COVID19 RETWEET PREDICTION CHALLENGE 2020

TEAM win bird a lucky name

December 2020

Jie WANG, Yujia FU, Zongmin LI

1 INTRODUCTION

Retweet is a basic function of Twitter. It's the main way of information dissemination in Twitter, its prediction has important research value and can be used in social marketing, hot event prediction and other fields. In this project, we are provided with a dataset which is a small subset that was extracted from the COVID19 Twitter dataset [1], and our aim is to predict the retweets count with the given data. The performance is evaluated by the minimum Mean Absolute Error (MAE) .

2 FEATURES

First of all, we extract 18 features from the provided data. The details are shown in Table 1 in Appendix, they can be classified into three categories: user features, tweet features and text features.

- User features

For the origin feature user_verified, we transform the boolean value into 1 and 0 to denote whether the user has been verified by Twitter. Verified user has higher credit and is likely to have more retweets.

We keep the features follower count, friend count and statuses count, these features present the social relationship of a user and whether he is active or not, higher count usually leads to more retweets(see Figure 2 in Appendix).

In addition to the original features, we include the friend-follower ratio of a user. According to a study[6] about follower-followee ratio on user characteristics determination, lower friend-follower implies a higher level of reputation and influence of a user, therefore likely to lead more retweets(see Figure 2 in Appendix).

- Tweet features

For url, hashtag and user mentioned in a tweet, we use correspondingly one feature to denote its existence and another feature to note the specific count. The existence of these elements and a higher count could lead to more retweets.

From the timestamp, we extract a feature to demote whether a tweet is posted on weekend. People spend longer time using Tweet on weekend so a tweet is likely get more retweets if it's posted on weekend(see Figure 4 in Appendix).

- Text features

We use a feature to note the length of text. A tweet with longer text expresses more information and usually has more retweets.

We use Vader(Valence Aware Dictionary for sEntiment Reasoning) [2] to analyse the sentiment of text. We choose Vader because it's specifically attuned to sentiments expressed in social media. Vader

[1] http://www.lix.polytechnique.fr/dascim/software_datasets/projects/covid-twitter-analytics/

calculates three scores for positive, negative and neutral sentiments. It also provides a compound score. Since the theme is Covid-19, text with stronger negative sentiment is retweeted more (see Figure 3).

At last, we use TF-IDF [5] to determine the word relevance in a text. We set the maximum number of tokens to 100. The result is a sparse matrix, so we think of making dimension reduction. Inspired by the paper [3], we use SVD(singular value decomposition) to reduce the dimension and after cross validation(see Fig 1 in Appendix) we chose 3 as the number of components.

- Feature selections

As shown in Table 3, We test the performance on different combination of features on using cross validation with the mode **RF after logarithmization** proposed in section 3. We choose to drop the count of hashtag, url and mentions. Few tweet contain these elements so their counts can't bring useful information to all data. What's more, we keep only the compound sentiment score for text sentiment control, because it's a combination of other three scores.

3 PROPOSED APPROACH

We have considered 3 basic models : Support Vector Machine (SVM), Gradient Boosting Regressor(GBR) and Random Forest Regressor(RF). We then proposed 4 enhanced models: RF after logarithmization, GBR after logarithmization, SVM-enhanced RF and RF-enhanced SVM. We firstly proprocess our training input by normalizing some features because the scales of features are different. The criterion is that the columns with a big scale (such as $user\_followers\_count$) will be normalized. See Table 1 for more details. The parameters can be redefined or determined with grid search and cross-validation. Also, we use cross-validation to prevent overfitting. We note $\mathbf{X}$ the normalized feature set and $\mathbf{y}$ the set of targeted feature values.

Basic models

• **Support Vector Machine(SVM)**:

The use of SVMs in regression is known as Support Vector Regression (SVR). The fit time complexity of SVR is more than quadratic with the number of samples which makes it hard to scale to datasets with more than a couple of 10000 samples. To solve this problem, we choose SVR with parameter kernel='linear' instead of 'rbf' or 'ploy'.

However, this method has certain limitations. The Linear kernel is given by the inner product <x,y>plus an optional constant c. Kernel algorithms using a linear kernel are often equivalent to their non-kernel counterparts. As a result, it works fine if target $\mathbf{y}$ is linearly related to features. The MAE of cross validation for this method is 146.14.

• **Random Forest Regressor**:

Random forest Regressing constructs a multitude of decision trees at training time and outputs mean prediction of all individual trees. The MAE of cross validation for this method is 227.82. This may be because that target value $\mathbf{y}$ is too sparse.

• **Gradient Boosting Regressor(GBR)**:
Gradient Tree Boosting produces a prediction model in the form of an ensemble of decision trees by using weighted averages in a stage-wise fashion. The MAE of cross validation for this method is 230.99. This may also be because that target value $\mathbf{y}$ is too sparse.

Enhanced models

• **RF after logarithmization**: since we have normalized some columns of feature sets, their scales are small. But the target, number of retweet has a big scale (from 0 to hundreds of thousands). We suppose that is has a negative impact on the result. We would like to decrease the scale of target by logarithmization. For a given number of retweet y_i in the training set $\mathbf{y}_{train}$, we note $y_i^{'} = log(y_i + 1)$. The new training set of target becomes: $\mathbf{y}_{train}^{'} = \{y_1^{'}, \cdots, y_n^{'}\}$ where n is its size. And we perform random forest on $(\mathbf{X}_{train}, \mathbf{y}_{train}^{'})$ which have both small scale.

• **GBR after logarithmization**: to contrast, we proposed a GBR model after logarithmization, with the same idea as RF after logarithmization, to study if smaller scale of target feature would improve the MAE.

• **SVM-enhanced RF** : inspired by previous research[4], we would like to enhance the best basic model (according to the testing results, see table 2) SVM with random forest. Firstly, we train a SVM model $S(\mathbf{X}_{train})$ on using $(\mathbf{X}_{train}, \mathbf{y}_{train}^{'})$. Let $\epsilon = \mathbf{y}_{train}^{'} - S(\mathbf{X}_{train})$ the residual. We then create a new training dataset $C = \{(x_i, \epsilon_i) | x_i \in \mathbf{X}_{train}\}$. Secondly, we train a random forest R on C. Finally, given the trained model S and R, the prediction $\hat{\mathbf{y}}$ for the input $\hat{\mathbf{X}}$ is $\hat{\mathbf{y}} = exp(S(\hat{\mathbf{X}}) + R(\hat{\mathbf{X}})) - 1$.

• **RF-enhanced SVM**: by comparison, we proposed a RF-enhanced SVM whose principle is similar to SVM-enhanced RF. We firstly train a random forest, and then train a SVM using the residual.

Model Comparison

From Table 2, we can see that RF after logarithmization performs the best score on test with cross validation and submission. Compared with RF without logarithmization, its MAE reduced 39.94% which proves that the scale of target feature has an impact on prediction accuracy. And the contrast between GBR with and without logarithmization also proves that smaller target scale will get better MAE.

Also, the results of SVM-enhanced RF are at the same level as RF after logarithmization. However, during testing, the first part of model, SVM, does not predict well $\mathbf{y}_{test}^{'}$ with MAE$>$ 300, which shows that RF plays an important role in the combination model. What's more, SVM-enhanced RF perform as well as RF-enhanced SVM. However, RF-enhanced SVM degrades a little MAE, compared with RF after logarithmization, which shows that adding SVM could not improve the results.

In this challenge, the good performance of RF (after logarithmization) could also be found in a related problem which is predicting if a tweet will be retweeted or not[1].

REFERENCES

[1] Hendra Bunyamin and Tomáš Tunys. "A Comparison of Retweet Prediction Approaches: The Superiority of Random Forest Learning Method". In: Sept. 2016. DOI: 10.12928/TELKOMNIKA.v14i3.3150.

[2] CHE Gilbert and Erric Hutto. "Vader: A parsimonious rule-based model for sentiment analysis of social media text". In: *Eighth International Conference on Weblogs and Social Media (ICWSM-14). Available at (20/04/16) http://comp. social. gatech. edu/papers/icwsm14. vader. hutto. pdf.* Vol. 81. 2014, p. 82.

[3] Ammar Kadhim et al. "Improving TF-IDF with Singular Value Decomposition (SVD) for Feature Extraction on Twitter". In: Jan. 2017. DOI: 10.23918/iec2017.16.

[4] Guangyuan Piao and Weipeng Huang. "FOCUS: Regression-enhanced Random Forests with Personalized Patching for COVID-19 Retweet Prediction". In: *Conference: CIKM Analyticup at CIKM 2020*. Oct. 2020.

[5] Juan Ramos et al. "Using tf-idf to determine word relevance in document queries". In: *Proceedings of the first instructional conference on machine learning*. Vol. 242. New Jersey, USA. 2003, pp. 133–142.

[6] Weiwei Yan, Yin Zhang, and Wendy Bromfield. "Analyzing the follower–followee ratio to determine user characteristics and institutional participation differences among research universities on ResearchGate". In: *Scientometrics* 115.1 (Apr. 2018), pp. 299–316. ISSN: 1588-2861. DOI: 10.1007/s11192-018-2637-6.

APPENDIXES

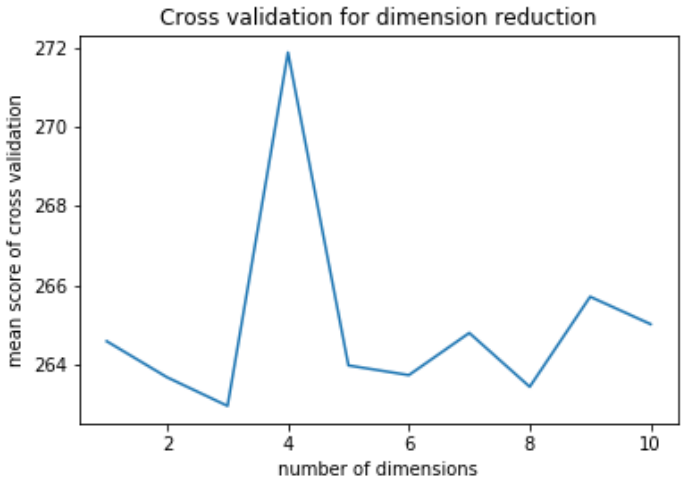

Figure 1: Cross validation scores for dimension reduction on it-idf with SVD

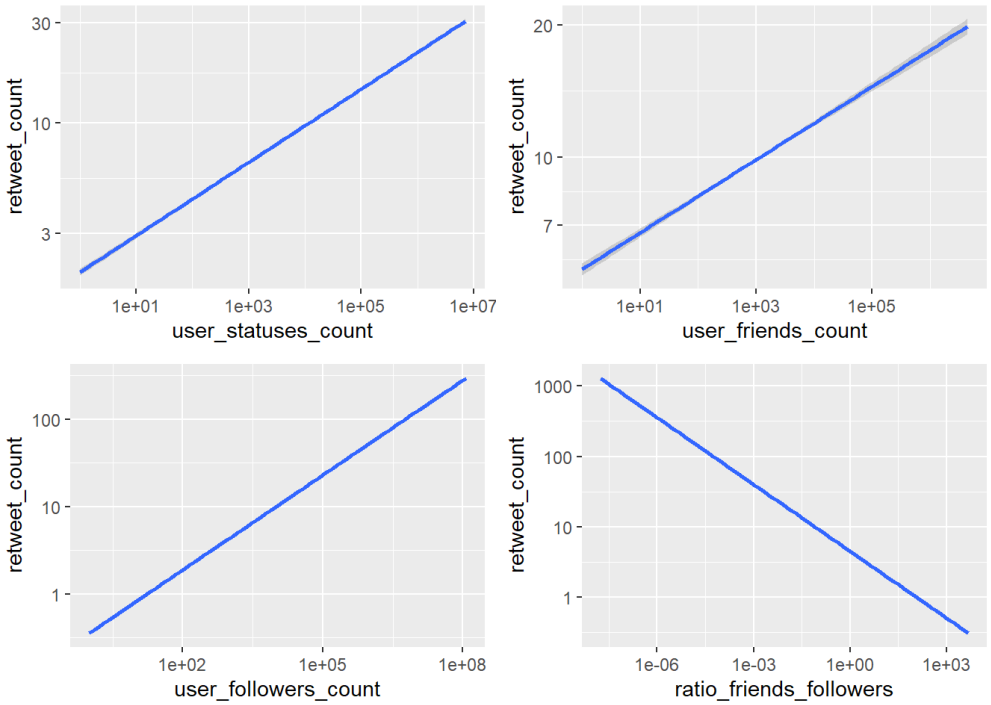

Figure 2: Smoothed line for retweet count and four user features: number of friends, number of followers, number of statuses and friend-follower ratio. Use linear model to smooth the line.

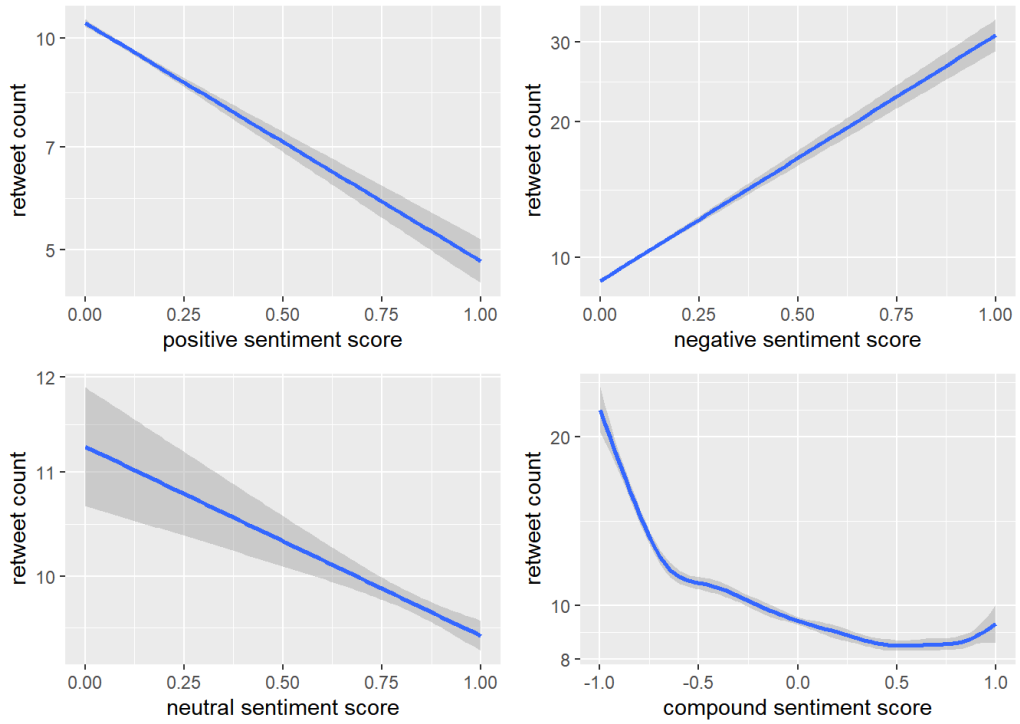

Figure 3: Smoothed line for retweet count and four text sentiment features: positive score, neutral score, negative score, compound score.

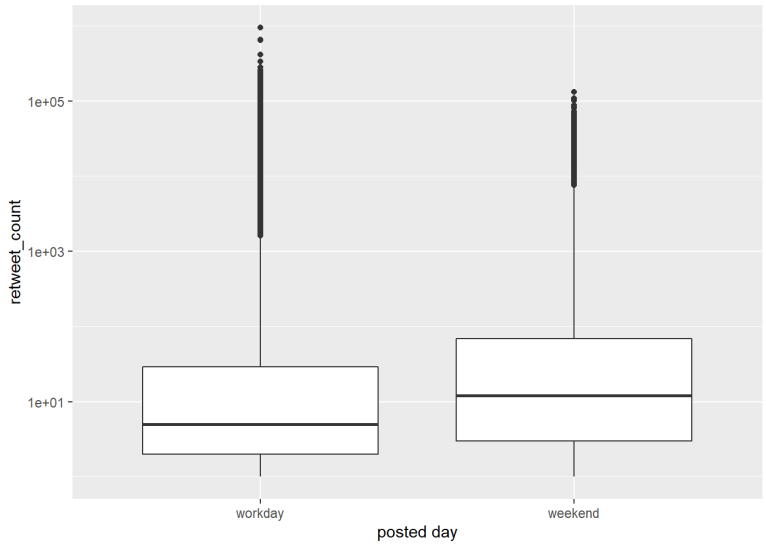

Figure 4: Distribution of retweet count group by whether a tweet is posted on weekend

Category	Feature	Description
User(5)	Has been verified	1 or 0 to denote whether the user has been verified by Twitter
	No. of statuses ⋆	The total number of tweets (statuses) the user has published
	No. of followers ⋆	The number of followers the user has
	No. of friends ⋆	The number of users that the user is following
	No. of friends / No. of followers ⋆	The ratio of those two numbers
Tweet(7)	No. of user mentioned ⋆ △	The users that are mentioned within the tweet (e.g. "@someuser").
	Has mention	1 or 0 to denote whether a tweet mentions other users
	No. of URLs ⋆ △	The total number of URLs in a tweet
	Has URL	1 or 0 to denote whether a tweet contains any URL
	No. of hashtags ⋆ △	The total number of hashtags in a tweet
	Has hashtag	1 or 0 to denote whether a tweet contains any hashtag
	Weekend	1 or 0 to indicate whether a tweet is posted on a weekend or not
Text(6)	Text length ⋆	The length of the text
	TF-IDF of the text	Term frequency and inverse, a vector of size 10 after dimension reduction
	Positive sentiment △	A score for positive sentiment for a tweet
	Negative sentiment △	A score for negative sentiment for a tweet
	Neutral sentiment △	A score for neutral sentiment for a tweet
	Compound sentiment	A normalized, weighted composite score for sentiment for a tweet

Table 1: Features extracted (⋆: features need to be normalized, △: features dropped for the best solution)

Category	Model	Testing Score	Submission Score
Basic	SVM	146.137482	159.44258
	GBR	230.9974437	-
	RF	227.8201377	-
Enhanced	RF after logarithmization	136.8241345	149.69948
	RF after logarithmization ⋆	**136.3319522**	**149.21560**
	GBR after logarithmization	140.7546662507321	-
	SVM-enhanced RF	137.8798042	153.68301
	RF-enhanced SVM	137.6657129	152.79456

Table 2: Models' testing results with cross validation and submission results (by MAE, model without ⋆ means that it's trained on all features, model with ⋆ means that it's trained on selected features)

#	Dropped features	Testing score
1	None	136.8241345
2	No. of URLs, No. of hashtags, No. of user mentioned	136.798267517576
3	Positive sentiment, Negative sentiment, Neutral sentiment	136.430670234709
4	Features dropped in 2 and 3	136.331952215745

Table 3: Performance of mode **RF after logarithmization** on dropping different features. Use cross validation to calculate the MAE score.