INF554- Report

Team: AGoodTeam, Authors: Wen YANG, Ranyu FAN, Wenqing QU

Section 1 .Feature Selection/Extraction

(1) Time related features

Timestamps in the database are transformed to date, day of week and hour because those features are more meaningful to us. As is shown Fig 1, day of week and hour could have an impact on the number of retweets.

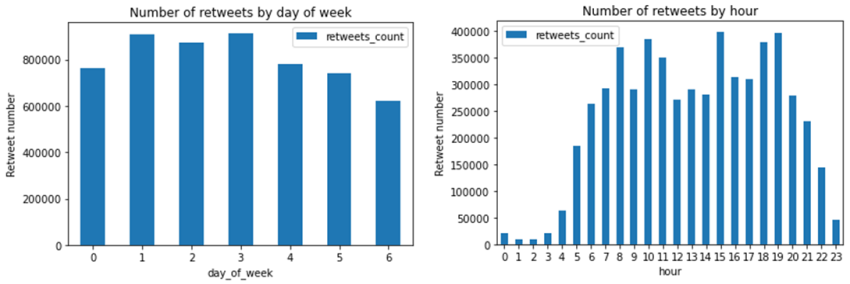

Fig1: The number of retweets according to day of week(left) / hour(right)

We found that the number of retweets varies in each day of the week , and in different hours. There are more retweets from 6 AM to 21 PM, Less retweets from 22 PM to 5 AM, so we are going to create categorized features : Monday to Sunday, day hour and night hour. At first, we assumed that it could have more retweets during day time than at night. Also ,we might have more retweets in the middle of the week than on weekends.

(2) Length of the text

In reference[1], it is recommended to study the length of text. We plot the cumulative distribution probability of the length of text in our database (Fig 2), and created 3 categories according to their length: Short for length in (0,45), Medium for length in (45,80), Long for (80,250+).

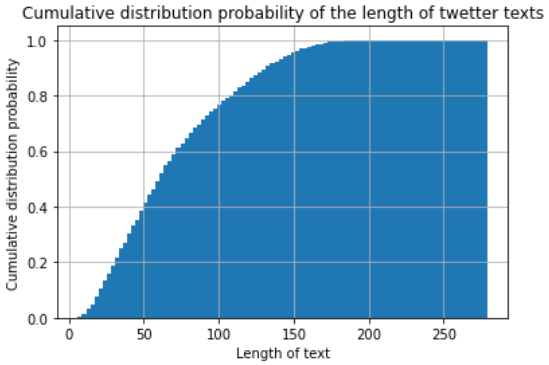

Fig 2 : Cumulative distribution probability of the length of text

(3) Followers-Friends Ratio

It is also recommended in reference[2] to study the Followers-Friends ratio, we computed the ratio of followers number and friends number, then we use Z-score normalization to rescale the features.

(4) Correlation matrix/ feature importance

We plot the correlation matrix of all the numerical features (Fig 3-right). Number of retweets has a rather strong correlation with some of the features. Correlation matrix shows the correlation coefficients between variables, the closer to 1 (or -1), the more correlated between the variables.

Considering that some features are obtained by one hot encoding, we also draw the 3 features with the best feature importance score(Fig 3- left). Feature importance gives a score for each feature, the higher the score, the more important or relevant is the feature towards output. Feature importance is an inbuilt class that comes with Tree Based Classifiers, we used Extra Tree Classifier for extracting the top 3 features for the dataset.

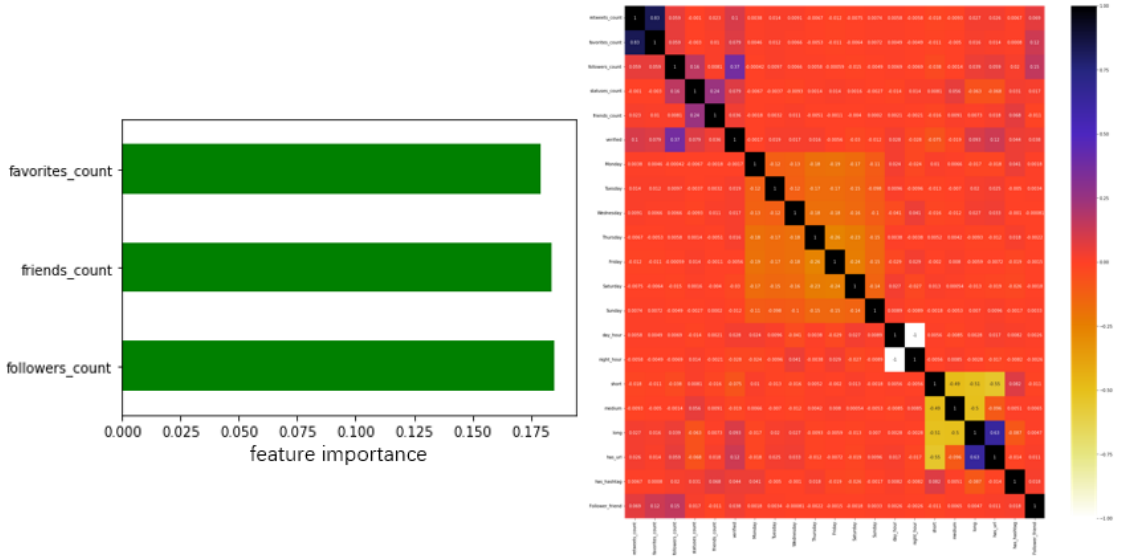

Fig3: Feature importance (Left) and correlation matrix (Right)

(5) Feature selection

At last, we selected : 'favorites_count', 'followers_count', 'friends_count', 'verified', 'long'(this refers to the text with length more than 80).

(6) Word Embedding

We used vectors to represent the texts in the database, so we used the gensim doc2vec library on python to train the texts, then we combined the vectors with the other features together as the final selected features.

Section 2: Model Choice, Tuning and Comparison

We selected five different models, including gradient boosting, linear regression, regression trees, random forests and neuron network, and we used mean absolute error (MAE) to compare the models.

(1) Gradient boosting

In gradient boosting algorithms, predictions are made by a group of decision trees. However, trees are fitted sequentially, so that each new tree is designed to reduce the overall error as much as possible.

(2) Linear regression

For linear regression, we didn't perform the hyperparameter tuning, because it has the worst results compared to other models.

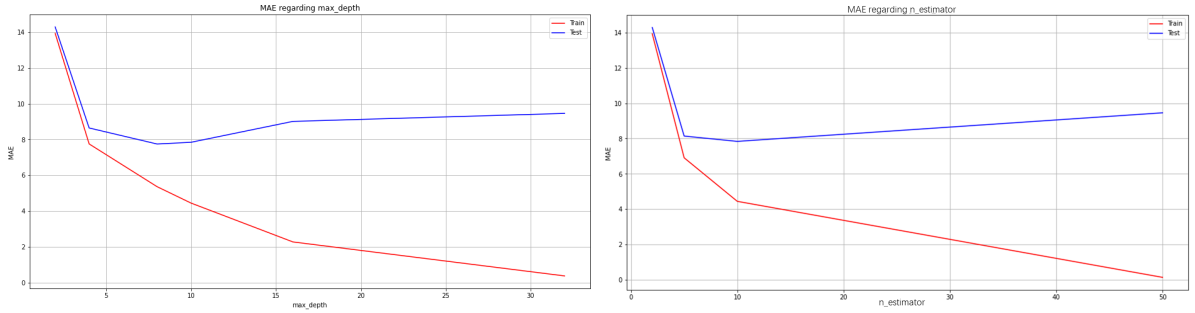

Fig 4: MAE regarding to max_depth in regression trees (left) and MAE regarding to n_estimators in random forests

(3) Regression trees

For regression trees, We studied the variation of MAE when max_depth changes. From Fig 4 (left) , it's found that for max_depth > 8 ,Training MAE decreases while Testing MAE increases, which might be an overfitting. As a result, we decided to use max_depth=8 as the optimized parameter for regression trees.

(4) Random forests

For random forests, we studied the variation of MAE when n_estimator changes. From Fig 4 (right), it's found that for n_estimators > 5 ,Training MAE decreases while Testing MAE increases, which might be an overfitting. In order to prevent overfitting, we decided to use n_estimators = 5 as the optimized parameter for random forest.

(5) Neuron networks

We've also implemented a neuron network as our model. We built a 3-layer neural network with 4, 5, 1 hidden nodes in each layer and relu as the activation function.

(6) Comparing the results:

The obtained results on MAE are presented in Table 1. Linear regression has the worst MAE compared with other models, both gradient boosting and neuron networks performed well. Neuron network, regression tree and random forest model have slightly higher errors than gradient boosting model.

We decided to use the Gradient Boosting model to predict the number of retweets in the evaluation file, because it has a good performance on reducing MAE and we could run this model on our own laptops. Although Neuron networks also gave us good results, it takes too much time, which is inconvenient for our project.

Table 1: MAE for different model on training data set

	Gradient Boosting	Linear regression	Regression tree	Random forest	Neuron networks
MAE	7.18	10.71	7.70	7.82	7.61

In conclusion, considering the MAE and the computing time on our computers, we chose gradient boosting at last.

We applied gradient boosting model on the evaluation dataset "evaluation.csv" file, the obtained results are stored in "gbr_predictions.txt" .

Reference

[1] Nelson Joukov Costa de Oliveira. (2018) Retweet Predictive Model in Twitter. University of Coimbra, Department of Informatic Engineering.

[2] T Vinayaka Ra. (2020) KM AnalytiCup 2020: COVID-19 Retweet Prediction with Personalized Attention. 2020 International Conference on Information and Knowledge Management AnalytiCup, CIKM AnalytiCup.