Dcard ML Intern Homework Report

國立台灣大學資訊工程學系 胡材溢

Introduction

This report is for the Dcard ML Intern Homework. It covers the design of an algorithm, including processing the provided data, training machine learning models, and post-processing the predicted results to forecast the number of likes after 24 hours given post's title, time, likes and comments in the first 6 hours, forum id, author id, and forum state.

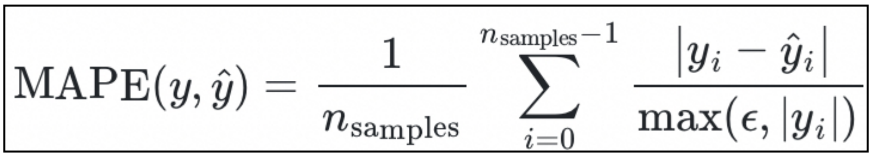
Data Preprocessing

The "title" and "created_at" columns in the data are both non-numeric data and cannot be directly used for training. For the "title" column, there is no effective way to obtain its information, so it is decided to be deleted directly. For the "created_at" column, after observing the "created_at" data in "train_set", "public_test", and "private_test", it is found that there is no overlapping month. Therefore, the month information is discarded. In addition, the "created_at" column is split into "created_at_date" and "created_at_time", representing the date and time of the post, respectively. "created_at_time" recorded in seconds is numerical data, which can be used directly.

Evaluation

(a) Evaluation Matrix

I use the mean_absolute_percentage_error() function from the Scikit Learn package to calculate the required mean absolute percentage error(MAPE). The formula is shown in the Figure 1 below.



$$\text{MAPE}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} \frac{|y_i - \hat{y}_i|}{\max(\epsilon, |y_i|)}$$

Figure 1. The formula of mean absolute percentage value in Scikit Learn

Where ϵ is an arbitrary small yet strictly positive number to avoid undefined results when y is zero.

(b) K-Fold Cross-Validation

"train_set.csv" and "public_test.csv" are initially provided and can be used directly for training and evaluating the model. However, since it is uncertain whether the two datasets are general enough, the data from both files will be merged. In addition to directly training with "train_set.csv" and evaluating the predictive algorithm using "public_test.csv", 6-fold cross-validation is performed to calculate the validation score five times, taking the average as the final evaluation score.

Method

(a) Training Model

After the data preprocessing is completed, the data is fed directly into different models for training. This experiment is run on a MacBook Pro 2019 laptop. The results are presented in Table 1.

Model	MAPE of Default Dataset	Mean of MAPE of 6-Fold	Average Runtime (Second)
Linear Regression	1.13	1.23	0.049
AdaBoost Regression	19.58	17.48	2.813
Random Forest	0.66	0.722	42.308
Linear SVR	2.44	1.50	9.459
ϵ—SVR	0.68	0.68	189.860
XGBoost Regression	0.63	0.68	3.058

Table 1. Performance of Different Regression Models

Based on the analysis of the MAPE and runtime, XGBoost Regression has the lowest MAPE, while still maintaining an acceptable execution efficiency compared to ϵ—SVR. Therefore, XGBoost Regression is chosen as the final model to be used.

(b) Fine Tuning

By calculating $\frac{|\, y_i - \hat{y}_i|}{\hat{y}_i}$ from the prediction and testing data, it was found that the reason for the larger MAPE is due to overestimation of the target variable in cases where the actual value is small. Therefore, only the data with "like_count_24h" value lower than the threshold is used as training data.

Threshold	300	400	500	600	700	800	∞
MAPE of Default Dataset	0.49	0.52	0.53	0.54	0.56	0.56	0.63
Mean of MAPE of 6-Fold	0.52	0.55	0.57	0.58	0.59	0.60	0.68

Table 2. Performance of Different Threshold

(c) Postprocessing of Final Prediction

Because the data type of "like_count_24h" is integer, the final prediction is rounded. In addition, even if the training data only includes data which "like_count_24h" values below a certain threshold, overestimation is still a significant issue. Therefore, if the "like_count_6h" of the data to be predicted is less than or equal to 20, the prediction is set to the mode of the "like_count_24h" of the data with the same "like_count_6h" in the training data plus one. The result of the final algorithm is shown in the Table 3 below.

	MAPE of Default Dataset	Mean of MAPE of 6-Fold	Average Runtime (Second)
Final Algorithm	0.33	0.31	9.075

Table 3. The Result of the Final Algorithm

● Discussion

(a) Evaluation Matrix

When calculating MAPE, we take the absolute difference between the predicted value and the true value, then divide by the true value. Therefore, the impact on MAPE is lower when the real value is larger. However, when the true value is smaller, if the predicted value deviates too much, it can easily cause MAPE to become larger. As a result, it is important to avoid overestimation as much as possible.

(b) The Performance of XGBoost Regression

XGBoost is a gradient boosting algorithm that uses decision trees as base models and ensembles multiple decision trees to improve prediction performance. In addition, XGBoost is highly parallelizable, making it efficient in handling large, high-dimensional, nonlinear, structured data.

(c) The Relationship between all columns and "like_count_24h"

After data preprocessing and merging the "train_set" and "test_set" datasets, 60,000 data were used to calculate the correlation coefficients between each column and the "like_count_24h", as shown in the Table 4 below. It can be observed that "like_count_24h" is more strongly correlated with the number of likes in the previous six hours, and less correlated with other fields.

Column	Correlation Coefficient	Column	Correlation Coefficient
like_count_1h	0.387858	comment_count_1h	0.036304
like_count_2h	0.459146	comment_count_2h	0.046216
like_count_3h	0.548945	comment_count_3h	0.058850
like_count_4h	0.635143	comment_count_4h	0.071177
like_count_5h	0.699237	comment_count_5h	0.082475
like_count_6h	0.746386	comment_count_6h	0.090705
forum_id	0.028817	created_at_date	-0.005939
author_id	-0.001060	created_at_time	0.004150
forum_stats	0.048151	like_count_24h	1.000000

Table 4. the correlation coefficients between all columns and "like_count_24h"

(d) SHAP Values of XGBoost Regression

By using the SHAP package, we can analyze the impact of each feature on the predicted results in the XGBoost regression model. The results are shown in Figure 2.

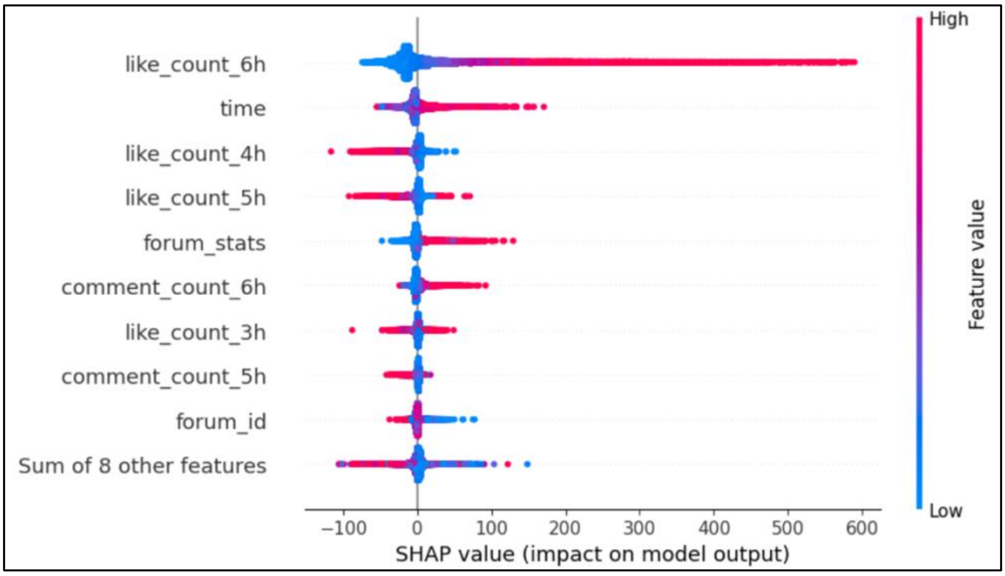


Figure 2. The SHAP Value of Features

In addition to the expected features such as the number of likes and comments in the first 6 hours having a greater impact on the model, the post time and forum stats also have a significant influence on the model's predictions.

(e) The Overestimated Results

From the model's predictions, it can be seen that the current model cannot accurately predict whether a post will receive a higher number of likes after 24 hours if the number of likes in the first six hours is less than 20. In most cases, the post will continue to have a lower number of likes after 24 hours, but there are some exceptions where posts with higher likes have no indication in the data that they will perform this way.

Conclusion

From the study, I find that extracting the post time from "created at" have a positive impact on the model's predictions. After comparing several models, it was found that XGBoost Regression was the most suitable as a basic prediction model. Since the data did not have enough information to reveal under what conditions a post might suddenly receive a higher number of likes, choosing a lower prediction result was an effective way to reduce MAPE.

The title information is discarded in this study. In the future, we can try using a language model to encode the title information and incorporate it into the training process, which may lead to better performance.

Reference

1. ChatGPT: https://openai.com/blog/chatgpt
2. scikit-learn: https://scikit-learn.org/stable
3. XGBoost: https://xgboost.readthedocs.io/en/stable/#
4. SHAP: https://shap.readthedocs.io/en/latest/