Models for Predicting Like Count in 24 Hours

Kai-Hsin, Chen

April 12, 2023

Contents

List of Figures

List of Tables

Chapter 1

Introduction

Social media platforms provide users with the ability to post content and receive feedback in the form of likes and comments. Predicting the engagement of a post, such as the number of likes it will receive, can be useful for marketers. In this report, multiple models are used to predict the like count of a post on Dcard 24 hours after it was published.

Chapter 2

Methodology

2.1 Data Preprocessing

Data preprocessing is performed in order to transform the original data into a format that is better suited for model training. In the experiment, techniques such as filling NaN values with the mean, applying min-max normalization, and utilizing z-score standardization are attempted.

2.1.1 Min-max Normalization

$$x' = \frac{x - min(x)}{max(x) - min(x)} \tag{2.1}$$

Equation 2.1 shows the formula for min-max normalization, which scales the original x values to fall between 0 and 1. The reason for using min-max normalization is that it standardizes the scale of all features, ensuring that they contribute equally to the model and helping to prevent the introduction of bias. However, since the training data contains outliers, min-max normalization tends to compress most of the normalized data near 0, rendering it less useful for training. As a result, min-max normalization is abandoned in the experiment.

2.1.2 Z-score Standardization

$$x' = \frac{x - x_{average}}{x_{std}} \tag{2.2}$$

Equation 2.2 shows the formula for z-score standardization, which transforms the features into a Gaussian distribution. This enables the model to more easily learn the weights and also preserves valuable information about outliers.

2.2 Model Selection

For this experiment, two models were utilized to predict the like count of a post 24 hours after it was published. The first model was a Multi-Layer Perceptron (MLP), while the second model was a Gated Recurrent Unit (GRU). Both models shared the following configurations:

- **learning rate**: 0.0001, which can be set by adding an argument in the command line before training.
- **batch size**: 32, which can be set by adding an argument in the command line before training.
- **epochs**: 100, which can be set by adding an argument in the command line before training.
- **early stopping**: 10. If the validation loss fails to improve over a period of 10 epochs, the model will cease training.

2.2.1 Multi-layer Perceptron

MLP is commonly used for supervised learning test. It is highly flexible and can learn complex nonlinear relationships between inputs and outputs. However, it can suffer from overfitting if the network is too large or the training data is not representative of the underlying population.

In the experiment, an MLP with 3 hidden layers was selected, with each layer being followed by a ReLU activation layer. Additionally, since the like count is always above 0, a ReLU layer was included before the output. The model was trained using different features, as described below:

- **Model I**: 12 features - 'like_count_1 $\sim$ 6h', 'comment_count_1 $\sim$ 6h'
- **Model II**: 13 features - Model I with 'title'
- **Model III**: 13 features - Model I with 'created_at'
- **Model IV**: 16 features - Model III with 'forum _stats', 'forum_id', 'author_id'.
- **Model V**: 15 features - Model IV without 'author_id'

In Model I, the model was trained simply on 12 numeric features. This model was actually trained to get the baseline of the experiment.

In Model II, the information in 'title' is obtained through fine-tuning Bert. Two approaches were employed for fine-tuning Bert. First, a dropout layer and a dense layer with a shape of (768, 1) were added after Bert to directly predict 'like_count_24h' based on 'title'. Second, a dropout layer, a dense layer with a shape of (768, 1), and a sigmoid activation layer were added after Bert to classify 'title' into two categories: above or below median of the target. The reason for adding 'title' as a feature is that readers tend to click on posts with interesting titles instead of dull ones.

In Model III, the time at which the posts were created was divided into 4 time bins: night (0 to 6), morning (6 to 12), afternoon (12 to 18), and evening (18 to 24), and then transformed into a one-hot vector. As a result, there were 16 columns in total. The reason for adding this feature is that the number of readers tends to differ during different periods of time. Hence, this feature can help the model capture any potential variation in reader engagement at different times of the day.

In Model IV, the author of the post, the forum at which the post was published, and the forum stats were added to the model. Although these three features were numeric in the source file, they were treated as categorical data in the experiment. The average 'like_count_24h' was calculated based on different authors, different forums, and different stats. The reason for adding these features is that some authors may tend to receive more likes based on their writing styles, and some forums tend to be more popular than others. Therefore, incorporating these features into the model can help to capture any potential variations in engagement that may arise as a result of these factors.

In Model V, the author id was removed from the model. This was because adding the author id as a feature led to fluctuations in validation loss. Therefore, in this model, the information regarding the author id was not considered.

2.2.2 Gated Recurrent Unit

The GRU is a type of recurrent neural network (RNN) architecture that was developed to tackle the vanishing gradient problem that often arises in traditional RNNs. Furthermore, the GRU is generally easier and faster to train than the Long Short-Term Memory (LSTM) model since it has fewer parameters. Specifically, the GRU only has 2 gates, a reset gate and an update gate, whereas the LSTM has three gates, an input gate, an output gate, and a forget gate.

The GRU model was chosen for this task because it is well-suited for handling sequential data, which is present in the dataset. Specifically, the dataset contains two features - 'like_count' and 'comment_count' - with a sequence length of 6, corresponding to the first 6 hours after a post is published. Thus, the input of the model is with shape(2, 6, sample size) and the output of the model is the prediction of the like count 24 hours after a post is published.

2.3 Evaluation

Evaluating the models is crucial in selecting the best model for predicting the targets. In this experiment, the evaluation metric used is the mean absolute percentage error (MAPE), which is calculated using the following formula:

$$MAPE = \frac{1}{n} * \sum_{t=0}^{n} (|\frac{y - \hat{y}}{y}|) \tag{2.3}$$

Chapter 3

Result

The following paragraph presents the results of each model used in the experiment, including their training and validation losses during training. The loss function utilized in the experiment is the mean absolute percentage error.

3.1 MLP Model I

The baseline for this experiment involved using 12 numeric features, including the like count and comment count of a post in the first 6 hours.

Epoch	20	40	60	80	100
Training Loss	0.2945	0.2932	0.2911	0.2895	0.2884
Validation Loss	0.3146	0.3097	0.3109	0.3118	0.3112

Table 3.1: Model I loss

The training and validation losses are illustrated in Figure 3.1 and summarized in Table 3.1. The validation loss of the best model is 0.3085.

3.2 MLP Model II

For Model II, which is the only model that incorporates information from the post's title, the training and validation losses are displayed in Figure 3.2 and Table 3.2. The best validation loss of the model is 0.3089. From Figure 3.2, the model was not overfitting nor underfitting.

Epoch	13	26	39	52	65	78
Training Loss	0.3001	0.2946	0.2933	0.2928	0.2913	0.2905
Validation Loss	0.3183	0.3121	0.3129	0.3105	0.3113	0.3139

Table 3.2: Model II loss

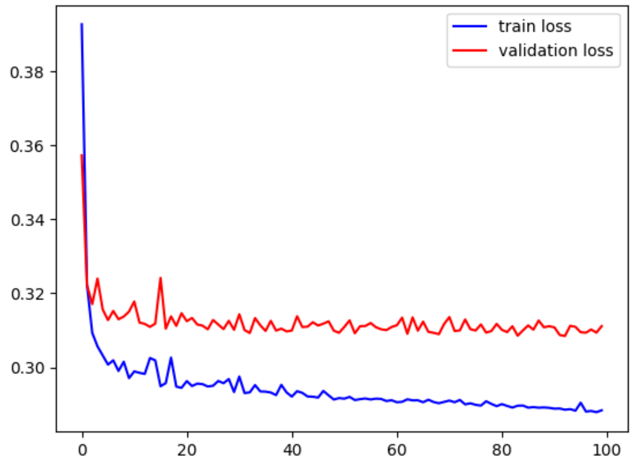

Figure 3.1: Model I training and validation loss curve

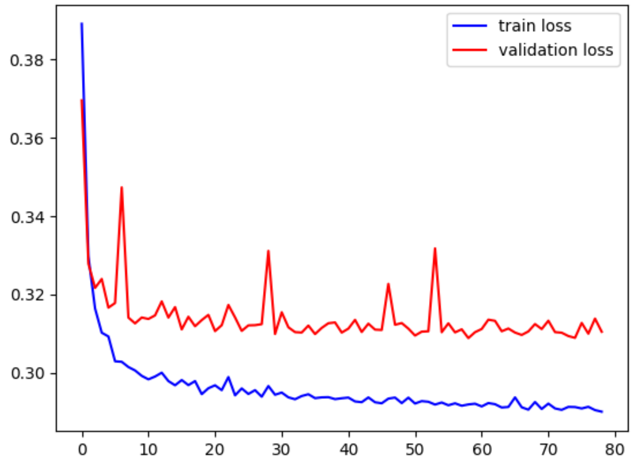

Figure 3.2: Model II training and validation loss curve

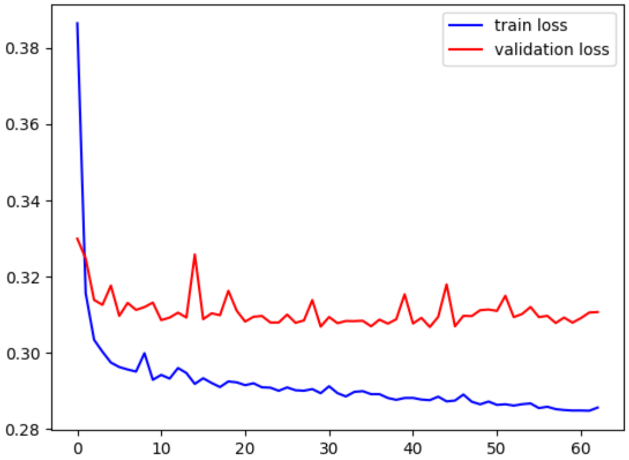

Figure 3.3: Model III training and validation loss curve

3.3 MLP Model III

Model III comprises 13 features, which include the 12 features present in Model I, in addition to the time at which a post was published. The training and validation losses for this model are depicted in Figure 3.3 and summarized in Table 3.3. The best validation loss obtained was 0.3068, and the model appeared to have converged and was about to overfitting. Thus, the training stopped.

Epoch	10	20	30	40	50	60
Training Loss	0.2930	0.2923	0.2894	0.2882	0.2872	0.2849
Validation Loss	0.3132	0.3110	0.3069	0.3154	0.3114	0.3079

Table 3.3: Model III loss

3.4 MLP Model IV

Model IV utilized all features except for the post title. The training and validation loss are shown in Figure 3.4 and Table 3.4. The validation loss of the model is notably higher than the other models. Possible reasons for this outcome will be discussed in Chapter 4.

Epoch	10	20	30	40	50	60
Training Loss	0.2347	0.2216	0.2140	0.2103	0.2080	2049
Validation Loss	0.4325	0.3995	0.4157	0.3684	0.3817	3931

Table 3.4: Model IV loss

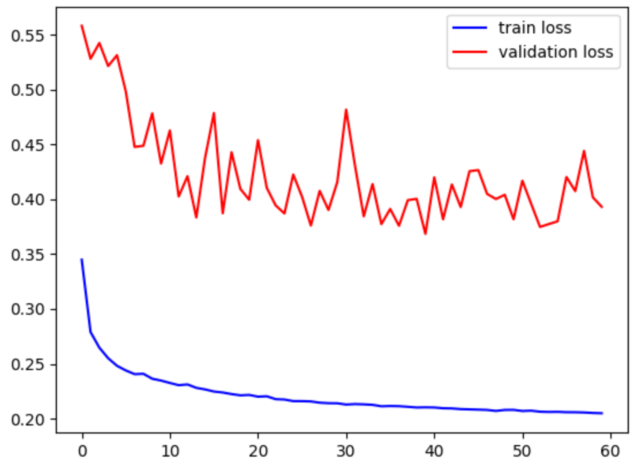

Figure 3.4: Model IV training and validation loss curve

3.5 MLP Model V

Model V contains every features except 'title' and 'author_id'. The graph shows this model was neither overfitting nor underfitting. The best validation loss was 0.3158.

Epoch	7	14	21	28	35	42
Training Loss	0.2966	0.2922	0.2886	0.2866	0.2853	0.2839
Validation Loss	0.3245	0.3228	0.3182	0.3311	0.3187	0.3232

Table 3.5: Model V loss

3.6 GRU Model

The GRU model used in this project only considered the like and comment count in the first 6 hours after the posts were published. The loss curve graph (Figure 3.6) for this model suggests that the model is neither overfitting nor underfitting. However, the best validation loss obtained was 0.3265, which is worse than the baseline.

Epoch	14	28	42	56	70	84
Training Loss	0.3300	0.3206	0.3160	0.3130	0.3106	0.3086
Validation Loss	0.3446	0.3335	0.3307	0.3306	0.3293	0.3279

Table 3.6: GRU Model loss

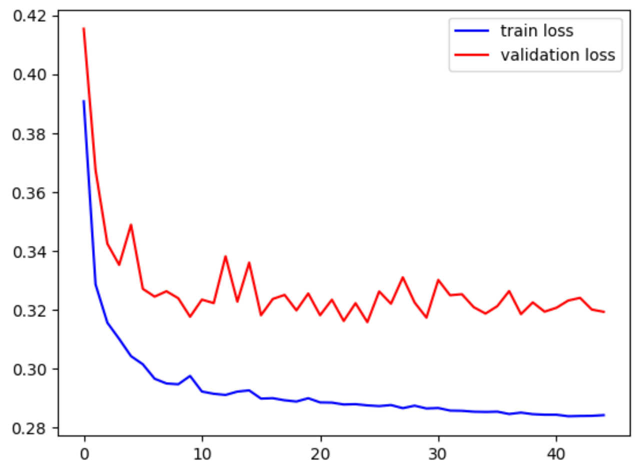

Figure 3.5: Model V training and validation loss curve

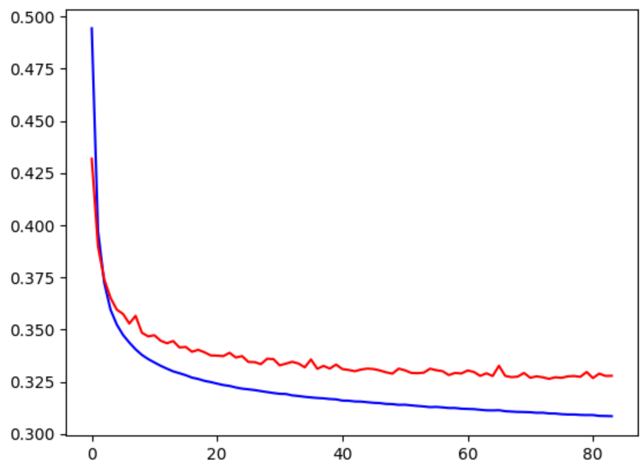

Figure 3.6: GRU Model training and validation loss curve

Chapter 4

Discussion

During the data preprocessing step, it was observed that the model with unstandardized training data outperformed the model with standardized data. This may be attributed to the fact that all the features in the dataset are in the similar scales, which is the like count and comment count. Therefore, standardizing the data may introduce noise to the model, although it can help the model converge faster. Consequently, in the following models of the experiment, the original training data without standardization was used.

Model I serves as the baseline for the experiment, with a validation loss of 0.3085. In typical scenarios, this value is considered unsatisfactory as the predicted value may deviate significantly from the true value. For instance, if the true value is 100, the predicted value may be around 70 or 130 on average, indicating a considerable degree of error. However, it is noteworthy that in the experiment, more than 40% of the data is smaller than 10 and more than 84% is smaller than 50, implying that a 30% error may not be considered significant in this particular context.

After incorporating the information from the post titles, the model did not exhibit any improvement. Notably, when training the BERT classifier, the accuracy could only reach approximately 70%. This is primarily because the post titles are short and the vocabulary is diverse, thus, only a few vocabularies can provide useful information. As a result, the post titles do not offer accurate information and consequently did not contribute to any improvement in the model.

After adding the information about the time the posts were created, the model exhibited an improvement of 0.002 in validation loss. This finding suggests that the created time of posts does provide some useful information, as the experiment was conducted multiple times, and Model III consistently outperformed Model I.

Surprisingly, with the addition of information about the author, the forum at which the posts were created, and forum stats, Model IV performed the worst among all the models. After conducting further experiments, it was concluded that adding the author's information led to a worse model. Possible reasons for this outcome include overfitting due to the added information, multicollinearity with other features, and irrelevance of the information. However, as shown in Figure 3.4, the model is not overfitting since the trend of the validation loss is still descending. Moreover, different features were selected to train the model, and regardless of the other features, including the author's information resulted in poor performance. Thus, the poor performance of Model IV is not due to multicollinearity with other features. Therefore, it is likely that the information about the author added noise to the model and thus, should be removed.

Model V, which excluded the information of the author, still failed to outperform the baseline, indicating that using the mean of the like count for each forum and forum stats is not an effective method for obtaining information from these two features. It is possible that adopting alternative approaches to extract information from these features could lead to improved results.

Chapter 5

Conclusion

The results of the experiment suggest that using only the count of comments and likes in the first 6 hours is sufficient for predicting the count of likes in the first 24 hours. Additionally, the MLP model outperforms the GRU model, and the only feature that improves performance is the time the posts were created. Interestingly, adding certain features may not necessarily lead to better performance.

Future research could explore alternative methods of separating the created time of posts, such as based on the count of likes in the first 24 hours instead of randomly. Additionally, new approaches could be explored for obtaining information about the author and forum, as the current method did not improve the model's performance.