# Group Members:

XINGQUAN GUAN, UID: 804 759 254

DONG WANG, UID: 804 363 328

PENG-YU CHEN, UID: 704 732 316

ZHENGXIANG YI, UID: 304 761 570

# Acknowledgement:

All group members **have made equal contributions** to the project and same grading scale is expected for individual.

# Part 1 Popularity Prediction:

## Question 1Report the following statistics for each hashtag, i.e. each file:

## • Average number of tweets per hour

## • Average number of followers of users posting the tweets per tweet (to make it simple, we

## average over the number of tweets; if a users posted twice, we count the user and the user’s

## followers twice as well)

## • Average number of retweets per tweet

A screenshot of a cell phone

Description automatically generated

## Question 2 Plot the number of tweets in hour over time for #SuperBowl and #NFL (a bar plot with 1-hour bins). The tweets are stored in separate files for different hashtags and files are named as tweet [#hashtag].txt.

A screenshot of a cell phone

Description automatically generated

Figure 1. Bar plot of #nfl hashtag.

A screenshot of a social media post

Description automatically generated

Figure 2. Bar plot of #super

## Question 3 For each of your models, report your model’s Mean Squared Error (MSE) and R-squared measure. Also, analyse the significance of each feature using the t-test and p-value. You may use the OLS in the libarary statsmodels in Python.:

## • Go hawks

A screenshot of a social media post

Description automatically generated

Top 3 significant based on p-value (5% level): number of tweets, number of retweets, and followers.

## • Go patriots

A screenshot of a social media post

Description automatically generated

Top 1 significant based on p-value (5% level): number of retweets.

## • nfl

A screenshot of a social media post

Description automatically generated

Significant features based on p-value (5% level): number of tweets, number of retweets, flowers, and maximum followers.

## • patriots

A screenshot of a cell phone

Description automatically generated

Significant features based on p-value (5% level): number of tweets.

## • sb49

A screenshot of a social media post

Description automatically generated

Significant features based on p-value (5% level): number of tweets, number of retweets, followers, and maximum followers.

## • superbowl

A screenshot of a social media post

Description automatically generated

Significant features based on p-value (5% level): number of tweets, number of retweets, followers, and maximum followers.

## Question 4 Design a regression model using any features from the papers you find or other new features you may find useful for this problem. Fit your model on the data of each hashtag and report fitting MSE and significance of features.

## • Go hawks

A screenshot of a social media post

Description automatically generated

Top 3 significant based on p-value: followers, hashtags, and urls.

## • Go patriots

A screenshot of a social media post

Description automatically generated

Top 3 significant based on p-value: mentions, retweets, and urls.

## • nfl

A screenshot of a social media post

Description automatically generated

Top 3 significant based on p-value: hashtags, tweets, and urls.

## •patriots

A screenshot of a social media post

Description automatically generated

Top 3 significant based on p-value: followers, followers\_max, and metions.

## • sb49

A screenshot of a social media post

Description automatically generated

Top 3 significant based on p-value: mentions, followers, urls.

## •superbowl

A screenshot of a social media post

Description automatically generated

Top 3 significant based on p-value: mentions, retweets, tweets.

## Question 5 For each of the top 3 features (i.e. with the smallest p-values) in your measurements, draw a scatter plot of predictant (number of tweets for next hour) versus value of that feature, using all the samples you have extracted, and analyze it.

## Do the regression coefficients agree with the trends in the plots? If not, why?

## • Go hawks

A screenshot of a cell phone

Description automatically generated

Figure 3. Scatter plot of top 3 significant features for #gohawks

## • Go patriots

A screenshot of a social media post

Description automatically generated

Figure 4. Scatter plot of top 3 significant features for #gopatriots

## • NFL

A screenshot of a cell phone

Description automatically generated

Figure 5. Scatter plot of top 3 significant features for #nfl

## • patriots

A picture containing screenshot

Description automatically generated

Figure 6. Scatter plot of top 3 significant features for #patriots

## • sb49

A screenshot of a cell phone

Description automatically generated

Figure 7. Scatter plot of top 3 significant features for #sb49

## • superbowl

A screenshot of a social media post

Description automatically generated

Figure 8. Scatter plot of top 3 significant features for #superbowl

For #gohawks: all features seem to have positive correlation with the prediction, but this cannot be validated by the regression coefficient. Follower has a negative coefficient. It seems that the regression result may be dominated by most of the data that located at the left and bottom corner. Using log scale for data and training may give a better result. For #gopatriots, similar situation can be observed, the retweets’ coefficient is negative, which could come from the same reason as #gohawks. For #nfl, similar result. The #tweets has negative coefficients, which is different from the plot. For #patriots, the trend in the plot is not so obvious, especially the feature “followers\_max”, which has negative coefficient in the regression. We should check the result by using the log scale. For #sb49 and #superbowl, still some features have negative coefficients which is different from the plot. However, it may because the result is affected by the outliers. Although previous cases may also have some outliers, the total number of data is relative smaller in #sb49 and #superbowl, which results in a sensitive model.

## Question 6: We define three time periods and their corresponding window length as follows:

## 1. Before Feb. 1, 8:00 a.m.: 1-hour window

## 2. Between Feb. 1, 8:00 a.m. and 8:00 p.m.: 5-minute window

## 3. After Feb. 1, 8:00 p.m.: 1-hour window

## For each hashtag, train 3 regression models, one for each of these time periods (the times are all in PST). Report the MSE and R-squared score for each case.

## QUESTION 7: Also, aggregate the data of all hashtags, and train 3 models (for the intervals mentioned above) to predict the number of tweets in the next time window on the aggregated data. Perform the same evaluations on your combined model and compare with models you trained for individual hashtags.

All mean squared error and R2 results for individual hashtag and all hashtags are summarized in Table 1 and 2. As can be observed from the tables, piecewise linear model performs the best the for data after Feb 1st 8pm and worst for the data before Feb 1st 8am. Among all hashtags, ‘#superbowl’ has revealed the best prediction. Aggregate all hashtags and make the prediction cannot make the model perform better, but similar trend can be found there. The data after Feb 1st 8am has revealed more linear dependency to the studied features by significantly higher R2 values.

Table 1. Piecewise Function Prediction MSE

|  |  |  |  |
| --- | --- | --- | --- |
| Tag | Before Feb 1. 8:00 am | Between Feb 1st 8:00 am to 8:00 pm | After Feb 1st 8:00 pm |
| #gohawks | 706378 | 69692 | 1859 |
| #gopatriots | 1504 | 14042 | 94.48 |
| #nfl | 64306 | 21485 | 17647 |
| #patriots | 326738 | 659333 | 9191 |
| #sb49 | 6573 | 1270512 | 63723 |
| #supberbowl | 480441 | 54.2 | 3.2 |
| All hashtags | 4426924 | 4096895 | 207439 |

Table 2. Piecewise Function Prediction R2

|  |  |  |  |
| --- | --- | --- | --- |
| Tag | Before Feb 1. 8:00 am | Between Feb 1st 8:00 am to 8:00 pm | After Feb 1st 8:00 pm |
| #gohawks | 0.329 | 0.543 | 0.855 |
| #gopatriots | 0.685 | 0.501 | 0.797 |
| #nfl | 0.533 | 0.827 | 0.797 |
| #patriots | 0.590 | 0.735 | 0.905 |
| #sb49 | 0.888 | 0.877 | 0.838 |
| #supberbowl | 0.429 | 0.741 | 0.890 |
| All hashtags | 0.415 | 0.821 | 0.826 |

## QUESTION 8: Use grid search to find the best parameter set for RandomForestRegressor and GradientBoostingRegressor respectively. Analyze the result of the grid search. Do the test errors from cross-validation look good? If not, please explain the reason.

Grid search with K-fold cross validation is performed for both Random Forest Regressor and Gradient Boosting Regressor. When training the regressor, the entire dataset is split into 80% training and 20% testing. All training data is used in the grid search. The selected best parameters are listed in Table 3 and the mean squared errors for each time periods are shown in Table 4. The test errors look fine.

Table 3. Best Parameter from Grid Search Cross Validation

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Max\_depth | Max\_features | Min\_samples\_leaf | Min\_samples\_split | N\_estimators |
| Random Forest | 70 | Sqrt | 4 | 10 | 200 |
| Gradient Boosting | 200 | Sqrt | 2 | 5 | 1400 |

Table 4. Grid Search Cross Validation Testing Error of Best Models

|  |  |  |
| --- | --- | --- |
| Split | Best Random Forest | Best Gradient Boosting |
| 0 | -948718.61 | -3374181.71 |
| 1 | -3864431.39 | -3784610.10 |
| 2 | -3768871.22 | -1895304.99 |
| 3 | -11507700.4 | -10399348.18 |
| 4 | -1769011.32 | -2509366.64 |

***QUESTION 9: Compare the best estimator you found in the grid search with OLS on the entire dataset.***

Table 5 presents the mean squared error for the three time-windows for random forest, gradient boosting and piecewise linear function, respectively. From the table, we found Gradient Boosting has the best performance in predicting data before Feb 1st 8pm and piecewise linear function has the best prediction for data after Feb 1st 8pm. Both tree-base models are significantly superior than piecewise linear function for time window before Feb 1st 8am and between Feb 1st 8am to 8pm. Fig 9 shows the predicted vs actual value for the three models. Still, Gradient Boosting shows great power in the prediction, which is consistent with the mean squared residuals.

Table 5. Mean Squared Error of Investigated Models

|  |  |  |  |
| --- | --- | --- | --- |
| Model | Before Feb 1. 8:00 am | Between Feb 1st 8:00 am to 8:00 pm | After Feb 1st 8:00 pm |
| Random Forest | 2461030.40 | 2507343.11 | 268158.63 |
| Gradient Boosting | 234520.04 | 1697624.25 | 37824.70 |
| Piecewise OLS | 4336167 | 3810399.48 | 193506 |

A close up of a map

Description automatically generated A close up of a map

Description automatically generated A close up of a map

Description automatically generated

Figure 9. Predicted value vs. original value: before Feb 1st 8am; between Feb 1st 8am and 8pm; after Feb 1st 8pm

***QUESTION 10: For each time period described in Question 6, perform the same grid search above for GradientBoostingRegressor (with corresponding time window length). Does the cross- validation test error change? Are the best parameter set you find in each period agree with those you found above?***

In question 10, all data is served for training the gradient boosting regressor. The searched best parameter sets for the time windows, testing error for the best model and corresponding mean squared errors are listed in Table 6, 7 and 8. In this case, the testing errors are significantly lower than what we got in Table 4. Comparing Table 6 and Table 4, the best parameter sets in this case are not the same as training the entire dataset. The main reason contributing to this difference is the data in each time period are not of the same distribution. As shown in Table 5, data after Feb 1st 8pm can be easily fitted by piecewise linear function, while data before Feb 1st 8am is harder to fit. Training on the entire dataset would sacrifice some generalizability for an individual time window. As shown in Table 8, Gradient Boosting Regressor can give almost 0 **training** error. However, whether this is overfitting or not requires further verification.

Table 6. Gradient Boosting Model Setting for Each Time Period

|  |  |  |  |  |  |
| --- | --- | --- | --- | --- | --- |
| Model | Max\_depth | Max\_features | Min\_samples\_leaf | Min\_samples\_split | N\_estimators |
| Before Feb 1. 8:00 am | 50 | Auto | 2 | 2 | 200 |
| Between Feb 1st 8:00 am to 8:00 pm | 50 | Sqrt | 4 | 10 | 1600 |
| After Feb 1st 8:00 pm | 10 | Sqrt | 3 | 5 | 1600 |

Table 7. Best Gradient Boosting Model Testing Error for Each K-fold Split

|  |  |  |  |
| --- | --- | --- | --- |
| Split | Before Feb 1. 8:00 am | Between Feb 1st 8:00 am to 8:00 pm | After Feb 1st 8:00 pm |
| 0 | -1743280.09 | -4122915.17 | -53299.02 |
| 1 | -12727782.32 | -2094465.37 | -50056.30 |
| 2 | -2179075.81 | -5355223.93 | -75856.53 |
| 3 | -5441038.67 | -3446577.73 | -811444.87 |
| 4 | -2339613.60 | -5231515.34 | -97370.40 |

Table 8. Mean Squared Error for the Best Gradient Boosting Regressor

|  |  |
| --- | --- |
| Model | Mean Squared Error |
| Before Feb 1. 8:00 am | 0.218 |
| Between Feb 1st 8:00 am to 8:00 pm | 0 |
| After Feb 1st 8:00 pm | 0 |

***QUESTION 11: Now try to regress the aggregated data with MLPRegressor. Try different architectures (i.e. the structure of the network) by adjusting hidden layer sizes. You should try at least 5 architectures with various numbers of layers and layer sizes. Report the architectures you tried, as well as its MSE of fitting the entire aggregated data.***

Nine different multi-layer neural network structures are investigated. The basic model setting is shown in Table 9. The logic behind investigation are trying different number of layers and increasing hidden dimensions. No changes are made to activation layer. Still, entire dataset without training testing split is used for training the neural network. The training errors of with and without standard scaler model are listed in Table 10. As from Table 10, without applying standard scaler transformation, even the least training MSE is significantly higher than what we have from tree-based model, which illustrates that neural network may not be a good choice for the data.

Table 9. Investigated Multi-layer Neural Network Structure

|  |  |  |  |  |
| --- | --- | --- | --- | --- |
| Model | Layer 1 | Layer 2 | Layer 3 | Layer 4 |
| 1 | 128 |  |  |  |
| 2 | 256 |  |  |  |
| 3 | 128 | 128 |  |  |
| 4 | 128 | 256 |  |  |
| 5 | 128 | 256 | 256 |  |
| 6 | 128 | 256 | 512 |  |
| 7 | 128 | 512 | 512 |  |
| 8 | 256 | 256 | 512 |  |
| 9 | 128 | 256 | 512 | 1024 |

Table 10. Training Mean Squared Error of Neural Networks

|  |  |  |
| --- | --- | --- |
| Model | Without Standard Scaler | With Standard Scaler |
| 1 | 15356098.32 | 15798924.43 |
| 2 | 23427420.35 | 11542214.22 |
| 3 | 15213171.82 | 4144304.87 |
| 4 | 62327279.66 | 3969008.44 |
| 5 | 19601617.91 | 2852213.82 |
| 6 | 11607296.06 | 2822091.26 |
| 7 | 119554412.84 | 2344872.35 |
| 8 | 11753296.68 | 2758967.23 |
| 9 | 12196259.54 | 1145791.46 |

***QUESTION 12: Use StandardScaler to scale the features before feeding it to MLPRegressor (with the best architecture you got above). Does its performance increase?***

In question 12, a StandardScaler is first used to standardize all input features, then multi-layer neural network structures are fitted. By comparing the training mean square error, there is an obvious increasing in the performance after applying standard scaler transformation. Also, deeper network is preferable than shallower networks. Consequently, for grid search the best MLP structure, we’ll focus on 6-layer structures.

***QUESTION 13: Using grid search, find the best architecture (for scaled data) for each period (with corresponding window length) described in Question 6.***

From question 12, the preferable neural network structure is 6-layer (4-hidden layer) structure. Here, grid search with cross validation is performed to find the best 6-layer neural network model. The best models and corresponding training MSE are listed in Table 11. Fig 10 presents the prediction vs actual value for the piecewise linear model, gradient boosting, random forest and neural network structure. As can be observed, neural network performs closely to random forest for the three time-period. It also performs the best in the time window after Feb 1st 8pm.

Table 11. Best Neural Network Structure and Corresponding Training MSE

|  |  |  |
| --- | --- | --- |
| Model | Hidden Layer Size | Training MSE |
| Before Feb 1. 8:00 am | [128, 128, 128, 128] | 2958253 |
| Between Feb 1st 8:00 am to 8:00 pm | [1024, 1024, 1024, 1024] | 1983465 |
| After Feb 1st 8:00 pm | [1024, 1024, 1024, 1024] | 10995 |

A close up of a map

Description automatically generatedA close up of a map

Description automatically generatedA close up of a map

Description automatically generated

Figure 10. Predicted value vs. original value: before Feb 1st 8am; between Feb 1st 8am and 8pm; after Feb 1st 8pm

***QUESTION 14: Report the model you use. For each test file, provide your predictions on the number of tweets in the next time window.***

***Note: Test data should not be used as a source for training. You are not bounded to only linear models. You can find your best model through cross validation of your training data.***

From aforementioned algorithms, Gradient Boosting model from question 10 is used in testing since it gives the lowest training error. The detailed model setting is shown in Table 6. The model is trained on separate time windows and entire time period. The testing mean squared errors are reported in Table 12 and predicted testing values are plotted in Fig 11. From Table 12 and Fig 11, Gradient Boosting Model seems to be overfitting since it gives almost zero training errors. It has the lowest prediction error for testing sample 2, and it performs the worst for data between Feb 1st 8am and 8pm.

Table 12. Gradient Boosting Model Testing Mean Squared Error

|  |  |  |  |
| --- | --- | --- | --- |
|  | Sample 0 | Sample 1 | Sample 2 |
| Before Feb 1. 8:00 am | 71734.22 | 15785.99 | 1269.94 |
| Between Feb 1st 8:00 am to 8:00 pm | 901874.69 | 3552169.72 | 19850.64 |
| After Feb 1st 8:00 pm | 1153.95 | 4718.06 | 6450.73 |

A close up of a map

Description automatically generatedA close up of a map

Description automatically generatedA close up of a map

Description automatically generated

Figure 11. Gradient Boosting Regressor predicted value vs. original value on testing set: before Feb 1st 8am; between Feb 1st 8am and 8pm; after Feb 1st 8pm

Besides Gradient Boosting Regressor, the developed multi-layer neural network is also investigated in testing. The model settings are shown in Table 11. As observed, neural network performs slightly better in the time window before Feb 1st 8am but worst in the other two periods compared to Gradient Boosting Regressor. However, it does not reveal overfitting issues.

Table 13. Neural Network Testing Mean Squared Error

|  |  |  |  |
| --- | --- | --- | --- |
|  | Sample 0 | Sample 1 | Sample 2 |
| Before Feb 1. 8:00 am | 22727.56 | 19714.70 | 49801.57 |
| Between Feb 1st 8:00 am to 8:00 pm | 1374785.93 | 836698.92 | 395142.24 |
| After Feb 1st 8:00 pm | 1520.92 | 31464.24 | 55235.58 |

A screenshot of a map

Description automatically generated A close up of a map

Description automatically generated A close up of a map

Description automatically generated

Figure 12. Multi-layer Neural Network predicted value vs. original value on testing set: before Feb 1st 8am; between Feb 1st 8am and 8pm; after Feb 1st 8pm