 * 1-^-33 Num retweets = 1.4043 * 10^1 Num followers = 9.6398 * 10^-1 Max followers = 7.7308 * 10^-2 Hour = 6.5327 * 10^-1	Num tweets = 12.878 Num retweets = -1.4761 Num followers = -0.045171 Max followers = 1.7696 Hour = 0.449446
#sb49	4471	0.809	Num tweets = 6.9992 * 10^-32 Num retweets = 1.4458 * 10^-2 Num followers = 1.8300 * 10^-1 Max followers = 3.4654 * 10^-2 Hour = 8.0547 * 10^-1	Num tweets = 12.4960 Num retweets = -2.4530 Num followers = 1.3331 Max followers = 2.1173 Hour = -0.24638
#superbowl	8004	0.805	Num tweets = 8.0613 * 10^-115 Num retweets = 4.3899 * 10^-15 Num followers = 6.2275 * 10^-12 Max followers = 7.1871 * 10^-8 Hour = 1.9138 * 10^-1	Num tweets = 28.961 Num retweets = -8.0591 Num followers = -7.0196 Max followers = 5.4567 Hour = -1.3080

The best features are the ones with the lowest p-value. **Number of tweets and number of retweets** are, a majority of the time, the most significant features.

Problem 1.3

We tried using adding a few features to get a better prediction. The features we used were: ["num_of_tweets","num_of_retweets","Sum of followers"," max of followers","hour","count of favorites","sum of verified","sum of status","sum of friends","sum of ranking scores","sum of impressions","sum of momentum"].

Motivation behind using these features

Count of Favorites:

It is the sum of number of likes received to the tweets produced in a period of 1 hour. This might be directly proportional to the number of tweets in the next hour.

Sum of verified users tweets:

If the number of verified users post tweets in a particular hour is more, there might be a high probability of it getting retweeted and hence might have a correlation with the number of tweets in the next hour.

Sum of statuses:

It is the the number of Tweets (including retweets) issued by the user. We thought this might have a direct relation with the number of tweets in the next hour.

Sum of friends:

It is the the number of users this account is following. This might be directly proportional to the number of tweets in the next hour.

sum of ranking scores:

Collective score which determine which posts users are more likely to be recommended to. A number of features go into creating the ranking score including a tweet's overall engagement, the tweet's engagement relative to other tweets by the same author, how recently the tweet was published, etc. Higher ranking scores should mean that the tweet will be recommended to more users and thus should bring about more tweets in the next hour.

sum of impressions:

Impression is the delivery of a post to an account's Twitter stream. Impressions are related to the engagement after the tweet. The higher number of impressions, the more likely it will spur discussion in the next hour.

sum of momentum:

Momentum measures the potential for users to see the tweet and respond. The higher the momentum the more likely that the number of tweets in the next hour will also be high.

We used the same linear model on these features and the same methods to evaluate the models. The graphs show the number of tweets per hour on the x axis and the feature value on the y axis. **It appears that there is a cluster towards a smaller number of tweets per hour and a smaller feature value.**

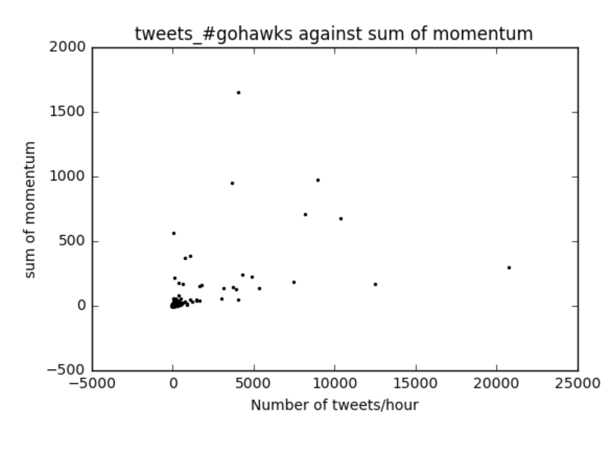
#gohawks

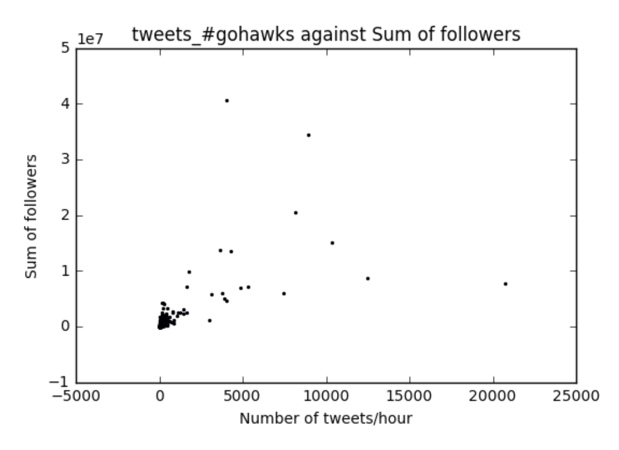
RMSE = 792.0449492104833

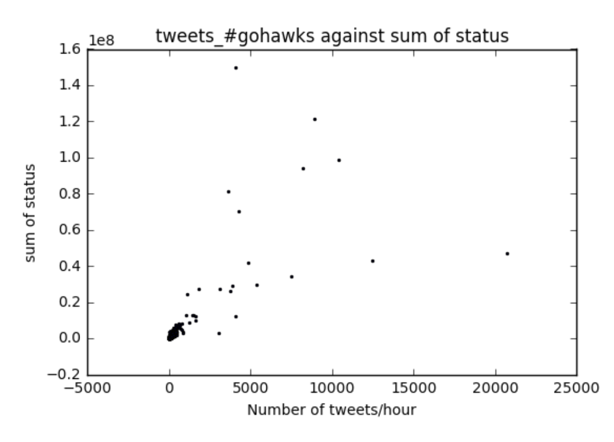
Pvalues

Num of tweets	5.1182 * 10^-1
Num of retweets	4.3986 * 10^-1
Sum of followers	1.1185 * 10^-14
Max of followers	6.4586 * 10^-2
Hour	1.0459 * 10^-1
Count of favorites	3.1136 * 10^-4
Sum of verified	9.1787 * 10^-3
Sum of status	2.8897 * 10^-14
Sum of friends	3.6032 * 10^-1
Sum of ranking scores	4.6166 * 10^-1
Sum of impressions	1.4884 * 10^-7
Sum of momentum	9.0632 * 10^-15

From these p-values, we can conclude that the sum of followers, sum of status, and sum of momentum tend to have the most influence and are therefore the most significant features.







#gopatriots

RMSE = 158.48601891541676

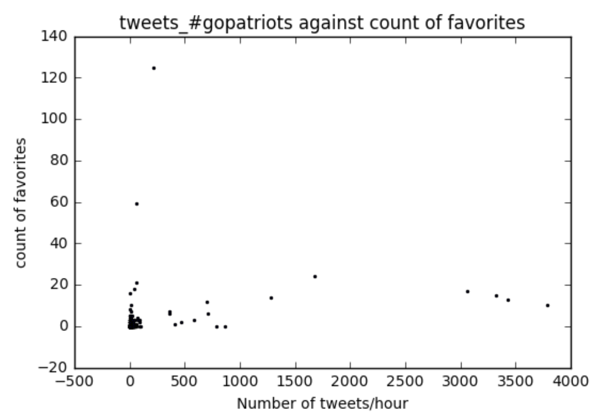
The gopatriots hashtag has the lowest RMSE. This makes sense because it appears to be the most specific of the hashtag. Not only does it refer to a specific team, but it also adds a positive connotation to the phrase by adding the term “go” to it.

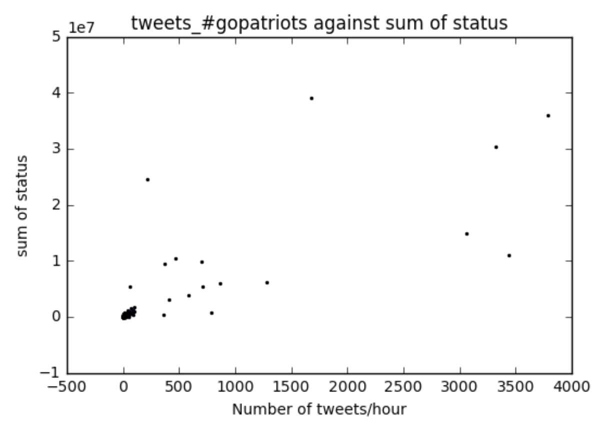
Pvalues

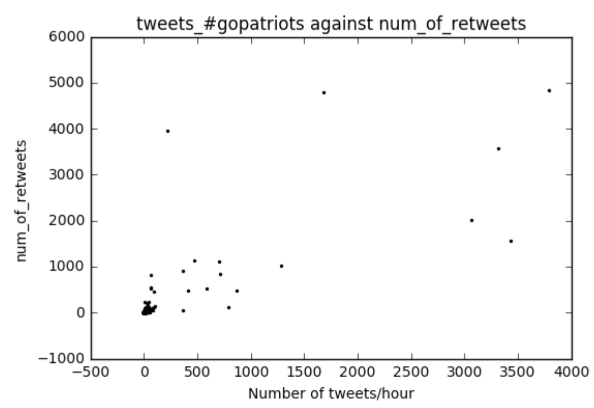
Num of tweets	1.3028 * 10^-1
Num of retweets	4.5912 * 10^-7
Sum of followers	6.9414 * 10^-3
Max of followers	3.5117 * 10^-1
Hour	1.5424 * 10^-1

Count of favorites	7.6387 * 10^-3
Sum of verified	1.9430 * 10^-1
Sum of status	5.3940 * 10^-8
Sum of friends	4.9576 * 10^-4
Sum of ranking scores	4.6166 * 10^-2
Sum of impressions	5.9488 * 10^-2
Sum of momentum	7.2835 * 10^-4

The most significant features are the number of retweets, the count of favorites, and the sum of statuses.







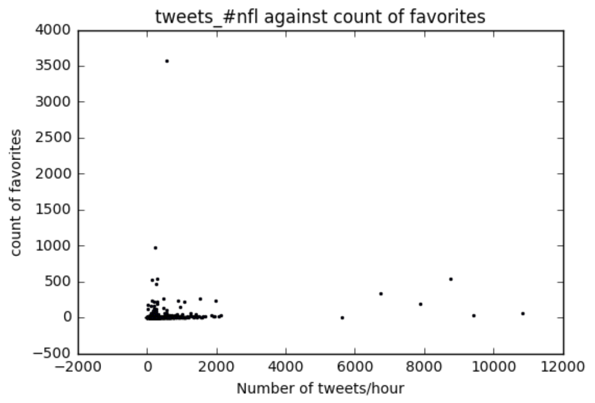
#nfl

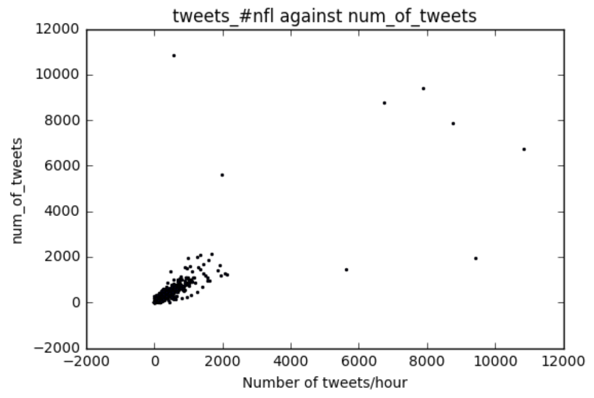
RMSE = 482.1178277028538

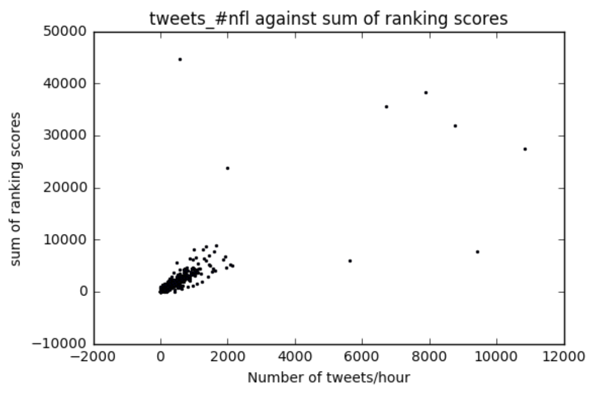
Pvalues

Num of tweets	2.0363 * 10^-7
Num of retweets	1.0588 * 10^-2
Sum of followers	9.3521 * 10^-1
Max of followers	3.2270 * 10^-1
Hour	3.8004 * 10^-1
Count of favorites	2.0122 * 10^-32
Sum of verified	7.7456 * 10^-3
Sum of status	2.3615 * 10^-1
Sum of friends	7.8425 * 10^-2
Sum of ranking scores	5.0788 * 10^-5
Sum of impressions	8.9558 * 10^-1
Sum of momentum	4.6379 * 10^-1

From these p-values, we can conclude that the most significant features are the number of tweets, the count of favorites, and the sum of the ranking scores.







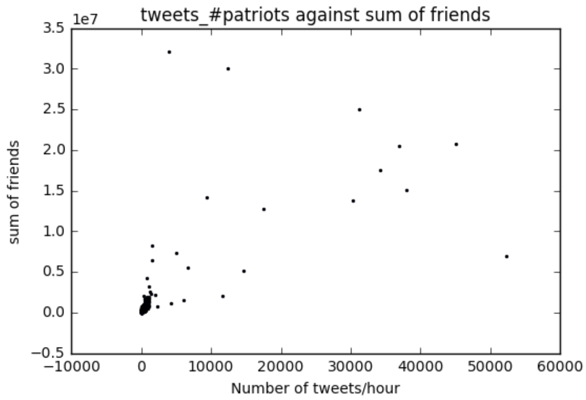
#patriots

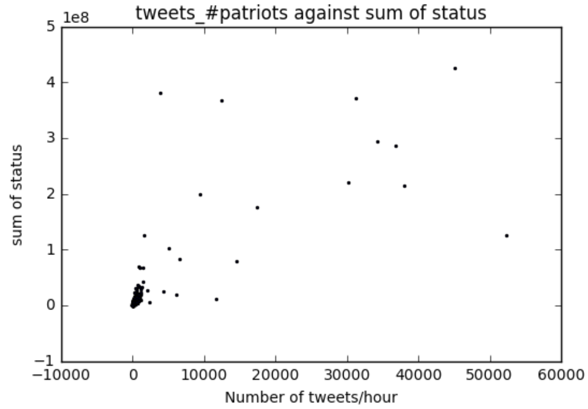
RMSE = 2230.457675306663

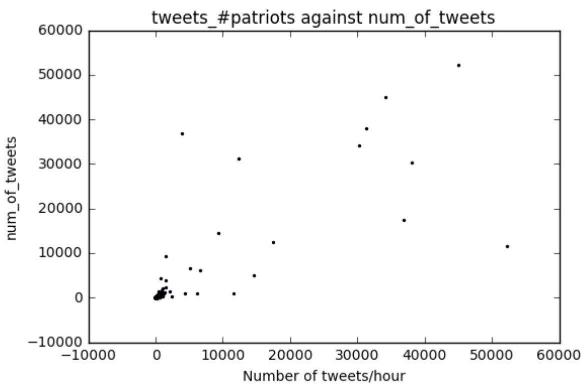
Pvalues

Num of tweets	6.7298 * 10^-5
Num of retweets	2.6254 * 10^-4
Sum of followers	6.6184 * 10^-2
Max of followers	1.5731 * 10^-1
Hour	3.5830 * 10^-1
Count of favorites	3.4884 * 10^-1
Sum of verified	2.1002 * 10^-4
Sum of status	7.9945 * 10^-14
Sum of friends	5.5865 * 10^-24
Sum of ranking scores	6.2442 * 10^-4
Sum of impressions	1.5356 * 10^-1
Sum of momentum	1.1098 * 10^-2

For the patriots hashtag, the most significant features are the number of tweets, the sum of statuses, and the sum of friends.







#sb49

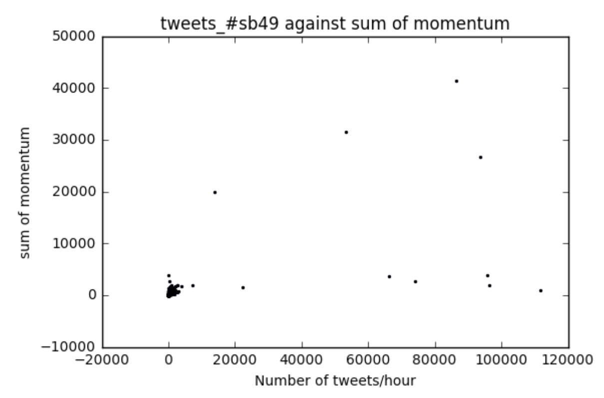
RMSE = 4344.454965441463

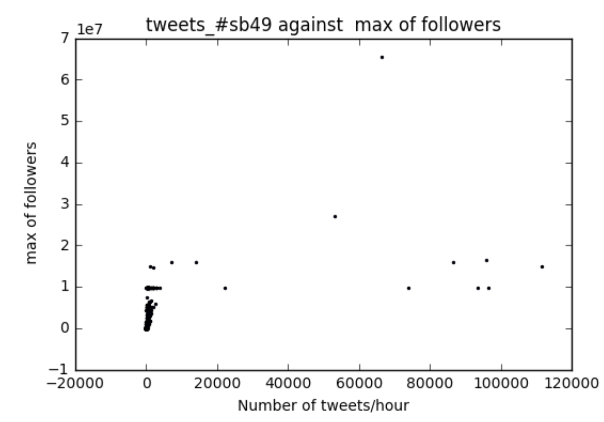
Pvalues

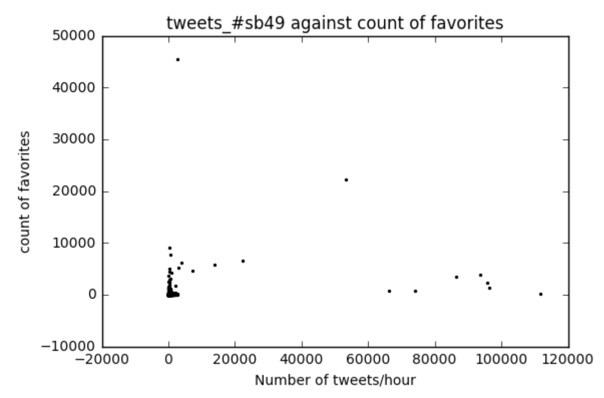
Num of tweets	7.4229 * 10^-2
Num of retweets	9.3652 * 10^-1
Sum of followers	1.4463 * 10^-1
Max of followers	3.2166 * 10^-4
Hour	9.6206 * 10^-1
Count of favorites	4.4397 * 10^-3
Sum of verified	1.6799 * 10^-1
Sum of status	1.6201 * 10^-1
Sum of friends	9.9837 * 10^-1

Sum of ranking scores	3.5933 * 10^-2
Sum of impressions	8.6364 * 10^-2
Sum of momentum	4.6098 * 10^-7

The lowest p-values are the max of followers, count of favorites, and the sum of momentum, making those the most significant features.







#superbowl

RMSE = 6058.779193236606

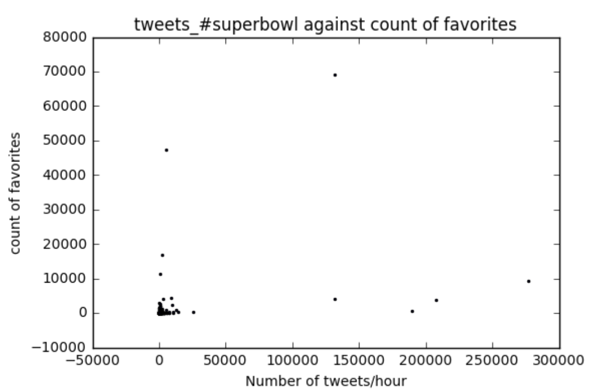
The superbowl hashtag has a relatively high RMSE compared to the other hashtags. This makes sense because it would have a lot of variety in types of tweets because the word is very generic.

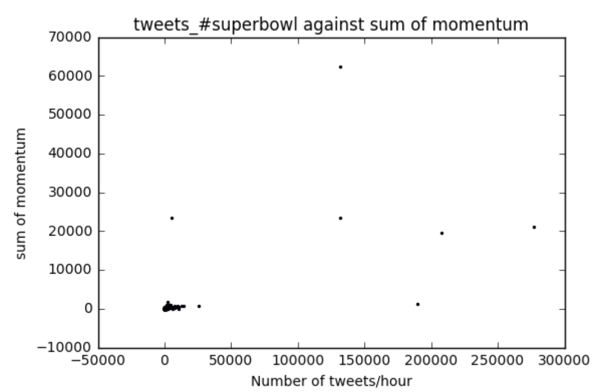
Pvalues

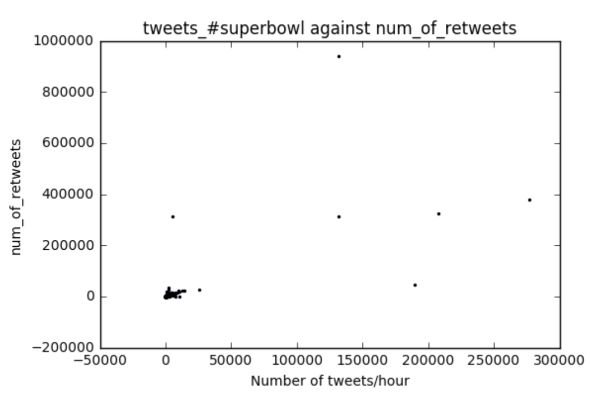
Num of tweets	2.1681 * 10^-9
Num of retweets	1.3327 * 10^-9
Sum of followers	3.5076 * 10^-4

Max of followers	2.4773 * 10^-1
Hour	1.8719 * 10^-1
Count of favorites	1.5901 * 10^-13
Sum of verified	3.2934 * 10^-1
Sum of status	1.3752 * 10^-3
Sum of friends	8.1041 * 10^-1
Sum of ranking scores	4.1594 * 10^-9
Sum of impressions	3.7931 * 10^-4
Sum of momentum	1.6801 * 10^-13

The most significant features of the superbowl hashtag are the number of retweets, the count of favorites, and the sum of momentum.





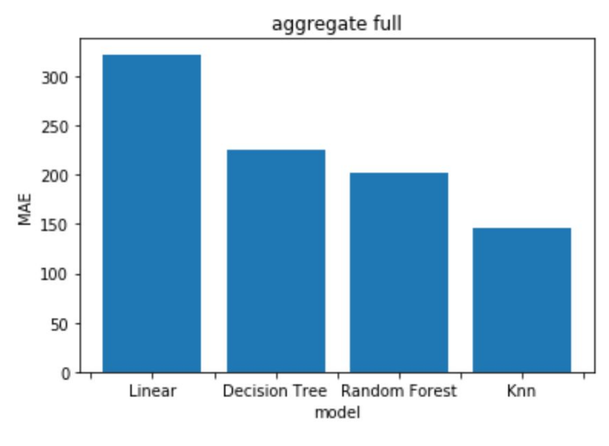


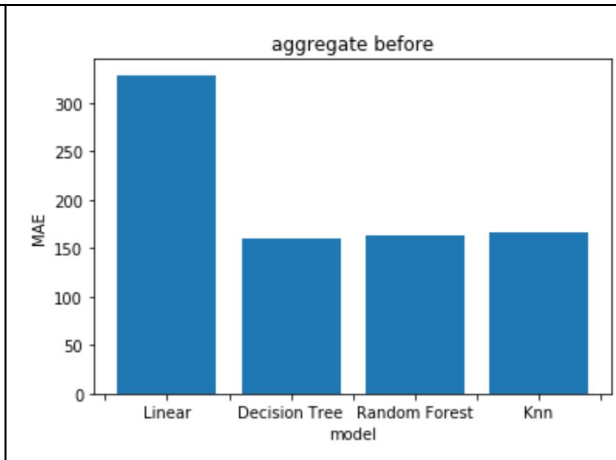
Problem 1.4

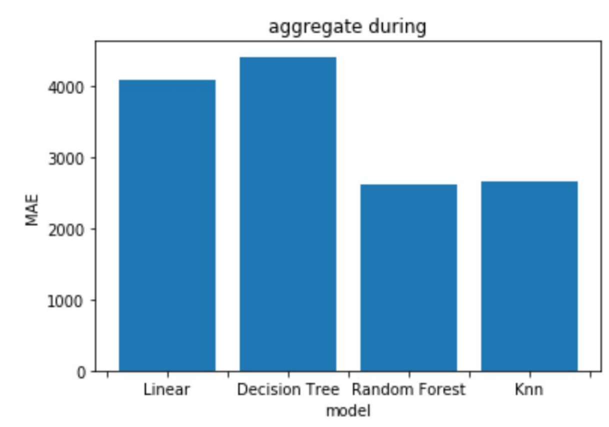
For each hashtag, we split the dataframe into 4 dataframes, one of the tweets that occurred before the superbowl, another of the tweets that occurred during the superbowl, a third of the tweets posted after the superbowl, and the last with all the tweets. For each time period, we changed the data frame so that there is one feature row per hour. All the tweets that occur in that particular hour were combined ie. for sum_followers, we took the sum of all the followers for all the tweets in that hour and for max_followers we took the max of the number of followers for that hour. We then applied cross validation using each of the following models: linear, polynomial regression using degree of 2 and 3, decision tree, random forest, and k nearest neighbor models. The linear model uses the linear regression line to classify points. The polynomial regression models transforms the data using the polynomial specified. The decision tree predicts using a different decision rule at each branch based on the feature. The random forest fits decision tree classifiers on subsets of the data. KNN takes the average value of the nearest n neighbors. Here are the results of **MAE** we found for each method:

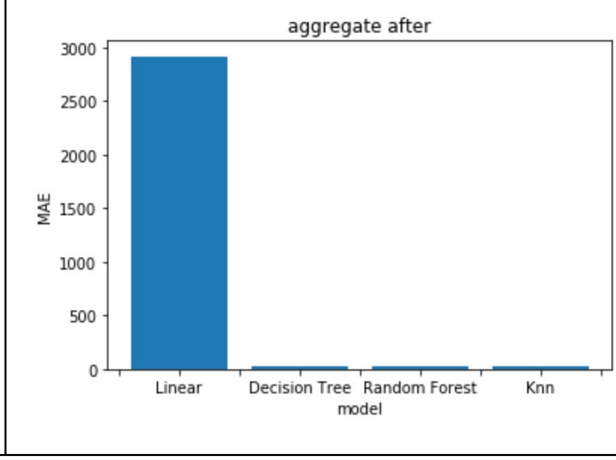
#gohawks	Full Set	Before Super bowl	During Super Bowl	After Super Bowl
Linear Model	**322.4608707993786**	329.04846369612306	4108.213213332476	2916.650503773027
Polynomial Regression, degree 2	2958.663033928536	**1108.0802447799672**	15144.495774913044	247030747.9883082
Polynomial Regression, degree 3	183799.14126350038	2805621.305257548	386311.2890385686	266582745125.4477
Decision Tree	229.91948672207042	163.9255025466888	4467.55	**22.372652532389246**
Random Forest	181.92498245614033	169.53673890063425	3123.77	**24.157838369963372**
K nearest neighbors	146.07338777979427	166.66996828752642	2662.37	**31.067051282051278**

The following graphs show the MAE of the linear, decision tree, random forest regressor, and knn model for each time period. We wanted to present this so that it is easy to see which one is the lowest and is the best model. We aren't including the polynomial regression classifiers because they don't perform well and skew the graph.





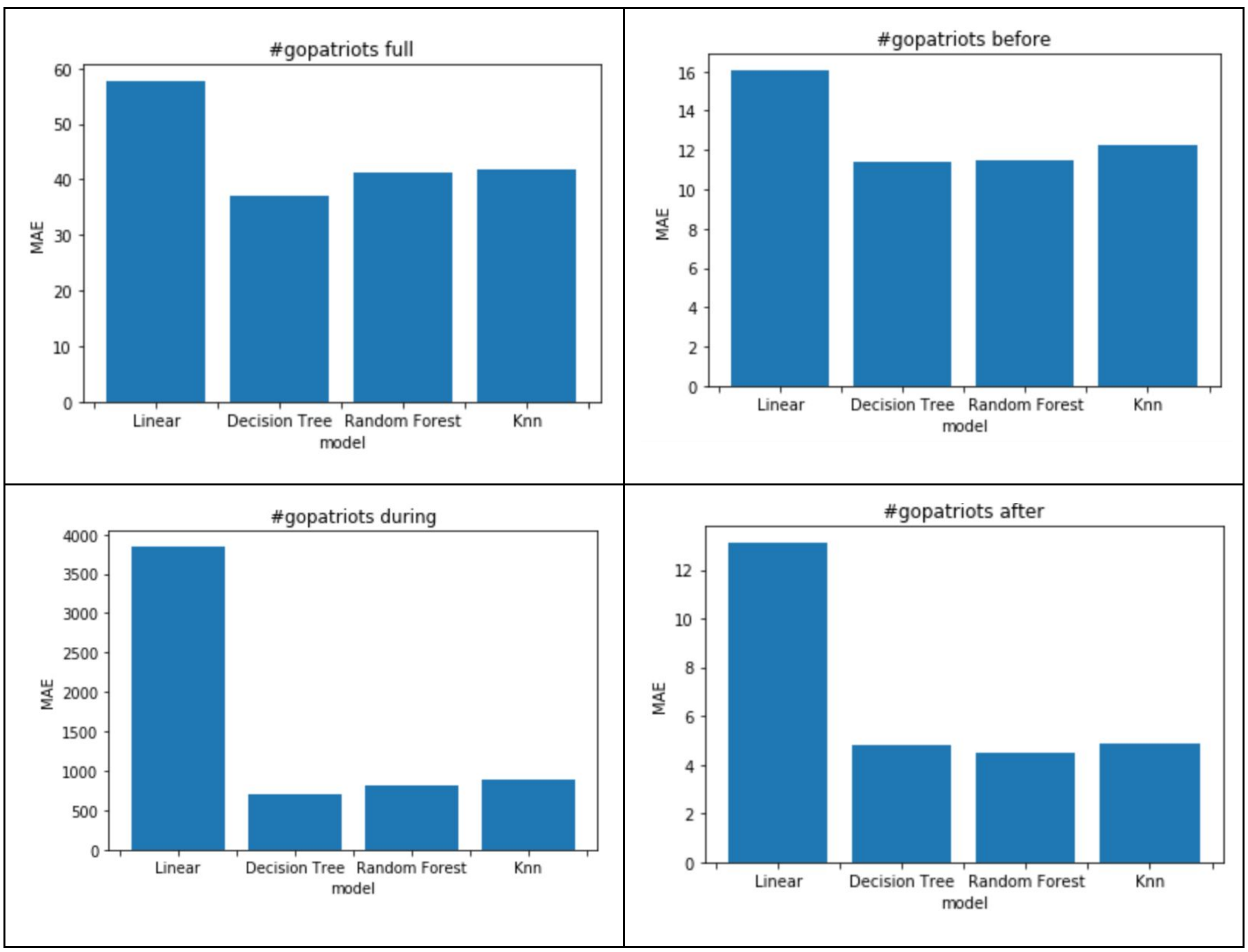




For the #gohawks dataset, we found that in the linear model, it was better to use the full set of data as opposed to splitting it into groups. We also found that the decision tree had the minimum MAE at 22.372 after the superbowl and on average, the KNN approach had the lowest MAE.

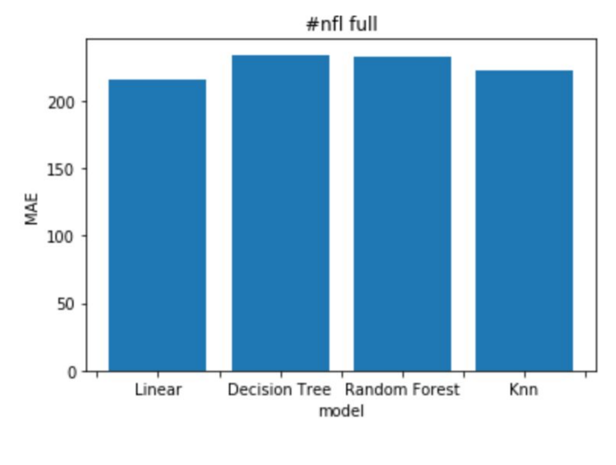
#gopatriots	Full Set	Before Super bowl	During Super Bowl	After Super Bowl
Linear Model	57.771245550286 19	16.079097557108 504	3851.30307081771 25	**13.12930432404 489**

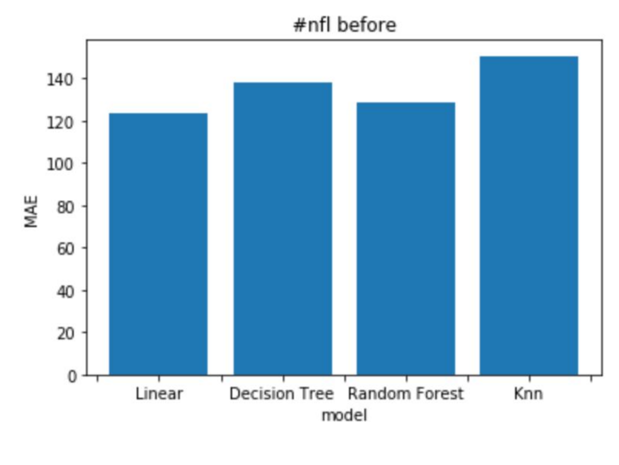
Polynomial Regression, degree 2	3228.346308462222	2300.8603916762313	153849.30252861936	24509.47709950451
Polynomial Regression, degree 3	8539377.714733407	70986856.61140174	2368910.4559301077	649947447.3944753
Decision Tree	37.51605471809563	11.473476061229753	548.45	**3.033484611885084**
Random Forest	41.921909628147304	11.601034370140079	846.8300000000002	**3.699600656288157**
K nearest neighbors	41.68020568663037	12.274746300211417	883.7100000000003	**4.848461538461537**

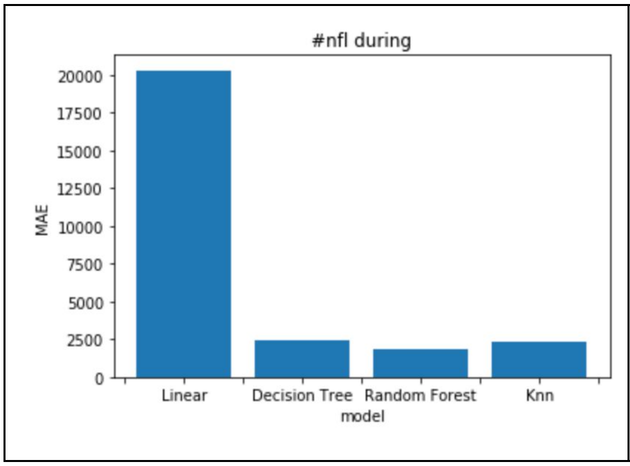


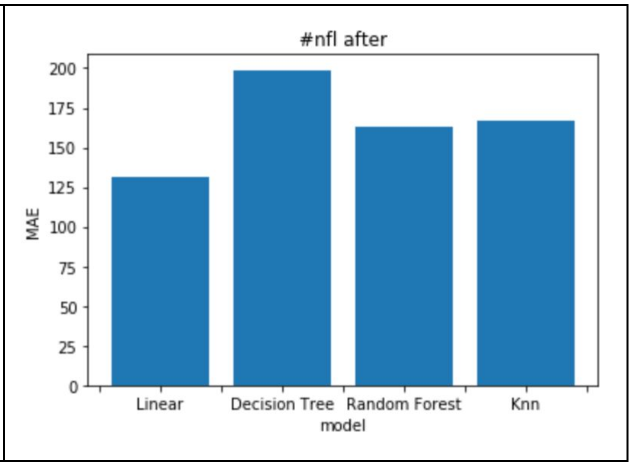
For the #gopatriots dataset, we found that the MAE's were relatively lower. The linear model performed well but the best again were the decision tree, random forest, and KNN models, with the decision tree model having the lowest overall MAE in the period after the superbowl and the decision tree on average, having the lowest MAE.

#nfl	Full Set	Before Super bowl	During Super Bowl	After Super Bowl
Linear Model	215.35366159011534	**123.84311675806654**	20296.968973825355	131.02641982856647
Polynomial Regression, degree 2	1384.9815046389829	**289.7477339173423**	121302.18001809041	1339.0199902691982
Polynomial Regression, degree 3	104103.74098460449	32871.061368566545	890854.8639945819	390137.6986756129
Decision Tree	258.5923482782674	**148.38522877606488**	1823.7	207.4869532618508
Random Forest	220.84290765634128	**131.87182346723046**	1850.1199999999997	162.84648351648352
K nearest neighbors	222.7222443015781	**150.6560042283298**	2359.6600000000003	167.0621978021978



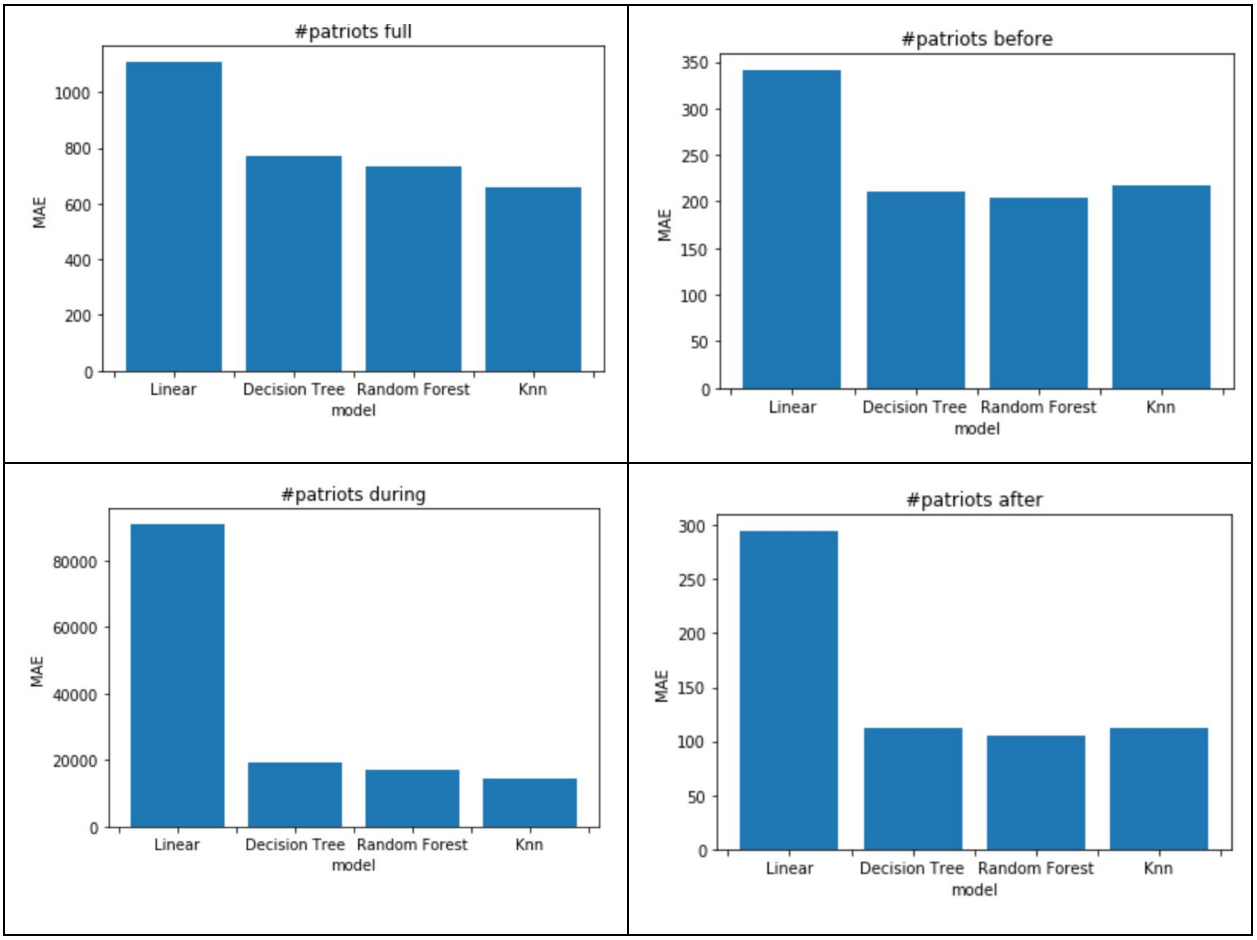






For the #nfl dataset, we found that the linear model had the overall minimum MAE. It also had the lowest MAE's in the periods before and and after the superbowl but performed poorly in the period during the superbowl, so the overall minimum average MAE comes from the random forest model.

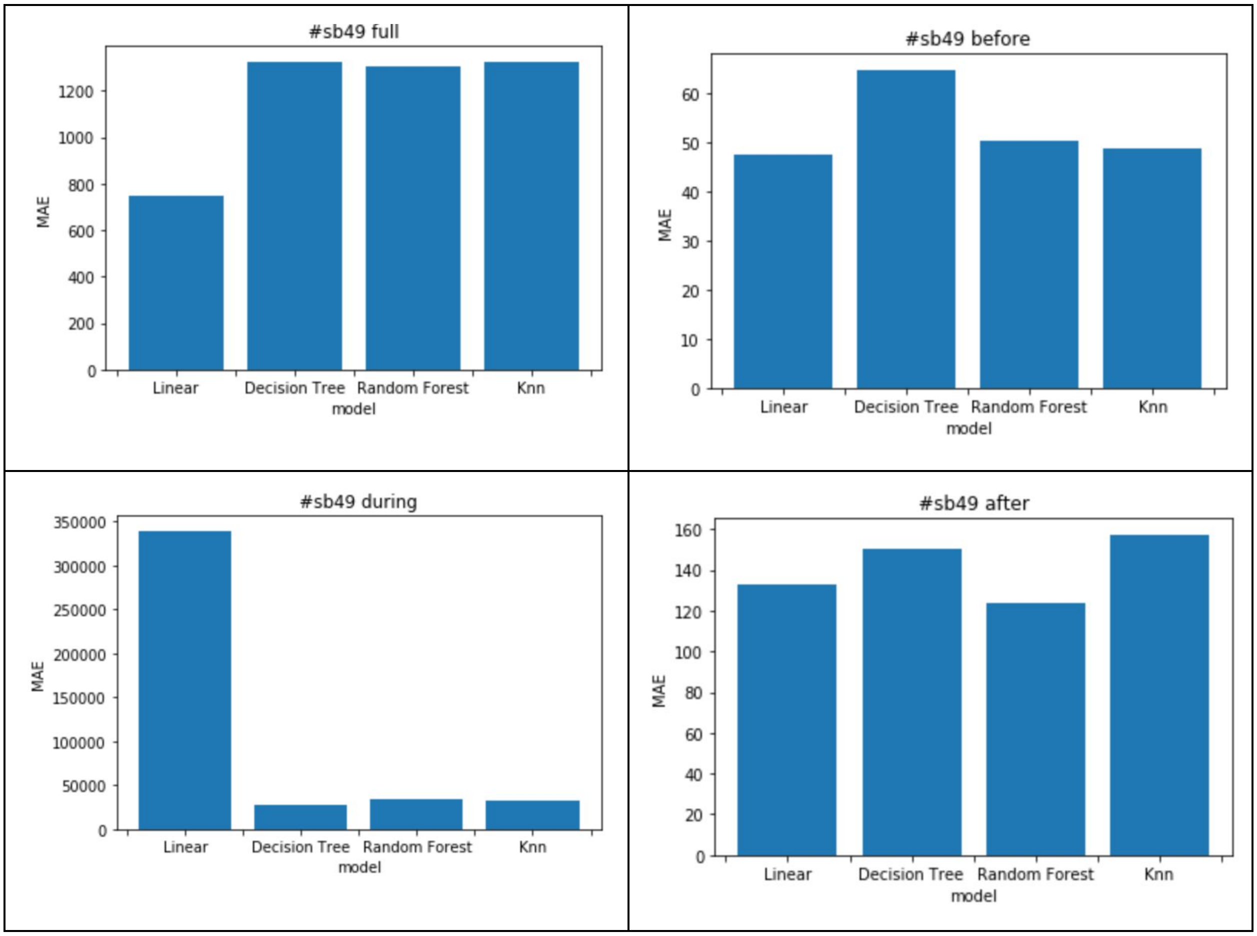
#patriots	Full Set	Before Super bowl	During Super Bowl	After Super Bowl
Linear Model	1109.8351259928922	341.4366190594668	91151.07401134752	**295.07884049225237**
Polynomial Regression, degree 2	12797.513118364106	**1532.375268955779**	392710.2350008624	280643.66239594505
Polynomial Regression, degree 3	896577.7588463376	152080.25880733324	1514398.7797180384	49853284.88963523
Decision Tree	758.0216952087349	213.29279574156558	20800.9	**112.50952141013438**
Random Forest	715.6053243717123	192.08594608879494	14154.805000000002	**104.52637362637361**
K nearest neighbors	658.3159380479253	218.01414376321355	14561.179999999998	**112.68307692307692**



Using the #patriots dataset, we found that the minimum MAE came from the random forest model in the last period. We also found that on average KNN had the lowest MAEs.

#sb49	Full Set	Before Super bowl	During Super Bowl	After Super Bowl
Linear Model	749.64691142286 03	**47.66240900966 33**	339489.93456735 65	132.801219117052 85
Polynomial Regression, degree 2	244481.747048614 02	801.828881143501 7	2227835.4431037 246	101874.84569482 98
Polynomial Regression, degree 3	293683902.5378 581	5077660.3384736 87	1489630.2949264 52	875507335.19844 69
Decision Tree	1355.03173816365 55	**66.08219045245 525**	38894.1	136.91303911828

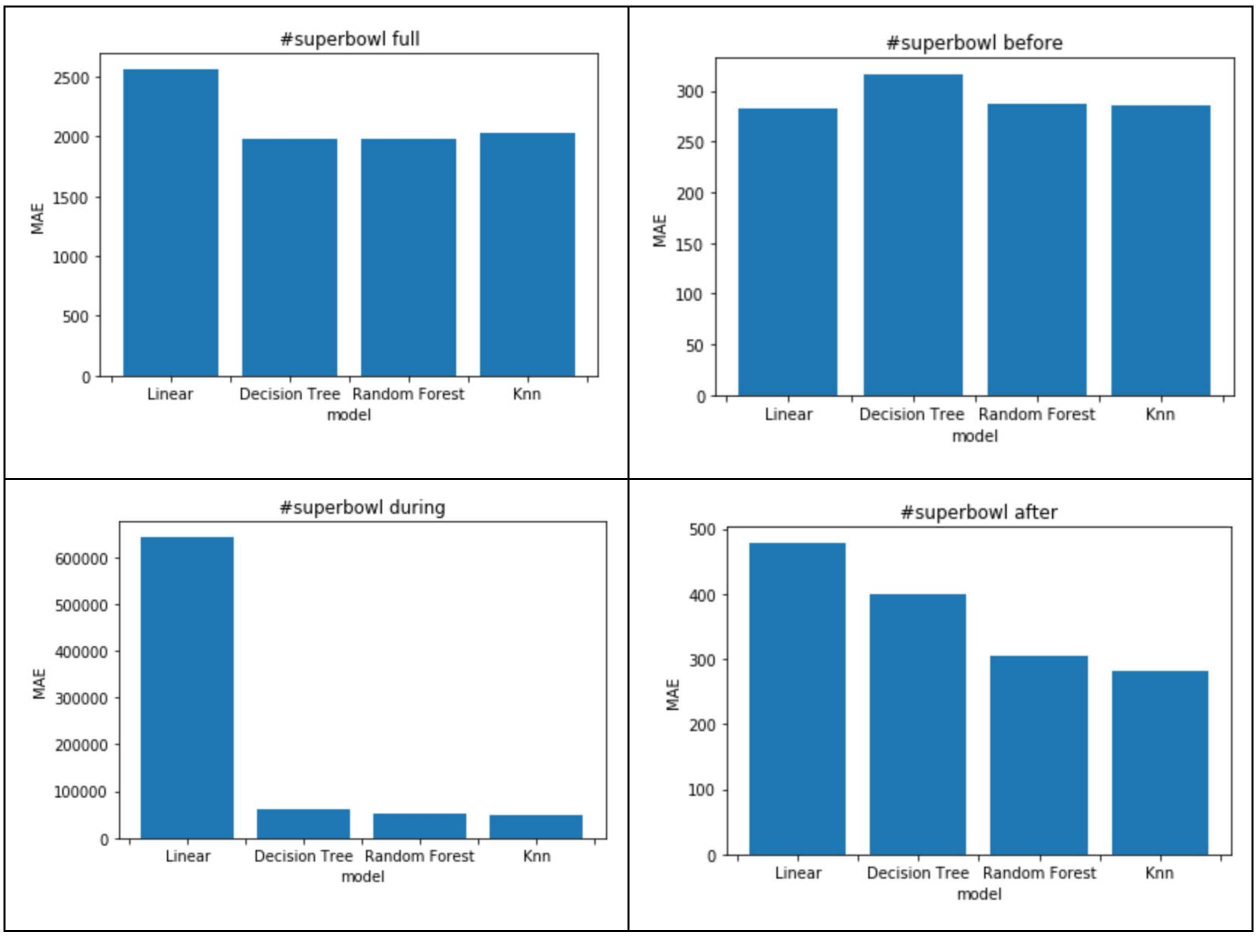
Random Forest	1301.623177365413	**51.70486120759087**	32228.21	128.09604395604399
K nearest neighbors	1325.932600818235	**48.65845665961946**	33062.46	157.37252747252745



From the #sb49 dataset, we found that the lowest MAE came from the linear model before the super bowl. We also found that on average the lowest MAE's were coming from the random forest model.

#superbowl	Full Set	Before Super bowl	During Super Bowl	After Super Bowl
Linear Model	2566.4340095747366	**283.0478083047684**	643477.6165349832	478.8983147506161
Polynomial	915361.54843704	2044.4656016051	4879039.580459	26568.73260786

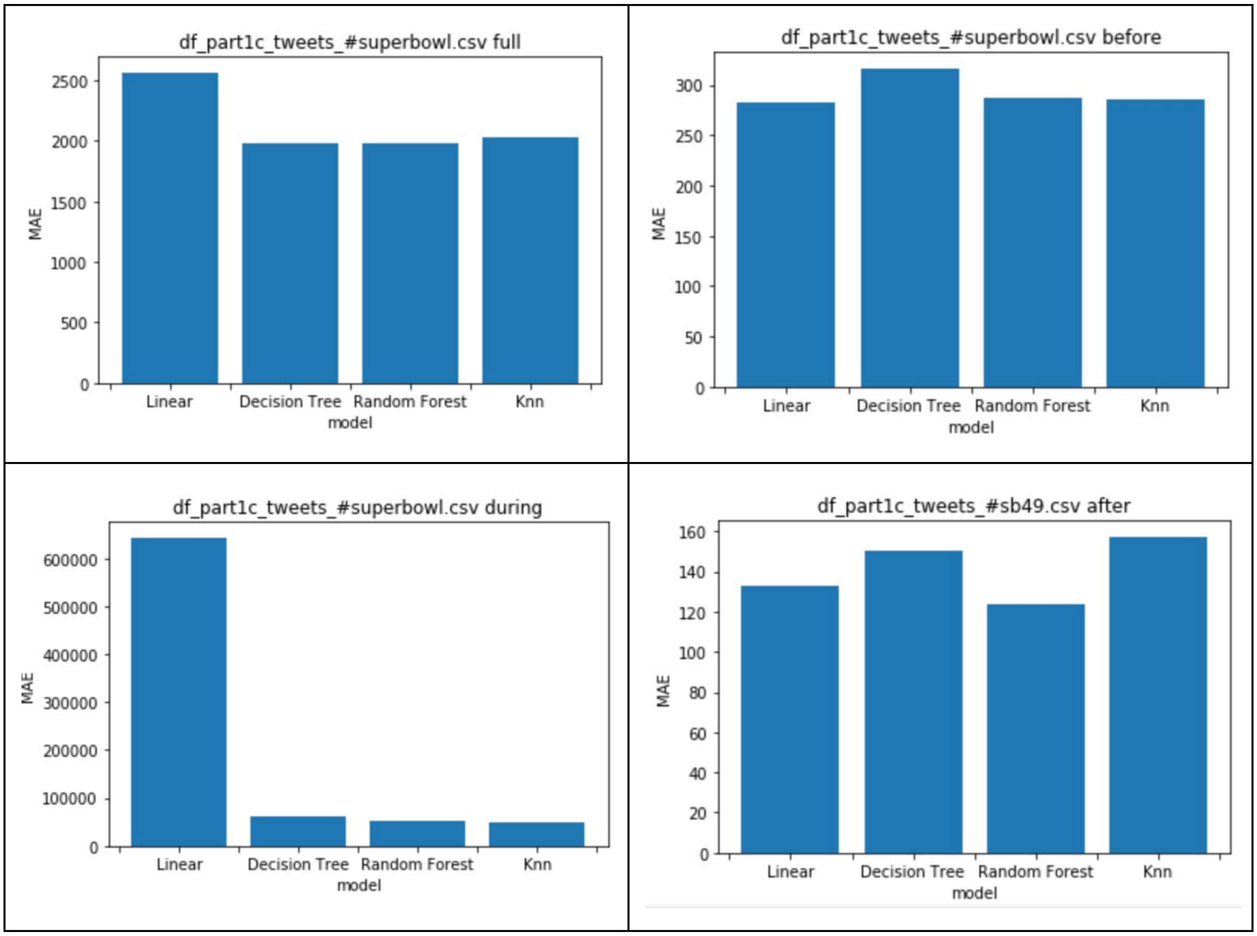
Regression, degree 2	24	91	865	6394
Polynomial Regression, degree 3	1095899702.968749	521754.5917996969	60658811.36241505	1458592.1759869447
Decision Tree	2032.0513579947954	**371.7510414064403**	55185.1	406.97719273743655
Random Forest	1980.5847574517825	**263.2831131078224**	55731.07000000001	309.2171978021978
K nearest neighbors	2031.3475043834012	285.4768921775899	50170.66000000001	**282.79131868131867**



In the #superbowl dataset, we found that on average the random forest model had the lowest MAE values. It also had the overall smallest MAE values out of all the periods.

From this data, we can conclude that the random forest model on average had the lowest MAEs. The best models were typically the random forest mode, KNN, and decision tree. They tended to have high MAE's when using the entire dataset, but when the initial split along periods were made, the MAE's significantly decreased. The linear model tended to be the next best predictor but tended to have an extremely high MAE for the period during the superbowl. The polynomial transformation typically performed the worst and this might be because the data was probably already linear.

aggregate	Full Set	Before Super bowl	During Super Bowl	After Super Bowl
Linear Model	2566.434	**283.047**	645522.783	478.898
Polynomial Regression, degree 2	915343.600	2043.845	4870383.398	27071.254
Polynomial Regression, degree 3	286835750.697	507734.387	60973704.195	1458592.177
Decision Tree	2015.515	429.430	64690.0	**403.238**
Random Forest	1989.315	295.108	47316.815	**286.421**
K nearest neighbors	2031.347	285.476	50170.66	**282.791**



For the aggregate of all the hashtags, the the random forest model performed the best on averages. The top models again were the random forest, KNN, and decision tree with the linear model closely following. The linear model wasn't quite as good as the top three because it predicted the middle period very poorly.

The majority times random forest gave better results. So we will use it for the second part of the question where we have to combine all the hashtags.

Question 2

Metrics for Full set
MAE = 1968.6978638223268
Metrics for 1st Interval
MAE = 295.53273255813946
Metrics for 2nd Interval
MAE = 51820.59

Metrics for 3rd Interval
MAE = 310.06615384615384

We can see that the model performs badly when all the hashtags are combined.

Problem 1.5

To complete this problem, we organized all of the data from all of the hashtags into one dataframe. Then we sorted according to time, taking the cumulative sum of the retweets, followers, momentum, etc. to get the features for each hour. We copied the features for each hour and the features for the following 4 hours so that each feature vector had 5 * the number of features per hour. This allows us to predict using 5 hours worth of data. We split the training data according to period. We then trained a model for each period using the best model from part 1.4 which was the random forest regressor. Using those models, we tested each sample and got the following results:

Hashtag aggregate

	Model trained on	Random Forest
sample1_period1	Period 1	204
	combined	186
sample2_period2	Period 2	139914
	combined	137591
sample3_period3	Period 3	558
	combined	448
sample4_period1	Period 1	251
	combined	283
sample5_period1	Period 1	307
	combined	309
sample6_period2	Period 2	116521
	combined	37307
sample7_period3	Period 3	142

	combined	81
sample8_period1	Period 1	12
	combined	19
sample9_period2	Period 2	66928
	combined	4584
sample10_period3	Period 3	20
	combined	19

Part 2

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

To predict the location of the author of a tweet (Washington or Massachusetts) given only the textual content of the tweet, we have tried various classifiers like Support Vector Machines, Logistic Regression and Random Forest. Additionally, we used Term Frequency - Inverse Document Frequency (TF-IDF) to extract features from the text. As using TF-IDF results in a high dimensional sparse feature space, we experimented with two dimensionality reduction techniques namely Latent Semantic Indexing and Non Negative Matrix Factorization to see how they perform on this prediction task. Here we have given the class Washington label 0 and Massachusetts label 1. The results of the techniques we tried are presented below.

Using Latent Semantic Indexing

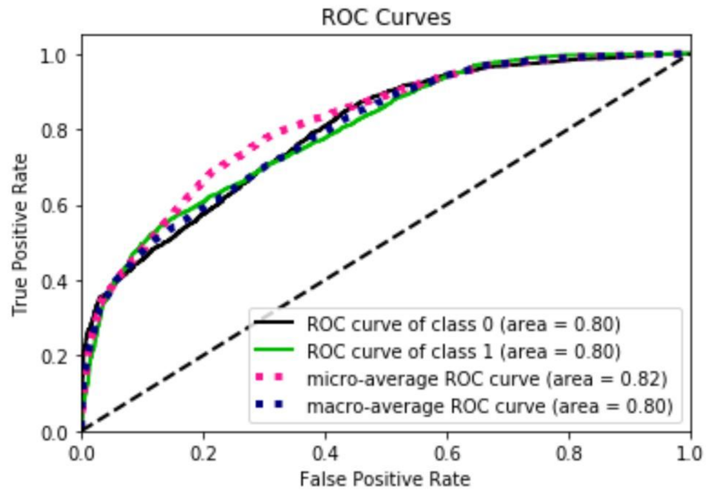
In this part, we used latent semantic indexing to perform dimensionality reduction on the tf-idf features and then trained SVM Hard Margin, SVM Soft Margin, Logistic Regression with and without regularization and Random Forest Classifiers.

SVM Hard Margin Classifier

In this we trained our model on SVM Hard Margin Classifier by setting parameter C to 1000. While the precision and accuracy were good with values around 0.8087941372418388 and 0.7321259629679686 the recall score was low. As can be seen from the classification report while the recall for Washington was high the recall for Massachusetts was low implying the classifier was poor at predicting the Massachusetts class.

Confusion Matrix

	Predicted False	Predicted True
Actual False	4203	287
Actual True	1695	1214



Accuracy	0.7321259629679686
Precision	0.8087941372418388
Recall	0.41732554142316947

Classification Report

	Precision	Recall

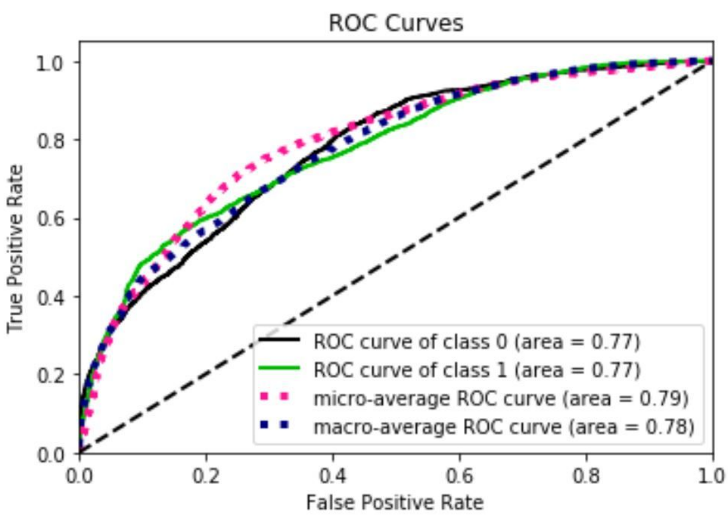
Washington	0.71	0.94
Massachusetts	0.81	0.42
Avg / Total	0.75	0.73

SVM Soft Margin Classifier

In this we trained our model on SVM Soft Margin Classifier by setting parameter C to 0.001. Even though the accuracy for this classifier was 0.6068387619948642 it predicted every instance as Washington leading to the predicted true values to be zero as can be seen from the confusion matrix. Hence as we can see in the classification report the recall of Washington is 1 while that of Massachusetts is 0 and similarly precision while predicting Massachusetts is low.

Confusion Matrix

	Predicted False	Predicted True
Actual False	4490	0
Actual True	2909	0



Accuracy	0.6068387619948642
Precision	0
Recall	0

Classification Report

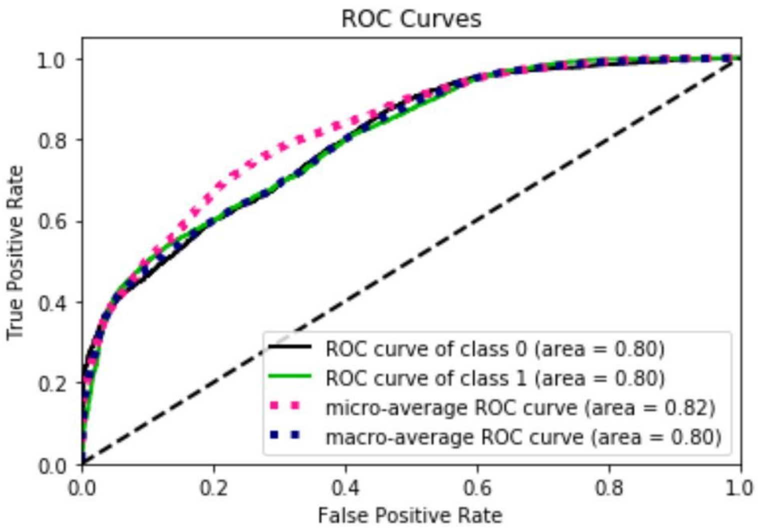
	Precision	Recall
Washington	0.61	1
Massachusetts	0	0
Avg / Total	0.37	0.61

Logistic Regression without regularization

In this we trained our model on Logistic Regression without regularization. The precision and accuracy for this classifier were around 0.7 the recall was low. As can be seen from the classification report while the recall for Washington was high the recall for Massachusetts was low implying the classifier was poor at predicting the Massachusetts class.

Confusion Matrix

	Predicted False	Predicted True
Actual False	4208	282
Actual True	1640	1269



Accuracy	0.7402351669144479
Precision	0.8181818181818182
Recall	0.43623238226194566

Classification Report

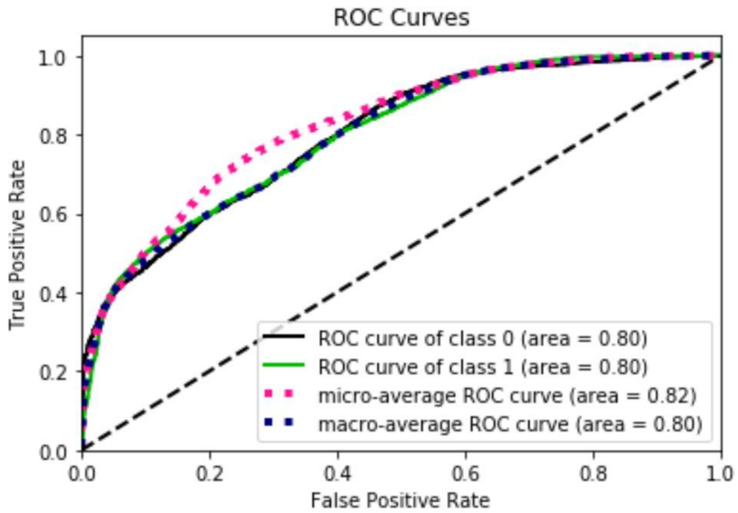
	Precision	**Recall**
Washington	0.72	0.94
Massachusetts	0.82	0.44
Avg / Total	0.76	0.74

Logistic Regression with L2 regularization

In this we trained our model on Logistic Regression with L2 regularization by setting the value of C to 1000 . The precision and accuracy for this classifier were around 0.7 the recall was low. As can be seen from the classification report while the recall for Washington was high the recall for Massachusetts was low implying the classifier was poor at predicting the Massachusetts class.

Confusion Matrix

	Predicted False	**Predicted True**
Actual False	4208	282
Actual True	1640	1269



Accuracy	0.7402351669144479
Precision	0.8181818181818182
Recall	0.43623238226194566

Classification Report

	Precision	**Recall**
Washington	0.72	0.94
Massachusetts	0.82	0.44
Avg / Total	0.76	0.74

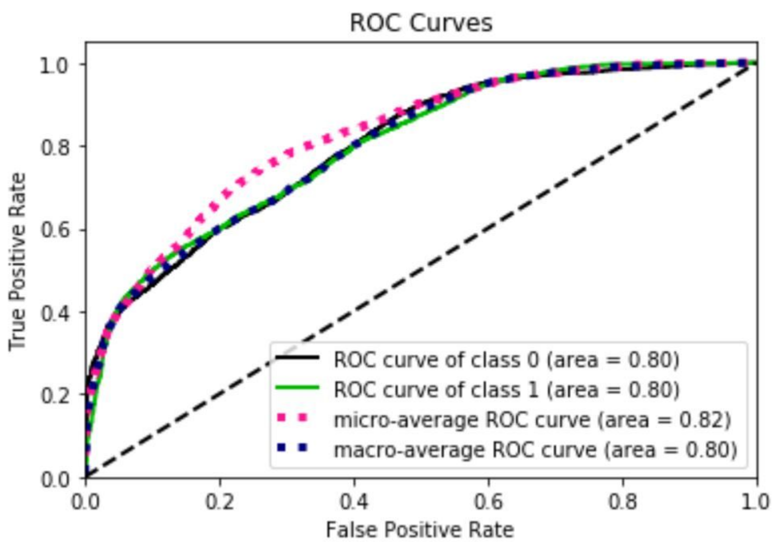
Logistic Regression with L1 regularization

In this we trained our model on Logistic Regression with L1 regularization by setting the value of C=1000. The precision and accuracy for this classifier were around 0.7 the recall was low. As can be seen from the classification report while the recall for Washington was high the recall for Massachusetts was low implying the classifier was poor at predicting the Massachusetts class.

Confusion Matrix

	Predicted False	**Predicted True**

Actual False	4209	281
Actual True	1642	1267



Accuracy	0.7401000135153399
Precision	0.8181818181818182
Recall	0.43623238226194566

Classification Report

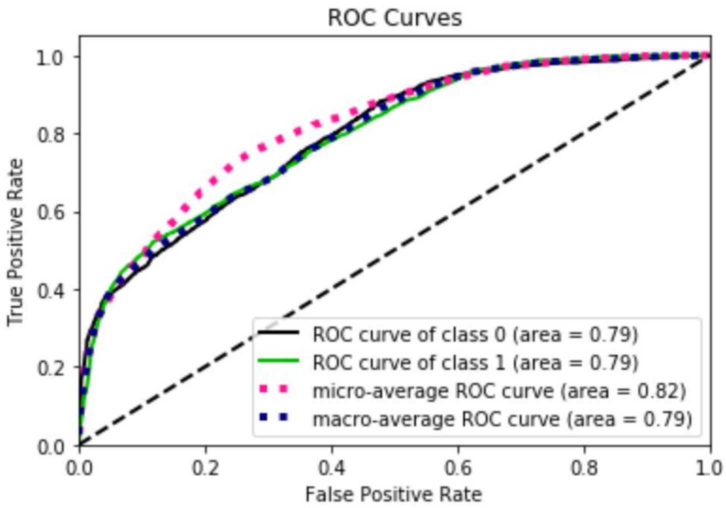
	Precision	**Recall**
Washington	0.72	0.94
Massachusetts	0.82	0.44
Avg / Total	0.76	0.74

Random Forest Classifier

In this we trained our model on Random Forest Classifier by setting the number of estimators to 150. As you can see from the results below this classifier gave good accuracy and precision and additionally it had slightly better recall for class Massachusetts with a value of 0.50 as compared to the recall values obtained in the previous classifiers.

Confusion Matrix

	Predicted False	Predicted True
Actual False	4015	475
Actual True	1434	1475



Accuracy	0.7419921611028517
Precision	0.7497441146366428
Recall	0.5036094877964936

Classification Report

	Precision	Recall
Washington	0.74	0.89
Massachusetts	0.76	0.50
Avg / Total	0.74	0.74

Using Non Negative Matrix Factorization

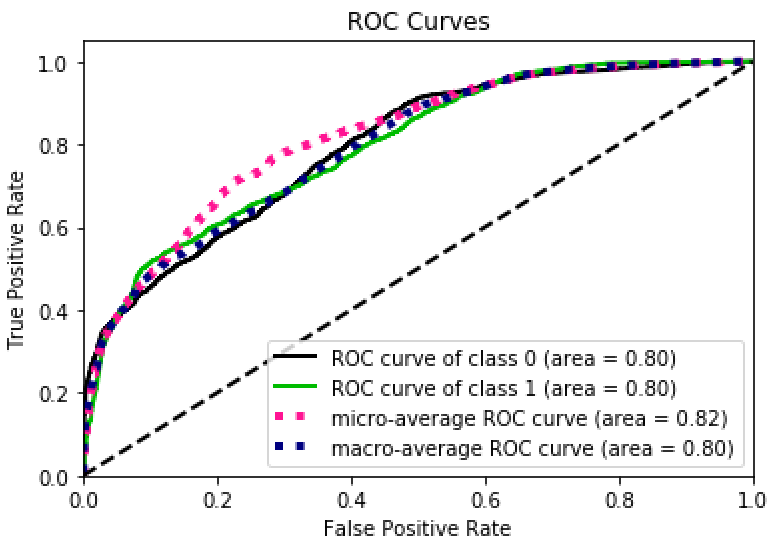
In this part, we used non negative matrix factorization to perform dimensionality reduction on the TF-IDF features and then trained SVM Hard Margin, SVM Soft Margin, Logistic Regression with and without regularization and Random Forest Classifiers. The results of each are reported below.

SVM Hard Margin Classifier

In this we trained our model on SVM Hard Margin Classifier by setting parameter C to 1000. While the precision and accuracy were good with values around 0.79827471798274718 and 0.73275628888287803 the recall score was low. As can be seen from the classification report while the recall for Washington was high the recall for Massachusetts was low implying the classifier was poor at predicting the Massachusetts class.

Confusion Matrix

	Predicted False	Predicted True
Actual False	4215	304
Actual True	1672	1203



Accuracy	0.73275628888287803
Precision	0.79827471798274718

Recall	0.41843478260869565

Classification Report

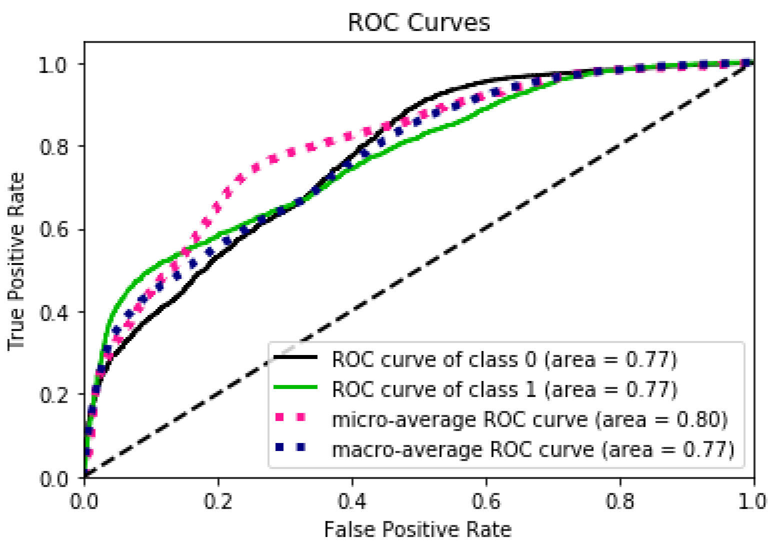
	Precision	Recall
Washington	0.72	0.93
Massachusetts	0.80	0.42
Avg / Total	0.75	0.73

SVM Soft Margin Classifier

In this we trained our model on SVM Soft Margin Classifier by setting parameter C to 0.001. Even though the accuracy for this classifier was 0.61117121990803358 it predicted every instance as Washington leading to the predicted true values to be zero as can be seen from the confusion matrix. Hence, as we can see in the classification report the recall of Washington is 1 while that of Massachusetts is 0 and similarly precision while predicting Massachusetts is low.

Confusion Matrix

	Predicted False	Predicted True
Actual False	4519	0
Actual True	2875	0



Accuracy	0.61117121990803358
Precision	0.0
Recall	0.0

Classification Report

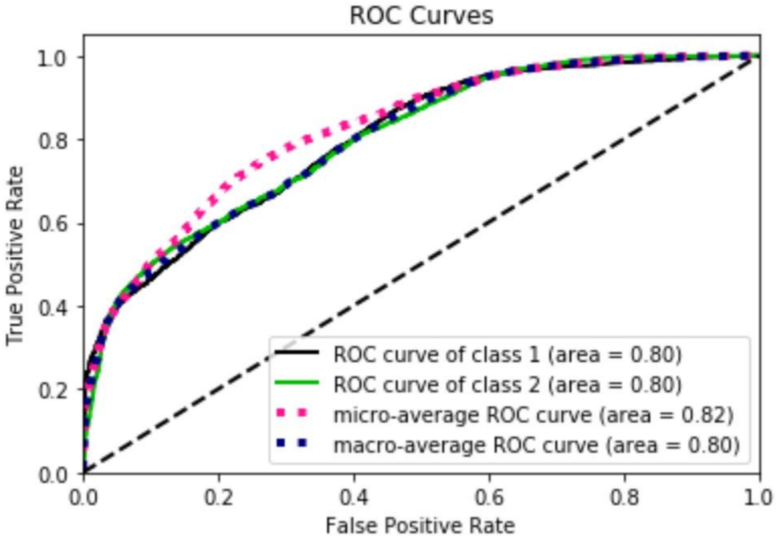
	Precision	Recall
Washington	0.61	1.00
Massachusetts	0.00	0.00
Avg / Total	0.37	0.61

Logistic Regression without regularization

In this we trained our model on Logistic Regression without regularization. The precision and accuracy for this classifier were around 0.83 and 0.74 respectively and the recall was low. As can be seen from the classification report while the recall for Washington was high the recall for Massachusetts was low implying the classifier was poor at predicting the Massachusetts class.

Confusion Matrix

	Predicted False	Predicted True
Actual False	4274	245
Actual True	1662	1213



Accuracy	0.74208817960508522
Precision	0.83196159122085045
Recall	0.42191304347826086

Classification Report

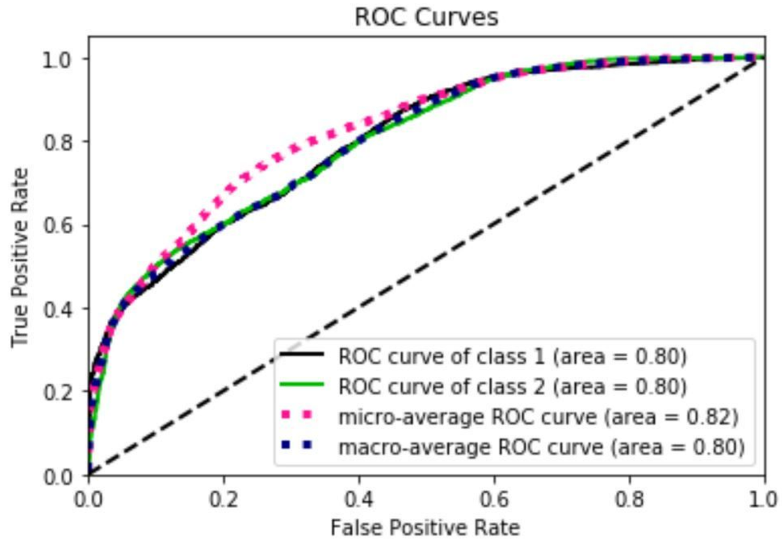
	Precision	**Recall**
Washington	0.72	0.95
Massachusetts	0.83	0.42
Avg / Total	0.76	0.74

Logistic Regression with L2 regularization

In this we trained our model on Logistic Regression with L2 regularization by setting the value of C to 1000 . The precision and accuracy for this classifier were around 0.83 and 0.74 respectively and the recall was low. As can be seen from the classification report while the recall for Washington was high the recall for Massachusetts was low implying the classifier was poor at predicting the Massachusetts class.

Confusion Matrix

	Predicted False	Predicted True
Actual False	4274	245
Actual True	1663	1212



Accuracy	0.74195293481200975
Precision	0.8318462594371997
Recall	0.42156521739130437

Classification Report

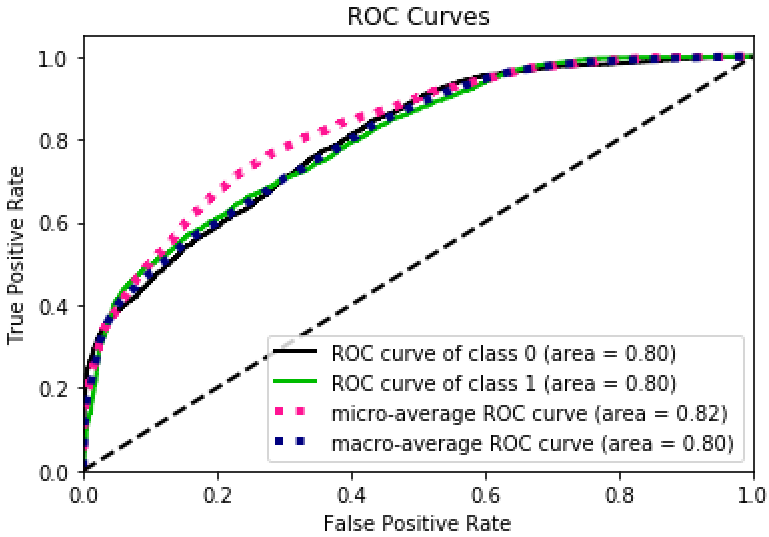
	Precision	Recall
Washington	0.72	0.95
Massachusetts	0.83	0.42
Avg / Total	0.76	0.74

Logistic Regression with L1 regularization

In this we trained our model on Logistic Regression with L1 regularization by setting the value of C=1000. The precision and accuracy for this classifier were around 0.83 and 0.74 respectively and the recall was low. As can be seen from the classification report while the recall for Washington was high the recall for Massachusetts was low implying the classifier was poor at predicting the Massachusetts class.

Confusion Matrix

	Predicted False	Predicted True
Actual False	4274	245
Actual True	1661	1214



Accuracy	0.74222342439816069
Precision	0.83207676490747084
Recall	0.42226086956521741

Classification Report

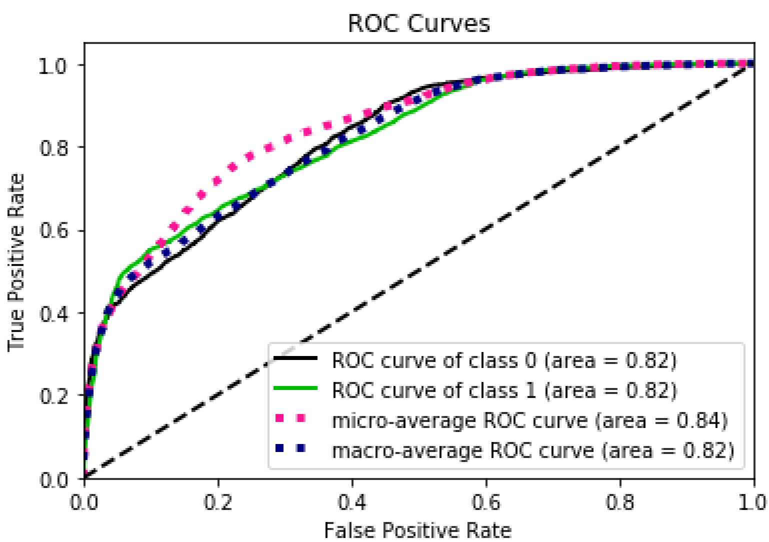
	Precision	Recall
Washington	0.72	0.95
Massachusetts	0.83	0.42
Avg / Total	0.76	0.74

Random Forest Classifier

In this we trained our model on Random Forest Classifier by setting the number of estimators to 150. As you can see from the results below this classifier gave good accuracy and precision and additionally it had slightly better recall for class Massachusetts with a value of 0.53 as compared to the recall values obtained in the previous classifiers.

Confusion Matrix

	Predicted False	Predicted True
Actual False	4111	408
Actual True	1339	1536



Accuracy	0.7637273464971599
Precision	0.79012345679012341
Recall	0.53426086956521734

Classification Report

	Precision	Recall
Washington	0.75	0.91
Massachusetts	0.79	0.53
Avg / Total	0.77	0.76

Part 3: Define Your Own Project

In this part of the project we chose to perform sentiment analysis on the twitter dataset. Sentiment analysis is the process of determining whether a piece of text is positive, negative or neutral. It is also known as opinion mining, deriving the opinion or attitude of the speaker or in our case the author of the tweet. A common use case of this analysis is to discover how people feel about a particular topic.

Using sentiment analysis, we look forward to analyze two things mainly.

1. Analysis of brands - The Super Bowl games are among the United States most watched television broadcast, with the viewership of Super Bowl XLIX estimated to be viewed by 114.4 million viewers. The game's extremely high viewership and wide demographics results in the television broadcast featuring many high profile television commercials, informally known as Super Bowl ads. Advertisers use these commercials as a means of building awareness for their products among this wide audience, while also trying to generate a buzz around the ads so they receive additional exposure. It is also not uncommon to find many viewers watching the game just to see the commercials as a result of Super Bowl commercials being a cultural phenomenon of their own alongside the game itself. A number of major brands, including Budweiser, Doritos, Microsoft, Fiat have been known for making repeated appearances during Super Bowl. So in this part we try to analyze the sentiments of people tweeting on twitter once they watch these ads. This analysis could be useful for the advertisers to get an understanding of viewers reactions to the commercials. Given the price of commercials to be extremely high, it provides the advertisers a good estimate of whether they were able to achieve their outcome of getting positive reactions for their products from the viewers.

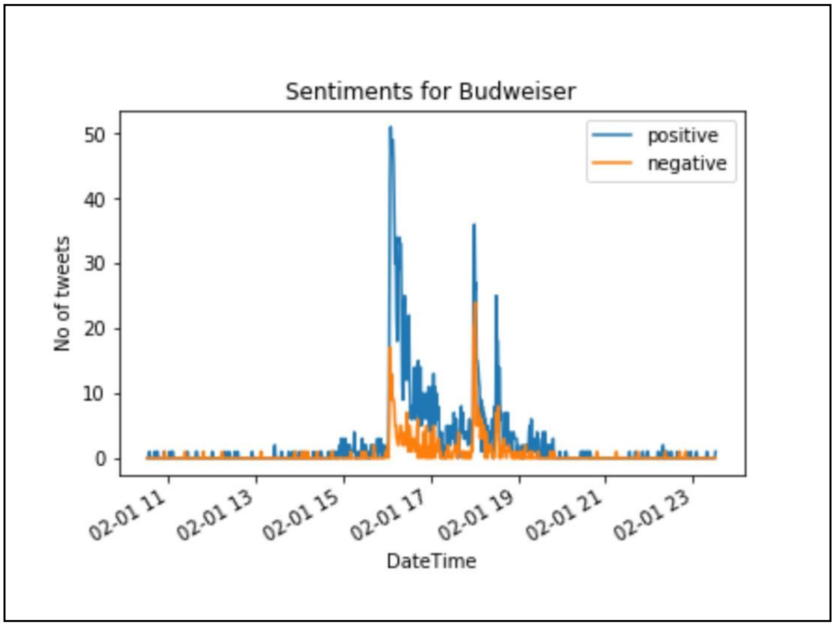
2. Analysis of game day tweets - As with any major sporting event, the Super Bowl creates an incredible amount of hype. There are times when the viewers are drawn to the edge of their seats as viewers go through an emotional rollercoaster during the course of the game. This results in a buzz on social media. All of this social chatter around the Super Bowl results in a lot of data being available to perform a study and analyze the emotions and reactions of the fans to the event. In this problem we try to get an insight of how fans of both the Patriots and Hawks felt at any given point throughout the event.

To perform sentiment analysis we use the TextBlob package for python. It is a convenient way to do a lot of Natural Language Processing (NLP) tasks. We are particularly interested in the sentiment analyzer module. This analyzer uses a pattern based approach for performing sentiment analysis by using a dictionary lookup with a set of rules. It uses the pattern.en module from CLIPS which comes bundled with a lexicon of adjectives which frequently occur in product reviews and are annotated with sentiment polarity scores.

Sentiment Analysis of Brands

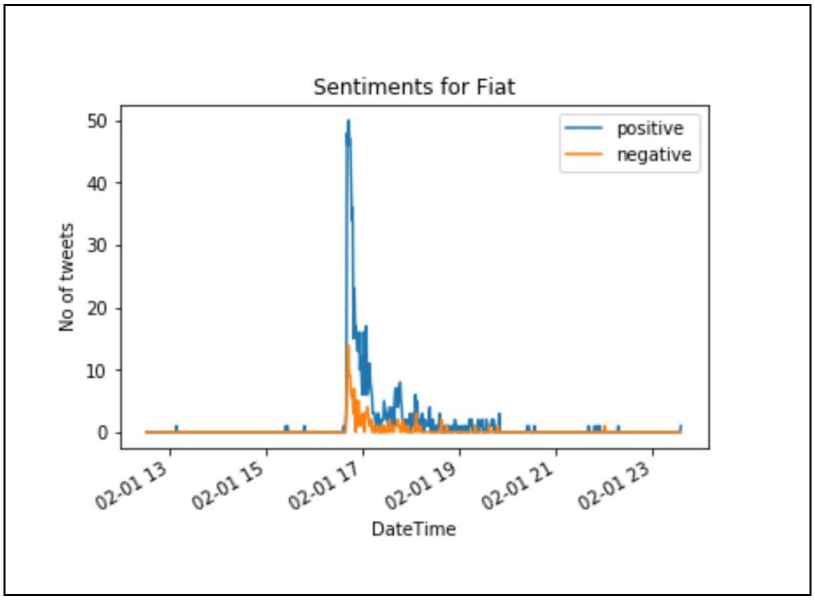
Sentiments for Budweiser

As can be seen from the graph for budweiser overall the positive sentiment is more for their ads. Additionally we can see that the peaks in sentiments were observed between time 16:05 and 16:10 pm once and then again between 17:59 to 18:07 and 18:30 to 18:35 pm. Budweiser ran three ads during the game and the three rises in the sentiments explains that the ads were well received on the twitter with overall positive sentiments.



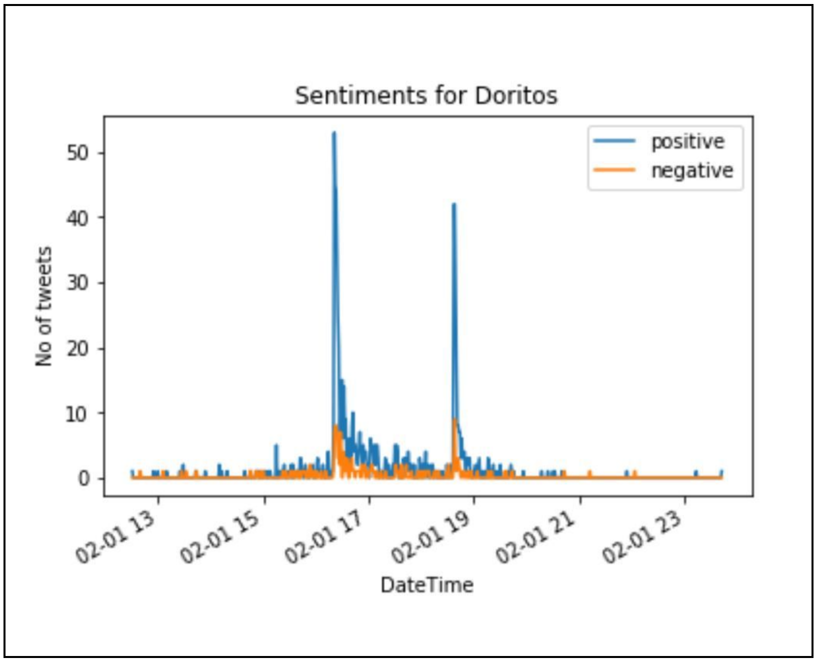
Sentiments for Fiat

As can be seen from the graph for fiat overall the positive sentiment is more for their ads. Additionally we can see that the peaks in sentiments were observed between time 16:40 to 16:50 pm once. While the The fiat commercial was played during the two minute warning break after second quarter and hence the peak around that time indicates that it was received by the audience positively.



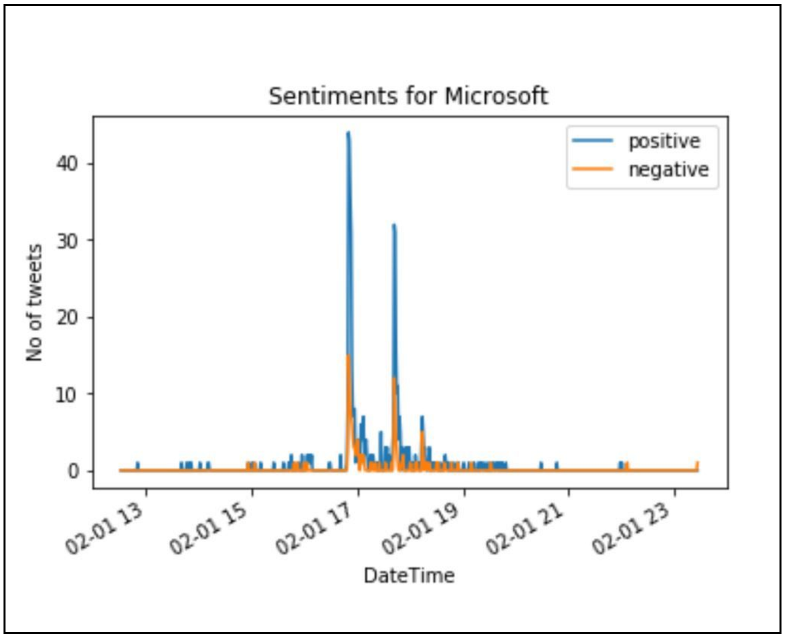
Sentiments for Doritos

As can be seen from the graph for doritos overall the positive sentiment is more for their ads. Additionally we can see that the peaks in sentiments were observed between time 16:21 to 16:30 pm and 18:37 to 18:40 pm . Doritos played the crash the Super Bowl commercial in 2015 too and as this commercial is created by inviting the participants to submit content online, we can see that this ad has very less negative sentiments.



Sentiments for Microsoft

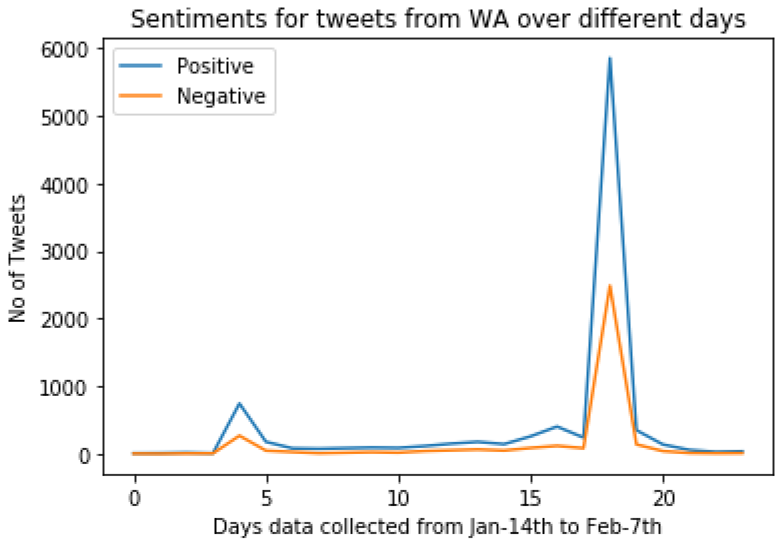
As can be seen from the graph for microsoft overall the positive sentiment is more for their ads. Additionally we can see that the peaks in sentiments were observed between time 16:03 to 16:10 pm with the sentiments continuing for around 10 - 15 mins based on the analysis of tweets and another peak at 17:59 to 18:07 pm . Microsoft released two ads during the 2015 Super Bowl and we can see from the sentiments that they were well received based on the peaks in sentiments around that time.

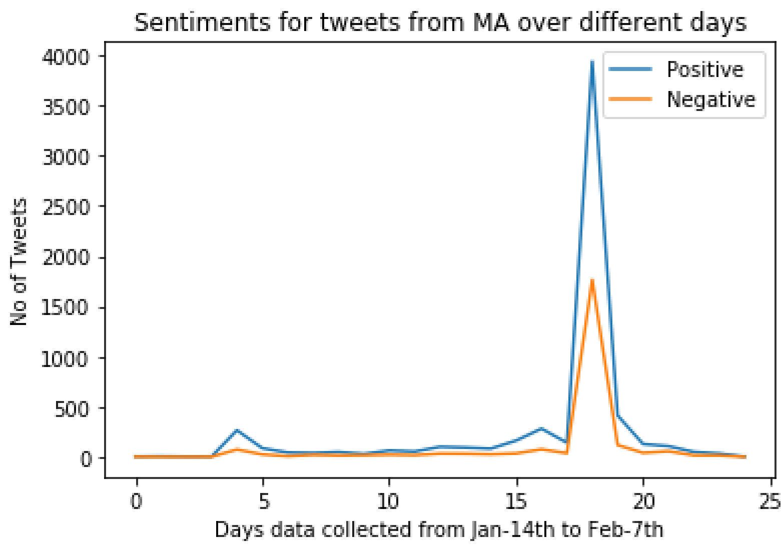


Overall based on our analysis of the twitter data for each of the four brands mentioned above, we could see that ads of each of them were well received. We could also observe some correlation between the time at which the ads were released and the change in sentiments regarding the brands on twitter data. This correlation is expected because on seeing the ad, twitter users usually post their reactions as to how much they liked or disliked the ads. This analysis of user reactions for brands could be very useful for the companies as they will be able see if they were able to promote their brand and additionally they could use that experience to further improve their brand image by improving their commercials.

Sentiment Analysis of Game Day Tweets

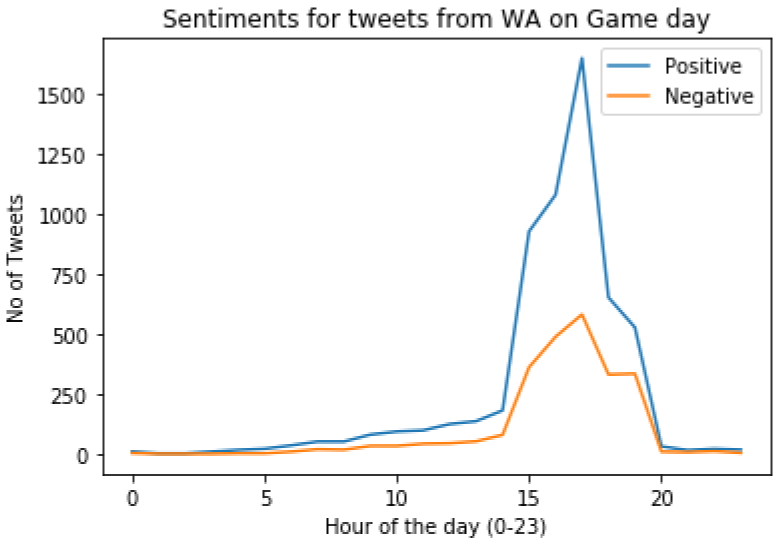
As we look at sentiments of viewers on game day it would not be a bad idea to have a general overview of people's sentiments during the build of the event and after the event. It gives us a good idea of whether a certain emotion is prevalent. We analyze the tweets that mention the superbowl hashtag and classify them based on the location of the tweet being either Washington or Massachusetts and then perform sentiment analysis on these two sets independently. We chose Washington and Massachusetts respectively based on the origin of the two finalist teams.

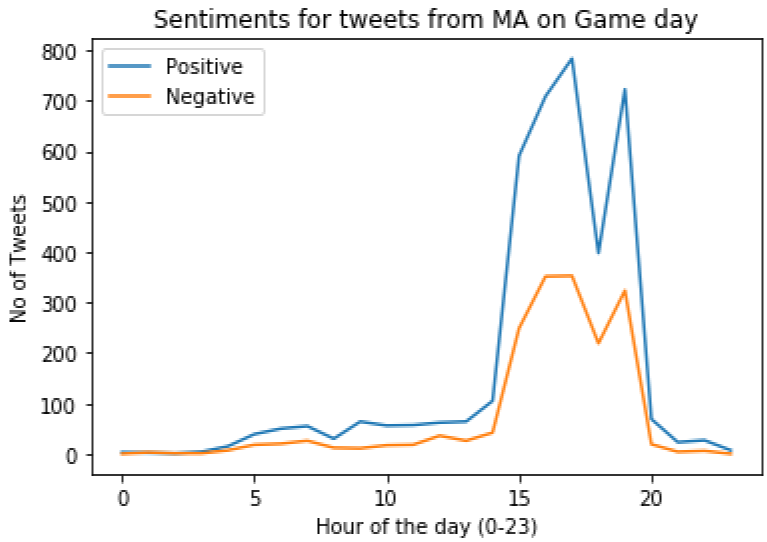




The above two graphs represent sentiments of tweets over the range of the days from January 14th to February 7th with 0th day representing January 14th. We observe that there is peak on one day, that is the game day and the rest of the days have relatively less or no twitter activity with the exception of pre-game day and post-game day where we observe a minimal spike. Looking at the sentiments of the tweets the positive sentiment tweets dominate the negative ones in both Washington and Massachusetts with WA state having a higher proportion tweets.

In the next section, we look in detail at the twitter activity pertaining to the game day and how the sentiment of viewers tweet tend to vary during the game and a spread over the whole day.





In the above graphs we see that again in general there is a higher proportion of positive sentiment tweets compared to negative ones in both the graphs. During the course of the game the positive sentiment tweets completely dominate the negative tweets but there is an interesting observation where we see a sudden dip in the positive tweets in case of Washington and it almost approaches the closer to the number of negative tweets. The explanation of this behavior may be attributed to an event in the game that would have likely favored Patriots over the Hawks. The time stamp leading to this observation is also around the time of the ending of the game and we know that in the game the Patriots defeated the Hawks this could be the reason for the sudden dip in the positive tweets for Washington and eventually the twitter activity reduces with almost equal proportion of positive and negative tweets post completion of the game.