COMP0124 Multi-Agent Artificial Intelligence Group Project (Group 17)

Su Shen
University of London
sushen.18@ucl.ac.uk

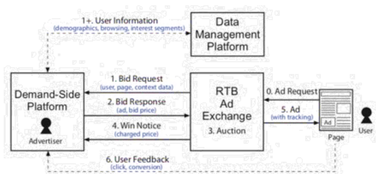
Qiqi Zeng
University of London
qiqi.zeng.18@ucl.ac.uk

Yijun Wang
University of London
yijun.wang.18@ucl.ac.uk

1. INTRODUCTION

Real-time bidding (RTB) for online display advertising has promoted the development of the advertising filed in recent years. Business Insider Intelligence estimated that RTB would grow 16% in 2013 to 33% in 2018 [1]. It is possible for advertisers to buy an individual impression in real time with RTB mechanism. The process of RTB is as follows: first, when a user is loading on a website, the advertisement request will be sent to the DSP (Demand-Side Platforms) bidding agent. Then, advertiser give their bid response back to the RTB Ad Exchange. Since the auction is the second price auction model, the winner of the auction will pay the second highest offer. Finally, the advertisement will be present to the user through RTB Ad Exchange and the user will response. Figure 1.1[2] briefly illustrates the interactions described above. The whole process always takes place within 10-100ms [3].

Figure 1.1: Real-time Advertisement bidding process [2]



To generate maximum profit in such real-time bidding, bidding strategies play an important role.

This report proposes bidding strategies consisting two components to allocate a bid for each impression:

- **CTR prediction model based on auction features:** *using Logistic regression model, Random Forest model, XG Boosting model (Extreme Gradient Boosting), Adaboost model, Factorization Machine model and Neural Network*
- **Bidding price estimation model based on predicted CTR:** *using linear bidding model, non-linear bidding model (Squared bidding strategy, Exponential strategy and Optimal Real Time Bidding Strategies) and Multiagent bidding strategy*

The CTR prediction model components need to estimate CTR with various features from RTB dataset such as the time, location, user-agent, publisher's ad slot size, ad exchanges, and user tags. Then, based on CTR prediction, the most suitable bid prices from the bidding price estimation model are offered for bidding.

From the result of CTR prediction model, *XG Boosting model*, which obtains highest Area Under Curve (AUC) of ROC, is the best model to predict CTR. Based on the evaluation metrics, among all bidding strategies including basic bidding strategies, linear bidding strategy and non-linear bidding strategy, the best model is *squared bidding strategy*, generating the highest number of clicks (171 clicks) on the validation set.

In this paper, we first do exploratory data analysis on the RTB dataset. Then, we figure out the optimal constant bidding strategy, the optimal bound for random bidding strategy and multiagent random bidding strategy separately. Besides, we compare the performance metrics of above basic strategies on the RTB validation set. Then, we apply several machine learning models to train the CTR prediction. Furthermore, based on CTR prediction, we apply linear and non-linear bidding strategy and tune parameters for model optimization on the validation set. We compare and conclude the best bidding price estimation model, which generates highest number of clicks. The code and the report are uploaded to Github[1].

2. LITERATURE REVIEW

For online advertising, the prediction of Click-Through-Rate (CTR) is a widely research topic. There are different models to predict CTR. For example, Gouthami et al. used Logistic regression model to predict Ad click on the Avazu data [4]. Kathryn et al. used random forest model to predict clicks in mobile advertising [5]. Ilya et. al used boosted tree model to predict CTR for sponsored search [6]. Oentaryo et al. applied Factorization Machine model to predict response in mobile advertising [7]. Jun Wang et al. proposed Product-based Neural Networks (PNN) with an embedding layer for user response prediction [8]. Sen Zhang et al. built Ada boost model for the imbalanced learning sample distribution to predict CTR [9]. Besides, XGBoost model is an open source implementation of gradient boosted trees that demonstrated state of art results in many domains and achieves high performance [10].

For bidding price estimation, linear bidding strategy is widely used in recent advertisement area [3]. Besides, many researches adapted complex bidding strategy to adapt real-world bidding and consider a series of game processes. Zhang et al. concluded that non-liner bidding strategy is practically most effective strategy [3]. Jian Xu et al. built lift-based bidding strategy to price biding request based on lifted value [12]. These contributions of these researches enrich the bidding strategies in real time bidding.

For the real-world bidding environment involving completion among different bidders, there is a phenomenon of information asymmetry. In response to it, Panos adapted Postpone strategy [13]. Begemann and Pesendorfer concluded the optimal mechanism that discriminately disclose the private information with regard to its accuracy to other bidders [14]. Besides, Eyster and Rabin defined and applied Cursed equilibrium in a Mixed Bidder Population environment including rational and irrational bidders [15].

3. APPROACH AND RESULTS

3.1 Data Exploration

In this paper, the basic statistical analysis of the dataset includes the number of impressions, number of clicks, cost, CTR (Click-Through-Rate), eCPC (Effective Cost per Click), average CPM (Cost per Mile), and also includes further analysis on user feedback

[1] Github URL: https://github.com/ucabs06/Multi-agent-AI_group-assignment

and bidding behaviors.

3.1.1 Basic Statistical Analysis on the Dataset

In the given training set, based on the column advertiser, there are nine advertisers. Table 1 shows the basic statistical analysis on the training set. Column 1 shows the advertiser ID. Column 2 shows the number of impressions of individual advertiser. Column 3 shows the number of clicks that users click on each advertisement. Column 4 shows the cost of each advertiser spent following the second price auction model. Column 5,6,7 shows CTR, eCPC and Avg CPM respectively, which are calculated using the following formulas:

Click-Through-Rate (**CTR**) is:

$$CTR = \frac{\text{Total number of Clicks}}{\text{Total number of Impressions}} \qquad (1)$$

Effective Cost per Click (**eCPC**) is

$$eCPC = \frac{\text{cost}}{\text{Total number of Clicks}} \qquad (2)$$

Average Cost per Mile (**Avg CPM**) is

$$AvgCPM = \frac{\text{cost} \times 1000}{\text{Total number of Impressions}} \qquad (3)$$

The Cost use the currency of CNY, and the unit is Chinese fen.

Table 1. Basic Statistical Analysis on the training set

Advertiser	Imps	Clicks	Cost	CTR	eCPC	Avg CPM
1458	492353	385	33968.7	0.08%	88.23	68.99
2259	133673	43	12428.2	0.03%	289.03	92.97
2261	110122	36	9873.8	0.03%	274.27	89.66
2821	211366	131	18828	0.06%	143.73	89.08
2997	49829	217	3129.3	0.44%	14.42	62.8
3358	264956	202	22447.2	0.08%	111.12	84.72
3386	455041	320	34931.8	0.07%	109.16	76.77
3427	402806	272	30458.7	0.07%	111.98	75.62
3476	310835	187	23918.8	0.06%	127.91	76.95

The higher CTR and lower eCPC, the better bidding strategy, because the objective is to maximize the number of clicks on advertisements and minimize the cost each advertiser spends.

From Table 1, the Advertiser 2997 stands out with CTR of 0.44%. CTR of other eight advertiser are less than 0.1%. The factors of influencing CTR of each advertiser are shown in section 3.1.2.

The average CPM of the nine advertisers are similar to the range of 60 to 95. However, for the eCPC, among nine advertisers, the Advertiser 2997 has the lowest eCPC of 14.4. The Advertiser 2259 has the highest eCPC of 289.03.

3.1.2 Analysis on the User Feedback

In this section, we show some statistics of user feedback. For clarity of presentation, we just choose the Advertiser 1458 and 3358. Specifically, Figure 1 shows the CTR distribution against different features such as the time, location, user-agent, publisher's ad slot size, ad exchanges, and user tags.

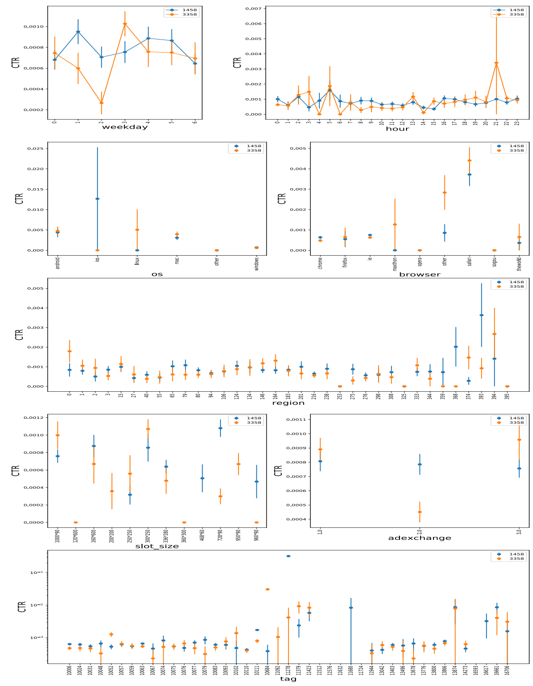


Figure 1: CTR distribution against different features for the Advertiser 1458 and 3358.

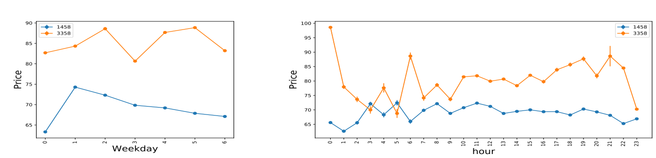
From Figure 1, we can observe some features that influence CTR for advertisers differently. It can be concluded that:

- We observe that Advertiser 1458 has received the highest CTR on Tuesday and at 5 AM, while Advertiser 3358 has the highest CTR on Thursday and at 9 PM.
- For the Advertiser 1458, iOS as the user platform appears has the highest CTR, while Linux for the Advertiser 3358 has the highest CTR. The two advertisers have the highest CTR through the Safari browser.
- The trend of CTR of the two advertisers across different region location are different.
- For the slot size, the 1000*90 and 300*250 have relatively higher CTR for both advertisers.
- The CTR distribution of two advertisers are not same on these ad exchanges.
- The CTR is different for the various tags of the users.

In conclusion, from these observations of the user feedback, CTR distributions are particular for different advertisers.

3.1.3 Analysis on the Bidding Behaviours

In this section, we show some statistics of bidding behaviours. Figure 2 shows the pay price distribution of Advertiser 1458 and 3358 against different features such as the time, location, user-agent, publisher's ad slot size, ad exchanges, and user tags.



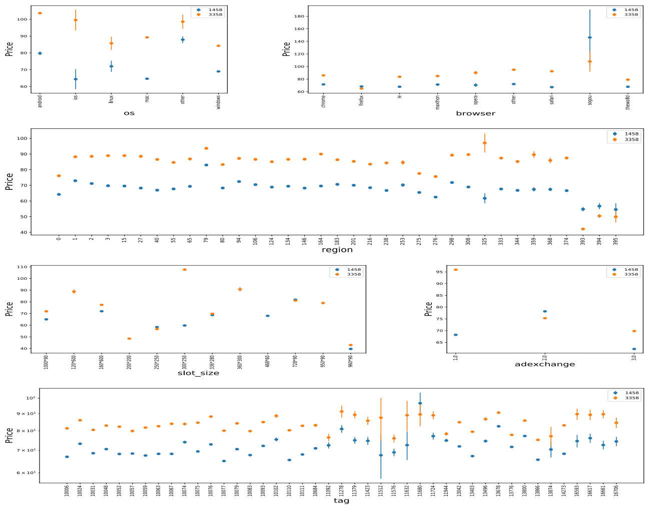


Figure 2: Pay price distribution against different features for the Advertiser 1458 and 3358.

From Figure 2, it can be observed that the bidding behaviours of each advertiser are different. For example:

- The pay price of Advertiser 1458 is higher in the morning than that in the afternoon and evening. While, the pay price is highest for Advertiser 3358 at 0:00. The high price may reflect the time period of high CTR for each advertisement.
- The pay price is influenced by the slot size. The suitable slot size determines the location of the advertisements on the web page leading high attention of users.
- For Advertiser 1458, the competitiveness in ad exchange 1 is highest compared with ad exchange 2 and 3, while the competitiveness in ad exchange 2 is highest for Advertiser 3358.

Compared with CTR distribution against different features, we observe that the volatility of the pay price distribution is lower, because the prices are integers, while the clicks are binary.

3.2 Basic Bidding Strategies

The performance metrics of basic bidding strategies on the validation set is exhibited on 3.2.4.

3.2.1 Constant Bidding Strategy

For constant bidding strategy, the bid price is a predefined constant value, which is the only parameter for this strategy.

To obtain the optimal constant value, the first step is to test the constant in the range from 0 to 300 with the step length 1. To improve optimization time, we apply vector calculation instead of for loop. Then, we plot the Click graph to find the optimal constant value with the highest Click. Based on the training RTB dataset, the Click graph is exhibited on Figure 3. The performance metric is exhibited on Table 2.

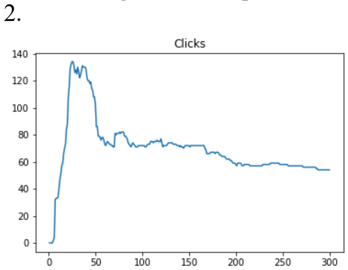


Figure 3: Click graph on the training set

bid price	Imps	Click	CTR	eCPC	CPM	Spend
25	411,910	134	0.03253%	46.642	15.173	6249.991
26	404,647	134	0.03312%	46.642	15.445	6249.984

Table 2: Constant bidding strategy Performance metric on the training set

From the above click graph, there exists a peak in the bid price range between 20 and 30. It means for other RTB dataset, the optimal bid price with the highest clicks is likely to be picked from 20 to 30. With the constant 25 and 26 bidding strategy, they both generate the highest clicks number 134. Since 26 bid price generates higher CTR compared with 25 bid price, the optimal bid price is 26 CNY fen.

3.2.2 Random Bidding Strategy

In terms of random bidding strategy, the bid prices are generated randomly in a certain range. The lower bound and the upper bound are the parameters for such strategy.

To figure out the optimal parameters, the first step is to generate the parameters between 0 and 300 with the step length 5, setting the lower bound smaller than the upper bound. To improve optimization time, we apply vector calculation instead of for loop. Then CTR heatmap with combinations of lower bounds and upper bounds is exhibited on the Figure 4. The performance metric is exhibited on Table 3.

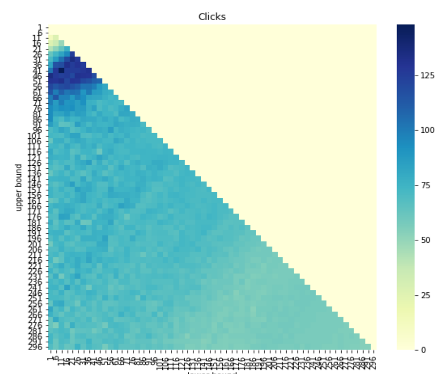


Figure 4: Click heatmap on the training set (Random bidding strategy)

lower bound	upper bound	Imps	Click	CTR	eCPC	CPM	Spend
16	**41**	373,411	142	0.03803%	44.014	16.738	6249.999

Table 3: Random bidding strategy Performance metric on the training set

From above analysis, the optimal lower bound is 16, and the optimal upper bound is 41 on training dataset. The random bidding strategy generates highest number of clicks (142 clicks).

3.2.3 Multiagent Random Bidding Strategy

Based on the random bidding strategy, this strategy considers the competitions among homogeneous random bidding agents. n players bidding at the same time and each of them bids a random price ranging from a defined lower bound to upper bound with 6250 CNY fen and stop bidding as long as they run out of the budget.

To implement this strategy, it will check if the bidder with the highest bids has enough budget, and if not, the right of purchase will be passed to the next bidder until the budgets of all n bidders run out. Therefore, it is hard to avoid for loop, and the optimization speed slows down. To reduce the optimization time, we first check the lower bound and upper bound ranging from 1 to 300 with step length of 30, based on training data. Then, we plot the CTR heatmap with combinations of lower bounds and upper bounds in this range. The CTR heatmap is exhibited on the Figure 5.

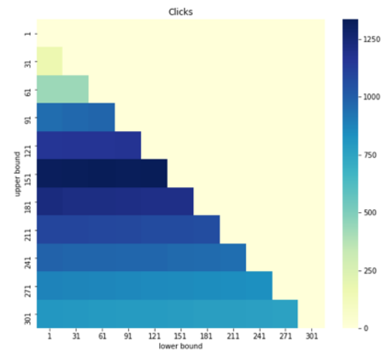


Figure 5: Click heatmap on the training set (Multiagent Random bidding strategy)

As the figure above shows, the best performance interval of upper bound is from 121 to 181. Thus, we search again within a more specific interval, ranging from 120 to 175 with step length of 5. The new CTR heatmap is exhibited on the Figure 6.

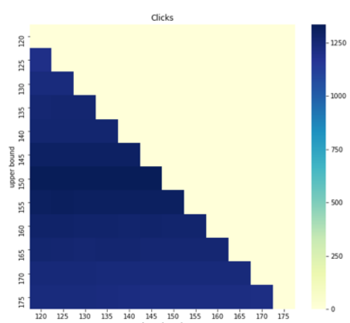


Figure 6: New Click heatmap on the training set (Multiagent Random bidding strategy)

From above Figure 6, at first glance, it seems that the highest click number (about 1250 clicks) is obtained at 150 regardless of the value of lower bound. In fact, the multiagent random bidding strategy generates the highest number of clicks (1336 clicks), when lower bound is 145 and upper bound is 150, which is quite different from the random bidding strategy.

This might because the highest bidding of each round is picked out from n biddings, so it is close to the upper bound, and it is not affected by the lower bound. Besides, compared with single random bidding strategy, this strategy can win much more total clicks due to the more abundant budget. When a single bidder runs out of the budget, another bidder may purchase that impression so that they can get more total impressions, increasing the probability of getting more clicks. However, the average clicks of each bidder drop significantly, this is because it becomes more difficult to bid not only higher than pay price but also bid prices from other competitors.

To investigate the impact of number of bidders (n), we test the total clicks, average clicks of each agent, and their CTR on the validation set, and the result is showed on Figure 7.

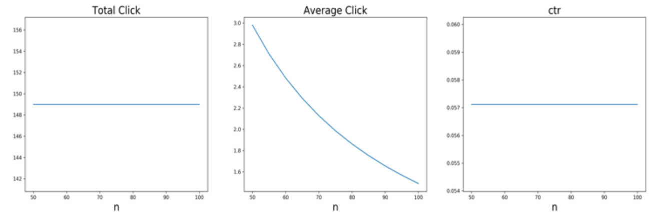


Figure 7: Result of impact of number of bidders

As this Figure 7 shows, the bidding agent size n is ranging from 50 to 100. As the agent size increases, it seems that the total clicks and CTR are not affected by n. However, the average click decreases gradually due to the increase of n.

3.2.4 Performance Metrics on validation set

Table 4 tabulates the performance metric for all basic bidding strategy on the validation set.

	Constant Bidding	**Random Bidding**	**Multiagent Random Bidding (Average)**	**Multiagent Random Bidding (n=50)**
Imps	60,444	60,955	1,219	260,857
Click	16	20	3	149
CTR	0.0265%	0.0328%	0.0571%	0.0571%
eCPC	58.416	50.963	5.217	260.857
CPM	15.463	16.722	2.980	148.999
Spend	934.656	1019.261	777.35	38,867.63

Table 4: Basic bidding strategies Performance metric on the validation set

Above performance metrics shows that constant bidding strategy and random bidding strategy have similar performance in terms of the number of clicks. However, the multiagent random bidding strategy generates more clicks than the other two strategies. The reason is that the increase in bidders has increased the overall budget, thus increasing the total number of clicks. However, due to competition between bidders, the average number of clicks is reduced.

In these three strategies, bidding for impressions is blind without any mechanism, resulting in missing the impressions with high probability of generating clicks.

Therefore, in order to increase the number of clicks and buy the best-matched impressions, the bidding strategy should use different features of advertisers, historical data and learning patterns through machine learning.

3.3 Linear Bidding Strategy

For the Linear Bidding Strategy, it is assumed that the bid value is linearly proportional to the predicted CTR. The formula is

$$bid = \text{base}\_bid \frac{pCTR}{\text{avg}\, CTR} \qquad (4)$$

pCTR is the predicted probability of the user click on its impression, which can be determined with a pre-trained CTR estimator. We applied *Random Forest and XGBoosting (Extreme Gradient Boosting), Adaboost, Factorization Machine, Neural Network, Support Vector Machine* methods to train the CTR prediction model. For model optimization, the hyperparameters tuning is implemented through an exhaustive grid search and the metric Area Under the Curve (AUC) of Receiver Operating Characteristic (ROC) is used for performance evaluation. Average CTR is the average click through rate on the training dataset and the base_bid is the parameter of the linear bidding strategy that would be tuned.

3.3.1 Data Preparation

Training, validation and test datasets contain the same attributes except the test only includes data prior to bidding. We preprocess them to ensure a better data manipulation in the following model construction. Some variables such as IDs, IP, domain, URL, keypage, creative are filtered out as they are so specific and if these features are included for model training, the model may prone to overfitting. The variable useragent is split into two binary features the system and the browser. Each Usertag feature contains multi-value separated with commas, we split them into binary features and totally found 68 unique usertags to categorize impressions. All other categorial variables such as weekday, hour, region etc. are processed with one-hot encoding, contributing to 872 features finally for building pCTR estimator. When training model, variables appear after the bidding are excluded, namely biding prices, paying prices.

3.3.2 Down Sampling

Through initial data exploration, it was found that the dataset is extremely large and indicates a high imbalance of non-clicks against clicks. Xinran He et al. investigated down sampling method to solve the imbalance problem about class of Facebook RTB dataset [11]. With unbalanced data sets, machine learning classification algorithms are supposed to produce degenerated models that do not take into account the minority class and bias the prediction model towards the more common class [16]. In this case, it may predict non-clicks for all samples and can still result in really high accuracy (99%). An efficient and most common way to mitigate the problem is either over-sample the minority class or down-sample the majority class [17]. In this case, we down-sampled the majority class which is the non-clicks data.

3.3.3 CTR Prediction

3.3.3.1 Modelling

Training a robust CTR estimator is rather significant because an accurate prediction would support for developing effective bidding strategy and obtaining proper bid prices that can maximize the clicks within a given budget (6250 CNY fen). Based on initial literature review and research, several machine learning technics including *Random Forest and XGBoosting (Extreme Gradient Boosting), Adaboost, Factorization Machine, Neural Network, Support Vector Machine* are applied to train the model. Among them, XGBoosting, Adaboost and Random Forest yield best three results after parameters tuning, with AUC of ROC score at 0.897, 0.849, 0.827 respectively.

Random Decision Forests are a typical ensemble method, which aims at making up the inherent defects of a single model or a mode with a particular set of parameters with integrating multiple learning models and gain better predictive results. Decision trees utilize a top-down method where the root node of the tree creates binary splits until certain criteria is met. Each node represents an attribute of a given advertisement impression. From the root node to the terminal node, the binary split of nodes provides a predicted value that serve as a criterion for classifying it into a particular target class. With the method of bootstrap aggregating, it contributes to a 'forest' of such decision trees and generate a more accurate prediction with relatively lower variance through averaging their results.

XGBoosting algorithm is a highly effective machine learning method that becomes quite popular and attracted much attention because of its outstanding performance in many Kaggle competitions [18]. It is a sequential technique that works on the tree ensembles, which combines a set of weak learners. At first, assuming we have K tress, the predicted output of a tree ensemble is

$$\hat{y}_i = \phi(\mathbf{x}_i) = \sum_{k=1}^{K} f_k(\mathbf{x}_i), \quad f_k \in \mathcal{F} \qquad (5)$$

Where $\mathcal{F}$ is the space of classification and regression tree (CART). The objective model for the tree ensemble is the sum of training loss and complexity of the trees.

$$Obj = \sum_{i=1}^{n} l(y_i, \hat{y}_i) + \sum_{k=1}^{K} \Omega(f_k) \qquad (6)$$

The equation (6) contains functions instead of just numerical vectors as parameters so that it cannot be optimized with traditional optimization methods, that is why Boosting method is introduced. Several feature algorithms contribute to its effective outcome. XGBoost takes into account sparse values in the training data, helping to specify the default direction of the branch for missing or specified values, which can significantly improve the efficiency. Also, its block structure supports the parallelization of tree construction.

Adaboost (Adaptive Boosting) classifier is a meta-estimator that at first fit a primary classifier on the dataset and constantly add new classifiers with adjusted weights. The final classification equation is as follows:

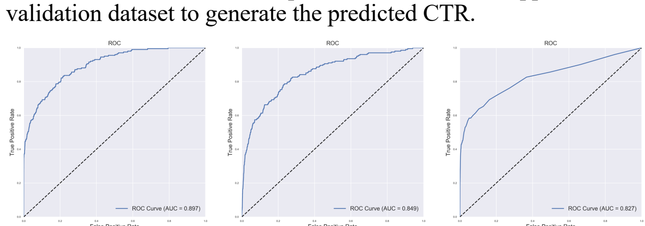
$$F(x) = \operatorname{sign}\left(\sum_{m=1}^{M} \theta_m f_m(x)\right) \qquad (7)$$

Where f_m represents m^{th} weak classifier with the corresponding weight θ_m.

To optimize the prediction precision and generate a high AUC of ROC score, we built a grid search model for every combination of hyperparameters specified, to search for the best combination. It was determined that for the Random Forest model, number of estimators 100, maximum depth 3 are the final best parameter set. With regard to XGBoosting, the number of estimators is 200, the max-depth is 5 and the learning rate is 0.1.

3.3.3.2 Evaluation

To evaluate the performance of CTR estimator model, we employed the AUC (Area Under the Curve), which is the area under the ROC (Receiver Operating Characteristic) curve, it ranges from 0 to 1 and larger value represents better prediction precision. Figure 8 shows the AUC-ROC curves for Random Forest and XGBoosting model. It was determined that XGBoosting obtained the highest AUC score 0.897. Then, this pre-trained model was applied on the validation dataset to generate the predicted CTR.



XGBoosting AUC = 0.897 Adaboost AUC = 0.849 Random Forest AUC = 0.827

Figure 8: AUC Score

3.3.3 Tuning Parameter – Basebid

The linear Bidding Strategy is defined based on formula 4 and the current objective is to optimize the basebid value so that the bid price obtained with this linear bidding strategy may generate the best performance within a limited budget 6250 CNY fen. This process starts with a relatively large basebid range to gain initial idea and narrow the interval gradually. At first, we obtain 50 evenly spaced numbers over 1 to 1000. It is found that higher click amount occurs in the range 40-350. This is set as a new interval for next trial and we expend basebid value amount to 100 through decreasing the step-length. Figure 9 shows the trend of click amount and CTR among different basebids. The highest number of clicks is 170, with the basebid equals to 190 and the CTR equals to 0.135. The performance of Linear Bidding Strategy is significantly better than constant and random bidding strategies.

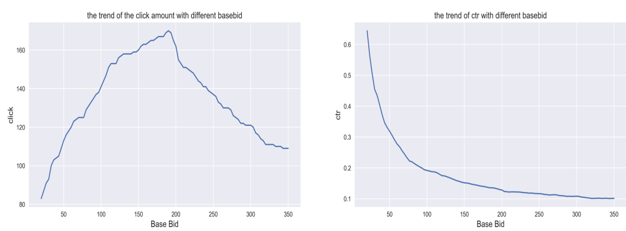


Figure 9: The trend of click amount and CTR with different basebids

3.4 Non-Linear Bidding Strategy

With regard to Non-Linear Bidding strategy, three models *squared bidding strategy, exponential strategy and Optimal Real Time Bidding Strategies* are applied. The squared bidding mechanism is characterized by the following equation.

$$bid = \text{base\_bid} \times \left(\frac{pCTR}{\text{avg}\,CTR}\right)^2 + \text{m} \times \frac{pCTR}{\text{avg CTR}} + n \qquad (8)$$

With the basebid, m and n are parameters supposed to be optimized on the validation data using grid search method mentioned before. It is found that a list of parameter combinations can result in the largest click amount (171), among them the highest CTR is 0.146%, with the basebid set as 166.67, m equals to 105.55 and n equals to 2.

Second, the bidprices obtained with the exponential strategy would be as follows:

$$bid = \text{base\_bid} \times e^{pCTR/avgCTR} \qquad (9)$$

In this case, basebid is the only parameter to be tuned by iteratively narrowing its range.

Third, the idea of Optimal Real Time Bidding Strategy is mainly derived from “Optimal Real-Time Bidding for Display Advertising” [14]. The objective of the campaign is to maximize the direct visits, i.e. the number of clicks, which can be formulated as:

$$b()_{\text{ORTB}} = \arg\max_{b()} N_T \int_{\boldsymbol{x}} \theta(\boldsymbol{x}) w(b(\theta(\boldsymbol{x}))) p_x(\boldsymbol{x}) d\boldsymbol{x} \qquad (10)$$
$$\text{subject to } N_T \int b(\theta(\boldsymbol{x})) w(b(\theta(\boldsymbol{x}))) p_x(\boldsymbol{x}) d\boldsymbol{x} \le B$$

Where N_T is the estimated amount of bid requests, every bid request is represented by x, which contains information of given impression. θ(x) is the predicted click through rate. And w(b(θ(x)) is the probability of winning the impression for a particular bid value with the feature x.

If the Lagrangian mechanics is introduced, we can obtain

$$\lambda w(b(\theta)) = [\theta - \lambda b(\theta)] \frac{\partial w(b(\theta))}{\partial b(\theta)} \qquad (11)$$

According to the paper, the probability of winning the auction is actually a concave function, this means the growth of winning rate at a low bid price is larger than that of a relatively higher bid. Therefore, the winning function would be

$$w(b(\theta)) = \frac{b(\theta)}{c + b(\theta)} \qquad (12)$$

If we get its derivation and plug together with (11) and (12) into (10), the final bid function can be

$$b_{\text{ORTB1}}(\theta) = \sqrt{\frac{c}{\lambda}\theta + c^2} - c \qquad (13)$$

Weinan[3] also mentioned another scenario, such as a highly competitive market or faced with a high reserve price, where the winning rate is not likely to rise dramatically until the bid value reaches a certain level. In such case the winning function is changed to

$$w(b(\theta)) = \frac{b^2(\theta)}{c^2 + b^2(\theta)} \qquad (14)$$

After solved it with the same method as the previous one,

$$b_{\text{ORTB2}}(\theta) = c \cdot \left[\left(\frac{\theta + \sqrt{c^2\lambda^2 + \theta^2}}{c\lambda}\right)^{\frac{1}{3}} - \left(\frac{c\lambda}{\theta + \sqrt{c^2\lambda^2 + \theta^2}}\right)^{\frac{1}{3}}\right] \qquad (15)$$

3.5 Multiagent Bidding Strategy

The strategies above can obtain a quite high click amount with a limited budget, however, they only considered the single agent case. In the multiagent system, they may performance poorly, and there are two main potential reasons. One reason might be that an agent can lose some impressions when his bids is lower than any other bids. Thus, he gains relatively less chance to get clicks. Another reason might be the increase in final pay price. As the second price rise, a single bidder would pay more for a single impression and he may run out of his budget without covering a wide range of users and impressions, resulting in the decrease of clicks probability. To perform well in multiagent system, we design some strategies mainly focusing on these two problems.

3.5.1 Opponent Simulation

It tends to be hard to gain impressions in the multiagent system than that in the single agent system. Therefore, to mitigate the impact of the other agents, the bid prices and parameters of strategies should be re-adjusted.

Specifically, the best performance strategy in the problem 4, the squared bidding strategy, will be applied as our main bidding strategy in this multiagent system. Another 5 well-performed bidding strategies are selected to simulate the bids of the other players. According to criterion 2, the new evaluation method defines that a bidder can get an impression if his bid price is higher than both the pay price and the other bids. Optimizing parameters guided by the new evaluation method, the parameters of the squared bidding strategy are updated (basebid = 450, m = 40, n = 3) and compared to the original parameters, this combination is expected to gain more clicks in the multi agent system.

3.5.2 Postpone bidding

According to Hong and Shum [21], since bidders do not receive distinct signals in an auction, they only observe the signals for each component, and the bid price can increase with the signal. In this real-time bidding we can obtain the bidding signals from the 'Leaderboard website' and observe the bidding price of other groups. On the 'Leaderboard', roughly 20 out of 29 groups finally run out of their budget, and it can be inferred that their performance was limited by their budget. In the multi agent system, as the second price increase, the budgets of many bidders are likely to run out quicker, and then they would exit this game. After the majority of bidders exiting, the second price might fall back close to the pay price. Therefore, this strategy postpones our biddings, avoiding high priced impressions as much as possible. By this strategy, we are likely to get more impressions with a limited budget, and thus get higher probability of clicks. In the implementation, we start bidding at the 50000^{th} ads, and before that, all of our bids are divides by 10 (to deliberately lose those games or get impressions with a very low price).

3.5.3 Game theory

Understanding the competitive bidding from a game theoretic perspective has provided a powerful and elegant analytical approach [14]. In this part, we discussed the asymmetric environment and the corresponding bidding mechanism we applied. Then, considering the mixed bidder population with both rational and irrational bidders, we adopted the cursed equilibrium.

- **Asymmetric environment**

The bidding environment would be extremely competitive, and information is asymmetric, which means one party possesses greater material knowledge than the other parties. In this project, possible situations might be some players may be aware of the valuation or bidding strategies of other opponents, while for the rest of bidders they not. It also derives an interesting and tricky question that whether a bidder should disclose his/her private information. Panos considered that it seems more rational for a bidder to hide their positive signal about the value of the auction objective and release a negative attitude at the same time so that opponents are more likely to lower their bids [14]. In this case, we are supposed to reveal the information that we may bid high, stimulating other bidders bid higher to win the impressions and run out of the budget at an early period and create an optimal environment for implementing the Postpone Bidding strategy mentioned before. Begemann and Pesendorfer concluded the optimal mechanism for such circumstance that is to discriminately disclose the privately information with regard to its accuracy to other bidders [15].

- **Cursed equilibrium in mixed bidder population**

Rational decision making is the cornerstone of game theoretic approach of competitive bidding in a Mixed Bidder Population environment, which includes both rational players who follow Nash Equilibrium strategies as well as irrational bidders [16].

We assume the percentage of irrational bidders in these 29 groups is χ, those who layers considered that there is no correlation between the type of bidders and corresponding actions, and believe rivals adopt average strategies for their Opponent Bidding Strategies Simulation. On the other hand, 1-χ represents the size of rational players, who are capable of adopting a Bayesian Nash equilibrium strategy. In a second price auction, for such mixed bidder population, the cursed equilibrium strategy maximizes the expected payoff [21]. It can be characterized as follows.

$$b_i = (1-\chi)E[v(S,Y)|S = s_i, \max_{j\neq i} S_j = s_i] + \chi E[v(S,Y)|S = s_i] \qquad (16)$$

Where s_i is the signal of each bidder for the value of impression. Although, the value of a given ad value is identical among all bidders, information about they hold about it may be varying, either higher or lower. In the values function v(S,Y), Y contains the bid information such as slot width, slot height, slot visibility etc. The value of χ usually ranges between 0% and 60%. In our model, what we assumed it that 5 groups are irrational, with χ set as 17.2%. The cursed equilibrium is confirmed to provider better fit compared with the pure Bayesian Nash equilibrium [14].

4 CONCLUSION&FURTHER STUDY

The validation results of all the models are presented in Table 5:

Bidding Strategy	Imps	Click	CTR	eCPC	CPM	Spend
Constant	60,444	16	0.0265%	58.416	15.463	934.656
Random	60,955	20	0.0328%	50.963	16.722	1019.261
Multiagent Random(Avg)	1,219	3	0.0328%	5.217	2.980	777.35
Linear	126,328	170	0.14%	36.202	48.718	6154.46
Squared	**117,148**	**171**	**0.15%**	**36.523**	**53.313**	**6245.571**
Exponential	145,768	161	0.11%	36.307	40.101	5845.474
OBTB1	148,827	163	0.11%	38.245	41.887	6233.976
OBTB2	149,874	148	0.10%	38.161	37.684	6129.515
Multiagent	84,867	130	0.15%	480,76	73,651	6249.845

Table 5: All bidding strategies Performance metric on the. validation set

In this project, the main objective is to help advertisers to build bidding strategies in a real-time bidding system. It started with implementing exploratory data analysis on the RTB dataset to gain initial insights. It is typically considered that an accurate prediction of CTR is rather significant for developing effective bidding strategies. Thus, we applied several machine learning algorithms for CTR estimator, and it was determined that *XGBoosting* model obtained the highest precision with the AUC of ROC score equals to 0.897. Based on pre-trained CTR prediction model, Linear and Non-Linear Bidding Strategies are created and optimized through parameter tuning. Finally, it was found that among all 5 Linear and Non-linear Bidding Strategies, the *Squared Bidding Strategy* yields the highest number of clicks (171 clicks) on the validation dataset. We suppose the strategy performed well on the validation set can similarly obtain a high click amount on the test dataset.

Taking into account we are competing among different groups, we developed a game theory (multiagent) approach. In the multiagent system, a same strategy may lead to a poorer performance, because a bidder is harder to get impressions, and he tends to pay more for a same impression. To solve above two problems, we simulated opponents' bids to update model parameters, and we postponed bids to reduce our impression payment as much as possible. In addition, we also considered asymmetric environment and cursed equilibrium in mixed bidder population to make some subjective adjustments to our strategy.

In terms of further research, we could ensemble two or several well-performed CTR estimator models to improve the precision. We also intend to introduce the **Multi-Agent Reinforcement Learning**, as Du et al. stated that the interaction between the bidders and the ADX shows great similarity with the interaction between an agent and the corresponding environment, similar to the reinforcement learning framework [20].

5 REFERENCES

[1] Mark Hoelzel. The Programmatic Advertising Report: Real-time bidding is taking over the digital ad market. http://uk.businessinsider.com/the-programmatic-and-rtb-ad-report-2014-8

[2] Shuai Yuan, Jun Wang, and Xiaoxue Zhao. 2013. Real-time bidding for online advertising: measurement and analysis. In *Proceedings of the Seventh International Workshop on Data Mining for Online Advertising*. ACM, 3.

[3] Zhang, W., Yuan, S., & Wang, J. (2014). Optimal real-time bidding for display advertising. In *Proceedings of the 20th ACM SIGKDD international conference on Knowledge discovery and data mining* (pp. 1077-1086). ACM.

[4] Kondakindi, G., Rana, S., Rajkumar, A., Ponnekanti, S. K., & Parakh, V. (2014). A logistic regression approach to ad click prediction. *Mach Learn Class Project*.

[5] Bryant, K., Ton, P., Owen, M., Rosen, B., & Justice, S. (2017). Predicting clicks in mobile advertising: An experiment. Retrieved from https://nycdatascience.com/blog/student-works/predicting-clicks-in-mobile-advertising-an-experiment/

[6] Trofimov, I., Kornetova, A., & Topinskiy, V. (2012, August). Using boosted trees for click-through rate prediction for sponsored search. In *Proceedings of the Sixth International Workshop on Data Mining for Online Advertising and Internet Economy* (p. 2). ACM.

[7] Oentaryo, R. J., Lim, E. P., Low, J. W., Lo, D., & Finegold, M. (2014, February). Predicting response in mobile advertising with hierarchical importance-aware factorization machine. In *Proceedings of the 7th ACM international conference on Web search and data mining* (pp. 123-132). ACM.

[8] Qu, Y., Cai, H., Ren, K., Zhang, W., Yu, Y., Wen, Y., & Wang, J. (2016, December). Product-based neural networks for user response prediction. In *2016 IEEE 16th International Conference on Data Mining (ICDM)* (pp. 1149-1154). IEEE.

[9] Zhang, S., Fu, Q., & Xiao, W. (2017). Advertisement click-through rate prediction based on the weighted-ELM and adaboost algorithm. *Scientific Programming*, *2017*.

[10] Tianqi Chen and Carlos Guestrin. 2016. XGBoost: A Scalable Tree Boosting System. CoRR abs/1603.02754 (2016). http://arxiv.org/abs/1603.02754

[11] He, X., Pan, J., Jin, O., Xu, T., Liu, B., Xu, T., ... & Candela, J. Q. (2014, August). Practical lessons from predicting clicks on ads at facebook. In *Proceedings of the Eighth International Workshop on Data Mining for Online Advertising* (pp. 1-9). ACM.

[12] Jian Xu, Xuhui Shao, Jianjie Ma, Kuang-chih Lee, Hang Qi, and Quan Lu. Lift-based bidding in ad selection. In *AAAI,* pages 651-657, 2016.

[13] Lorentziadis, P. L. (2016). Optimal bidding in auctions from a game theory perspective. *European Journal of Operational Research*, *248*(2), 347-371.

[14] Bergemann, D., & Pesendorfer, M. (2007). Information structures in optimal auctions. *Journal of economic theory*, *137*(1), 580-609.

[15] Eyster, E., & Rabin, M. (2005). Cursed equilibrium. *Econometrica*, *73*(5), 1623-1672.

[16] Ganganwar, V. (2012). An overview of classification algorithms for imbalanced datasets. *International Journal of Emerging Technology and Advanced Engineering*, *2*(4), 42-47.

[17] Hoens, T. R., & Chawla, N. V. (2013). Imbalanced datasets: from sampling to classifiers. *Imbalanced Learning: Foundations, Algorithms, and Applications*, 43-59.

[18] Chen, T., & Guestrin, C. (2016, August). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining* (pp. 785-794). ACM.

[19] Hong, H., & Shum, M. (2003). Econometric models of asymmetric ascending auctions. *Journal of Econometrics*, *112*(2), 327-358.

[20] Du, M., Sassioui, R., Varisteas, G., Brorsson, M., & Cherkaoui, O. (2017, November). Improving real-time bidding using a constrained markov decision process. In *International Conference on Advanced Data Mining and Applications* (pp. 711-726). Springer, Cham.

[21] Avery, C., & Kagel, J. (1997). Second-price auctions with asymmetric payoffs: an experimental investigation. Journal of Economics and Management Strategy, 6, 573–603.