UCLA ECE219 - Project 5

March 22nd, 2019

Authors

Ning An - 205034447
Zaurbek Tsorojev - 805029443
Endi Xu - 005030030
Minya Yao - 704161217

Introduction

The goal of this project was to analyze the tweet activity for given hashtags from Twitter in order to predict its future popularity. We train a regression model with the given data, and use it to make predictions for other hashtags. The dataset that we are using was obtained by querying popular Twitter hashtags related to the 2015 Super Bowl spanning a period starting from 2 weeks before the game to a week after the game.

Question 11

The MSE obtained from various neural network architectures we test on are summarized below, with hidden layer sizes listed, grouped by the number of hidden layers. For each architecture, we test on both logistic and tanh activation functions. We did not use relu, because the resulting MSEs were very high.

1. **One hidden layer** : [2,5,10,50,100,200,300,400,500]
   - **logistic** :
     [28754.204105085813, 28753.855694454414, 28788.59475232106, 28740.597246997637, 28711.898292, 28719.499150106578, 28641.6375867408, 28649.84564102266, 28540.37562984672]
     Minimum MSE = 28540.37562984672
   - **tanh** :
     [28775.593376809276, 28759.156362876012, 28788.4528120922, 28753.442763458068, 28706.560488, 28753.052086402895, 28750.588593469223, 28752.683491968422, 28749.134405383822]
     Minimum MSE = 28706.560488627467

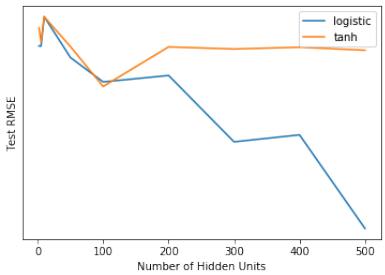


FIGURE 1 – RMSE vs # Hidden Units (for one hidden layer)

2. **Two hidden layers** : [(10,10), (20, 10), (20,20), (50,50), (100,100), (200,200)]
   - **logistic** :
     [28443.981591592787, 28444.09875841569, 28380.38820623823, 28378.480785872398, 28417.989732970007, 28384.012590969585]
     Minimum MSE = 28378.480785872398
   - **tanh** :
     [28788.103795861836, 28443.812738672033, 28380.381475064314, 28575.403698808364, 28457.392518130186, 28367.00906982105]
     Minimum MSE = 28367.00906982105

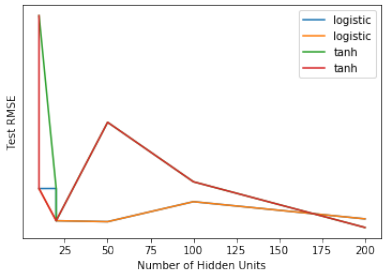


FIGURE 2 – RMSE vs # Hidden Units (for two hidden layer)

We see that the architecture that gives the lowest MSE is using tanh activation function, with hidden layer sizes of (200, 200).

Question 12

We use StandardScaler to scale the data before feeding it to MLPRegressor with tanh activation function and two hidden layers with sizes (200, 200). The resulting MSE is 13684.990, which is much less than the MSE obtained without preprocessing the input by scaling it. To understand this, we know that the idea behind StandardScaler is that it transforms the data so that the distribution will have zero mean and unit variance. This helps to train the model better in the sense that scaling avoids the situation when one or several features dominate others in magnitude.

Question 13

We use gridSearchCV to perform grid search on MLPRegressor to find the parameters of the architecture that yields the best performance, for each window length described in Question 6. We use StandardScaler to scale the data before feeding it to the grid search function. The parameters we fine-tuned are shown in the table below :

Parameter	**Range**
hidden layer sizes	(100,100,100), (20,20,20), (200,200), (100,100), (20,20), 20, 50, 100, 200, 500
activation functions	"logistic", "relu", "tanh"
alpha	0.01, 0.001, 0.0001
maximum number of iterations	1000, 2000, 3000, 5000, 10000

The optimal parameters for MLP Regressor for each period and the performance of the optimized MLP Regressor models are shown below :

Period	Optimal Parameters	MSE
1 (before Feb 1, 8 AM, 1-hour window)	Optimal parameters for MLP Regressor : {'activation' : 'tanh', 'alpha' : 0.0001, 'hidden_layer_sizes' : 100, 'max_iter' : 1000, 'random_state' : 42}	1700.923
2 (between Feb 1, 8 AM and Feb 1, 8 PM, 5-min window)	Optimal parameters for MLP Regressor : {'activation' : 'tanh', 'alpha' : 0.0001, 'hidden_layer_sizes' : (20, 20), 'max_iter' : 1000, 'random_state' : 42}	4181.166
3 (after Feb 1, 8 PM, 1-hour window)	Optimal parameters for MLP Regressor : {'activation' : 'tanh', 'alpha' : 0.01, 'hidden_layer_sizes' : (100, 100), 'max_iter' : 1000, 'random_state' : 42}	643.155

Question 14

In this question, we use the Random Forest Regressor to predict the number of tweets on the 6-th window, using the data from the previous 5 windows to train the model. The time span of the dataset for the all samples across different periods are shown below.

Period 1

- Sample 0 start and end date :
  PST dataset start date : 2015-01-31 05 :00 :37-08 :00
  PST dataset end date : 2015-01-31 10 :30 :38-08 :00
- Sample 1 start and end date :
  PST dataset start date : 2015-02-01 01 :00 :10-08 :00
  PST dataset end date : 2015-02-01 05 :40 :26-08 :00
- Sample 2 start and end date :
  PST dataset start date : 2015-01-26 17 :05 :59-08 :00
  PST dataset end date : 2015-01-26 20 :49 :01-08 :00

Period 2

- Sample 0 start and end date :
  PST dataset start date : 2015-02-01 19 :00 :05-08 :00
  PST dataset end date : 2015-02-01 19 :29 :48-08 :00
- Sample 1 start and end date :
  PST dataset start date : 2015-02-01 12 :30 :18-08 :00
  PST dataset end date : 2015-02-01 12 :59 :51-08 :00
- Sample 2 start and end date :
  PST dataset start date : 2015-02-01 08 :30 :12-08 :00
  PST dataset end date : 2015-02-01 08 :59 :15-08 :00

Period 3

- Sample 0 start and end date :
  PST dataset start date : 2015-02-04 00 :00 :16-08 :00
  PST dataset end date : 2015-02-04 05 :46 :07-08 :00
- Sample 1 start and end date :
  PST dataset start date : 2015-02-05 20 :04 :00-08 :00
  PST dataset end date : 2015-02-06 01 :50 :04-08 :00
- Sample 2 start and end date :
  PST dataset start date : 2015-02-05 17 :00 :04-08 :00
  PST dataset end date : 2015-02-05 22 :35 :12-08 :00

We see that the time span for each sample is not identical to each other, and not exactly 6 hours for periods 1 and 3, and not exactly 30 minutes for period 2. Therefore, we would expect to see that the predicted value deviates from the true value.

We yield the following results :

Period #	Period 1 (1-hour window)				Period 2 (5-min window)				Period 3 (1-hour window)			
Sample #	0	1	2	Mean	0	1	2	Mean	0	1	2	Mean
Prediction	182.1	486.2	199.4	289.2	3846.6	3725.6	967.3	2846.5	3387.7	3387.7	3387.7	3387.7
Normalized Prediction	3.0	8.1	3.3	4.8	3846.6	3725.6	967.3	2846.5	56.5	56.5	56.5	56.5

In the last row, we normalized our predictions to a rate of number of tweets per 5 min. We observe from the results that for the period where the event of Super Bowl is included, the model predicts a significantly higher number of tweets in the 5-minute window, compared to the 1-hour windows for the other periods. This is reasonable, since people tend to discuss about the news, and it is likely that more people tweet about the Super bowl.

Question 15

In this question, we use the textual content of the tweet to predict the location of the author of a tweet (Washington or Massachusetts), by training a binary classifier. We consider all the tweets including `#superbowl`.

Point 1

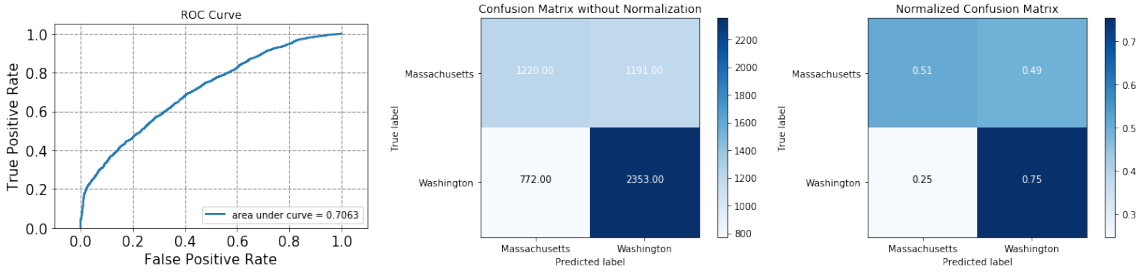
In order to only select tweets whose location is in Washington or Massachusetts, we got through all the tweets and check if their location is any of the cities of these states. If it is, we assign the tweet a label of "0" for Massachusetts and "1" for Washington. Processing this way, we collected 55,355 tweets.

Point 2

In this part, we will train 4 binary classifiers to predict the location of the author of a tweet.

a. Linear SVM Classifier

We start with a linear SVM classifier. The ROC curve and confusion matrix are reported below.

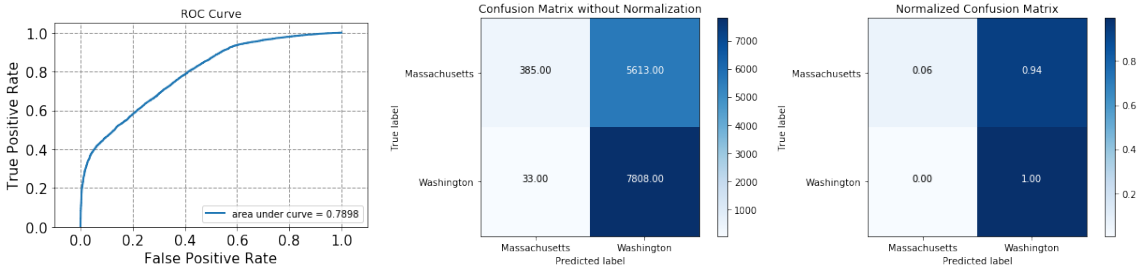


We get the following metric values :

Accuracy	0.64541
Recall	0.75296
Precision	0.66394

b. MultinomialNB

Next, we train a Naïve Bayes classifier with the MultinomialNB model.



Accuracy	0.59202
Recall	0.99579
Precision	0.58177

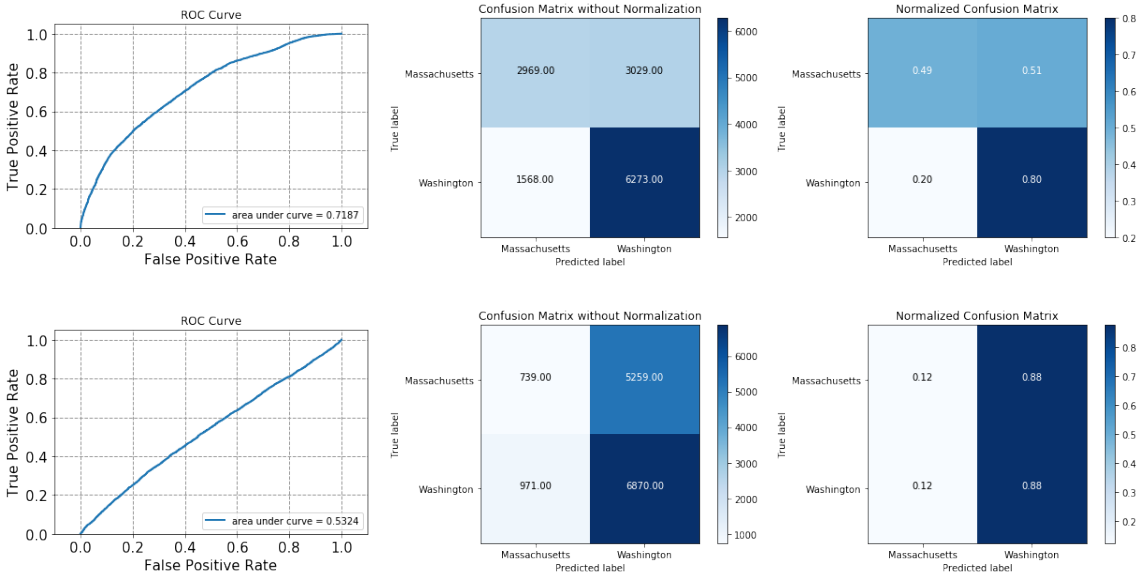
c. GaussianNB

We try a Naïve Bayes classifier again, but this time with the GaussianNB model.

Accuracy	0.66782
Recall	0.80003
Precision	0.67437

d. Logistic Regression

Finally, we try a logistic classifier.



Accuracy	0.54982
Recall	0.87616
Precision	0.56641

Conclusion

From these 4 classifiers, the best results were achieved with the Linear SVM and GaussianNB classifiers. All of the classifiers were able to predict the location of Washington with good accuracy, however they all predicted Massachusetts with bad accuracy. A reason for this could be that we didn't perfectly assigned the right labels when selecting the cities in the two states.

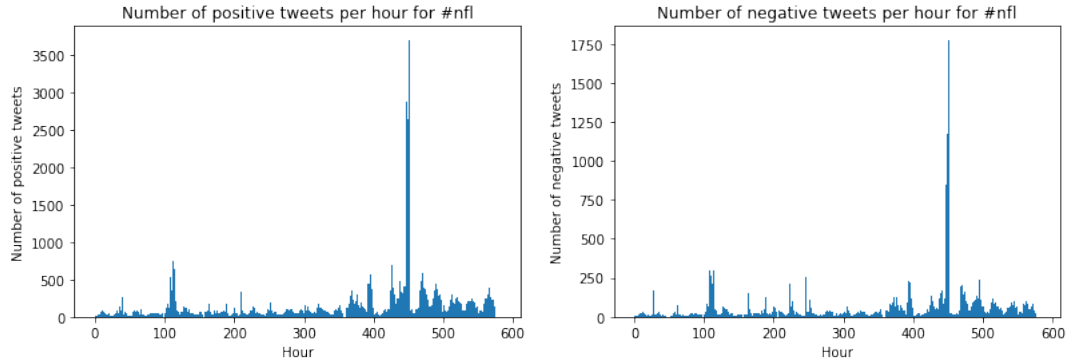
Question 16

In this last part, we will analyze the tweet sentiments for several hashtags in order to understand how people are feeling (positve or negative). The tweet sentiments are obtained using a Python library called Textblob, which allows us to apply NLP techniques.

We chose 4 different hashtags : `#nfl, #superbowl, #gopatriots, #gohawks`
For each hashtag, we will plot the number of positive and negative tweets over the two weeks of data that we have.

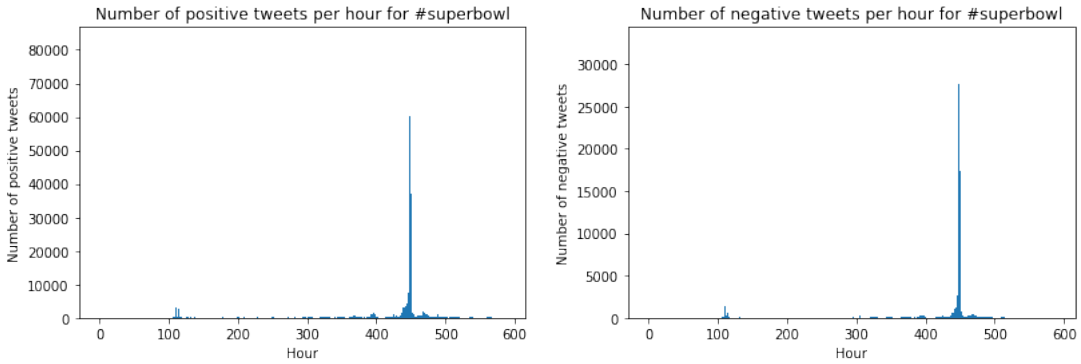
Hashtag #nfl

We see that we have two main spikes for both plots. This show a high number of tweets during or after an important match. There is double number of positive tweets than negative tweets, which shows that the majority of people had a positive feeling about the event.



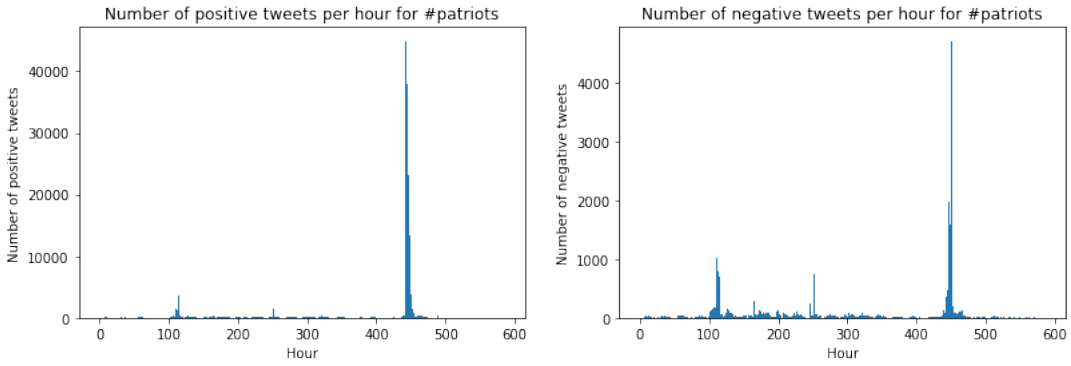
Hashtag #superbowl

Here we have one main spike. At that moment in time, there are about 85,000 positive tweets and 35,000 negative tweets. The moment corresponds to the final of the Superbowl. The positive tweets are probably all the people happy that their team won, and the negative ones are those who are disappointed.



Hashtag #patriots

We know that the patriots won the Superbowl 2015, however it was very intense matchup and surprising win as everything changed in the last 2 minutes of the game. This explains the huge spike of tweets that we see. The spike is big both in positives and negatives, which shows that it was a controversial win.



Hashtag #sb49

Finally, with the hashtag `#sb49`, we see an overwhelming amount of positive reactions during the finals. There are 10 times more positive tweets than negative ones. This shows us that overall people enjoyed the event and had fun watching it.