COMPGW02 Web Economics Coursework Group Report

Sven Sabas
sven.sabas.17@ucl.ac.uk

Parham Oghabi
parham.oghabi.17@ucl.ac.uk

Niren Patel
zctpatb@ucl.ac.uk

1. INTRODUCTION

This report studies bid optimisation for real-time bidding (RTB) based display advertising. It is built on a data set consisting of a feedback log of approx. 2.7 Million RTB advertising auctions. The analysis can largely be divided into two - click-through-rate (CTR) prediction, where the focus is on predicting the probability of a click for an unseen advertisement, and the derivation of bidding strategy, with an aim to devise cost-effective RTB strategy for paying for the display advertising.

The prediction task uses a number of machine learning models to maximise the predictive accuracy. For the optimal bidding, various strategies were evaluated against key performance indicators (KPIs) within a constrained, pre-defined budget. Overall, it was found that basic bidding strategies, such as constant and random, were outperformed by more elaborate techniques that relied on the predicted CTR (pCTR).

2. RELATED WORK

There is a growing amount of literature in the computational advertising space, see [10] for further reference, but a bulk of the bidding strategies implemented in this report are derived from [13]. Furthermore, [4] discusses the practical complexities of predicting clicks and offers a number of solutions for implementing it at scale. Some of the recommendations, such as negative downsampling and model re-calibration, are also implemented in this analysis. Lastly, for a reader new to the field, it is recommended to look at [12] to get better grasp of the iPinYou data set that is the foundation of the whole report.

3. SIMPLE STRATEGIES

3.1 Constant Bidding Strategy

The simplest strategy, constant bidding, ignores all the available information offered in the log data and bids a constant amount for each advertisement. This naive strategy does not require CTR prediction and treats all the auctions equally. For the purpose of this report, the optimal model parameter (constant value to bid) was found by line-search. The tuning was done using the validation set and the value was chosen, such that it would maximise the KPI (number of clicks) achieved on the set, whilst satisfying the budget constraint.

By implementing line-search, an optimal parameter value of 77 achieves 68 clicks on the validation set (Figure 1). Since this strategy does not take into account pCTR, it doesn't exploit all the information available for an impression and performs poorly in comparison to more sophisticated models. The complete evaluation metrics on the validation set are shown in Table 1, which compares the constant bidding strategy with a random bidding alternative described in next section.

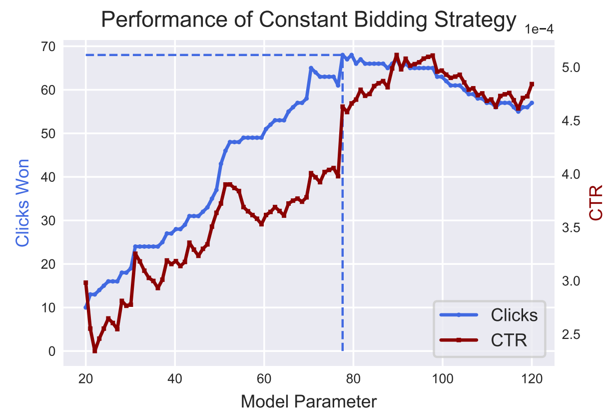

Figure 1: Hyper-parameter Tuning Process of the Constant Strategy (Budget 6,250 CNY). Optimal Bid is Shown by Dashed Line.

3.2 Random Bidding Strategy

Another simple strategy is the random bidding strategy that bids a random amount, usually drawn from a uniform distribution, for each auction. The randomness can be tamed by setting lower and upper bounds for the bids. Grid-search method can be applied to find the optimal hyperparameters for the lower and upper bounds. Similar to the constant bidding strategy, no additional information is required as an input.

	Constant	Random
Clicks	68	83
Impressions	146,865	154,245
CTR	0.046%	0.050%
Budget (CNY)	6,249	6,034
Avg CPM	42.54	39.12
Avg CPC	91.89	72.69

Table 1: Evaluation of Basic Bidding Strategies

By implementing line-search, optimal parameter values of 24 and 108, achieved 83 clicks on the validation set. However, it must be noted that due to the inherent random nature of the bidding strategy, the realised results can vary vastly, and in practice it is difficult to fine-tune the hyper-parameters.

These naive strategies are ought to underperform in practice as they do not take into consideration the metadata about the users as well as the market conditions. This means that the advertisements are not adopted to specific users, partially defeating the purpose of tailored advertising, and hence limit the effectiveness of targeted advertising. Furthermore, simple strategies rely solely on the back-log of prices in the back-testing data and thus are inflexible to accommodate to changes in market conditions such as increased competition. In practice, basic strategies tend to either underbid, which results in very few bids being won, or overbid, which quickly depletes the allocated budget.

4. pCTR BASED STRATEGIES

To be more effective in targeting users likely to click on the ads and hence increase the effectiveness of one's campaign, it is possible to leverage on the user metadata. The metadata can be used as an input in statistical models to compute pCTR, a probability measure that reflects users likelihood of clicking on the advertisement, conditional on the features.

Using the output of the aforementioned statistical approach advertisers can increase the effectiveness of their targeted campaigns by focusing on the auctions that are *a priori* likely to receive positive responses, i.e a click. This opens a whole new avenue for bidding strategies, which are referred to as pCTR based strategies in this report. In this section some of the statistical models that are used for the prediction are highlighted, in addition to describing pCTR based bidding strategies.

4.1 CTR Prediction

The efficiency of a CTR based advertising strategy relies on the precision and calibration of CTR predictions. The prediction system needs to be flexible, robust and capable of learning from large scale data sets. In the following section the main steps of the prediction process are highlighted and comparative results of model predictive results are given.

4.1.1 Feature Engineering

CTR prediction is a challenging task and relies a lot on the quality of input data. The data set used for the particular analysis has 22 raw variables (see individual report for description), all of varying importance for prediction. Figure 2 shows how specific features can impact the CTR. For example, on Wednesdays, the CTR for advertiser 3358 drops significantly. Whereas on Thursdays, it has the highest CTR.

In order to train the CTR prediction model, the raw variables have to be transformed to an appropriate format for the inference. Using the insights from the exploratory analysis and from previous work with the particular dataset [13] as well as the output from model diagnostics, the following features were used for the final CTR prediction: *weekday, hour, OS, browser, region, slot width and height, slot format, slot visibility, slot floor price bucket, and user tags.*

The *user agent* column was parsed to obtain the *operating system* and *browser*. Each *user tag* is also considered as

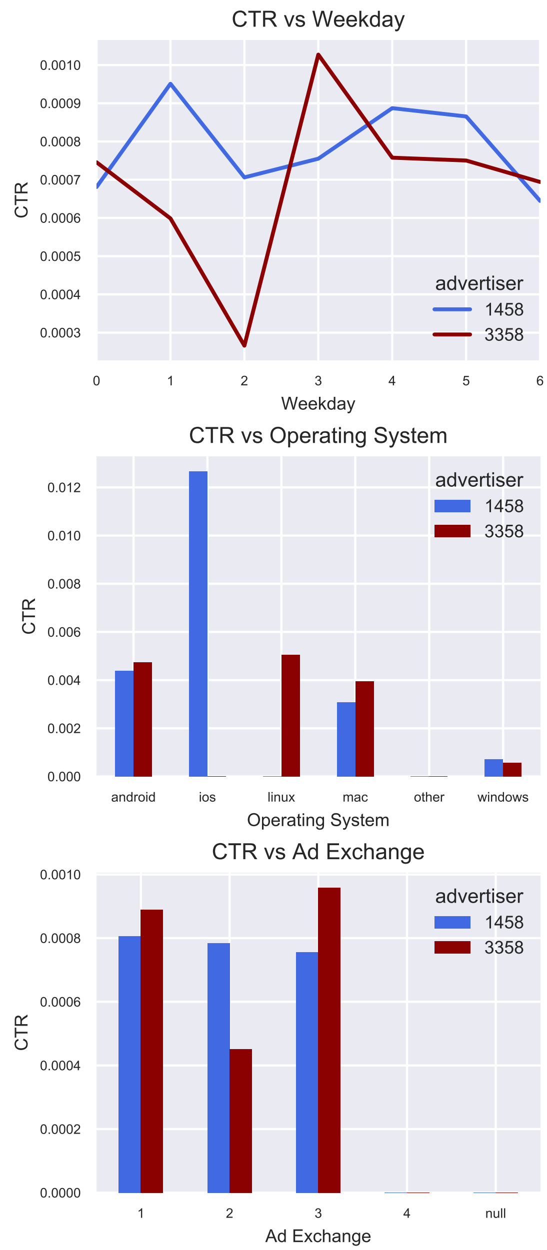

Figure 2: Effect of Various Bid Request Features on the CTR of Advertisers 1458 and 3358

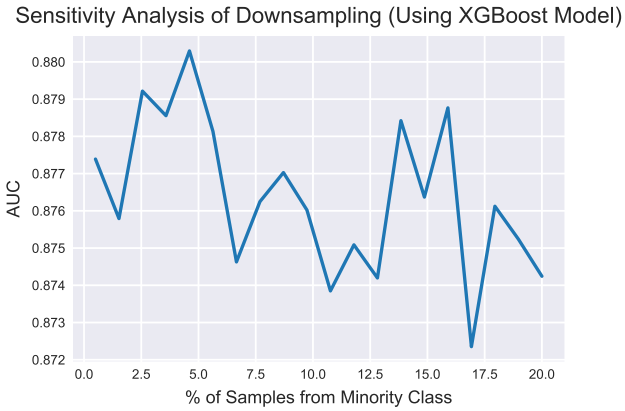

Figure 3: Sensitivity Analysis of Downsampling

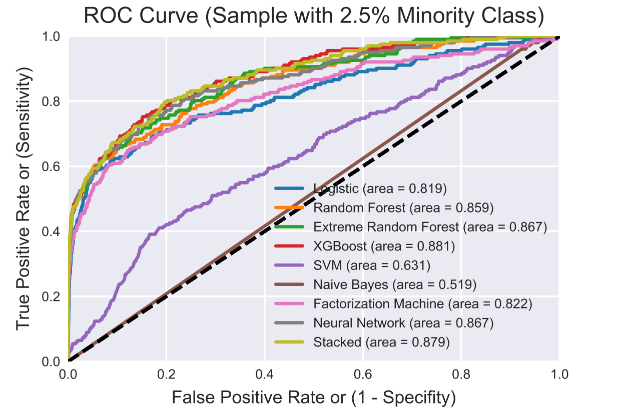

Figure 4: Prediction Accuracy Comparison of Various Machine Learning Models Based on AUC

a separate feature. The *slot floor price* is split into one of the five possible buckets: [0], [1, 10], [11,50], [51, 100], [101, 1000). Higher order combinations of features were not computed. All features were categorical, which resulted in one-hot encoding of the input features. There were a total of 198 features used for CTR prediction.

4.1.2 Downsampling

From the individual statistical analysis done on specific advertisers and the whole data set, it is clear that the data set is imbalanced with only 0.073% of the won impressions being clicked. This causes a number of problems in training the statistical models. In particular, the prediction models used for this problem rely on loss minimisation for parameter optimisation. Loss minimisation for the current problem is minimisation of classification error, but if 99.9% of the samples come from one class, the model would very likely classify all of the samples as majority class, which is not clicked. This limits the practical use of the CTR based bidding strategies as it would be difficult to extract extra information if the prediction set is uniformly negative.

To mitigate the issue, [4] have suggested negative downsampling the majority class to improve the class imbalance. Based on the paper, the initial training data set was downsampled to 62,522, which resulted in 2.5% samples being drawn from the minority class. In addition, a sensitivity analysis of the downsampling rate was conducted (Figure 3) using the most accurate machine learning model. The analysis further confirmed that 2.5% was the optimal minority class rate.

Another important point to note is that since downsampling was performed, the mean of the predicted CTRs of the models will be much higher than the average CTR of the original training data. Therefore, all the model pCTRs have to be calibrated to be comparable with the initial data set using the following equation:

$$\frac{p}{p + \frac{1-p}{w}},$$

where p is the model predicted CTR and w is the negative downsampling rate.

4.1.3 CTR Prediction Models

For the actual CTR prediction task, a plethora of models were trained ranging from logistic regression and factorization machine [8], to variations of decision trees [2] and stacked models [14]. The model hyper-parameters[1] were tuned using grid-search and a majority of models used regularisation terms to avoid overfitting such as L_1 or L_2 penalty for linear regression, active drop-out rate for neural networks or early stopping for gradient boosting models.

Model	AUC	Confusion Matrix	
Logistic Regression ($\mathbf{L_2 = 0.01}$)	85.11	252,191	51,532
		58	144
XGBoost (Trees=50, Depth=3, L.Rate=0.1)	88.07	303,169	554
		135	67
Factorization Machines	82.22	303,060	663
		144	58
Multi-layer Perceptron (16 Hidden Nodes)	86.70	303,303	420
		135	67

Table 2: AUC Scores of Various CTR Prediction Models

Logistic regression is a linear model, which can scale to billions of samples and parameters. It is incapable of extracting complex and non-linear features from the bid request features without manually inputting higher order combination of features. Therefore, it can be observed from the confusion matrix in Table 2 that even though logistic regression predicted the most number of clicks correctly, it also overpredicted the non-clicks and this can result in depleting the budget wastefully. Therefore, non-linear models such as decision trees and deep neural models tend to perform better in this particular task.

The receiver operating characteristic (ROC) curves and respective areas under the ROC curve (AUC) of all the tested models are shown in Figure 4 and in Table 2. There are some clear outliers, such as SVM and Naive Bayes models, that performed extremely poorly, but a good proportion of the models had reasonable accuracy after hyperparameter tuning. The two best performing models were

[1]Hyperparameters differed across models, but for the purpose of grid-search based optimisation, included a majority of the tunable model parameters. See full list in the GitHub repository [9].

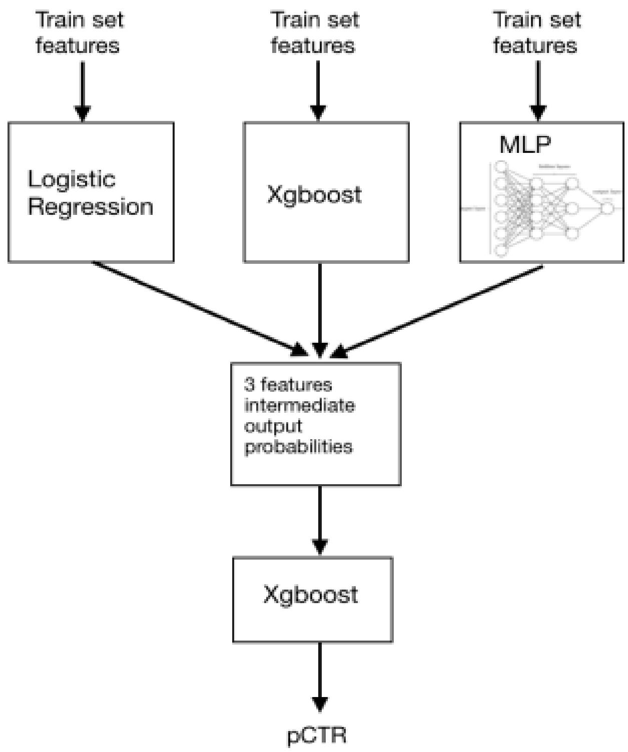

Figure 5: Diagram of the Stacked Ensemble CTR Prediction Model.

stacked model[2] with AUC of 0.879 and gradient boosted decision tree (XGBoost)[3] with AUC of 0.881.

It must be highlighted that XGBoost is a flexible non-linear model and is susceptible to overfitting. To decrease the risk and to tame the model complexity, the final XGBoost model had two regularisation terms ($L_1 = 0.8$ and $L_2 = 1$) and early stopping (maximum tree depth was set to 5 and total of 50 trees were fitted) were implemented. The whole training processes used XGBoost wrapper library for sklearn in Python [3].

4.2 Linear Bidding Strategy

As mentioned earlier, constant and random bidding strategies do not take into account the metadata received for each auction and hence ignore potentially important sources of information. One way to leverage on the information is include the pCTR computed from the metadata to be a form of signal that can then be inputted to a bidding strategy. Simplest such strategy is a linear bidding strategy, where

[2]Stacked model used 3 base level classifiers - logistic regression, XGBoost and multilayer perceptron - as additional feature extractors. The outputs of the base models are combined to form a three-dimensional intermediate input into the second level classifier. These intermediate inputs are the predicted probabilities from the base classifiers. The second level classifier, another XGBoost classifier outputs the final pCTR based on the 3-dimensional input vector received from base classifiers. The final stacked ensemble model is shown in Figure 5. The model was implemented using *mlxtend* [7] Python package.

[3]Gradient boosted decision tree is a additive, tree based supervised learning algorithm, see [5] for full details.

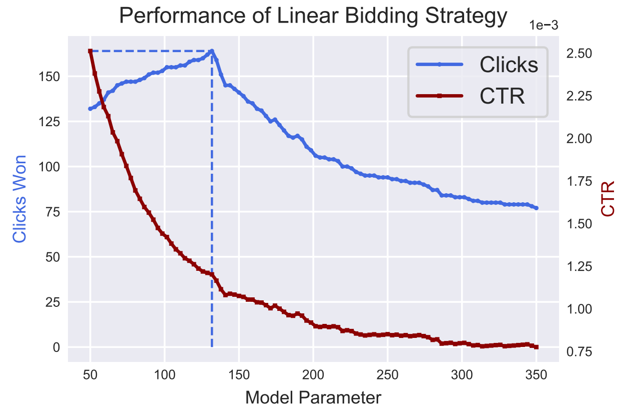

Figure 6: Hyper-parameter Tuning of the Linear Strategy. Optimal Bid is Shown by the Blue Line.

the bid amount is linearly related to the pCTR signal.

More formally, the linear bidding strategy can be expressed as:

$$bid = parameter * \frac{pCTR}{avgCTR},$$

where the bid amount (bid) is a linear function of the ratio of predicted CTR and the average CTR of the data set. Average CTR is simply the fraction of minority samples in the data set. This model bids more for the auctions where the likelihood of user clicking on the ad is increased. Due to its simplicity, the linear bidding strategy is one of the most commonly used strategies by Demand Side Platforms (DSPs).

Similarly to the previous strategies line-search was used to optimise the model slope parameter to maximise the potential number of clicks. Figure 6 depicts the tuning process with a peak for the optimal parameter value at 131 and total of 164 clicks are achieved on the validation set. To note, the average CTR was set to the mean CTR of the training set which was 0.073%.

Compared to the simple bidding strategies described earlier, using the pCTR and linear strategy clearly improves the performance and helps to maximise the targeted KPI. One potential drawback of using a linear bidding strategy is that if competition uses higher order relationship and non-linearities in their pCTR based bidding function, then they might bid more for the highly probable advertisements and leave the linear bidding strategy users empty-handed. This is further exacerbated, when very few clicks are being observed, but there is a strong predictive signal to identify them. This means that more complex non-linear strategies could bid high for the probable cases, winning all the clicks, but low (lower than linear) for other cases, enabling them to efficiently manage their budget without wasteful bidding. The next section offers some solutions.

4.3 Non-Linear Bidding Strategies

Non-linear bidding strategies are natural extensions to the linear bidding strategies that exploit the competitive dynamics of the market. The exact functional expression of the bidding function depends on a number of factors and can take various forms. The simplest alternatives are power functions ($bid = parameter_1 * (\frac{pCTR}{avgCTR})^{parameter_2}$),

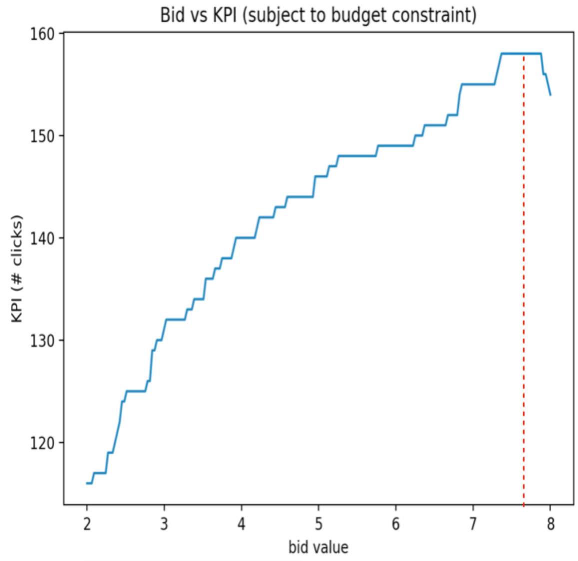

Figure 7: Hyper-parameter Tuning Process of ORTB 1. The Optimal Value of C is Shown by the Red Dashed Line

where it is possible to optimise for two sets of parameters - $parameter_1$ and $parameter_2$. Alternatively one can keep the power function constant, i.e set $parameter_2$ to 2, and use line-search to optimise for $parameter_1$.

In addition, there has been more extensive research to find optimal functional forms for the non-linear bidding strategies. [13] highlights two alternative forms for optimal real time bidding (ORTB):

1. ORTB1:

$$bid = \sqrt{\frac{c}{\lambda} * \theta + \lambda * 2} - \lambda,$$

where c and λ are two parameters to be optimised and θ is pCTR.

2. ORTB2:

$$bid = c * [(\frac{\theta - \sqrt{c^2\lambda^2 + \theta^2}}{c\lambda})^{\frac{1}{3}} - (\frac{c\lambda}{\sqrt{\theta - c^2\lambda^2 + \theta^2}})^{-\frac{1}{3}}],$$

where c and λ are once again two parameters to be optimised and θ is pCTR.

Three of the aforementioned non-linear bidding function were fitted and hyperparameters optimised (using grid-search), in addition to to the linear bidding strategy (see ORTB1 example in Figure 7). The comparative results can be seen in Table 3. To note, the validation set had a total of 202 clicks.

It is clear that using pCTR in the bidding strategy benefits advertisers in terms of the targeted KPI. The linear bidding strategy achieves twice as many clicks as the random strategy and yet spends almost identical amount on bidding. Even though the linear strategy won less auctions compared to the other two basic strategies, this is non-detrimental; since the goal of most advertisers is to maximize the number of clicks and CTR and also to reduce their effective Cost Per Click (eCPC).

	Linear	**Square**	**ORTB1**	**ORTB2**
Clicks	164	169	160	147
Impress.	136480	108,452	152,146	157,603
CTR	0.120%	0.155%	0.105%	0.09%
Budget	6249	6249	6249	6249
Avg CPM	45.79	57.69	41.01	39.65
Avg CPC	38.1	36.8	39.06	42.5

Table 3: Evaluation of CTR Based Bidding Strategies

The results of non-linear bidding strategies were mixed. ORTB1 and ORTB2 strategies enabled the winning of more auctions than a linear strategy, but the main metric, clicks won, was lower than the alternatives. Squared bidding strategy, being the most aggressive, won the most clicks, but least amount of auctions. This shows that highly targeted bidding can be lucrative. On the contrary, due to high sensitivity to hyperparameter tuning, square bidding function performed badly on the live test set as it quickly depleted the budget due to wrongfully tuned hyperparameters. Hence, one must be careful using such aggressive strategy as changing conditions and even a slight tweak in parameter settings can have material effect on bidding,

4.4 Hybrid Bidding Strategy

4.4.1 Overview of the Submitted Bidding Strategies

For the final test submission three different bidding strategies were attempted by the group: ORTB1, ORTB2 and a hybrid ORTB1+Linear [13]. To note, the best performing non-linear bidding strategy on the validation set, square bidding strategy, was excluded as initial tests showed the aggressive behaviour depleted the budget too rapidly and resulted in sub-par performance on the test set.

All three attempted strategies are non-linear bidding functions, with the first two being strictly concave. The hybrid strategy, which is a mixture of ORTB1 and a linear bidding strategies, is depicted in Figure 8. It essentially uses the ORTB1 bidding strategy until a certain threshold and then follows the linear strategy once the calibrated predicted CTR exceeds that threshold.

The calibration of ORTB1 and ORTB2 models is described in section 4.3. Similarly, for the hybrid ORTB1 + Linear strategy a grid-search optimisation approach was used to tune the hyperparameters: C, λ, the threshold value and the base bid value to meet the budget constraint and to maximize the number of clicks. The average CTR was chosen as the training set average (0.073%). The experiments showed that setting the threshold value close to the intersection of ORTB1 and the linear function yielded the highest number of clicks on the validation set (Figure 8).

4.4.2 Evaluation of the Submitted Bidding Strategies

Table 4 outlines the various evaluation metrics for the three different bidding strategies on the validation set. To note, the validation set had a total of 202 clicks. It can be seen that ORTB1 + Linear outperforms ORTB1 and ORTB2 in terms of maximizing the target KPIs and achieved the highest score for both total clicks and CTR. As a result, the eCPC for the hybrid model is also the lowest, which would translate to cost effective marketing strategies for the

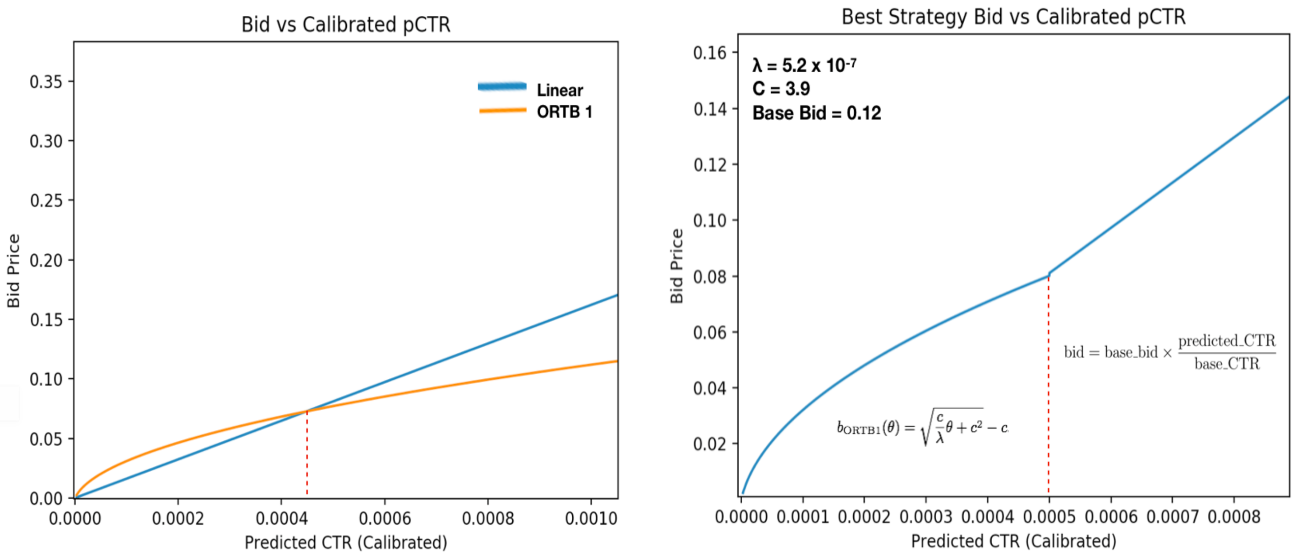

Figure 8: The Left Plot is the Intersection of the lLinear and ORTB 1 Bidding Strategy Against the Calibrated pCTR. The Right Plot is the Optimal Strategy Which is a Combination of the Linear and ORTB 1 Functions.ADv

	ORTB1	**ORTB2**	**ORTB1+ Linear**
Clicks	160	147	163
Impressions	152,146	157,603	141,939
CTR	0.105%	0.090%	0.115%
Budget Spent	6,249	6,249	6,249
Avg CPM	41.01	39.65	44.03
Avg CPC	39.06	42.50	38.34

Table 4: Evaluation Metrics of the ORTB based Bidding Strategies

	ORTB1	**ORTB2**	**ORTB1+ Linear**
Clicks	160	147	163
Budget	5,928	6,249	5,733
Impressions	147,179	157,603	130,435

Table 5: Minimum Budget Required by Each Strategy to Achieve Maximum Number of Possible Clicks on the Validation Set

advertisers. All in all, the hybrid model was the best performing model out of the aforementioned three on the validation set.

On the other hand, ORTB2 achieved the lowest number of clicks. This is due to the concavity characteristics of the function, meaning it has a rapid increase in the bid price at low pCTR levels, causing it to deplete the budget quickly on the validation set. Therefore, even though it won the most number of auctions, it only did so because it bid very high on impressions with a low pCTR. Hence, ORTB2 can be a useful strategy to maximise total number of impressions (not clicks) in competitive markets.

Another important point to make about the hybrid strategy is that it can achieve its maximum number of clicks (163) without fully depleting the budget (Table 5). Even though the number of won impressions drops, the strategy achieved the maximum number of clicks, the main KPI, with a budget of only 5,733 CNY. ORTB2 cannot save money due to its aggressive bidding behavior at low pCTR values.

All three aforementioned models were tried on the test set using the feedback API provided. Furthermore, two sets of CTR predictions were used as an input - one from the stacked model (AUC = 0.879) and another from XGBoost model (AUC = 0.881). The two best performing combination are shown in Table 6. It can be seen that the bidding strategy that used ORTB1 + Linear function with a stacked ensemble model prediction, performed the best and achieved 175 clicks on the test set.

Finally, to note that for criterion 1 (single-agent strategy) bids were placed sequentially through the default test set order. However, for criterion 2 (multi-agent strategy) the test set was reversed to allocate more budget to the auctions happening later in the set. This was due to the fact that criterion 2 accounts for competition among the bids from different groups. The reversion was done because it is likely that many groups will deplete their budget in the first half of the test set, therefore leaving them unable to bid on later bid requests. By reversing the test set and then placing bids, later auctions will be won and increase the probability of maximizing the KPI. Moreover, for multi-agent strategy the hyperparameter tuning was altered and instead of the payprice (second highest price), the bid price (highest price) was used as the cut-off to win the auction. This meant that the strategies are better tuned for more competitive environments, where naturally the bids would be higher.

4.4.3 Strategy Evaluation Under Various Budgets

In addition to testing the bidding strategies on the validation set with a pre-defined budget, a budget size sensitivity analysis was also conducted. Table 7 shows the KPI per-

	ORTB1 + Linear (XGBoost)	ORTB1 + Linear (Stacked)
Clicks	173	175
Impressions	142,052	141,942
CTR	0.121%	0.123%
Cost	6249	6245
CPC	36.12	35.68

Table 6: Comparison of a XGBoost CTR Model and a Stacked CTR Model with the Optimal Bidding Strategy. Results Were Evaluated on the Test Set.

Model	1/16*B	1/8*B	1/4*B	1/2*B
Constant	10	15	24	43
Random	9	15	24	38
ORTB1	29	46	93	136
ORTB1 + Linear	14	24	45	91

Table 7: Evaluation of Various Bidding Strategies Under Different Budgets Contraint (Number of Clicks)

formance (clicks won) of various strategies under different budget constraints, given as a fraction of original budget of 6,250 CNY.

The parameters for the ORTB1 strategy were once again tuned using grid-search on the validation set and were optimised for each budget size, such that it would maximise the number of clicks achieved. Whereas, for the hybrid ORTB1 + Linear, random and constant strategy, only one set of parameters were used - the optimised parameters based on the full budget. This was done to show that the optimisation of the bidding strategies is very sensitive and dependent on budgeting constraints. The tuned parameters can adversely affect the performance and the KPI under different budget constraints. Thus one must be careful in optimising the bidding strategies in changing circumstances such as a fluctuating budget.

5. CONCLUSIONS

In this report, a number of basic bidding strategies were implemented, such as constant and random. These strategies offer limited performance for advertisers due to the inherent inflexibility of the strategies. Additionally, it was shown that one can improve the number of clicks by combining bidding strategies with CTR prediction, to leverage the valuable information contained in the auction metadata. The best performing strategy (ORTB1 + Linear) achieved its goal of maximizing the KPI (number of clicks) by not only meeting the budget constraint but also being able to save money. Though, one must be careful to use the right settings for the optimisation, as the hyperparameters are sensitive to the bidding conditions such as competition and marketing budget.

The previous strategies are a static optimization problem and compute bids independently of preceding auctions. The feedback API offered by the course organisers was helpful to test the performance of the algorithms, but also allows groups to over-fit to the test set. For additional analysis, it would be interesting to implement a representational state transfer API (REST API) where students can feed their Python scripts and during test time all scripts are run simultaneously and bids are placed sequentially based on individual bid feedback by the REST API. This would allow for bids to change dynamically. In the current environment, if an auction is lost, the budget is not spent and will not be able to be re-invested since bids are pre-set whereas a real-time multi-agent feedback API would resolve that issue.

If the REST API would have been available and more time was given, the problem could have been approached using multi-agent reinforcement learning [1] or using feedback controller methods such as Proportional-Integral-Derivative (PID) or water-level [6], [11].

All code related to the analysis, CTR model, and bidding strategies can be found on Github [9].

6. REFERENCES

[1] H. Cai, K. Ren, W. Zhang, K. Malialis, J. Wang, Y. Yu, and D. Guo. Real-time bidding by reinforcement learning in display advertising. In *Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, WSDM '17, pages 661–670, New York, NY, USA, 2017. ACM.

[2] T. Chen and C. Guestrin. Xgboost: A scalable tree boosting system. *CoRR*, abs/1603.02754, 2016.

[3] T. Chen and C. Guestrin. XGBoost: A scalable tree boosting system. In *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '16, pages 785–794, New York, NY, USA, 2016. ACM.

[4] X. H. et al. Practical lessons from predicting clicks on ads at facebook. *ADKDD*, 2014.

[5] J. H. Friedman. Greedy function approximation: A gradient boosting machine. *The Annals of Statistics*, 29(5):1189–1232, 2001.

[6] W. Z. K. M. J. W. Han Cai, Kan Ren. Real-time bidding by reinforcement learning in display advertising. *In Proceedings of the Tenth ACM International Conference on Web Search and Data Mining*, pages 661–670, 2017.

[7] S. Raschka. Mlxtend, Apr. 2016.

[8] S. Rendle. Factorization machines. In *Proceedings of the 2010 IEEE International Conference on Data Mining*, ICDM '10, pages 995–1000, Washington, DC, USA, 2010. IEEE Computer Society.

[9] S. Sabas, N. Patel, and P. Oghabi. COMPGW02 Web Economics Coursework Repository. https://github.com/SSabas/ucl-webecon-cw, 2018.

[10] Z. Wang. Papers on computational advertising. https://github.com/wzhe06/Ad-papers, 2018.

[11] J. W. T. Z. X. W. Weinan Zhang, Yifei Rong. Feedback control of real-time display advertising. *WSDM*, 2016.

[12] J. W. X. S. Weinan Zhang, Shuai Yuan. Real-time bidding benchmarking with ipinyou dataset. *UCL Technical Report*, JULY 2014.

[13] S. Y. Weinan Zhang and J. Wang. Optimal real-time bidding for display advertising. *In Proceedings of the 20th ACM SIGKDD international conference on knowledge discovery and data mining*, pages 1077–1086, 2014.

[14] D. H. Wolpert. Stacked generalization. *Neural Networks*, 5:241–259, 1992.