COMPGW02 Web Economics

Group Report

Achillefs Sfakianakis[*]
UCL
ucabasf@ucl.ac.uk

Akis Thomas[†]
UCL
ucablat@ucl.ac.uk

Dwane van der Sluis[‡]
UCL
ucabdv1@ucl.ac.uk

1. INTRODUCTION

This paper deals with the pricing and sale of online advertising opportunities inside the a multi-billion[5] dollar industry. When a webpage is presented to a user, it contains a number of advertising slots. The advertisement displayed and it's pricing are determined via real-time bidding. This paper describes a solution to the online real-time advertising bidding problem, where the amount to bid for each advertising impression is optimised. Rather than presetting a fixed bid for every campaign or keyword, as happened in traditional online advertising, real-time bidding (RTB) has enabled the personalization of advertising per user. This gives the opportunity to advertisers to bid on individual advertisement impressions based on an individual user's profile. The data used was based on the iPinYou dataset[1], which consists bids, price paid, click info and many end user specific attributes for over 2 million auctions. Data was split into train, validation and test sets. The final aim of the assignment was to determine an optimal bidding strategy by predicting clicks and other agents bids. It is evaluated on number of clicks, click-through-rate (CTR), money spent, average cost-per-mile (CPM) and average cost-per-click (CPC).

The procedure that our team followed consisted of:
1) An initial data exploration to understand the data and determine useful variables
2) An initial experimentation with basic bidding strategies which exposed the dynamics of a real-time bidding system
3) The implementation of a linear bidding strategy based on the estimated probability of a user click
4) The development and optimisation of three separate components by individually team members (pCTR, base_bid and formula to combine both)
5) The implementation and evaluation of the final model, as well as the implementation of a system that could conduct real-time auctions for multiple entities, helping our team simulate real world competition, which was used to evaluate and improve the various models developed.

Our results show, that non-predictive strategies such as constant and constrained random achieve a CTR of around 5% (67 clicks) which is a lot better than the 0.07% of impressions that are clicks in the training set. To improve on this, three click prediction models were individually combined with a linear bidding formula, which resulted in a much better CTR of 0.14% (162 clicks). We then experimented with improved click predictors, predictions of pay price and non-linear formulas, achieving a higher CTR of 0.22% (163 clicks). Finally, after experimenting with different ensemble models, a combination of our previous two best models achieved a CTR of 0.22% (165 clicks). This high number of clicks, in combination with the extremely high CTR, allows us to perform well in both single agent and multi agent environments. The reasoning behind this is that with a better understanding of where clicks are, we can easily adapt to advertisers' different aims (e.g. high numbers of clicks or high number of impressions).

[1]https://goo.gl/aqGmDX

2. RELATED WORK

On-line advertising bid optimisation is a well studied problem[8]. On-line advertising broadly splits into i) sponsored search advertising[1], where adverts are selected according to the users search terms ii) contextual advertising, where advertisements themselves are selected and served by automated systems based on the identity of the user and the content displayed iii) display advertising, where general advertisements and brand messages are displayed to site visitors and iv) real-time biding auctions, where advertising inventory is bought and sold on a per-impression basis.

Originally much research focused on sponsored search keyword auctions [1] where often advertisers only pay if their ad is clicked [5]. This changed around the time Chen et al.[4] (2011) described a real-time bidding algorithm for performance-based display ad allocation. Funk et al. [7] give an excellent flow diagram[2] of the actors involved in real time advertising.

McMahan et al. [5] explore issues of computational costs, prediction confidence, calibration and feature management with in Ad click prediction.

Zhang et al.[9] investigate optimal real-time bidding strategies and suggest optimal strategies involve targeting many low cost impressions rather than strategies targeting highly coveted clicks, where competition is highest.

Zhang et al. [10] describe the iPinYou dataset, and conduct experiments on CTR estimation.

Yuan et al. [8] propose a game theory based algorithm for predicting the reserve price. The OneShot algorithm proposed performed the best with significant margin in most cases.

[2]http://aisel.aisnet.org/cgi/viewcontent.cgi?article=1303&context=bise

3. APPROACH AND RESULT

Table 1: Terminology

	Description
base_bid	pay price for an auction
CPM	Cost per mille (1000) impressions
CTR	Click Through Rate
DSP	Demand Side Platform - Represents advertisers
eCPC	effective cost per click
floor price	minimum price for each auction
pBB	predicted base bid
pCTR	probability of a click through
RTB	Realtime Bidding
Single-Agent	Competing against paid price only
Multi-Agent	Competing against paid price and 31 other teams on an unseen data set

3.1 Problem 1: Data Exploration

This section summarises the work produced in three individual reports produced by the authors of this paper. The data analysed was a subset of the iPinYou dataset, obtained from https://goo.gl/aqGmDX.

The *training* dataset consists of information on 2.4 million auctions. The *validation* data set consisted of 303,376 auctions. The dataset is very unbalanced, having 1 click per ˜1800 impressions. The unseen *test* dataset consisted of around 300,000 auctions.

To identify which features would be useful for click prediction, CTR was plotted against all features, of which examples are given in Figure 1. This allowed an appropriate selection of features to be used, as well as new features to be created.

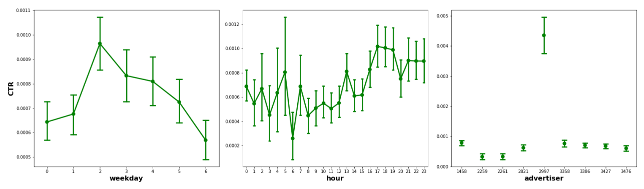

Figure 1: CTR distribution against different features

Figure 2 shows the large difference between bid prices and pay prices in the train data, highlighting the immense opportunity for improving bidding strategies. This paper aims to produce bidding strategies for two scenarios. In the first scenario the strategy should bid against the pay price in the dataset. It was found that the pay price in this dataset is extremely similar to the floor price, creating an environment that is similar to a single-agent scenario. As such, this scenario will be referred to under this terminology from here forth. The second scenario holds auctions between 31 teams on the test set. It was found that this multi-agent scenario is extremely competitive, resulting in average pay prices of at least 639 if all teams are to spend their budget. This is 8 times higher than the single-agent scenario which needs to be accounted for.

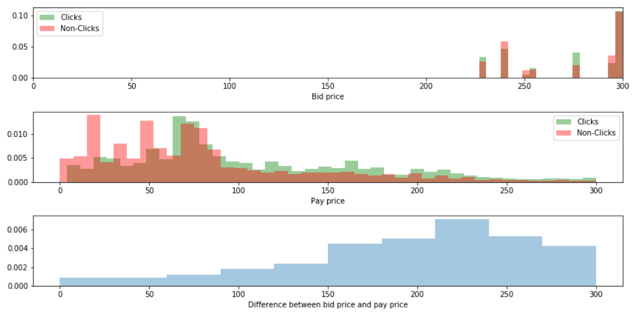

Figure 2: Distributions of Clicks and Non-Clicks (Impressions).

3.2 Problem 2: Basic Bidding Strategy

This section aims to evaluate two non-predictive bidding strategies, so as to provide a benchmark to compare future strategies against.
To evaluate these and future strategies, a function was created that stepped through auctions one by one, deducting the pay price from the budget if the auction was won and sufficient budget was remaining.

3.2.1 *Constant Bidding Strategy - Parameter Optimisation*

As preparation for the next step, we created a function that stepped through a range of constant bid values and calculated statistics, such as how many clicks would have been won. This function was then used to calculate the optimum constant bid for each statistic. This function can be adapted to the advertiser's needs, according to whether they would like to maximise impressions, clicks, CTR or another metric. Additionally, the function tries to create a similar environment as is expected from the validation set: we assume that, as the demand side platform (DSP), we know how many impressions we would like to bid on, as well as our budget. During training, we can thus adapt the budget to the number of impressions being trained on.

By setting the range from 0 to 10% higher than the highest paid price in the training set, the optimum constant bid was calculated. To avoid over-fitting, bootstrapping was used; Ten repetitions were made during which random 75% of the train data was used each time. Figure 3 shows how the statistics of the evaluation criteria change based on different constant bidding values. Taking the most frequently suggested value from all bootstraps we obtained an overall optimal constant bid value of 79.

3.2.2 *Constant Bidding Strategy - Evaluation*

As can be seen in Figure 3, until the bid value is high enough for the total budget to be spent, the number of clicks continues to increase. Beyond this point the number of clicks gradually decreases until it reaches a constant number where every single impression is bought at the payprice until the budget runs out. Table 3 shows the results having used an optimal value of 79 on the validation set. As can be seen the click through rate has improved dramatically by tuning only the constant bid value. This leads us to believe that significant improvements can be achieved by further developing

Table 2: Constant Bid Evaluation on Validation Set

	Bid	Imp.	Clicks	Cost	CTR	CPM	CPC
Const	79	146k	68	6,250	0.047	43	92
Rand	69-89	147k	64	6,250	0.044	43	98

our model. To confirm whether our methods produced the best constant bid value, we tried the same range of constant bid values on the validation set. We found the optimal constant bid value would have been 78, confirming the validity of the method used. We can thus conclude that the train set is representative of all the data combined. 68 clicks can now be used as a baseline to compare future models with. As a constant bid price is deterministic, this is the best performance that can be achieved on the validation set.

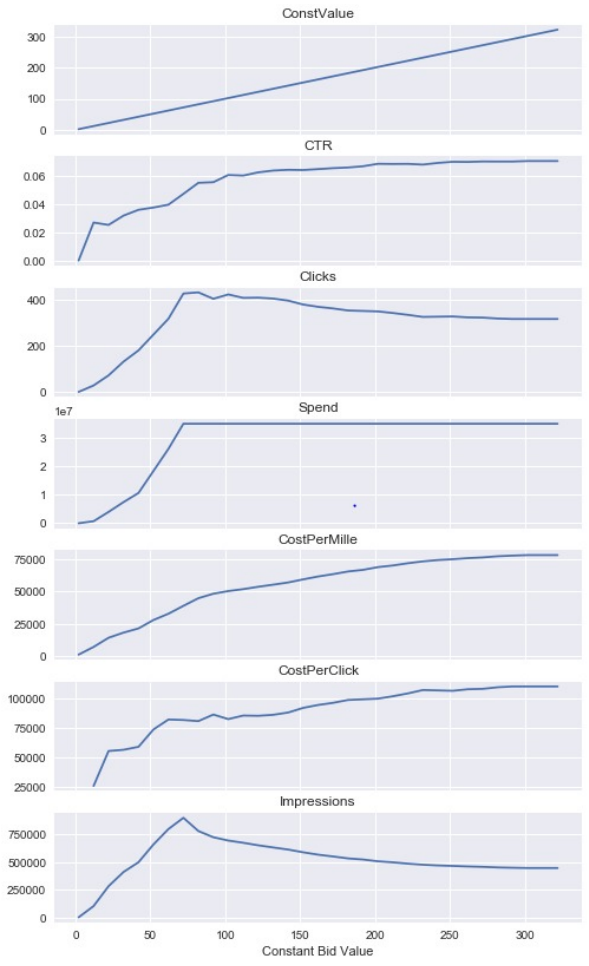

Figure 3: Results from varying the constant bid on the training set

3.2.3 Random Bidding Strategy - Parameter Optimisation

Figure 4 shows the number of clicks obtained for a random bidding strategy bounded between a lower and upper bound on the training set. It can be seen that the maximum number of clicks, around 300 on the training set, is obtained when the average of the lower and upper bounds is 79. This coincides nicely with the value found for a constant bidding strategy. These results were obtained using bootstrapping, with 5 iterations performed for each combination of lower and upper bounds, each time using random 50% of the data.

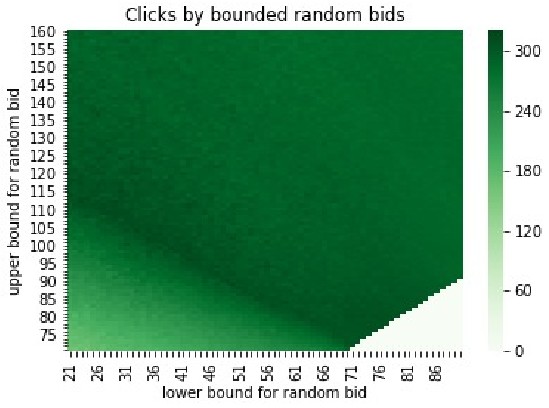

Figure 4: Results from varying a random bid between a lower and upper bound

3.2.4 Random Bidding Strategy - Evaluation

We then looked at the validation set, see table 2, and confirm the optimal lower and upper bounds for the random strategy to be any which average to 79. As can be seen, this configuration achieved an average of 64 clicks with a standard deviation of 2.8. This means that (assuming a normal distribution of results) approximately 5% of the time over 70 clicks will be achieved on the validation data set.

3.3 Problem 3: Linear Combination Bidding Strategy

The linear bidding strategy combines base_bid, pCTR and avgCTR to predict a bid value. E.g

$$bid = base\_bid \frac{pCTR}{avgCTR}$$

The objective is to calculate the pCTR as accurately as possible, as well as to optimise the base_bid constant. An accurate pCTR enables us to bid truthfully and not overspend on erroneous predictions. The pipeline followed consisted of: Data Preparation, Feature Selection, Model Selection/Tuning and Evaluation.

Data Preparation
Within this section, features were filtered out that contained no information, such as IDs, as well as some categorical features whose number of unique values approaching the order of magnitude of number of samples in the training set, such as IP or domain. If these features had been used in future stages, they would have resulted in a very large feature space and made the models prone to overfitting. After this, the data was cleaned by dealing with null values using medians and modes. As mentioned in Section 3.1 some experimenta-

tion resulted in new combined features that looked like they could prove to be stronger predictors, e.g hour of day. Section 3.1 also shows that a major particularity of data within the online advertising space is the extreme class unbalance in the dataset, which causes problems with many algorithms, as predicting the most common class will obtain a accuracy of over 99.99%. A crucial component to producing a robust model is dealing with this imbalance. To this purpose, we first down-sampled the majority class to a dataset of 20,000 impressions and then up-sampled the minority class from 1,793 to 20,000 clicks. This resulted in a perfectly balanced dataset. Down-sampling was performed by randomly removing data, which can result in information loss. Alternatively, if this had produced poor results in future sections, literature suggests down-sampling methods such as Near-Miss or Tomek Links. Up-sampling can cause overfitting by replicating data, which is why the SMOTE (Synthetic Minority Oversampling Technique) algorithm was used. "This algorithm manages to overcome the overfitting problem by creating 'synthetic' examples along the line segments joining the k minority class nearest neighbours, rather than creating actual copies of the data."[3] Finally, the categorical data was encoded using dummy variables, resulting in 701 features overall.

Feature Selection

This large feature space was reduced to minimise computational expense and the chance of overfitting. For this purpose, we used recursive feature elimination which is an instance of backward feature elimination. This performed multiple iterations of training a model where during each iteration, the least important features are removed. Feature selection was repeated using logistic regression and random forests, as we intended to use these models in following steps.

Model Selection/Tuning

We decided to use Logistic Regression, Random Forest and XGBoost methods, as these are general purpose models commonly suggested in literature. These could later be replaced if they resulted in unsatisfactory accuracies. To tune the hyper-parameters of these models, we performed an exhaustive grid search aiming for maximal precision. Precision was more important than recall or accuracy as it is representative of pCTR.

Base Bid Optimisation

To determine the optimal base_bid value, a similar procedure to Section 3.2 was used, where we iteratively used increasing values as inputs to the linear bidding formula. The increasing values were taken from the range [0.2*mean(paid_price), 6*mean(paid_price)]. (Ideally, this test would be performed using cross-validation on the train set, however as this was computationally expensive, we used the validation set).
Figure 5 shows the effect of base_bid and CTR predictor on Clicks, CTR and CPC. In general XGBoost yields the best results, with maximum clicks achieved at base_bid = 89. As we can see, up until this base_bid value, the entire budget can not be spent, and as such, the click count continues to grow. After this value, the multiplication of base_bid and pCTR tends to be high enough to buy impressions even if pCTR is low and causes us to run out of budget prematurely.

Table 3: Linear Strategy Evaluation on Validation Set

	Bid	Imp.	Clicks	Cost	CTR	CPM	CPC
LR	71	163k	93	6,067	0.0572	37	65
RF	71	138k	152	5,963	0.1098	43	39
XGB	89	116k	162	6,046	0.1396	52	37

Evaluation

Table 3 shows that all trained models perform significantly better than both constant and random bidding strategies. Among them, the random forest and XGBoost present by far the better performance, earning a significantly higher number of clicks (152 and 162 respectively), and a 40% lower cost per click (39 vs 65).

Additionally we can see the importance of adjusting the base bid constant individually for each CTR predictor.

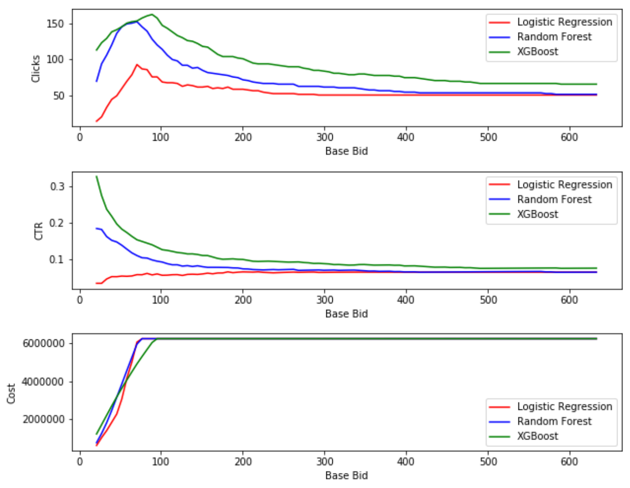

Figure 5: Linear Strategy results from varying the base bid on the validation set

In summary we almost tripled our CTR to 0.14% for a 40% lower cost per click, and a 21% higher Cost Per Mille (CPM). We feel this is a very significant improvement.

3.4 Problem 4: Your Best Bidding Strategy

The three authors of this paper individually worked on building better bidding strategies, of which the results are summarised in table 4. First of all, all three authors agreed to build different models to predict pay price, as opposed to optimising a constant value. One report additionally works on an improved method of dealing with the data imbalance, by creating multiple down-sampled datasets. The other two reports focus on developing improved non-linear strategies. These reports were developed with an aim to improve either clicks or CTR, with either the single-agent or multi-agent scenarios in mind. Our best model for the single-agent scenario achieved 1 additional click with a CTR of 0.22% (improvement from 0.14%). Additionally, for the multi-agent scenario, a strategy was developed that increases CTR dramatically by bidding on a subset of impressions with the highest pCTR.

Table 4: Individual Strategies on Validation Set

pCTR	pBB	Strategy	Imp	Clicks	Cost	CTR	CPC
MF	RFG	Linear	73k	160	5,798	0.2201	36.18
XGB	Lasso	Exp.	71k	157	6,240	0.2207	39.75
XGB	XGB	avgBB/pBB	125k	160	6,230	0.1279	49.51

3.5 Problem 5: Combine models

3.5.1 Comparison of individual models

The first report that developed a method to deal with the imbalanced dataset achieved high CTR and a high click count on the validation set. We believe this is our strongest model, that can be improved upon using ensemble methods. The second and third reports experimented with various formulas to combine pBB and pCTR. The experimental work produced by these reports showed that predicting pay price does not allow us to target cheaper clicks, but instead allows cheaper impressions to be targeted if an advertiser's aim is high exposure. The third report suggests a method for determining the relationship between pCTR and the actual CTR. This can later be used for targeted bidding for the multi-agent scenario, based on how many clicks we would like to aim for. This allows us to bid high for a low number of clicks with high certainty or spread our budget across less certain clicks.

3.5.2 Combined Bidding Strategy

As mentioned in sections 3.3 and 3.4 of our individual reports, we created different models for predicting CTR and pay price and determined various formulas to combine the two. We could hence create pCTRs and pBB for the validation set using each model individually. As such, we performed a 3-way grid search that combines all individual components together. With 4 pCTR files, 5 pBB files and 4 formulas, this resulted in 80 combinations, with the top 3 displayed in table 4. The final combined model consisted of an ensemble model for pCTR using the MF and XGBoost models and a linear strategy. As base bid we started from the mean pay price of the training set and tried to tune it for optimal performance. Results are presented in Figure 6 and Table 5

Table 5: Ensemble Strategy on Validation Set

pCTR	Strategy	Imp	Clicks	Cost	CTR	CPC
Ensemble	Linear	76k	165	6,149	0.2167	37.27
Ensemble	Linear	68k	160	5,335	0.2340	33.35
Ensemble	Linear	41k	142	2,841	0.3429	20.01

We concluded that accurate click prediction was the most important factor and hence worked on some ensemble models to produce more accurate pCTR. First of all, for each impression, the pCTR of our 4 best models were multiplied together and given as input to our best formula. The same was repeated for our top two pCTR predictors, which achieved better results. In order to try and improve this ensemble method, we tried additionally bidding on the top X% of impressions with highest pCTR in all other models, to see if clicks or CTR would improve. This however was not the case, indicating that the combination of our best two models cannot be improved by adding information provided by other models.

3.5.3 Multi-Agent Reinforcement Learning

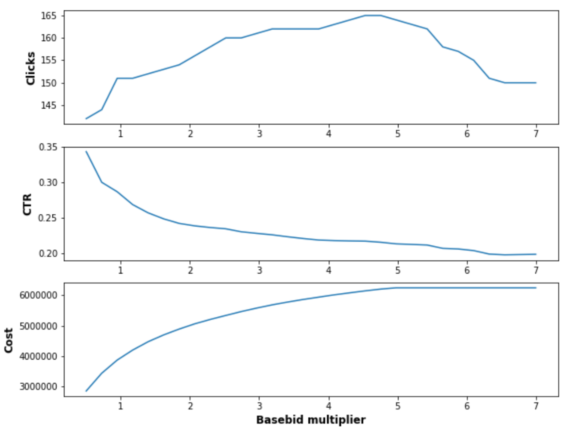

Figure 6: Ensemble Model results from varying the base bid on the validation Set

To evaluate models via multi-agent reinforcement learning (MARL), a script was written to simulate the environment that our strategy will be competing in. That script took multiple bidding strategies (agents) and budget constraints into consideration and evaluated them on the validation set. During each auction, the strategy with the highest bid won the auction at second price, unless that agent had already exhausted it's budget. This script then took our 3 best models from section 3.5.2 , the constant and random bidding strategies and our best linear model and held an auction with these.

As explained in section 3.1, the environment competing in is extremely competitive, requiring teams to use a much higher average bid. To determine the constant by which the bids should be multiplied for each model, an iterative process was implemented taking different values for this constant. Whilst this provided us with a constant that was most likely to work against the 31 other teams, it is impossible to determine a constant that is perfect for any scenario as we cannot know if other teams will account for the increased competitiveness of this environment. As such, we assumed that roughly 50% of teams would take this into account and we lowered our constant accordingly. This constant would later need to be adjusted iteratively in a real-world scenario based on latest performance.

pCTR would also need to be iteratively calibrated in the same way. It is evident that over time, actual CTR in the data could change, meaning that our pCTR needs to be scaled accordingly. To simulate this we used the API provided, to estimate the relationship between pCTR and CTR, in a changing landscape of other teams' best strategies. This relationship was thus analysed on the test set (see Figure 7) and would be re-created daily in a real-life scenario.

The multi agent simulation we built demonstrated the non-linearity in a multi-agent environment. Systems that were ostensibly the same at predicting clicks (XGBoost and Forest of Random Forests), when placed in the multi-agent performed dramatically different when competing against

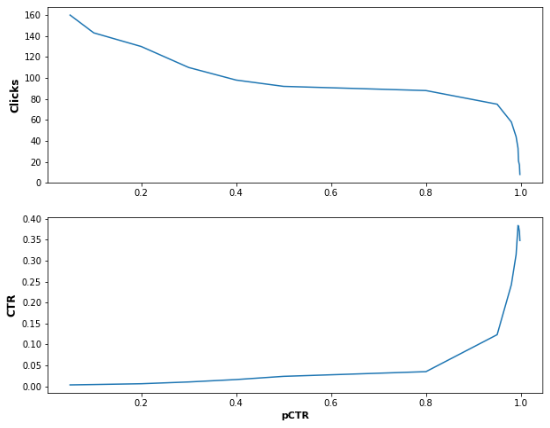

Figure 7: Clicks and CTR vs pCTR in Multi-Agent environment

one another.

3.5.4 *Performance on Test Set*

At the time of submission, the strategies developed performed extremely well. In the scenario where we bid against the pay price, we achieve a click count of 176 and a CTR of 0.0022. Although the click count is slightly lower than some teams may have achieved, we believe that the noticeably high CTR is extremely valuable in terms of adjusting to advertisers' needs. Our model excelled in the multi-agent scenario when bidding against 31 other teams. It consistently ranked first during the last two weeks before submission, with the final results on the night of submission being: Rank 1, Clicks 41, CTR 0.0029.

3.5.5 *Further Work*

A more effective way of combining the individual models for pCTR and base_bid could potentially be the use of stacked models. Stacked models use the predictions of individual models as new features to train a meta-model (or meta-learner) and is a technique that is vastly used in the general Data Science field (i.e. especially in Kaggle Competitions), which has also been proved to outperform individual models in many cases.

3.5.6 *Better Use of Existing Data*

Due to the in-balance in the datasets, much of the dataset was unused in our models. More advanced techniques could be explored to counter this.

4. CONCLUSION

In this assignment, we implemented a bidding strategy that allows us to compete in a real-time bidding system (single and multi-agent?). Our main intervention lies on the hypothesis that the optimal bidprice is a function of three independent components, namely pCTR, base_bid and formula (e.g. the model that combines pCTR and base_bid to calculate the final bidprice). In order to create models that could accurately predict or optimise the aforementioned three components, we went through an exhaustive procedure consisting of data cleaning and preparation, feature engineering and selection, experimentation with different machine learning algorithms and hyperparameter tuning. The validation of the results was performed on the validation set itself, as well as in a multiagent RL environment that could simulate the actual real-time bidding system environment.

Graph 6 shows the evaluation of the various bidding strategies on the validation set. We can see that from the constant bidding model through to the ensemble model , significant improvement were achieved. As all the training was performed on the train set and the validation set was used purely for evaluation purposes, we feel that our model should perform similarly well on the test set. It is difficult to predict how our strategy will perform in the multi-agent environment, however 30 clicks (replace with actual number) are a reasonable number to expect.
Regarding blocking factors, one of the main constraints was the size of the data in place. Having been able to access machines with greater computational power we would be able to train our models using larger proportion of the data (or even the whole dataset) rather than undersampling the dataset, as well as increasing the complexity of the machine learning model used (i.e. increase the number of decision trees in the final random forest model for pCTR).

Table 6: Individual Strategies on Validation Set

pCTR	pBB	Strategy	Imp	Clicks	Cost	CTR	CPC
		Const:79	146k	68	6250	0.047	92
		Rand:69-89	147k	64	6250	0.044	98
LR	71	Linear	163k	93	6,067	0.0572	65
RF	71	Linear	138k	152	5,963	0.1098	39
XGB 9	89	Linear	116k	162	6,046	0.1396	37
MF	RFG	Linear	73k	160	5,798	0.2201	36.18
XGB	Lasso	Exp.	71k	157	6,240	0.2207	39.75
XGB	XGB	avgBB/pBB	125k	160	6,230	0.1279	49.51
Ensemble	const	Linear	76k	165	6,149	0.2167	37.27

5. CODE LOCATION

https://github.com/DwanevanderSluis/COMPGW02_Web_Economics.git

5.1 Group Notes

The group worked very well together meeting for a full day at Dwane's house each week - with additional individual work between meetings. We created multiple Jupiter notebooks and used GitHub from the very start. We feel that this work structure enabled us to build a deeper understanding of the topic, as we shared and developed our individual and group ideas/intuitions weekly. We feel that this collaboration triggered out-of-the-box ideas, leading to high-level work and thus great results.

Achilleas
Co-ordinated data preparation including feature engineering, cleaning and modelling. Experimented with multiple base bid predictors, different loss functions and hyper parameter tuning. Created also code to evaluate performance on validation set and investigated dynamics when pCTR thresholds were altered.

Akis
Produced the code for constant and random bidding strategies as well as the code to evaluate these and other strate-

gies on the validation set. Developed non-linear formulas that obtained better results than the linear formula. Took Dwane's multi-agent script and adjusted it to recreate the competitiveness of the environment with all 31 teams. Experimented with different types of ensemble methods and took charge of frequent evaluations on the test set.

Dwane
Created the initial version of code that simulated multiple player auctions that did not allow any player to exceed budget. Created a multi forest for click prediction that used (to our knowledge) a novel way creating a (more) balanced dataset, while defending against over fitting, and extracting more information for the dataset.

6. REFERENCES

[1] A. Animesh, V. Ramachandran, and S. Viswanathan. Online advertisers bidding strategies for search, experience, and credence goods: An empirical investigation abstract.

[2] T. Chakraborty, E. Even-Dar, S. Guha, Y. Mansour, and S. Muthukrishnan. Selective call out and real time bidding. In A. Saberi, editor, *Internet and Network Economics*, pages 145–157, Berlin, Heidelberg, 2010. Springer Berlin Heidelberg.

[3] N. V. Chawla, K. W. Bowyer, L. O. Hall, and W. P. Kegelmeyer. Smote: synthetic minority over-sampling technique. *Journal of artificial intelligence research*, 16:321–357, 2002.

[4] Y. Chen, P. Berkhin, B. Anderson, and N. R. Devanur. Real-time bidding algorithms for performance-based display ad allocation. In *Proceedings of the 17th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '11, pages 1307–1315, New York, NY, USA, 2011. ACM.

[5] H. B. McMahan, G. Holt, D. Sculley, M. Young, D. Ebner, J. Grady, L. Nie, T. Phillips, E. Davydov, D. Golovin, S. Chikkerur, D. Liu, M. Wattenberg, A. M. Hrafnkelsson, T. Boulos, and J. Kubica. Ad click prediction: a view from the trenches. In *Proceedings of the 19th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining (KDD)*, 2013.

[6] A. More. Survey of resampling techniques for improving classification performance in unbalanced datasets. *arXiv preprint arXiv:1608.06048*, 2016.

[7] M. Stange and B. Funk. Real-time-advertising. 56:335–338, 09 2014.

[8] S. Yuan, J. Wang, B. Chen, P. Mason, and S. Seljan. An empirical study of reserve price optimisation in real-time bidding. In *Proceedings of the 20th ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, KDD '14, pages 1897–1906, New York, NY, USA, 2014. ACM.

[9] W. Zhang, S. Yuan, 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*, KDD '14, pages 1077–1086, New York, NY, USA, 2014. ACM.

[10] W. Zhang, S. Yuan, J. Wang, and X. Shen. Real-Time Bidding Benchmarking with iPinYou Dataset. *ArXiv e-prints*, July 2014.