# Research on advertising click-through rate prediction model based on Taobao big data

Leqi Chen

School of Southwest University of Finance and Economics, Sichuan 611100, China

13012998480@163.com

**Abstract.** Ad click-through rate prediction is a key problem in the field of computational advertising. In this paper, LR model, Random Forest model, GDBT model and LightGBM model are used to predict ad click-through rate respectively. In this paper, we adopt the Stacking-based LR and GDBT fusion model and the BP neural network model for deep learning prediction. The experimental results on the real dataset show that the deep learning based BP neural network model performs better than other models.

**Keywords:** click-through rate prediction; gradient boosting decision tree; BP neural network; fusion model.

## 1. Introduction

In the context of the Big Data era, along with the emergence and rapid boom of the Internet, online advertising has emerged and become the mainstream of operation and promotion in various industries on the Internet. Click-Through Rate (CTR) refers to the ratio of the number of times an advertisement is clicked to the total number of times it is displayed, given a user and web content[1]. For advertisers, a high click-through rate means that the advertisements are accurately delivered, which is conducive to the promotion of the advertiser's products; for advertising platforms, a high click-through rate increases the number of visitors to the website on which the platform is placed; for users, a high click-through rate means that the advertisements bring information that meets their needs, resulting in a good search experience[2].

The purpose of CTR forecasting is to predict how likely a person is to click on an ad or item, i.e., to predict how many users will click on an ad in the future based on their historical ad click data. Revenue from advertising comes from users clicking on ads, and the effectiveness of ad auctions depends on the accuracy of click-through predictions. Accurate ad click-through rate forecasting maximizes the benefits for advertisers, platform owners and users. Many internet applications require ad click-through rate forecasting, such as online advertising and recommendation systems. Internet companies such as Weibo and Google derive much of their revenue from advertising. In the e-commerce industry, the effectiveness of advertising depends on the amount of traffic brought to the platform after the ads are placed. Therefore, data-driven advertising accuracy has become the mainstream way of advertising, and it is particularly important to accurately predict the click-through rate of advertisements to achieve accurate advertising.

Richardson et al[3] used Logistic Regression (LR) and Multiple Additive Regression Trees (MART) to predict the CTR, and the results showed that the LR model outperformed the MART model. Chapelle et al [4]proposed a machine learning framework based on LR to address the specific problem of display ad click-through rate prediction. Although the LR model is simple, it lacks the ability to handle non-linear features, and in order to handle non-linear features, He et al[5] from Facebook introduced a model that combines a gradient boosting decision tree ( Gradient Boosting Regression Tree (GBDT) and logistic regression to achieve prediction of Facebook ad click-through rates.

To address the problem of sparse and unevenly distributed advertising data, Rendle et al [6] proposed the Factorization Machine (FM) model. FMs automatically capture second-order crossover features and model all interactions between variables using factorization parameters, allowing for parameter estimation with very sparse data. Since not all feature interactions are equally predictive,

and FMs model all feature interactions with the same weight, researchers have further proposed models such as field-aware factorization machines (FFMs) [7], attention factorization machines (AFMs) [8], and field-weighted factorization machines (FwFMs) [9]. Wei et al [10] proposed a combined GBDT and FMs model for advertising click-through rate prediction, and verified the effectiveness of the high-level features extracted by GDBT for CTR prediction models.

With the rise of deep learning, the field of CTR prediction has also started to use deep learning models. Google proposed the Wide & Deep model [11], which combines the generalized linear model and the deep neural network model to develop a recommendation system that combines the benefits of memory and generalization. Based on the Wide & Deep model, Guo et al[12] derived an end-to-end learning model DeepFM, which, in contrast to the Wide & Deep model, does not require feature engineering except for raw features. Lian et al [13] combined compressed interaction networks (CINs) and classical deep neural networks (DNNs) into a unified model, and named this new model named xDeepFM (eXtreme Deep Factorization Machine). Further, Hou Na et al [14]improved the xDeepFM model and proposed a research model for advertising click-through rate prediction based on the AP-XDeepFM model.

The data involved in CTR prediction tasks usually has multiple features, and the way in which important features are extracted can greatly affect the accuracy of CTR prediction. In contrast, the click-through rate prediction models in the previous paper often ignore the importance of different features. For this reason, the literature[15][16][17] each proposed a model for advertising click-through rate prediction based on feature importance. Zhang et al[18] proposed a new framework, called multi-scale and multi-channel neural network (MSMC), to learn the importance of features and the semantics of features to improve click-through rate prediction. Jose et al[19] proposed a neural plus factor model (NAFM) based on NAFM) for an interpretable, accurate and efficient CTR estimator. However, the relative importance of higher-order feature interactions was not considered in these models, so Li et al[20] proposed the logarithmic interaction network of attention (ALIN) for improving CTR validity and accuracy based on logarithmic neural networks (LNN), which can adaptively model arbitrary-order feature interactions and learn the importance of higher-order feature interactions.

In this paper, we use LR model, random forest model, GDBT model and LightGBM model to predict the click-through rate of advertisements for the large amount of rich ad log data provided by Tianchi Data Sets. Based on this, we innovatively improve the traditional LR model and GDBT model, and propose a Stacking-based LR and GDBT fusion model for ad click-through rate prediction research model, so that these relatively weak models are fused by Stacking to achieve a stronger model. In addition, we also adopted the BP neural network (BPNN) model for deep learning prediction. The final experimental results demonstrate the effectiveness of the deep learning based BP neural network model.

## 2. Relevant theoretical foundations

### 2.1 Logistic regression models

Logistic Regression (LR) is a classification or regression problem in which an objective cost function is established and the optimal model parameters are solved by an optimization iteration. Although the word "regression" is used in logistic regression models, it is actually a classification method and is mainly used for dichotomous problems. The LR model converts the predicted value from the linear regression model into a probability value between (0,1), which corresponds to the probability of a user clicking on an advertisement.

The LR algorithm is one of the simplest and fastest classification models and performs well on datasets with linear separation bounds, with the expression

$$h_\theta(x) = \frac{1}{1+e^{-z}} = \frac{1}{1+e^{-\theta^T X}} \tag{1}$$

$h$(x) represents the predicted output value, where $y = \frac{1}{1+e^{-x}}$ is a sigmoid function, so a logistic regression model is a mapping of the results of a linear model function into a sigmoid function.

### 2.2 Random Forest Models

Before introducing Random Forests (RF), let us first introduce Ensemble Learning (EL). Ensemble learning models use a series of weak learners (also called base models or base models) to learn and integrate the results of each weak learner to obtain better learning results than individual learners. Two common algorithms for integrating learning models are the Bagging algorithm and the Boosting algorithm. The Random Forest model is a typical machine learning model for the Bagging algorithm, while typical machine learning models for the Boosting algorithm include GBDT, XGBoost and LightGBM models, etc.

Bagging, also known as self-help aggregation, is a technique for repeatedly sampling from data based on a uniform probability distribution with put-back sampling. A base classifier is trained on each sampled set of self-help samples; the trained classifier is voted on and the test sample is assigned to the class with the highest number of votes. Each of these self-help sample sets is as large as the original data.

Random Forest is a classical Bagging model with a decision tree as the weak learner. The random forest model randomly samples n different sample data sets from the original data set, then builds n different decision tree models based on these data sets, and finally obtains the final result based on the mean of these decision tree models (for regression models) or voting (for classification models).

### 2.3 Gradient Boosting Decision Tree

GBDT (Gradient Boosting Decision Tree) is one of the traditional machine learning algorithms that fits the true distribution of the data better. Each classifier is trained on the basis of the residuals from the previous round of classifiers, using an additive model that continuously reduces the residuals of the training process to classify or regress the data.

The Boosting Tree is a boosting method using a decision tree as the base function, either as a classification tree or as a regression tree. The main idea is that each time a model is built it is in the direction of a gradient descent of the loss function of the previously built model. As the loss function keeps decreasing, the model can keep improving its performance, and the best way to do this is to make the loss function decrease in the direction of the gradient. The model for the boosting tree is shown below:

$$f_m(x) = \sum_{m=1}^{M} T(x;\theta_m) \tag{2}$$

where T(x;θm) denotes the decision tree, θm denotes the parameters of the tree, and M is the number of trees. The model is trained for a total of M rounds, and each round produces a weak classifier T(x;θm), with the following loss function:

$$\hat{\theta} = arg_{\theta_m} min \sum_{i}^{N} L\big(y_i, F_{m-1}(x_i) + T(x_i;\theta_m)\big) \tag{3}$$

Fm-1(x) is the current model, which determines the parameters of the next weak classifier by empirical risk minimization. The choice of the loss function itself is also the choice of L. There are squared loss functions, 0-1 loss functions, logarithmic loss functions and so on. When the squared loss function is chosen, the difference is the residual.

GBDT can be used for almost all linear and non-linear regression problems, and can also be used to set thresholds to solve dichotomous problems, making it suitable for a wide range of applications, and can be used to train logistic Stiff regression models to improve ad click-through rates.

### 2.4 LightGBM model

LightGBM (Light Gradient Boosting Machine) is a framework for implementing the GBDT algorithm, which supports efficient parallel training and has the advantages of faster training, lower memory consumption, better accuracy, and distributed support for fast processing of large amounts of data, etc. The main reason for LightGBM is to The main reason for LightGBM is to solve the problems encountered by GBDT in massive data, so that GBDT can be used in industrial practice better and faster.

### 2.5 Stacking fusion models

In the data mining process, the generalisation ability of a single model is often thin, and the method of model fusion can combine the advantages of multiple models to improve the prediction accuracy of the model. Typical methods of model fusion include weighted fusion, Bagging, Stacking/Blending, lifting trees, etc.

Stacking is a multi-layer model where multiple models that have been trained are used as base classifiers. The predictions of these several learners are then used as a new training set to learn a new learner. That is, it can be seen as a combination strategy, using another machine learning algorithm to combine the results of individual machine learners. We call the first layer of learners the primary learners and the second layer of learners the secondary learners. In general, simple models are preferred for the secondary learners to prevent overfitting.

The Stacking fusion approach is similar to the combinatorial approach in that each layer is a model and the next layer uses the output of the previous layer to obtain the results as input to the next layer, but the Stacking algorithm is divided into two layers. The first layer uses a different algorithm to form T weak classifiers, while generating a new dataset of the same size as the original dataset, and using this new dataset and a new algorithm to form the classifier in the second layer.

(1) First split the training set D into k subsets D1,D2,...,Dk of similar size but not intersecting each other;

(2) Let Dj′ = D-Dj and train a weak learner Lj on Dj′. use Dj as the test set to obtain the output Dj′ of Lj on Dj, where j = 1...n, i.e. there is 1 model corresponding to k weak learners;

(3) Step 2 yields k weak learners and k corresponding outputs Dj′, and this k outputs plus the original class labels form the new training set Dn.

(4) The sub-learner L is trained in Dn and L is the final learner.

### 2.6 BP Neural Network Models

Back Propagation (BP) is the most effective method for training Artificial Neural Networks (ANNs), which is essentially an application of the chain rule and is widely used for updating the parameters of neural networks. The algorithm replaces the numerical calculation method for solving the neural network parameters and reduces a large number of repetitive calculations in the neural network training process.

The main idea of the backpropagation algorithm is to input the input data to the hidden layer and finally calculate the result. Then, as the initial result is certainly in error with the target, the error between the estimated value and the target value is calculated and the error is backpropagated from the output layer to the hidden layer and the input layer. This process is iterated until the difference between the estimated value and the error value is relatively small, i.e. the model converges, and the parameters in each layer of the model are adjusted.

# 3. Experiment

## 3.1 Dataset

The main object of data analysis in this study is the Taobao ad click-through rate dataset provided by Alibaba's Aliyun Tianchi, including the raw sample skeleton table, the ad basic information table, the user basic information table and the user behavior log table.

Among them, the raw_sample is a random sample of ad display/click logs (26 million records) of 1.14 million users in Taobao website within 8 days, which constitutes the original sample skeleton. The ad_feature table covers the basic information of all the ads in the raw_sample table. The user profile table, user_profile, covers the basic information of all the users in the raw_sample table. user_behavior_log, which covers all purchases made by all users in the raw_sample table over a 22-day period (700 million records in total).

**Table 1.** Data category distribution table

| Data category | Field labels | Field explanation |
|---|---|---|
| **Basic data** | Noclik, clk | Whether the user did not click on the ad and whether the user clicked on the ad |
| **Advertising features** | adgroup_id, pid, cate_id, campaign_id, customer_id, brand, price | Desensitised ad unit ID, resource position, product category, ad plan ID, advertiser ID, brand ID, price of the item |
| **User characteristics** | user_id, cms_segid, final_gender_code, age_level, pvalue_level, shopping_level, occupation, new_user_class_level, btag | Desensitized user ID, WeChat group ID, gender, age level, spending class, shopping depth, whether college student, city level, user behavior type |
| **Time information** | time_stamp | Timestamp |

## 3.2 Evaluation metrics

In this paper, two evaluation metrics, Accuracy (ACC) and Mean Square Error (MSE), are used to evaluate the model.

Accuracy refers to how many of the judgments are correct, i.e. those that are positive and those that are negative. If an instance is positive but is predicted to be positive, it is a true class (TP); if an instance is negative but is predicted to be negative, it is a true negative class (TN); if an instance is negative but is predicted to be positive, it is a false positive class (FP); if an instance is positive but is predicted to be negative, it is a false negative class (FN).

$$ACC=(TP+TN) / (TP+TN+FN+FP) \quad (4)$$

Mean squared error, a measure reflecting the degree of difference between the estimated quantity and the quantity being estimated. Let t be an estimate of the overall parameter θ determined from a subsample. The mathematical expectation of (θ - t)2 is called the mean squared error of the estimate t. It is equal to σ2 + b2, where σ2 and b are the variance and bias of t respectively. It is equal to σ2+b2, where σ2 and b are the variance and bias of t. The MSE ranges from [0,+∞) and is equal to 0 when the predicted value exactly matches the true value, i.e. the perfect model; the larger the error, the larger the value. In summary, the smaller the MSE value, the more accurate the machine learning network model, and the opposite, the worse.

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(\hat{y}_i - y_i)^2 \quad (5)$$

In this equation: yi is the actual click-through rate of the ith sample; y ˆ i denotes the predicted click-through rate of the ith sample; N denotes the total number of samples.

### 3.3 Feature engineering

In this paper, for the raw data tables, we perform feature engineering in order to extract the maximum number of features from them.

(1) Missing value padding

We choose different methods to fill in the missing values. For the 'new_user_class_level' (city level) attribute in the user_data table, we choose to fill it with the plural; for the 'pvalue_level' (consumption class) attribute in the user_data table, we choose to fill it with the KNN algorithm (a filling algorithm based on K The 'brand' attribute in the ad_feature table is filled with the previous data for the missing data of 'brand'.

(2) Data merging

We merged the dataset, user and ads tables into a new data table using the dataset table as the skeleton, based on the primary key userid and adgroup_id respectively. for the data table obtained after the merge, we again checked the percentage of missing values in each column, and deleted the rows containing missing values in the data, and operated directly on the original data. For the data table obtained after merging, we again check the percentage of missing values in each column and delete the rows containing missing values in data, operating directly on the original data.

(3) De-dimensionalize

De-dimensionalize allows data with different specifications to be converted to the same specification. Common methods of dimensionless normalization are normalization and interval scaling. Standardization presupposes that the eigenvalues obey a normal distribution, and after standardization, its conversion to a standard normal distribution. The interval scaling method makes use of boundary value information to scale the value interval of a feature to a certain characteristic range, e.g. [0, 1], etc.

(4) Binarization of quantitative features

The core of the binarization of quantitative features lies in setting a threshold value, with those greater than the threshold value being assigned a value of 1 and those less than or equal to the threshold value being assigned a value of 0. The formula is expressed as follows:

$$x' = \begin{cases} 1, x > threshold \\ 0, x \leq threshold \end{cases} \tag{6}$$

(5) Feature selection

Once the data has been pre-processed, we need to select meaningful features to feed into the machine learning algorithms and models for training. Generally speaking, we will select features based on whether they are divergent or not and their relevance to the target, and features with high relevance to the target should be selected preferentially. In this paper we choose to use the variance selection method, where we first calculate the variance of each feature and then, based on a threshold, select features with a variance greater than the threshold.

### 3.4 Experimental Results

In this section, the original LR model, random forest model, GBDT model, LightGBM model, GBDT + LR model and BP neural network model are used to predict the click-through rate of online advertisements, and the accuracy of these models is evaluated using two metrics, ACC and MSE. The original data were randomly divided into a training set and a test set in the ratio of 7:3.

Five models, namely LR model, random forest model, GBDT model, LightGBM model and GBDT + LR model, were used to predict the online advertising rate, and the ACC and MSE obtained from the five sets of experiments are shown in the following graphs.

**Table 2.** Experimental results

| Model | Training set ACC | Test set ACC | Training set MSE | Test set MSE |
|---|---|---|---|---|
| **LR** | 0.949127045 | 0.950892706 | 0.050872955 | 0.049107294 |
| **RF** | 0.973616073 | 0.930001013 | 0.026383927 | 0.069998987 |
| **GDBT** | 0.949184904 | 0.950858956 | 0.050815096 | 0.049141044 |
| **LightGBM** | 0.949127045 | 0.950892706 | 0.049187679 | 0.049456607 |
| **LR+GDBT** | 0.949127045 | 0.950892706 | 0.050872955 | 0.049107294 |


**Figure 1.** Experimental results

We also used the BP neural network model to train the data 300 times, and the final prediction accuracy obtained was 0.9508926868438721, and the final loss function image was plotted as shown below.

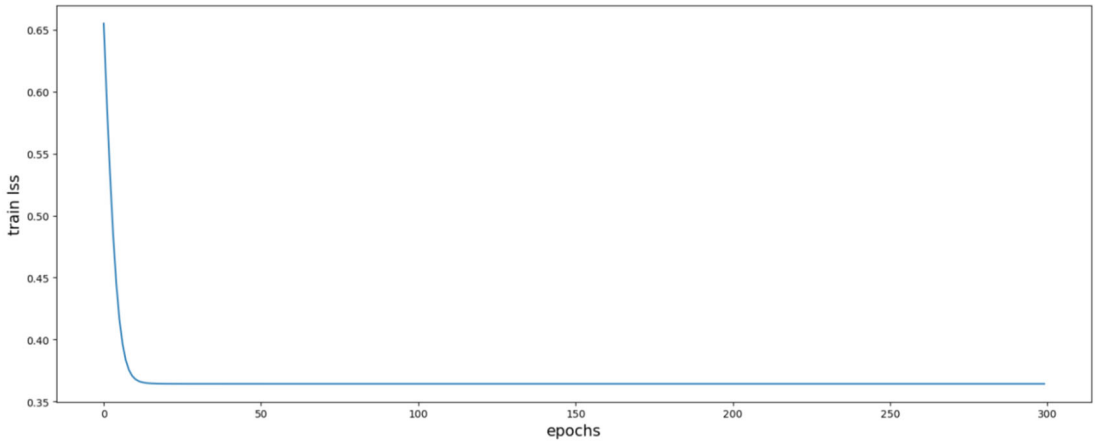


**Figure 2.** BP neural model

The final results of the LR model, the LightGBM model and the fusion model of LR and GDBT were not significantly different, and the GDBT model had slight differences compared to the above

three types of machine learning models, but the effect was still not significant. The performance of the random forest model on the training and test sets differed significantly, with the training set outperforming the other machine learning models, while the test set performed relatively worse. Meanwhile BP neural networks, as deep learning models, are more accurate than machine learning models.

## 4. Conclusion

In order to study the effectiveness of each model in predicting the click-through rate of advertisements, this paper first performed missing value filling, data cleaning and data merging on the data set, extracted features in the data and did feature engineering. And divided the training set test set and set the threshold value. We used LR model, Random Forest model, GDBT model, LightGBM model, and Stacking-based LR and GDBT fusion model to predict ad click-through rate respectively. To improve the results of click-through rate prediction, this paper also used a BP neural network model for deep learning prediction, and the final results outperformed the machine learning model and met expectations, completing the study of click-through rate prediction in a more complete way.

## References

[1] Zhou Ao-Ying, Zhou Min-Qi, Gong Xue-Qing. Computational advertising:A data-centric integrated application for the Web[J]. Journal of Computer Science,2011,34(10):1805-1819.

[2] Yang, Cheng. A click-through rate prediction algorithm based on real-time user feedback [J]. Journal of Computer Applications, 2017,37(10):2866-2870.

[3] Richardson M, Dominowska E, Ragno R. Predicting clicks: estimating the click-through rate for new ads[C]//Proceedings of the 16th international conference on World Wide Web. 2007: 521-530.

[4] Chapelle O, Manavoglu E, Rosales R. Simple and scalable response prediction for display advertising[J]. ACM Transactions on Intelligent Systems and Technology (TIST), 2014, 5(4): 1-34.

[5] He X, Pan J, Jin O, et al. Practical lessons from predicting clicks on ads at facebook[C]//Proceedings of the eighth international workshop on data mining for online advertising. 2014: 1-9.

[6] Rendle S. Factorization machines[C]//2010 IEEE International conference on data mining. IEEE, 2010: 995-1000.

[7] Juan Y, Zhuang Y, Chin W S, et al. Field-aware factorization machines for CTR prediction[C]//Proceedings of the 10th ACM conference on recommender systems. 2016: 43-50.

[8] Xiao J, Ye H, He X, et al. Attentional factorization machines: Learning the weight of feature interactions via attention networks[J]. arXiv preprint arXiv:1708.04617, 2017.

[9] Pan J, Xu J, Ruiz A L, et al. Field-weighted factorization machines for click-through rate prediction in display advertising[C]//Proceedings of the 2018 World Wide Web Conference. 2018: 1349-1357.

[10] WEI Xiaohang, YU Chongzhong, TIAN Changli et al. Internet CTR prediction model on big data platform [J]. Proceedings of the Computer Engineering and Design,2017,38(09):2504-2508.DOI:10.16208/j.issn1000-7024.2017.09.038.

[11] Cheng H T, Koc L, Harmsen J, et al. Wide & deep learning for recommender systems[C]//Proceedings of the 1st workshop on deep learning for recommender systems. 2016: 7-10.

[12] Guo H, Tang R, Ye Y, et al. DeepFM: a factorization-machine based neural network for CTR prediction[J]. arXiv preprint arXiv:1703.04247, 2017.

[13] Lian J, Zhou X, Zhang F, et al. xdeepfm: Combining explicit and implicit feature interactions for recommender systems[C]//Proceedings of the 24th ACM SIGKDD international conference on knowledge discovery & data mining. 2018: 1754-1763.

[14] Hou Na,Shao Xinhui. Ad click-through rate prediction based on AP-XDeepFM model [J]. Proceedings of the Computer Application and Software,2022,39(12):108-113+131.

[15] He Xiaojuan,Guo Xinshun. Research on advertising click-through rate prediction model based on feature optimization [J]. Journal of East China Normal University (Natural Science Edition),2020(04):147-155.

[16] Zhou F. A model for predicting the click-through rate of advertisements based on feature importance [J]. Proceedings of the Computer Knowledge and Technology,2020,16(36):12-14.DOI:10.14004/j.cnki.ckt.2020.3663.

[17] Jiang X.Y., Huang X.Y., Chen Yujing, Xu Fu. A prediction model for advertising click-through rate with dynamic feature importance extraction[J]. Proceedings of the Small Microcomputer Systems,2022,43(05):976-984.DOI:10.20009/j.cnki.21-1106/TP.2020-1021.

[18] Zhang J, Ma C, Zhong C, et al. Multi-scale and multi-channel neural network for click-through rate prediction[J]. Neurocomputing, 2022, 480: 157-168.

[19] Jose A, Shetty S D. Interpretable click-through rate prediction through distillation of the neural additive factorization model[J]. Information Sciences, 2022, 617: 91-102.

[20] Li S, Cui Z, Pei Y. A Dual Adaptive Interaction Click-Through Rate Prediction Based on Attention Logarithmic Interaction Network[J]. Entropy, 2022, 24(12): 1831.