**Project 5: Application - Twitter Data**

Prof. Vwani Roychowdhury

ECE 219 Winter 2020

UCLA, Department of ECE

Siyuan Peng 805426447 Haonan Huang 605322629

Yinghan Cui 005430212 Yunzheng Zhu 505231998

**Introduction**

In social networks, it is useful to predict future popularity of a subject or event. In this project, we are going to predict on Twitter, a good platform to perform such analysis. By knowing current (and previous) tweet activity for a hashtag, can we predict its tweet activity in the future or can we predict if it will become more popular. The data we are using is collected by querying popular hashtags related to the 2015 Super Bowl spanning a period starting from 2 weeks before the game to a week after the game. We will use the training data to create the model, and test data to make the predictions. Then, we will use the model to predict the number of tweets containing the hashtags posted within one hour immediately following the given time window.

**Part 1: Popularity Prediction**

1. **First look at the data**

The data consists of 6 text files, and the corresponding tweet hashtags are #gohawks, #gopatriots, #nfl, #patriots, #sb49, and #superbowl, as indicated in the filenames.

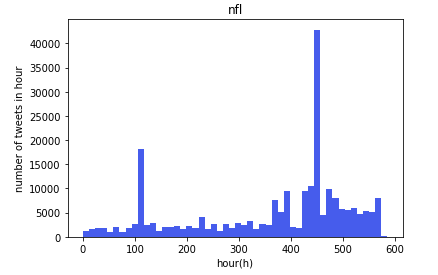
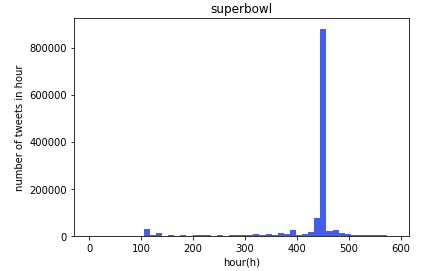
**Question 1 - Report the following statistics for each hashtag**

The basic statistics for each hashtag are shown in Table 1 below.

|  |  |  |  |
| --- | --- | --- | --- |
| Hashtag | Average # of tweets per hour | Average # of followers of users posting the tweets per tweet | Average # of retweets per tweet |
| #gohawks | 292.598615916955 | 2217.9237355281984 | 2.0132093991319877 |
| #gopatriots | 40.888695652173915 | 1427.2526051635405 | 1.4081919101697078 |
| #nfl | 397.64846416382255 | 4662.37544523693 | 1.5344602655543254 |
| #patriots | 751.9129692832764 | 3280.4635616550277 | 1.7852871288476946 |
| #sb49 | 1269.0255972696245 | 10374.160292019487 | 2.52713444111402 |
| #superbowl | 2071.353242320819 | 8814.96799424623 | 2.3911895819207736 |

Table 1. Basic statistics for each hashtag, including “average number of tweets per hour”, “average number of followers of users posting the tweets per tweet”, and “average number of retweets per tweet”.

**Question 2 - Plot “number of tweets in hour” over time for #SuperBowl and #NFL**

****Figure 1. “Number of tweets in hour” over time for #SuperBowl and #NFL.

As we can see in Fig. 1, both reach peak at around the 450th hour. It is exactly the time that the Superbowl occurs. For the NFL, there is a peak at the 100th hour as well, which might be another important game taking place (such as championships).

1. **Linear Regression**

In this section, we apply several features to predict “number of tweets in the next hour” for each hashtag. The first 5 features we use and the corresponding labels are shown below:

tweets: number of tweets

retweets: total number of retweets

followers sum: sum of the number of followers of the users posting the hashtag

followers max: maximum number of followers of the users posting the hashtag

hour of day: time of the day (24 values represent hours of the day)

**Question 3 - Report model’s Mean Squared Error (MSE) and R-squared measure.**

We train each linear regression model using the first five features to predict the “number of tweets in the next hour”. The MSE and R-squared measure are shown in Table 2.

|  |  |  |
| --- | --- | --- |
| Hashtag | MSE | R-squared |
| #gohawks | 759843.8445222704 | 0.4764399847953502 |
| #gopatriots | 27588.58568971645 | 0.6293383566038233 |
| #nfl | 270401.91406034445 | 0.5707688538937594 |
| #patriots | 5189695.980567309 | 0.6684079890987789 |
| #sb49 | 16107134.315916982 | 0.804588854407114 |
| #superbowl | 52573154.30093936 | 0.7998348436989455 |

Table 2. MSE and R-squared measure for each hashtag model.

R-squared measures the proportion of the variance for the dependent variable “number of tweets in the next hour”. As we can see from Table 2, it is relatively high for #sb49 and #superbowl, but moderate for the rest. MSE are relatively high for all models since it is bad for linear regression models to predict all the peaks of the hashtags well.

The fitted values v.s. True values plots and OLS regression results for each hashtag model are shown below in Fig. 2 and Fig. 3 below.

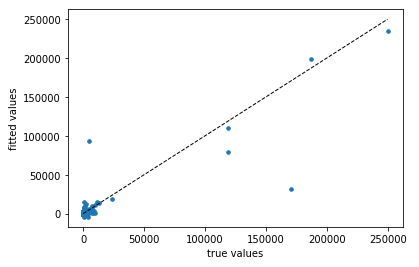
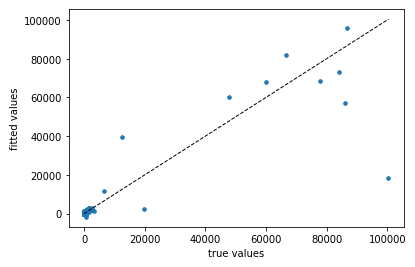
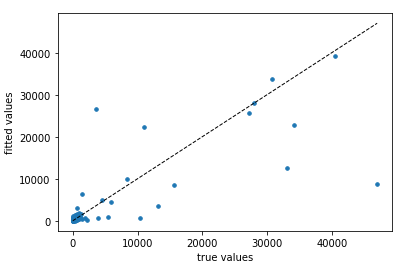
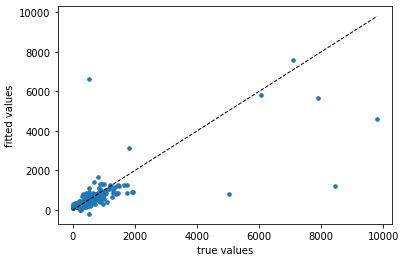
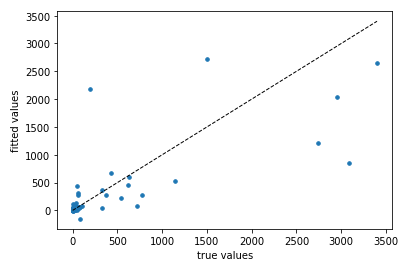
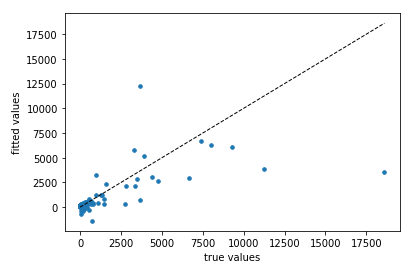


Figure 2. The fitted values v.s. True values plots for six hashtag model in order (#gohawks, #gopatriots, #nfl, #patriots, #sb49, #superbowl from left to right, then top to bottom).

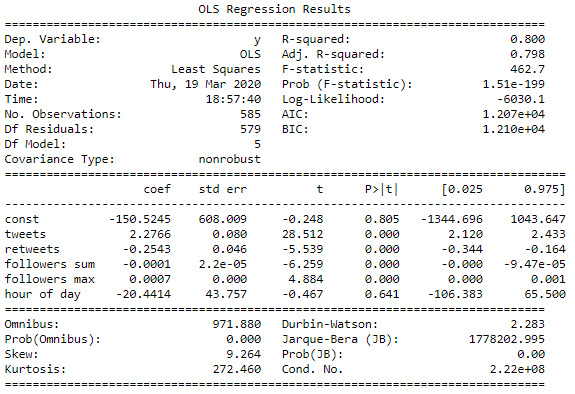
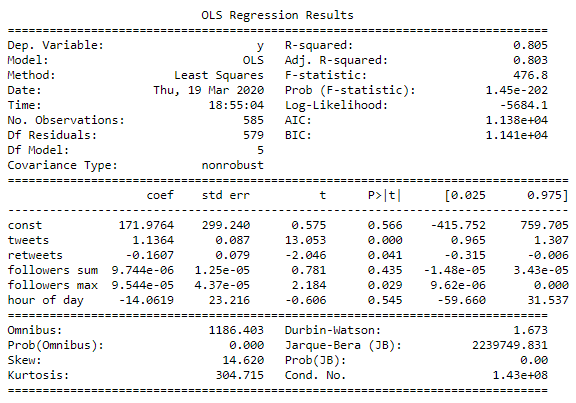
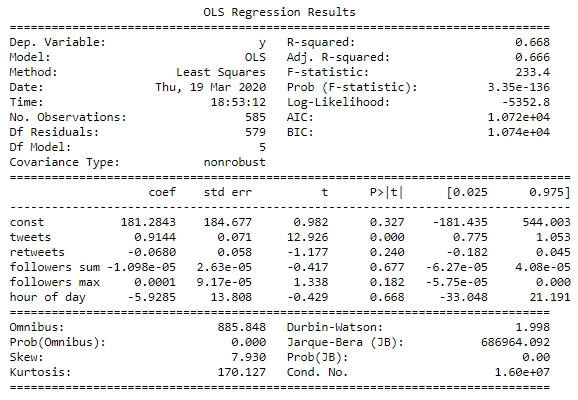
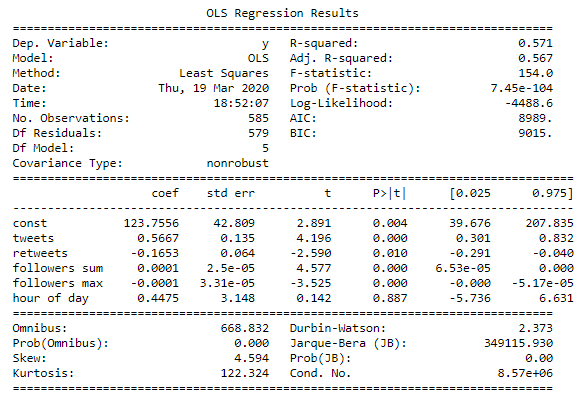
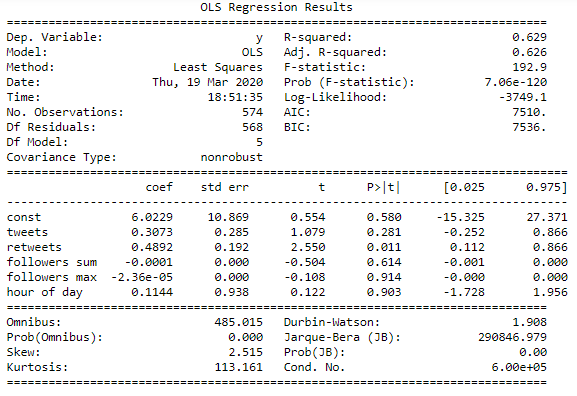
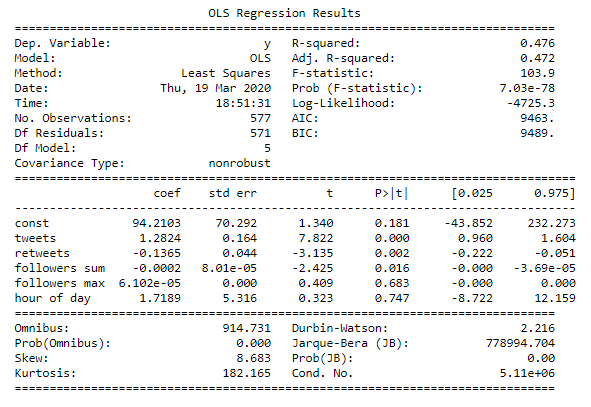


Figure 3. The fitted values v.s. True values OLS regression results for six hashtag model in order (#gohawks, #gopatriots, #nfl, #patriots, #sb49, #superbowl from left to right, then top to bottom).

As we can see from Fig. 2 and Fig. 3, for 5% significance level (significant: < 5%; insignificant: > 5%), “hour of day” is not significant for all six models. The three hashtags, #nfl, #sb49, and # superbowl, have all other four features with significance. The rest three team hashtags are having most of the features insignificant. To conclude, the five features are better for the general hashtags, but not the team hashtags.

1. **Feature Analysis**

In this section, we add six new features for predicting the “number of tweets in the next hour”, as shown below (hour of day is removed since it is insignificant to all the models):

mentioned: number of mentioned user in this tweet

media: number of media url attached

active: active index defined by year

author: author name

favourites\_count: user’s favourites count

title: length of this tweet’s title

**Question 4 - Design a regression model**

We train each linear regression model using the previous five features and the six new features to predict the “number of tweets in the next hour”. The MSE and R-squared measure are shown in Table 3.

|  |  |  |
| --- | --- | --- |
| Hashtag | MSE | R-squared |
| #gohawks | 536339.8552752357 | 0.6304423536400289 |
| #gopatriots | 8459.428739863255 | 0.8863448168682551 |
| #nfl | 189254.8008673475 | 0.6995803252181446 |
| #patriots | 3701906.9318352095 | 0.7634692343650118 |
| #sb49 | 14222241.832739882 | 0.8274562988719606 |
| #superbowl | 32004965.335808694 | 0.8781454342233084 |

Table 3. MSE and R-squared measure for each hashtag model.

R-squared measure in Table 3 increases for all, but it will definitely increase with the increment of features. Thus, R-squared cannot be used as the model selection tool here. The decrease of MSE indicates that with 10 features is better than with 5 features in predicting accuracy. However, the value are too large for prediction.

The fitted values v.s. True values plots and OLS regression results for each hashtag model are shown below in Fig. 4 and Fig. 5 below.

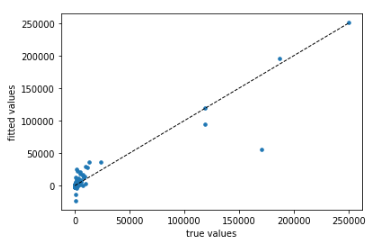
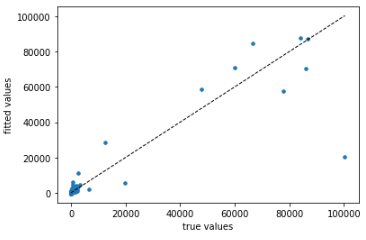
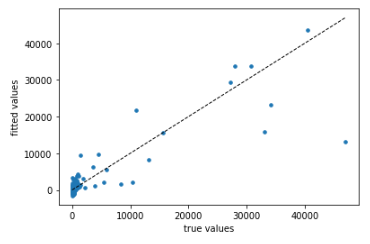
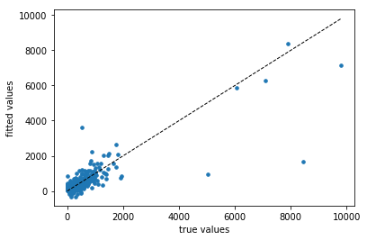
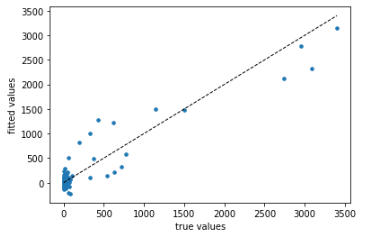
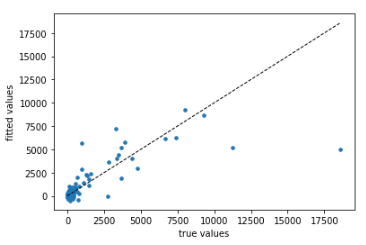


Figure 4. The fitted values v.s. True values plots for six hashtag model in order (#gohawks, #gopatriots, #nfl, #patriots, #sb49, #superbowl from left to right, then top to bottom).

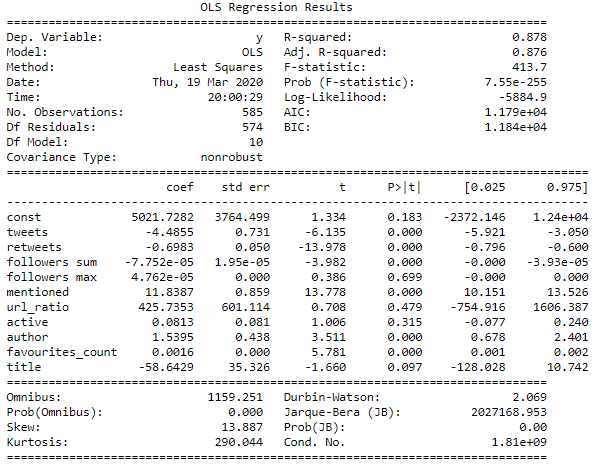
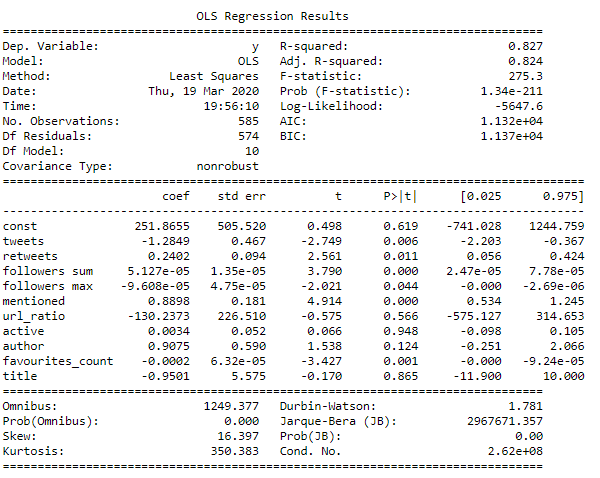
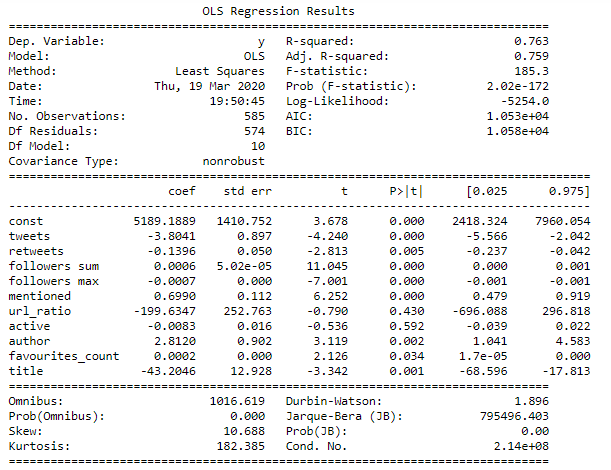
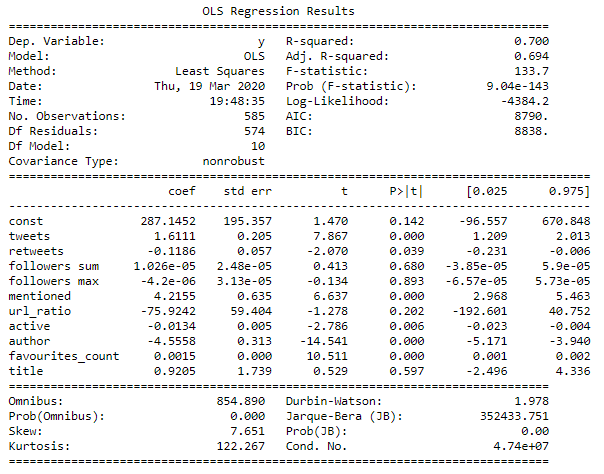
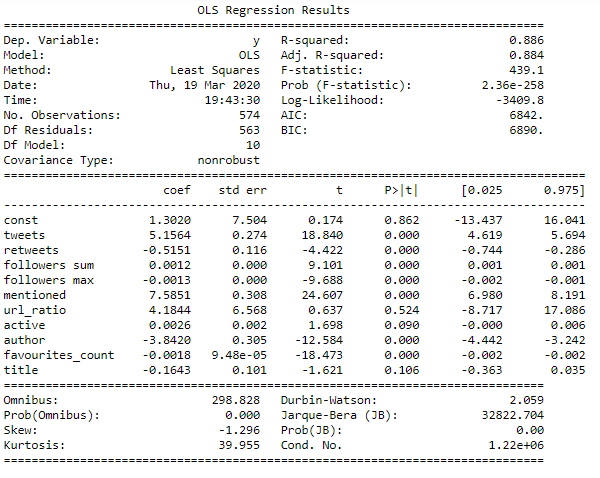
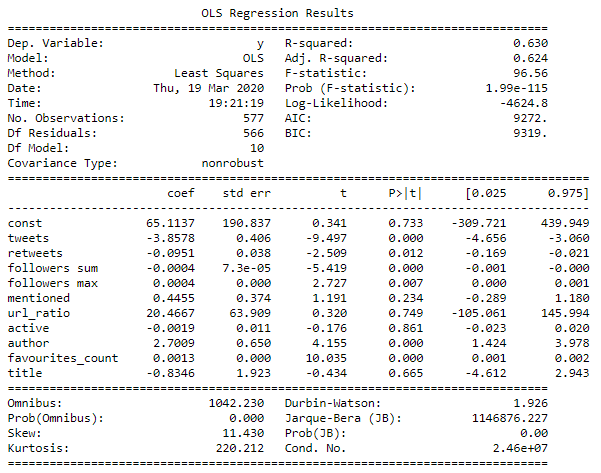


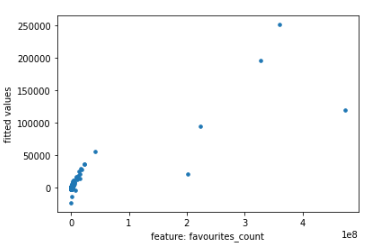
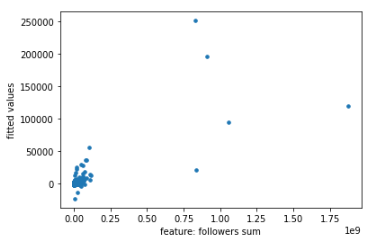
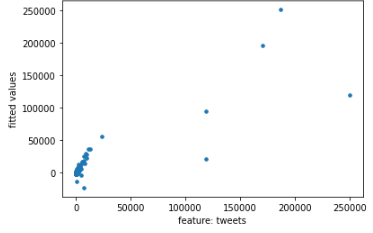
Figure 5. The fitted values v.s. True values OLS regression results for six hashtag model in order (#gohawks, #gopatriots, #nfl, #patriots, #sb49, #superbowl from left to right, then top to bottom).

As we can see from Fig. 4 and Fig. 5, for #nfl, #sb49, and #superbowl, the six new features are not significant. However, the new features are significant for the other three hashtags (#gohawks, #gopatriots, and #patriots), with a relatively high improvement in the significance level.

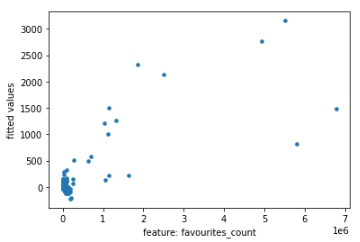
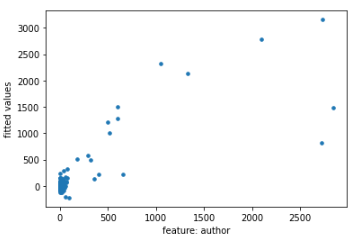
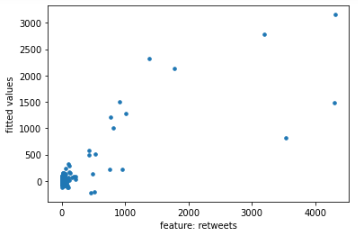
**Question 5 - Plot of predictant v.s. value of that feature for each of top 3 features**

In this section, we select the top 3 features to make the plot of predictant v.s. Value of that feature. Top 3 features are defined as the smallest p-value or the largest t-statistics. The predictant v.s. Top 3 features plots for each hashtag are shown in Fig. 6 below.

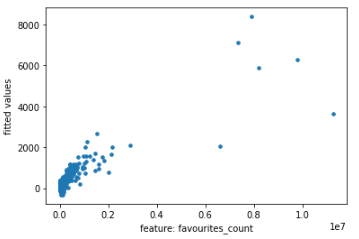
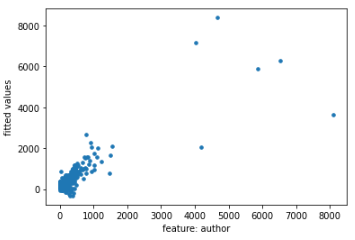
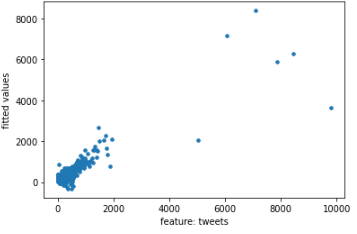
#gohawks: (Top 3 features: tweets, followers sum, favourites\_count)



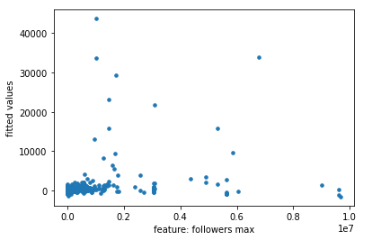
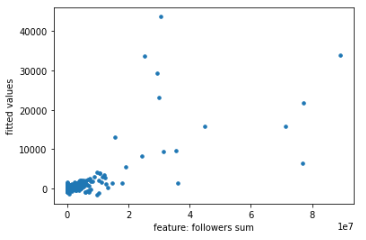
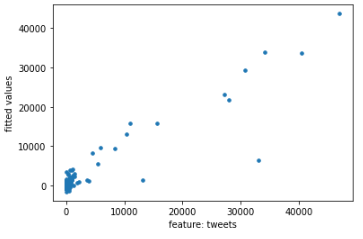
#gopatriots: (Top 3 features: retweet, author, favourites\_count)



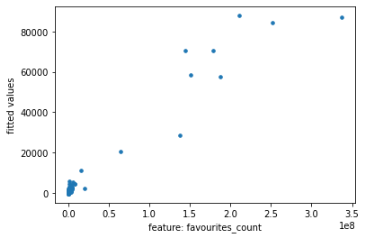
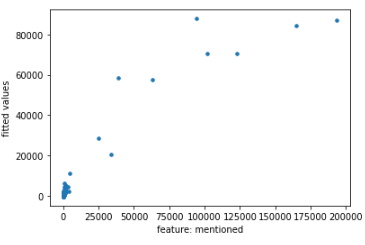
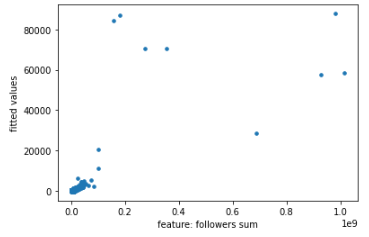
#nfl: (Top 3 features: tweets, author, favourites\_count)



#patriots: (Top 3 features: tweets, followers sum, followers max)



#sb49: (Top 3 features: followers sum, mentioned, favourites\_count)



#superbowl: (Top 3 features: tweets, retweets, mentioned)

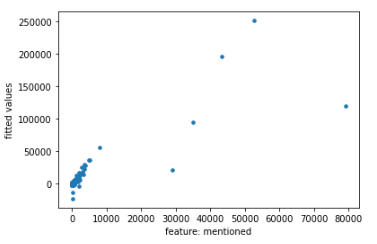
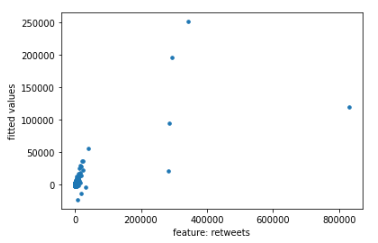
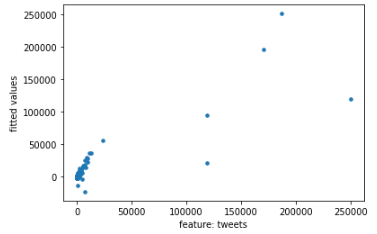


Figure 6. Scatter plots of predictant v.s. Top 3 features.

All 18 plots are having features and the fitted values positively correlated, but the regression coefficients are not fully positive. For #gohawks, the coefficient for “follower sum” is not positive because the plot measures the correlation between the variables fitted value and “followers sum”. The original correlation could be positive, but after controlling other features, the correlation becomes negative, which is related to the omitted variable bias problem.

1. **Piecewise Linear Regression**

**Question 6 - Report the MSE and R-squared score (3 regression models)**

For each hashtag, we train three regression models (when the hashtags haven’t become very active; their active period; and after they pass their high-activity time), one for each of these time periods.

The MSE and R-squared score for each case are shown in Table 4.

|  |  |  |
| --- | --- | --- |
| Hashtag | MSE | R-squared |
| #gohawks - before active | 1749.7512241246761 | 0.36638992462703057 |
| #gohawks - active | 16.310436890211516 | 0.9049370158124668 |
| #gohawks - after active | 1436.0436332460893 | 0.7302717479776322 |
| #gopatriots - before active | 159.77409040135905 | 0.12237779987801423 |
| #gopatriots - active | 3.165177855119004 | 0.8390871767246021 |
| #gopatriots - after active | 264.52418078314435 | 0.8794219548442211 |
| #nfl - before active | 221.8623394249195 | 0.5882669837719856 |
| #nfl - active | 9.482129404993081 | 0.8785199038474725 |
| #nfl - after active | 715.8543846504308 | 0.729333446542455 |
| #patriots - before active | 1813.7397130839088 | -0.30930773920851773 |
| #patriots - active | 499.7888384384575 | 0.835299896199698 |
| #patriots - after active | 9309.79244649938 | 0.19532579589260313 |
| #sb49 - before active | 62.11676241814218 | 0.8453967337775798 |
| #sb49 - active | 823.6397513656524 | 0.8744594773706587 |
| #sb49 - after active | 1129.910101201265 | 0.9786081345954118 |
| #superbowl - before active | 1412.7648120341032 | -0.668053205347308 |
| #superbowl - active | 34.51745313735777 | 0.93996522291428 |
| #superbowl - after active | 26602.64027637441 | 0.7960786289659055 |

Table 4. MSE and R-squared score for each case in three time periods.

**Question 7 - Compare with models trained for individual hashtags**

We aggregate the data of all hashtags, and train 3 models (when the hashtags haven’t become very active; their active period; and after they pass their high-activity time) to predict the number of tweets in the next time window on the aggregated data and perform the same evaluations on our combined model.

The MSE and R-squared score for each case are shown in Table 5.

|  |  |  |
| --- | --- | --- |
| Period | MSE | R-squared |
| - before active | 1191.2195695237467 | 0.28565661443676094 |
| - active | 87.39908530048552 | 0.8413176706991644 |
| - after active | 31465.931673357518 | 0.7504146144787548 |

Table 5. MSE and R-squared score for each case in three time periods.

The results for the aggregate data have relatively lower MSE for active period and higher MSE for periods before and after the active time.

1. **Nonlinear Regression**

**Ensemble methods:**

**Question 8 - Find the best parameter set and analyze the result**

In this part, we use RandomForestRegressor and GradientBoostingRegressor from sklearn as two examples of ensemble regressors. We still use the aggregated data in this part. We use the grid search to find the best parameter set for these two methods respectively.

|  |  |  |
| --- | --- | --- |
|  | Random Forest | Gradient Boosting |
| max depth | 10 | 20 |
| max features | sqrt | sqrt |
| min samples leaf | 1 | 1 |
| min samples split | 2 | 2 |
| n estimators | 1800 | 200 |
| test MSE | 4448.778163974289 | 4549.305838591513 |

Table 6. Grid search result for Random Forest and Gradient Boosting

We can see that the test errors from cross-validation do not look good. This might be caused by over-fitting, because the train MSEs are small but test MSEs are huge.

**Question 9 - Compare the best estimator**

We compare the best estimator you found in the grid search with OLS on the entire dataset. OLS results on the entire dataset is shown in figure 7. We can see that the features used near the root node are: media, mentioned, author. In OLS results, the most important 3 features with smallest p-values are media, retweets, mentioned, author. We can conclude that random forest regressor is consistent with OLS analysis.

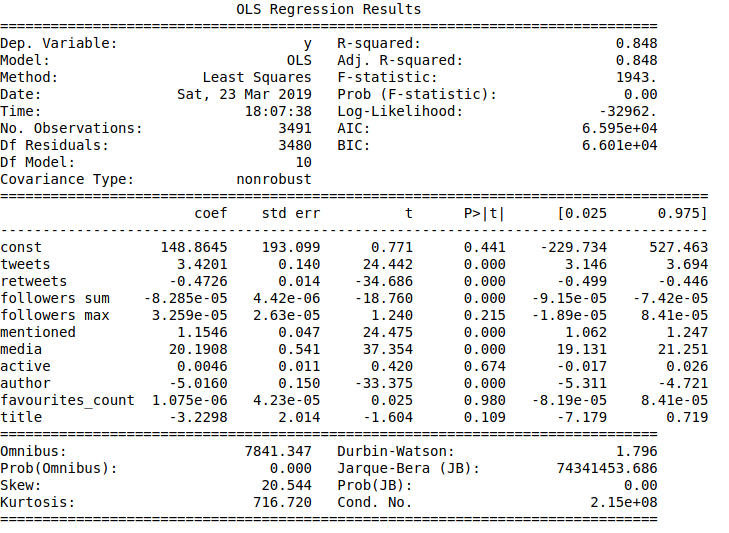


Figure 7. OLS on the entire dataset

**Question 10 - Cross-validation test error**

we perform the same grid search above for GradientBoostingRegressor for each time period described in Question 6. The optimal parameters set and the corresponding test MSE under different cases are shown in Table 7.

|  |  |  |  |
| --- | --- | --- | --- |
|  | - before active | - active | - after active |
| max depth | 40 | 10 | 20 |
| max features | auto | sqrt | sqrt |
| min samples leaf | 2 | 1 | 4 |
| min samples split | 10 | 5 | 2 |
| n estimators | 200 | 400 | 2000 |
| test MSE | 1885.9909330327248 | 1753.3871782573021 | 564.1801697048221 |

Table 7. Grid search result for Gradient Boosting in three periods

**Neural Network:**

**Question 11 - Different architectures**

We regress the aggregated data with MLPRegressor and try different architectures by adjusting hidden layer sizes. The MSE of the entire data under different architectures are shown in table 8.

|  |  |  |
| --- | --- | --- |
| numbers of layers | layer size | MSE |
| 1 | 50 | 128877.81326886978 |
| 1 | 100 | 130187.80518306098 |
| 1 | 200 | 128891.27577071772 |
| 2 | 50, 50 | 128150.9272828794 |
| 2 | 100, 100 | 127911.56828948876 |
| 2 | 200, 200 | 128675.70627595532 |
| 3 | 50, 50, 50 | 128454.21552308132 |
| 3 | 100, 100, 100 | 128788.89516006147 |
| 3 | 200, 200, 200 | 128804.43733649951 |
| 4 | 50, 50, 50, 50 | 128333.90922977257 |
| 4 | 100, 100, 100, 100 | 139226.34338444137 |
| 4 | 200, 200, 200, 200 | 129089.70387667182 |

Table 8. MSE for neural network under different architectures

**Question 12 - StandardScaler to scale the features**

We use the best architectures obtained above (which has layer sizes(100, 100)) and use StandardScaler to scale the features before feeding it to MLPRegressor. The MSE we got is 0.4647630696194375, which greatly increased the performance.

**Question 13 - Best architecture for grid search for each period**

We use grid search to find the best architecture (for scaled data) for each period described in Question 6. For each period, we try different architectures and select the best one for number of layers to be 1,2 and 3 using grid search. The sizes of each layer are 100, 200, 300. Other parameters used in Question 13 are shown as below: 'learning\_rate':['adaptive'], 'max\_iter':[200], 'learning\_rate\_init':[0.01], 'alpha':[0.0001], 'verbose':[10].

The results of Question 13 Grid Search are as below:

Window 1 MSE = -0.65

Window 1Best Parameters: {'alpha': 0.0001, 'hidden\_layer\_sizes': (300,), 'learning\_rate': 'adaptive', 'learning\_rate\_init': 0.01, 'max\_iter': 200, 'verbose': 10}

Window 2 MSE = -0.15

Window 2 Best Parameters: {'alpha': 0.0001, 'hidden\_layer\_sizes': (300, 300), 'learning\_rate': 'adaptive', 'learning\_rate\_init': 0.01, 'max\_iter': 200, 'verbose': 10}

Window 3 MSE = -0.64

Window 3 Best Parameters: {'alpha': 0.0001, 'hidden\_layer\_sizes': (100,), 'learning\_rate': 'adaptive', 'learning\_rate\_init': 0.01, 'max\_iter': 200, 'verbose': 10}

1. **Using 6x window to predict**

**Question 14 - Report the model of using and predictions on number of tweets in the next window.**

In this section, we use the previous 6 hours/30 minutes lag features to make better predictions. The test data has 3 sets of examples with 3 time periods of each. We splitted the data into 3 time periods for training and we used linear regression, random forest, neural network and gradient boosting for prediction. Results are shown in Table 9.

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Sample & Period | True value | Linear Regression | Random Forest | Neural Network | Gradient Boosting |
| sample 0 period 1 | 120 | 137.88 | 190.54 | 732117722 | 296.71 |
| sample 0 period 2 | 1123 | 1235.83 | 2390.38 | 7916311863 | 1781.41 |
| sample 0 period 3 | 87 | 76.75 | 1298.06 | 9683463354 | -607.08 |
| sample 1 period 1 | 846 | 557.13 | 572.61 | 1130647435 | 676.07 |
| sample 1 period 2 | 903 | 864.86 | 3300.64 | 4010775525 | 4547.88 |
| sample 1 period 3 | 46 | 44.14 | 111.91 | 24509478439 | 620.65 |
| sample 2 period 1 | 61 | 103.91 | 176.00 | 621087252 | 308.21 |
| sample 2 period 2 | 28 | 27.40 | 201.74 | 312884156 | 106.81 |
| sample 2 period 3 | 43 | 87.34 | 185.33 | 25775061040 | 1498.18 |

Table 9. Predictions for number of tweets in the next time window.

**Part 2: Fan Base Prediction**

In this part, we are trying to recognize the supporting team by the user location, and we want to predict the location based on the textual content of the tweet posted by the users. Here we consider all the tweets including #superbowl whose specified location is either in the state of Washington (WA) or Massachusetts (MA). We found that there are 67978 tweets for MA and 119391 tweets for WA. Each has a binary label as well (0 for “MA”, 1 for “WA”). 80% data are used for training, and 20% are used for testing. The keyword for locations is explained below:

WA: “Seattle, Washington”, “Washington, WA”, “Seattle, WA”, “Kirkland,Washington”, etc.

MA: “Boston, Massachusetts”, “Massachusetts, MA”, “Boston, MA”, etc.

**Question 15**

**Part 1 - Method to determine the location**

First, classification analysis on the textual data. Then, we do the feature extraction to construct the TF-IDF matrix. Next, dimension reduction by the truncated SVD. Then, we did the logistic regression with L-2 regularization, naive bayes classifier, and adaptive boosting classifier. (ROC curve and confusion matrix reported in Fig. 8)

**Part 2 - Train a binary classifier to predict the location of the author of a tweet**

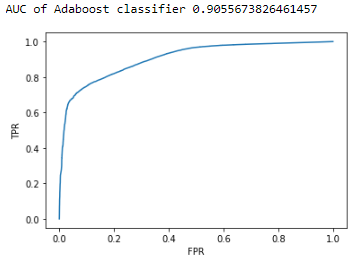
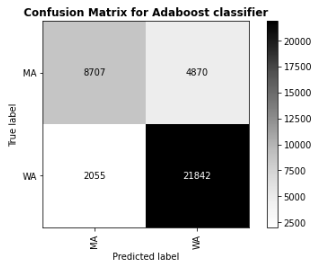
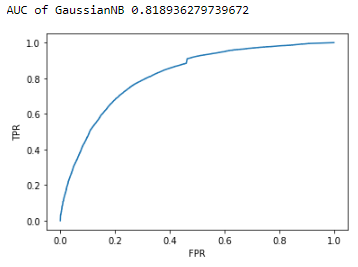
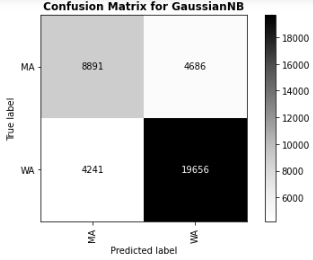
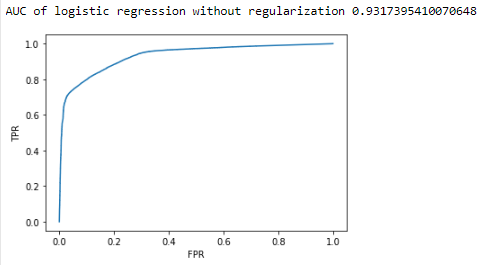
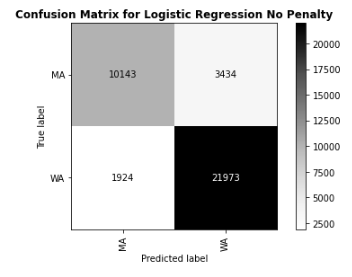


Figure 8. Confusion matrix and ROC curve for logistic regression classifier, naive bayes classifier, and adaptive boosting classifier.

The prediction results of the collected data is shown in Table 10 below:

|  |  |  |  |
| --- | --- | --- | --- |
| Measure | Logistic Regression | Naive Bayes | Adaptive Boosting |
| Accuracy | 0.8570208678016759 | 0.7617815018412766 | 0.8152052089448685 |
| Recall | 0.9194878018161275 | 0.8225300246892916 | 0.9140059421684731 |
| Precision | 0.8648403983154249 | 0.8074932215923096 | 0.8176849356094639 |
| F-1 score | 0.8913272756774298 | 0.8149422666307344 | 0.8631666304412259 |

Table 10. Prediction results for logistic regression classifier, naive bayes classifier, and adaptive boosting classifier.

**Part 3: Define your Own Project**

**Question 16 - Sentiment Analysis**

In this part, we analyzed the sentiment of textual contents from Twitter using the toolbox “TextBlob”. We extracted out all the tweets from 02/01/2015 12:00 pm to 02/02/2015 12:00 am. The game started at 3:30 pm and lasted for 3 hours and 37 minutes. The Twitter texts were used as the input of TextBlob, and outputs the polarity (range from [-1.0, 1.0], describing how negative or positive the sentiment is) and subjectivity score (range from [0.0, 1.0], 0.0 for objectivity, 1.0 for subjectivity) for each tweet. Mean scores of 5 minutes are used. The two files, “#gohawks” and “#gopatriots”, are used. The polarity and subjectivity of the fans on the game day is shown in Fig. 9. The polarity and subjectivity of the fans during the game is shown in Fig. 10.

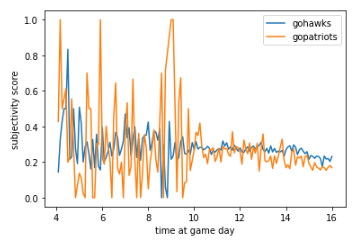
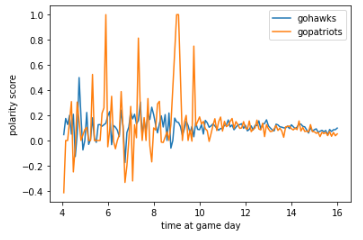


Figure 9. The polarity and subjectivity of the fans on the game day.

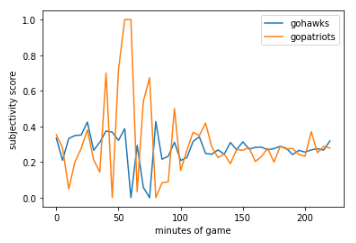
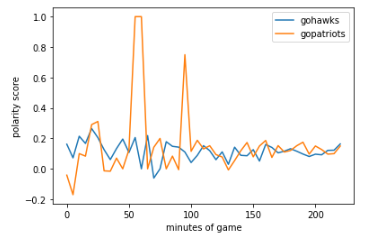
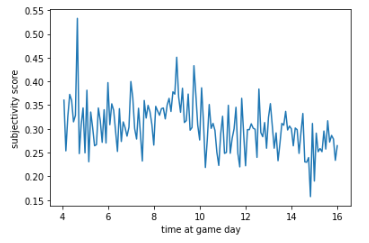
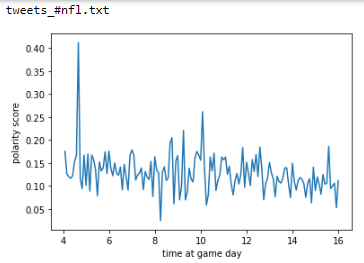


Figure 10. The polarity and subjectivity of the fans during the game.

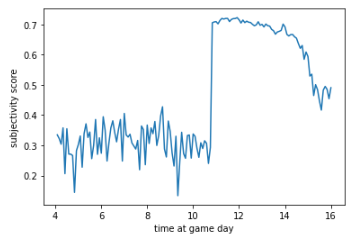
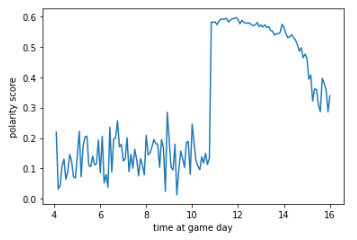
Based on the results shown in Fig. 9 and Fig. 10, before the game started, the sentiments were similar. After the game started, the scores started to oscillate. In the late night of the day, the Hawks fan had much weaker oscillations than the Patriots fan since the Patriots won the championship with a huge comeback. From the polarity, we can see that Hawks fans were more positive and subjective since they had the advantage for most of the game.

Fig. 11 below shows the sentiments for tweets to other hashtag files.

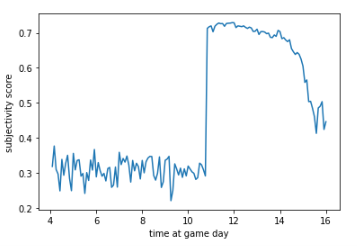
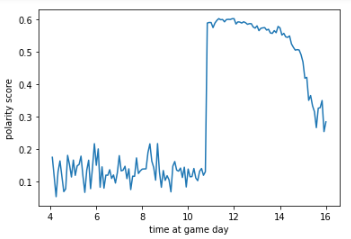
#nfl:



#patriots



#sb49



#superbowl

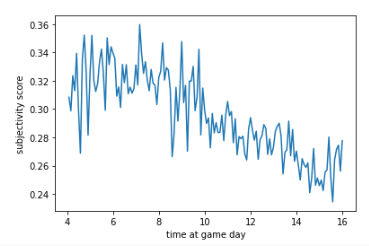
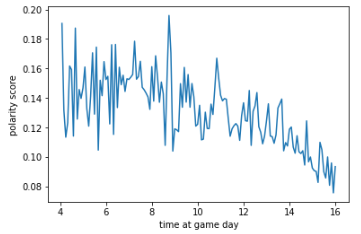


Figure 11. Sentiments for tweets to other four hashtag files.

Based on the results, tweets for “#sb49” and “#patriots” were both positive and subjective before the game, but they became less positive and more objective after the game started.