Dynamic RNN:An Effective Approach for User Retention Prediction

Chenfei Kang*
1132048823@qq.com
Department of Automation, Shanghai Jiaotong University
Shanghai, P. R. China

Wei Wang
wangw1228@163.com
University of Chinese Academy of Sciences
Beijing, P. R. China

Weiqiang Tang
quantbruce@163.com
Shenzhen Stock Exchange
Shenzhen, P. R. China

Sibei Li
3366212933@qq.com
Beijing University of Technology
Beijing, P. R. China

Sihan Lu
1257941923@qq.com
Shanghai Jiaotong University
Shanghai, P. R. China

ABSTRACT

User retention prediction is one of the most concerned issues for enterprises. The increase in prediction accuracy can not only improve the user's personalized product experience, but also reduce the cost of customer acquisition. In this competition, based on the data provided by the organizer iQIYI, the contestants were asked to develop a system that can accurately predict the number of days each user will visit the iQIYI APP in the next 7 days (7-day retention score). This paper describes our team's (Otaku and ACG Fans) solution to this challenge. We first analyzed the characteristics of the dataset, and constructed appropriate feature engineering accordingly; then developed the Dynamic Sliding Window-Based Recurrent Neural Network(RNN), which combines the advantages of RNN and sliding window method. Not only the prediction performance is excellent, but also the robustness is strong. Finally, Our solution achieved a public score of 86.3181(ranked 2nd) and a private score of 86.3019(ranked 3th)[1].

CCS CONCEPTS

• **Computing methodologies** → *Temporal reasoning*.

KEYWORDS

Recurrent Neural Network, User Retention Score Prediction, Time Series Prediction

ACM Reference Format:
Chenfei Kang, Wei Wang, Weiqiang Tang, Sibei Li, and Sihan Lu. 2022. Dynamic RNN:An Effective Approach for User Retention Prediction. In *Proceedings of The 15th ACM International WSDM Conference (WSDM '22).* ACM, New York, NY, USA, 4 pages. https://doi.org/10.1145/nnnnnnn.nnnnnnn

*The corresponding author
[1] https://github.com/Chenfei-Kang/2022_WSDM_iQiYi_Retention_Score_Prediction

WSDM '22, Februry 21–25, 2022,

ACM ISBN 978-x-xxxx-xxxx-x/YY/MM...$15.00
https://doi.org/10.1145/nnnnnnn.nnnnnnn

1 INTRODUCTION

User retention prediction[1, 7] is a combination of recommendation[4] and time series prediction[6]. The WSDM 2022 User Retention Score Prediction contest requires contestants to develop an algorithm to predict a user's 7-day retention score based on information provided by 5 different datasets, including user portrait data, app launch logs, video related data, user playback data, and user interaction data. For test set users, it is necessary to predict the "7-day retention points" of each user on a certain day. The value range of the 7-day retained score is from 0 to 7. The evaluation metric is described by the following function:

$$100 \times (1 - \frac{1}{n}) \sum_{t=1}^{n} |\frac{F_t - A_t}{7}|, \tag{1}$$

where n is the number of users in the test set, F_t is the predicted value of 7-day retention score of users, and A_t is the real value of 7-day retention score of users. However, because the competition dataset size is considerable (about 2.5G in total), as well as unfavorable factors such as unknown training labels, low feature correlation, and the existence of missing values, it brings great challenges to training and prediction. To address this challenge, we compress the data set size without losing information, and innovatively design the training labels, and at the same time adopt a more reasonable evaluation metric AUC[3], and then perform feature engineering on the data, and propose the Dynamic RNN model is used to predict the user's 7-day retention score. Our solution ended up with the 3rd best result(86.3019) in the private leaderboard.

The paper is organized as follows: Section 2 introduces the dataset of the competition. In Section 3, we describe our methodology which contains our Dynamic RNN model details. In Section 4, we show the experiment results of our model. Finally, we conclude our analysis of the contest, as well as some additional discussions of the future directions in Section 5.

2 DATASET

A total of 5 files are provided in the competition, namely User portrait data, App launch logs, Video related data, User playback data, and User interaction data. These data include the desensitization information of nearly 600,000 users and the video information they watched. The A-stage test set contains 15,000 users who need to be predicted, and the B-stage test set contains 35,000 users. The competition requires players to predict the number of days the user will start the APP within seven days after the given end date based on the given user information, which is the officially defined as user retention rate. There is no label in the provided data set. So the user's label needs to be constructed by the players themselves. The construction method of the label is also one important factors that have affects the final result.

3 METHODOLOGY

We first construct features based on App launch logs, User portrait data, Video related data, User playback data, and User interaction data, and then input them into neural network training.

3.1 Feature Engineering

For launch Sequence Features, we construct a user launch sequence based on App launch logs, up to 60 days before the forecast date. For playtime Sequence Features. The playtime in the original data is in seconds, we normalize it by:

$$P_{norm} = \frac{1}{1 + e^{3-p/450}}. \quad (2)$$

And then Construct a user Playtime sequence up to 60 days before the end date.

For user interest features, we counts the User playback data and video related data to construct the user's interest features. For video preferences features, we apply the target encoding of "father_id", "tag_id", and "cast_id" to get "father_score", "tag_score", and "cast_score". For these 3 features, we select the calculated average of the top three frequency as the feature of the user. The calculation method of target encoding is:

$$Encoding = weight * category + (1 - weight) \times overall, \quad (3)$$

where category is the mean of the labels belonging to the category of data, and overall is the mean of the labels of all the data. The value for weight is to compute as $weight = \frac{n}{n+m}$ where n is the total number of times that category occurs in the data. m is the "smoothing factor". Larger values of m put more weight on the overall estimate.

For user portrait features, user portrait data gives the basic information of the user. It should be pointed out that, for multi values of "device_ram" and "device_rom", choose the first one. And for "territory_code", we apply target encoding again.

3.2 Model Architecture

Most of the time series prediction models are based on RNN[2, 5] and Transformer[8]. In this paper, we apply Gated Recurrent Unit(GRU) to predict user retention score. GRU is a recurrent neural network, which has natural advantages for time series modeling. Therefore, Two GRUs are used to process our two time series features: launch sequence and playtime sequence. For other features, we use a fully connected layer to process them. After that, the output of the GRU and the fully connected layer is concatenated as the input of the final classification layer. In addition, we did not directly predict the retention score, but split it into seven dichotomous classification tasks, and finally add them together as the retention score. An overall pipeline of our method is shown in Figure 1.

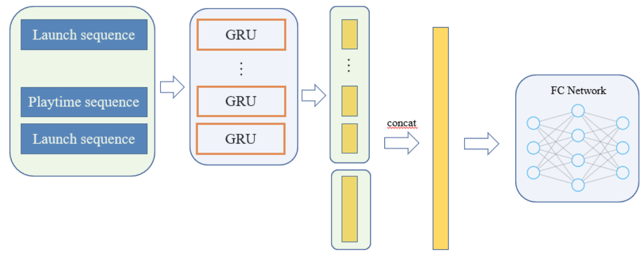

Figure 1: Model Pipline

4 EXPERIMENT

4.1 Preprocess

The date span of the entire dataset is from day 100 to day 222, but the end date specified by the test set is between 161 and 222. If it is randomly selected between 100 and 222, it will affect the effect of the model. In order to approximate the end date distribution of the test set, we will roughly control the range of random selection within the date range of the test set, according to the date of the last login of the user according to the established rules to judge. The distribution of end date before and after preprocess is shown in Figure 2 and 3.

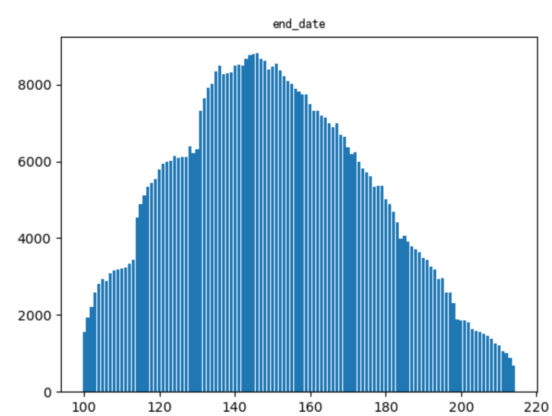

Figure 2: End date distribution based on random sampling

The user's launch sequence is an important feature, and we use a sliding window approach to generate the user's launch sequence. We select the login sequence 60 days before the end date. For users whose startup interval is less than 60 days, use -1 to fill in. In addition, we also construct other sequence features such as: the total number of videos watched per day, the video playback time sequence, the user login type sequence, and so on.

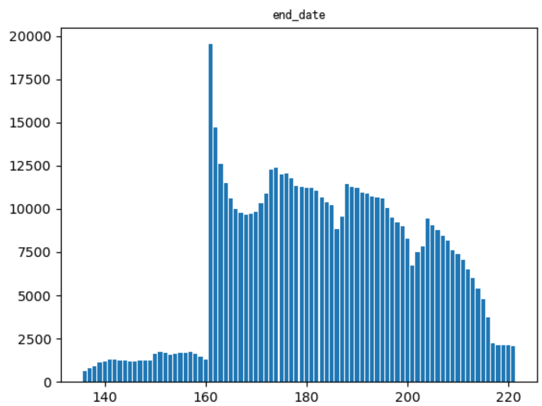

Figure 3: End date distribution based on preprocess

Generally speaking, to predict the user's retention rate within seven days, its label only needs to be defined as the total number of launches in the seven days. However, we modify it to predict the probability of everyday launching, corresponding to seven two-predicted values, that is to say, the label consists of a number becomes a 0-1 vector of length 7. The sum of the predicted vectors is the predicted value.

We tried a lot of feature engineering, including some temporal features (startup records in the first ten weeks before user's end date, etc.) and other non-temporal features. However, experiments show that too much feature engineering did not work for our model, on the contrary, sometimes even make it worse. Therefore, our final method does not require a lot of elaborate feature engineering.

4.2 Postprogress

The value predicted by the network is not directly used as the user's retention rate, but is processed using our post-processing method. We set those inactive users' predictions as 0: the predicted value of users in the test set who have only one launch record is directly set to zero; We clip the predicted value according to the rule, the specific rule we use is: if the fractional part of the predicted value is 0.5, keep it unchanged, otherwise it will be rounded. Through experiments, we find that this truncation strategy is very effective and can significantly improve the performance on the test set.

4.3 Experiment Settings

In the training process, we choose 24,000 users randomly as the validation set. At the same time, in order to ensure the stability of the model, we do not use the official evaluation index as the standard for evaluating the quality of the model, but use the AUC index. Experiments show that using the AUC index, the final result of the selected model is better than the model selected using the official evaluation metric.

The hyper parameters are as follows: The learning rate is 0.001, training epochs is 15 epochs, and batch size is 128. We use Adam optimizer and MSE Loss and trained on A100 GPU.

4.4 Experiment Result

We conducted experiments on the user retention dataset of WSDM Cup 2022, where we compared the performance of some tried methods. The experimental results are shown in the table 1 (using the B-stage test set). We only used the single model and do not use model ensemble.

Table 1: Experiment result

Method	Score
Dynamic RNN with preprocess and postprocess	**86.3012**
Cross validation with Dynamic RNN	86.1882
Dynamic RNN with preprocess	86.0136
RNN with preprocess and postprecross	85.8082
Baseline with RNN	84.5048

As can be seen from table 1, dynamic RNN, pre-processing and post-processing can improve the performance of the model. We recorded the training process with the best results, which is shown in Figure 4 to 7:

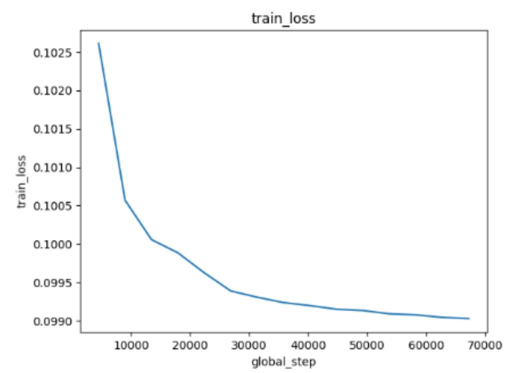

Figure 4: The curve of train loss during training

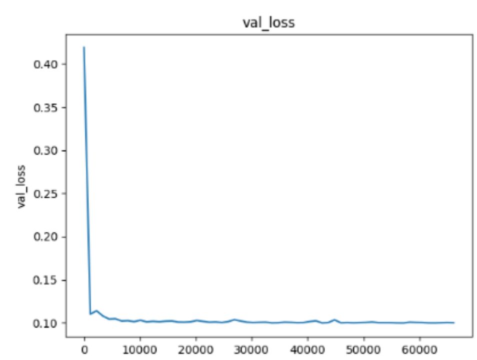

Figure 5: The curve of validation loss during training

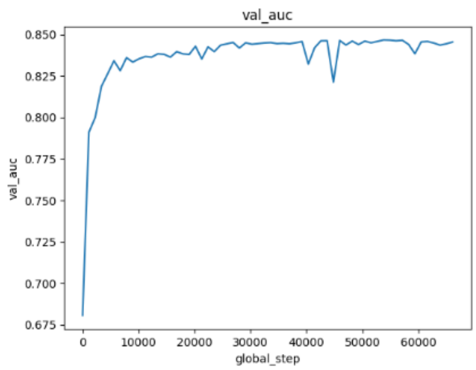

Figure 6: The curve of validation AUC during training

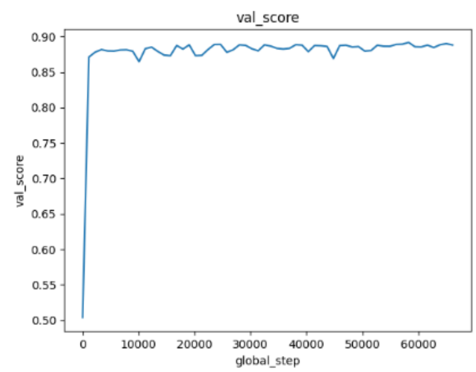

Figure 7: The curve of validation retention score during training

5 CONCLUSION

In this paper, we propose a method based on GRU and MLP for predicting the 7-day retention rate of iQiyi users, in which our model trains and predicts based on our constructed features. At the same time, we won the third place in the Retention Score Prediction competition (WSDM Cup 2022). Our solution can be summarized as follows: 1. We construct training labels, as well as user base features, login sequence features, interest features, etc. 2. We train our model, which is based on Dynamic RNN, using the constructed features and labels. 3. We truncate the predicted labels for more accurate predictions.

6 ACKNOWLEDGMENTS

We thank everyone associated with organizing and sponsoring the WSDM Cup 2022. Dataset was provided by iQIYI Technology Product Team. Challenge was sponsored and managed by the 15th ACM International Conference on Web Search and Data Mining (WSDM 2022). we are very grateful to WSDM Cup Chairs Kyumin Lee and Neil Shah for their great efforts during the challenge.

REFERENCES

[1] Anders Drachen, Eric Thurston Lundquist, Yungjen Kung, Pranav Rao, Rafet Sifa, Julian Runge, and Diego Klabjan. 2016. Rapid prediction of player retention in free-to-play mobile games. In *Twelfth artificial intelligence and interactive digital entertainment conference.*

[2] Rui Fu, Zuo Zhang, and Li Li. 2016. Using LSTM and GRU neural network methods for traffic flow prediction. In *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC).* IEEE, 324–328.

[3] Charles X Ling, Jin Huang, Harry Zhang, et al. 2003. AUC: a statistically consistent and more discriminating measure than accuracy. In *Ijcai*, Vol. 3. 519–524.

[4] Hao Ma, Irwin King, and Michael R Lyu. 2009. Learning to recommend with social trust ensemble. In *Proceedings of the 32nd international ACM SIGIR conference on Research and development in information retrieval.* 203–210.

[5] Larry R Medsker and LC Jain. 2001. Recurrent neural networks. *Design and Applications* 5 (2001), 64–67.

[6] William WS Wei. 2006. Time series analysis. In *The Oxford Handbook of Quantitative Methods in Psychology: Vol. 2.*

[7] Carl Yang, Xiaolin Shi, Luo Jie, and Jiawei Han. 2018. I know you'll be back: Interpretable new user clustering and churn prediction on a mobile social application. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining.* 914–922.

[8] Haoyi Zhou, Shanghang Zhang, Jieqi Peng, Shuai Zhang, Jianxin Li, Hui Xiong, and Wancai Zhang. 2021. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of AAAI.*