Web Economics Group Project

(Group 6)

Qiuru Dai
University College London

Weisi Han
University College London

Boyang Liu
University College London

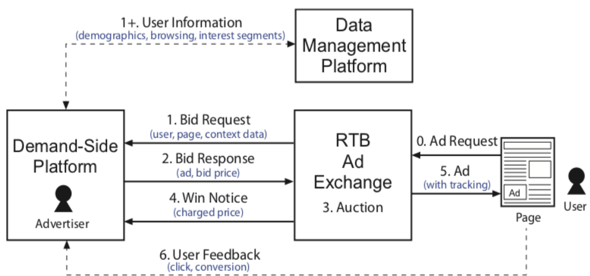

Figure 1: Procedure of RTB

1 INTRODUCTION

In the domain of computational advertising, RTB(Real Time Bidding) performs an important role. Compared with traditional bidding methods, this bidding system is designed to sell the specific chance of advertising to the advertiser with highest requirement, ensuring the maximum profit of both advertiser and the information provider(IP). The web users can also benefit in this way as their opportunities to view ads will be sold to the most suitable advertisers providing the most fitted ads.

Figure1 clearly shows the working procedure of RTB. When a web user requesting a web page the information related to the user and the page will be broadcast to advertisers. The advertisers need to claim a price based on the information they captured about this bidding, no matter from the IP or its own database. What need to be mentioned is that in RTB generally second price auction is applied, which means that the advertiser claiming the highest price will win the auction, while the second highest price will be payed. If the advertiser wins the auction, it will be informed and its advertisement will be shown to the user. The advertiser can also obtain the user feedback such as whether the ad is clicked by the user, which can be used to optimize their bidding algorithm further.

To obtain more profit in advertising, bidding strategies are the most important. Ideal bidding strategy can help the advertiser to maximize the click-through-rate(CTR), which means that their product linked to their ads will be viewed most.

In this project we are required to explore a best bidding strategy based on the given dataset, for the advertisers in it.Here the bidding strategy can be divided into two parts. One is the learning algorithm to better predict the CTR of different impressions. The other is the bidding algorithm calculating the bidding price based on the predicted CTR(pCTR). As a group a number different methods of both the two parts are implemented and tested. For CTR estimation algorithm we've tried Factorization Machine(FM), Gradient Boosting Decision Tree(GBDT implemented by XGBoost and LightGBM), and more state-of-the-art models such as deep neural networks(DNN) and convolutional neural networks(CNN). For the bidding algorithm non linear algorithms such as ORTB and modified linear bidding methods are tried. Tested and compared with baseline strategies such as constant bidding, random bidding and linear bidding, some of the methods are proved to be able to perform well while some are not. In the end we combined the selected methods with satisfied performance, which turns out to be a model with better performance. Based on this model designed for Winning criterion 1, which only considers the advertiser itself in the auction, another model for winning criterion 2 is also implemented, taking other advertisers as competitors into consideration. All of the code related to this report can be found in [8]

2 RELATED WORK

In bidding strategies, many advanced strategies can be found from the literature. The Optimal Real-Time Bidding strategy (ORTB) [1] including a approximation of winning probability and calculate the expectation of the total click number under the budget constraint as the objective function which would be maximized. The authors proposed a mathematical presentation of the real time bidding problem, which make this problem computable. Usually, the real time bidding problem is divided into two part: utility estimation (for example, CTR estimation) and prediction of market value. In [2], these two parts is combined as a whole to find the better global optimization. Moreover, a reinforcement learning method [3] is used in the real time bidding area, which can deal with budget constraints naturally.

The implementation of CNN is inspired by[4]and[6] a lot. Both of them are discussing applying Neural Networks in this field, together with the open-sourced code implemented by Theano and TensorFlow. The idea to turn to GDBT is inspired by [5] and [6].[5]obtains good results with a variation of GDBT, and [6] gives a detail comparison of different packages based on GDBT.

3 APPROACH AND RESULTS

3.1 Experiment Setup

3.1.1 *Data exploration*.

This part is detailed in the individual report.

3.1.2 *Feature Exploration, Selection and Engineering*.

There are 24 features for each impression in the dataset. however, not all of the features are equally significant for our prediction. Some of them are irrelevant attributes or unique identifiers for each bid, and such meaningless information could negatively affect our modelling power and cripple the predictive accuracy. Therefore, we just simply eliminate them from our training set. These features include *bid ID*, *user ID*, *user IP*, *url*, etc. The rest features comprises both numerical variables as well as categorical variables. Features like *slot width* and *height*, or the *floor price* are numerical variables which could be directly feed into our model without any processing.

For features like *hours* of a day, it is difficult to interpret it as a numerical variable or a categorical variable. On the one hand, 7pm is closer to 6pm or 8pm than it is to 2pm or 10pm, but on the other, there is discontinuity between 23pm and 0am. This attribute is actually an ordinal cyclic variable, so we would like to adopt trigonometric approach to transform time information by using the following equations so that the beginning hour is the same as the end.

$$xhr = sin(2\pi \times \frac{hr}{24}) \tag{1}$$

$$yhr = cos(2\pi \times \frac{hr}{24}) \tag{2}$$

For indicator features like city and region, we compute its frequency proportion of the whole training data. For example, there are 370 different cities in our dataset, and the frequency occurrence of city 2 is 1.55%, so each of the city id will be substituted by its occurrence proportion. The similar approach is also applied to feature region.

Dealing with other categorial variables could be more straightforward. What we did is to encode every category as a one hot encoding vector. This binary vector contains |i| (number of values in this category) elements where all columns are equal to zero except for the category column. Features like user agents, advertiser, user tag, advertiser will be processed in this way.

In sum, we have 127 features which for our training our model and click prediction.

3.1.3 *Down-sampling*.

Class imbalance is a common problem in online real-time bidding. Based on our data exploration, there are more than 2.4 million winning impressions in our dataset, however, the clicked ones only account for 0.073%. As most conventional classification algorithms are often biased towards the majority class, not taking the data distribution into consideration. The low number of clicks (1793) could make the training process extremely difficult. In order to deal with the problem and improve the bidding performance, our group decide to balance the proportion of class labels by under-sampling the majority class with 0 clicks. To explain in more detail, we empirically choose a set of negative sampling rate, ranging from 0.001, 0.01, 0.025, 0.05, 0.075, 0.1 and test the prediction accuracy of the learning model on the validation set. However, the performance of different models on different training size also varies. Our findings are summarised in Tablexx. The best performance for XGBoost achieved with 0.025 sampling rate, FM is at 0.1 and for DNN, it works better at 0.075 sampling ratio. Therefore, we train our individual model with different sized sample data. Ofparticularnoteis that after a model is trained in a negative sampled dataset, the output probability from a classifier should be calibrated. For example, if we use 0.1 sample rate to construct the training set, the empirical CTR will increase for about 10%. In order to re-calibrate the result to get back to the upsampled one, we re-calibrate the prediction according to the following equation:

$$q = \frac{p}{p + (1 - p)/w} \tag{3}$$

Where q is the calibrated probability, p is the prediction in undersampling space (predicted CTR) and w is the under-sampling rate.

3.1.4 *Evaluation metric.*

The whole bidding system is composed of two parts: a CTR estimator and a market price model. Different metrics will be applied to evaluate these two models.

CTR estimation model.

As the goal of a CTR Estimator is to predict whether a specific ad impression will be clicked under a given context, it can be simply treated as a binary classifier, in which the system is required to determine the click result. Therefore, percentage correct makes intuitive sense. However, although accuracy is a important to measure most classification model, it does not hold well against our highly imbalanced dataset. As more than 99.9% impressions are associated with 0 clicks, even if the model choose to always predict âĂIJlabel-0âĂİ, we can get an extremely high accuracy. Therefore, to avoid getting misleading result, we will first compute the confusion matrix for the true positive, true negative, false positive and false negative class for two types of predictions. For example, the true positive class exists when the actual click of is 1 and the prediction is 1 as well. Furthermore, based on the confusion matrix, we could calculate and discuss the precision, recall, F1 score and more importantly, plot the Receiver Operation Characteristic (ROC) curve and the Area Under the Curve (AUC) on a chart. The ROC curve plots the false positive rate (X-axis) against the true positive rate (Y-axis) (see equation 4) and 5)). If the AUC score is equal to 1, it implies all clicks are correctly predicted. Any values of AUC higher than 0.5 represents the modelsâĂŹ prediction is better than a random predictor.

$$False\ Positive\ Rate = \frac{FP}{FP + FN} \tag{4}$$

$$True\ Positive\ Rate = \frac{TP}{TP + FN} \tag{5}$$

Bid Price Estimation Model.

The performance of our bid price estimation model will be evaluated based on the following metrics, within a specified limited budget of 6250 CNY fen.

Clicks: This is the main metric of our bidding strategy, indicating the click number of our impressions won.

Click-Through Rate (CTR): CTR describes the the ratio of clicks to impressions won. A higher CTR means more visitors and better advertising performance.

Spend: This is the total amount of money we spent for all impression won. In our case, we have to spend all 6250 budget to win the impressions.

Average Cost Per Millie (CPM) and **Average Cost Per Click (CPC)**: This is the cost per 1,000 impressions and the average cost for every click. The lower CPM and eCPC are, the more cost efficiency an algorithm provides.

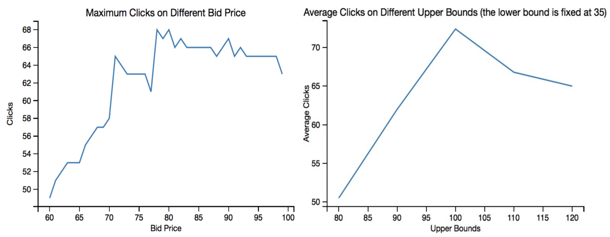

Figure 2: Constant Bidding and Random Bidding Strategy

Our main aim is to enhance the total number of clicks from the winning impressions under the limited budget. However, instead of solely based on that, we would also manage to obtain a higher CTR, and reduce the CPC as well as CPC to get the biggest economic benefit.

3.2 Basic Bidding Strategies

3.2.1 ***Constant biding strategy.***
The constant bidding strategy is a simple strategy which bids all impressions using a specific constant bid price. As the market price in the validation set ranges from 0 to 300, we decide to set the constant bidding price from 1 to 301 and count the total number of click. After testing all the 301 values on the validation set, we obtained the optimal bid price as 80 fen, which will generate 68 clicks (See Figure 2).

3.2.2 ***Random biding strategy.***
The random bidding strategy is optimized on the validation set with 6250 CNY fen budget, whose bid-price is generated from the uniform distribution. The upper bound and lower bound of the bidding range are tuned as parameters. The performance of this random bidding strategy is maximized with the lower bound equals to 35 and upper bound equals to 100 which can acquire 74 clicks on the validation set in average. The highest number of click achieved by this model could be 80.

When tuning the parameters, first, we fixed the lower bound and try different upper bounds in the range of lower bound plus 10 to 330 with an interval of 10. Then, different lower bounds in the range of 0 to 100 are tried. Since the performance of the random bidding strategy could vary from time to time, we run the model ten times for each pair of parameters and calculate the average clicks. The average clicks of each pair of parameters is plotted to help us to find the best pair of parameters.

Figure 2 shows the performance of random bidding strategy when the lower bound is fixed as 35. The x axis shows the value of upper bound and y axis shows the number of average clicks. We can see that the random bidding strategy obtains the highest average clicks, which equals to 74, with the lower bound equals to 35 and upper bound equals to 100.

3.3 Linear Bidding strategies

Besides using constant or random bidding strategies, the normal solution is to estimate or "learn" a bidding price from the information about the specific bidding. Among varieties of bidding strategies,

Table 1: Evaluation Metrics of Logistic Regression

TP	1
TN	303719
FP	4
FN	201
Accuracy	99.93%
Recall	0.50%
Specificity	99.99%
False Positive Rate	0.0013%
Precision	20%
F1 score	0.97%

linear bidding strategies are widely used, usually as a baseline due to its easy implementation, high efficiency and ability of interpretation. The linear bidding strategy consist of 2 parts, namely a linear CTR estimator and a linear bidding formula, of which the former is usually a linear machine learning algorithm to predict the CTR, and the later is a function linear to the predicted CTR to provide the bidding price.

3.3.1 *CTR estimation model.*
As a basic CTR estimator, Logistic Regression algorithm is applied. In this algorithm the prediction can be expressed as:

$$\hat{y} = \frac{1}{1 + e^{-w^T x}} \tag{6}$$

where x is the input data and y is the binary class to predict. W is the weight to be learned during the process of training, which is trying to minimize the cross entropy loss:

$$\mathrm{tt}\,(y, \hat{y}) = -y \log \hat{y} - (1 - y) \log (1 - \hat{y}) \tag{7}$$

After each iteration the weights are updated by Stochastic Gradient Descent Learning:

$$w \leftarrow (1 - \lambda)\, w + \eta\, (y - \hat{y})\, x \tag{8}$$

After the whole training process the weights will be optimized to predict the binary class in the prediction function. Here we use the implemented LR provided in Scikit-Learn package to simplify the implementing process. The data used to train the model here is preprocessed after the procedure described in section 3.1.2. And the hyper parameters are tuned by grid search method. Table1 shows the metrics of the LR model. The plot of its ROC is also provided.

3.3.2 ***Bid price estimation.***
Here we use the following equation to provide bidding prices, which is requires in the problem description and is simple to implement.

$$bid = base\_bid \times \frac{pCTR}{avgCTR} \tag{9}$$

The above learning model can output the probability whether the impression can be clicked, which is the predicted CTR(pCTR). Through this function the bid price will be linearly proportional to pCTR, where base bid is a parameter to be optimized on validation set. To find an optimal base bid price for linear bidding strategy, we run the bidding algorithm on the validation set with a range of different base bidding price, and choose the base bid with highest clicks.

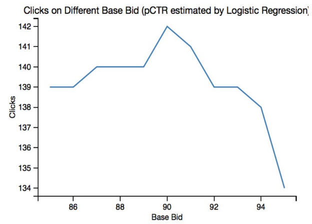

Figure 3: Constant Bidding and Random Bidding Strategy

Table 2: Click Performance of Linear Bidding Strategy

Strategy	Impressions	CTR	Click	Spend	eCPC
random	158194	0.00005	80	6101.29	76.27
linear	130789	0.00108	142	6189.07	43.59

Table 3: CTR Estimation Performance

Model	AUC	Accuracy	Precision	Recall	F1 Score
LR	0.794	99.93%	20.00%	0.50%	0.97%
FM	0.876	99.93%	46.94%	11.39%	18.33%
XGB	0.885	99.94%	81.25%	12.87%	22.22%
LGBM	0.892	99.85%	18.05%	36.63%	24.18%

3.3.3 *Logistic Regression+Linear Strategy*.

Figure3 shows different number of clicks gained when using different base bid with the LR model on validation set. It obviously shows that the optimal base bid is 90 with the highest clicks to be 142.

Table 2 clearly shows the much better performance of the linear strategy, with 62 more clicks and twice the CTR of random bidding. Therefore, the results here proves the solution of using machine learning algorithms to predict clicks and bidding based on the prediction.

3.4 Combined Bidding strategies

3.4.1 *Separate Bidding strategy Analysis*.

Our group have developed a total of four different models for pCTR estimation. The first one is a simple Logistic Regression Model, the second one is based on Factorisation Machine (FM), the last two applies scalable boosting methods XGBoost and LightGBM. Details of the model implementation can be found in individual report and their performance results have been summarised in Table 3.
Compared with the baseline LR model, all three improved models have displayed a far better performance in most aspects. Even though the LGBM does not achieve an accuracy higher than LR, this can be attribute to the characteristics of highly imbalanced dataset. However, if we focus on AUC, we could find the three models have generated an 10% higher scores than LR. As the AUC scores can be seen as useful metric for datasets with highly unbalanced classes,

Table 4: Click Performance of different Bidding Strategies

Strategy&Mode	LR	FM	XGB	LGBM
Constant			68	
Random			74	
Linear	142	169	167	167
ORTB1	-	-	166	-
ORTB2	-	-	166	-
PRUD	-	159	-	-

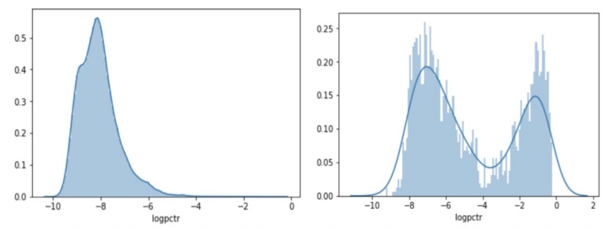

Figure 4: Log pCTR distributions by Xgboost

this reflect the overall performance of the models. Additionally, XGBoost has achieved the best precision, which means it has an advantage of identifying negative labels where as LGBM performs best in terms of finding all the positive samples (click=1).

To generate the final bidding price, five bid estimators have been applied and combined with these CTR predictors. The click number results of different strategy combinations has been shown in Table 4. Each row in 4 represents a single bidding price generation strategy and the columns list different CTR prediction models. The Values area displays the number of clicks generated by a particular strategy combination, which is the result tested on the validation set.

By comparing the click performance of different combinations, we could find the Linear, ORTB and PRUD algorithm generally works much better than Constant and Random bidding strategy. As the latter strategies takes the predicted CTR into consideration, this situation verifies the significance of our impression value evaluation. Furthermore, even though Linear bidding strategy is relatively straightforward, it can always obtain a higher click number than others by simply tunning the base_bid and to adapt different bidding circumstances. Therefore, we decide to use this biding strategy in our combined bidding model.

A novel strategy is proposed by observing the log pCTR distribution. In figure 2, the left distribution corresponding to the impressions that are not clicked and the right one corresponding to the impressions that are clicked by people. We can see that the pCTR distribution for clicked impressions can be divided into two parts in the middle of two peaks. The right part corresponding to the positive impressions that can be well distinguished by our model, while the left part is similar to the pCTR distribution of impressions without clicks. However, we can still observe a log pCTR threshold at around -9 below which the impression are highly unlikely to be clicked. Therefore, we decided to set the bid price of impressions whose log pCTR is smaller than -9 to 0. Another -8.5 threshold is also experimented. The experimental results shows

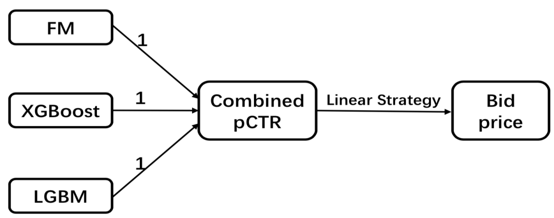

Figure 5: Pipeline of Bidding Strategy Combination

Table 5: Click Performance of Combined Bidding Strategy

Model	CTR	Click	Spend	eCPC
FM+XGB;Valid	0.00132	170	6246.84	36.75
FM+XGB;Test	0.00141	181	6232.27	34.43
FM+XGB+LGBM;Valid	0.00129	169	6229.99	36.86
FM+XGB+LGBM;Test	0.00140	183	6237.99	34.09

Table 6: Click Performance of Combined Bidding Strategy

Model	CTR	Click	Spend	eCPC
FM+XGB;Valid	0.00132	170	6246.84	36.75
FM+XGB;Test	0.00141	181	6232.27	34.43
FM+XGB+LGBM;Valid	0.00129	169	6229.99	36.86
FM+XGB+LGBM;Test	0.00140	183	6237.99	34.09

that this method can improve CTR significantly but may not be very helpful in terms of Clicks (see table 7).

3.4.2 ***Combined Bidding Strategy**.*

In order to further enhance the bidding performance, our team decide to combine different pCTR estimation models into one predictive model. With an aim of decreasing variance, reducing bias and improving the predictions, such ensemble method is commonly used in machine learning. In our CTR prediction case, each model can be seen as a weak classifier. By analysing their pCTR estimation performance, we find FM, XGBoost and LGBM share approximately the same prediction result. Therefore, we average together multiple estimations from these models (average the pCTR outputs from three models) and use this new pCTR to represent the final click-through rate of the impressions in the validation set. The pCTR of the impressions in the test set will be predicted in the same way.

After generating the pCTR, we need to choose a best bidding strategy to compute the final bidding price. Initially, we would like to apply the linear bidding strategy as it always provides us the best click performance when combined with individual pCTR estimators. The combined pCTR of the validation set will be used to search for an optimal base_bid. Once the base_bid has determined, the final bid price of the test set impressions can be obtained.

Table 7: Applying pCTR Threshold

threshold	impression	CTR	Click	Spend	eCPC
no threshold	136252	0.00131	179	6217.91	34.73
logpctr>-9	131749	0.00135	179	6165.07	34.44
logpctr>-8.5	116352	0.00153	5911.27	33.21	

Table 8: Original Bidding Model Performance

Impression	Spend	Clicks	CTR	avgCPC
1744	2461.99	12	0.006881	205.17

Table 9: Modified Model Performance (add constant)

Constant	Impression	Spend	Clicks	CTR	avgCPC
0	2040	2331.46	19	0.009314	122.71
10	2409	2448.19	19	0.007887	128.85
30	4115	2936.99	19	0.004617	154.58

3.5 Multi-agent Bidding strategies

Based on our best single-agent bidding model, a multi-agent bidding model is proposed. The single-agent model is improved to be more adaptable to the multi-agent situation.

Since our test environment is shared with other research groups, the test results could vary due to the different opponentsâĂŹ bid-price. Therefore, we always test new multi-agent models and the original single-agent model at the same time and compare their results to evaluate if the new models have better performance in this uncertain environment.

First, the original single-agent bidding model is tested in the multi-agent environment. From table 8 we can see that around 40% of budget is spent and the numbers of impressions and clicks are low. It seems that the multi-agent environment is more competitive, so our initial model cannot win much impressions and the budget is not fully utilized.

We assume that the bid-price distributions of all opponents are fixed. To improve the probability of wining, we can move our bid-price distribution to the right to some extent by adding a constant to our original bid-price. From table 9, we can see that, the impression increases significantly after adding the constants, while the number of clicks remains the same. Although, this method increases our probability of wining, the performance becomes worse regarding CTR and avgCPC. There are much more impressions with no click than impressions are clicked, so if we add the same constant to each price, we could waste much budget on impressions with no click.

Therefore, to maximum the number of clicks with limited budget, the bid-price of impressions are more likely to be clicked should be raised more. The pCTR shows the probability that an impression being clicked, so the increase of bid-price should be positively related to pCTR. As explained before, the bid-price generated from our single-agent model is positively related to the pCTR, so we multiply the original bid-price by a factor to generate the new bid-price. The table 10 could suggest that the model performance can be

Table 10: Modified Model Performance (Multiply Factor)

Factor	Impression	Spend	Clicks	CTR	avgCPC
1	1683	2489.67	22	0.013072	113.17
1.3	5956	6229.20	43	0.007220	144.87
1.5	5885	6250.23	39	0.006627	160.26

maximum with the factor equals to 1.3 when the number of clicks doubles, but the CTR and avgCPC become slightly worse than the original model. This 1.3 factor model also achieved the first rank in the test environment. When the factor is set as 1.5, the number of clicks begins to decrease due to the lack of budget.

Other factors like 1.1 and 1.2 are also tried, these results are not presented due to the space limitations. We cannot compare these results directly due to the variation of the test environment, but by comparing with the original model, the experimental results suggest that a factor in the range of 1.2 to 1.3 should be able to fully optimize our model in the multi-agent situation.

4 CONCLUSION

Based on the data exploration which is introduced in our individual report, the most simple constant bidding and random bidding strategies are implemented at first as a start point. Both of the 2 strategies are described in detail and well tuned, with 68 and 74 average clicks on validation set.

Then a linear bidding strategy consisting of logistic regression as the CTR estimator and a linear formula as the bidding algorithm is implemented. The detail description of linear strategy is provided in this report. The hyper parameters of both parts of the strategy are well tuned, resulting in a high number of clicks to be 142 on validation set.

Taking the linear strategy as the baseline to compared with, varieties of bidding strategies are implement and tested by us, which is described in detail in our individual reports. Then the performance of the models are evaluated and compared. Models with fairly good performance are selected and combined together, forming a better model with higher prediction ability. The combining method is described in detail in this report and result proves the ensemble theory in machine learning, that is, multiple "weak" learners can be combined together to form a "strong" learner, as the process of combining is in fact where different learners correcting faults of each other. Besides, multi-agent bidding strategy is also attempted. Both of our final strategies are well tuned with best results. Until the report is written up, our single-agent strategy ranked No.2 on the hidden test set and the multi-agent one ranked No.1. However, although no efforts is spared by us to obtain the best bidding model with best performance, there are still potential improvements due to the limited time. For example, ORTB should have better results than linear bidding algorithm, especially when the CTR estimator is becoming stronger, so ORTB may be tested further as we ran out of chances on the test set. Besides, the deep learning method doesn't bring good results described in many papers. This may be tested on more complex dataset.

REFERENCES

[1] Weinan Zhang, Shuai Yuan, and Jun Wang. Optimal real-time bidding for display advertising. In Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining, pages 1077âĂŞ1086. ACM, 2014.

[2] Ren K, Zhang W, Chang K, Rong Y, Yu Y, Wang J. Bidding Machine: Learning to Bid for Directly Optimizing Profits in Display Advertising. IEEE Transactions on Knowledge and Data Engineering. 2018;30(4):645-659. doi:10.1109/tkde.2017.2775228.

[3] Cai H, Ren K, Zhang W, et al. Real-Time Bidding by Reinforcement Learning in Display Advertising. Proceedings of the Tenth ACM International Conference on Web Search and Data Mining - WSDM 17. 2017. doi:10.1145/3018661.3018702.

[4] Zhang et al. Deep Learning over Multi-field Categorical Data âĂŞ A Case Study on User Response Prediction. ECIR 16] in Monday Machine Learning Track

[5] X. He, J. Pan, O. Jin et al., âĂIJPractical lessons from predicting clicks on ads at facebook,âĂİ in Proceedings of the Eighth International Workshop on Data Mining for Online Advertising. ACM, 2014, pp. 1âĂŞ9.

[6] Product-based Neural Networks for User Response Prediction https://towardsdatascience.com/catboost-vs-light-gbm-vs-xgboost-5f93620723db

[7] Weinan Zhang, Shuai Yuan, Jun Wang, and Xuehua Shen. Real-time bidding benchmarking with ipinyou dataset. arXiv preprint arXiv:1407.7073, 2014.

[8] https://github.com/BlaBlaPer/WebEconomics