Click prediction model

Data Science group

April 6, 2016

Abstract

Using data from Centro campaigns, we developed a click prediction model. This model can be used for a click prediction during campaign planning,and during a budget optimization when maximization of amount of clicks is considered as one of campaign goals.

1 Introduction

Number of clicks received during an advertising campaign often serves as one of important indicators of a campaign performance. For this reason, it would be preferable to predict the number of expected clicks provided different campaign conditions. To make such predictions we need a predictive tool that can be made using one of machine learning techniques and a set of variables (features) which potentially affect clicks.

It would be also desirable to apply this tool to optimize advertiser's budget allocation among different sites, platforms, ad classes, etc. To analyze a few different scenarios in a real time, the optimal budget allocation should be performed quickly for each new input, e.g. a set of sites. A gradient descent method used for such tasks requires the click prediction function to be smooth and differentiable. It significantly narrows a set of possible machine learning techniques, and led us to the model described in section 3 as a possible approach provided those limitations.

As the input variables for the click model we would need to use variables which are typically used in the campaign planning. Those variables are available through the Pentaho data base, and are discussed in Section 2.

2 Data selection

To build a click prediction model, we used data collected since January 1, 2013 to reflect most recent advertising tendencies, on the one hand, and to have enough statistics to train/build the model, on the other hand. The Python script used for the data retrieval from a data base can be found by this link `https://github.com/centro/Data-Science/blob/master/Recommendation-Engine/dev/Scores/MMS/mms_data_pull.py` We use data from monthly fact tables, so all the metrics (e.g. the number of impressions, clicks) for a given campaign are calculated per month.

The extracted data have to satisfy the following selection criteria:

- $5000 <$ delivered_impressions $< 10^7$;
- delivered_clicks > 1;

Table 1: Click-through-rate for different values of Platform, Ad class, Cost type and Cost subtype.

Platform	Ad class	Cost type	Cost subtype	CTR
Mobile	Video	CPC	CPC	0.558524
Desktop	Video	CPC	CPC	0.487362
			RTB CPC	0.175532
	Display	CPC	RTB CPC	0.025133
Mobile	Display	CPC	RTB CPC	0.023458
	Video	CPM	CPM	0.021617
Desktop	Display	CPC	CPC	0.017076
	Video	CPM	CPM	0.010680
Mobile	Display	CPC	CPC	0.010662
		CPM	CPM	0.006207
Desktop	Video	CPM	RTB CPM	0.004181
Mobile	Display	CPM	RTB CPM	0.001812
Desktop	Display	CPM	CPM	0.001177
			RTB CPM	0.000664

- $0 < \text{CTR} < 1$;
- delivered_gross_revenue / ordered_gross_revenue > 0.1;
- cost_type != 'Added Value' & cost_type != 'Flat Rate'.

We are building click model to take into account variations in amount of clicks as a function Platform, Ad class, Cost type and Cost subtype. Tables 1–3 show some summary statistics in the selected data. Table 1 shows average Click-Through-Rate (CTR) for 16 combinations of of Platform, Ad class, Cost type and Cost subtype, ordered by CTR. Here we did not include Cross-platform as an intermediate case between Desktop and Mobile (but did use it in the click prediction model). Table 2 shows average CTR versus Google verticals for two platforms and two ad classes, with cost type and subtype fixed as 'CPM'. Table 3 shows distribution of the ordered gross revenue for the same combinations Here the results are ordered by the revenue and normalized to the total revenue.

We have statistically compared CTRs for different ad classes, and found that there are just two distinct ad classes, Video and non-Video. The later includes Display, Audio, Text, and others (marked as Misc and TBD in the data base). Since the ad class Display significantly dominates both in terms of the number of campaigns and gross revenue, we call non-Video class as just Display, and use it in this way in our model.

Below we list all the variables used as an input to the click model:

features = number of impressions, CPM, flight_days, number of unique visitors, number of page views, ad class, Google vertical, platform, cost_type, cost_subtype.

Please note that first five variables are continuous, and the last five are categorical (non-numerical). We use the number of page views being normalized on the number of unique visitors, i.e. an average page views per user.

Table 2: Average CTR vs Google verticals for two platforms and two ad classes. The number of impressions is required to be within $[150000, 350000]$ (overall average is about 250000). Cost type and subtype fixed as 'CPM'.

Google vertical	Desktop Display	Desktop Video	Mobile Display	Mobile Video
Home & Garden	0.0029	0.0222	0.0047	--
Arts & Entertainment	0.0027	0.0580	0.0118	--
Beauty & Fitness	0.0021	0.1344	0.0043	0.0060
Shopping	0.0016	0.0524	0.0060	--
Business & Industrial	0.0014	0.0508	0.0065	0.2082
Computers & Electronics	0.0014	0.0375	0.0051	--
Finance	0.0014	0.0139	0.0061	0.1534
Jobs & Education	0.0014	0.0103	0.0072	0.0065
Travel	0.0014	0.0432	0.0063	0.0023
Sports	0.0013	0.0995	0.0062	--
Food & Drink	0.0012	0.0323	0.0069	0.0050
Health	0.0012	0.0318	0.0044	0.0045
Law & Government	0.0011	0.0175	0.0062	0.0052
Real Estate	0.0011	0.0166	0.0046	0.0125
Internet & Telecom	0.0009	0.0452	0.0048	0.0220
Autos & Vehicles	0.0008	0.0168	0.0048	0.0034

Table 3: Ordered gross revenue for different values of Platform, Ad class, Cost type and Cost subtype.

				ordered_gross_revenue (%)
platform	ad_class	cost_type	cost_subtype	
Desktop	Display	CPM	CPM	73.07
	Video	CPM	CPM	9.33
Mobile	Display	CPM	CPM	6.36
Desktop	Display	CPC	RTB CPC	4.68
	Video	CPC	RTB CPC	1.58
	Display	CPM	RTB CPM	1.56
		CPC	CPC	1.50
Mobile	Display	CPC	CPC	0.78
Desktop	Video	CPC	CPC	0.51
Mobile	Video	CPM	CPM	0.40
		CPC	CPC	0.10
Desktop	Video	CPM	RTB CPM	0.09
Mobile	Display	CPM	RTB CPM	0.04
		CPC	RTB CPC	0.02

3 Click model

Initial goal of the click model was an optimization of a budget allocation among a few selected sites (publishers) and maximizing total number of clicks on those sites. For such an optimization problem, we need *an explicit functional form* for the number of expected clicks as a function of important numerical metrics. Since the number of impressions (imp) is most influential variable, we have parametrized the number of clicks ($clicks$) as

$$clicks = \alpha(imp)^{\beta}, \tag{1}$$

where α and β are free parameters that have to be found from a fit to data points. Since imp is directly related to the total budget B and Cost-Per-Impression (CPI) as $imp = B/CPI$, Eq. 1 can be re-written as

$$clicks = \alpha(B/CPI)^{\beta}. \tag{2}$$

From Tables 1 and 2 it is clear that expected $clicks$, in addition to the number of impressions (or CPM), also depend on categorical variables such as Platform, Ad class, Cost type, Cost subtype and Google vertical. Basically, it means that we need to consider Eq. 1 for all of the possible combinations of those categorical variables. However, it also clear that not all of those combinations would be independent. That is, we should be able to describe some of them by the same curve (Eq. 1). In other words, in terms of expected $clicks$, for a given B and CPI, just a few clusters formed in the space of those categorical variables would be independent.

To follow this idea, we apply K-means clustering algorithm [1] to all the selected data and fit $clicks$ using the pints in all of those clusters. The clustering procedure starts from a random split of all the points among K clusters ($K = 10$ is chosen as default value). Then we perform a fit for $clicks$, and then re-cluster by finding a "closest" curve (cluster) for each point. This iterative procedure is converging, and continues until the α_i, β_i $(i = 1, K)$ coefficients changes by less than 1% as compared to a previous iteration.

Typically, β_i, found from the fit, varies within $0.7-1.0$. It means that the number of clicks saturates with the number of impressions, as shown on Fig. 2. It also automatically means that CTR should decrease with the number of impressions as

$$CTR = \alpha(imp)^{\beta} - 1. \tag{3}$$

Fig. 1 shows dependence of CTR vs impressions with $\beta = 0.8$ and 0.9 as examples. That is, with $\beta < 1$, we cannot compare click performances of different campaigns (or campaign classes) based on CTR unless we fix same range of the number of impressions. In Appendix I we discuss why we observe such a dependence of CTR vs impressions by analysing Sizmek pre-aggregated cookie-level data.

For each new event (campaign) with specific values of those categorical variables, there is a finite probability to belong to each of those K clusters. This is a typical classification task that can be solved using one of ensemble methods. To find a cluster assignment probability, we used $RandomForest$ (RF) [2] with a feature vector, shown in the end of Section 2, as an input. The RF was trained to classify events by the K clusters. The output from RF is the classification probabilities $p_i,\ (i = 1, K)$.

Using these probabilities and the predicted number of clicks each cluster, the total number of clicks

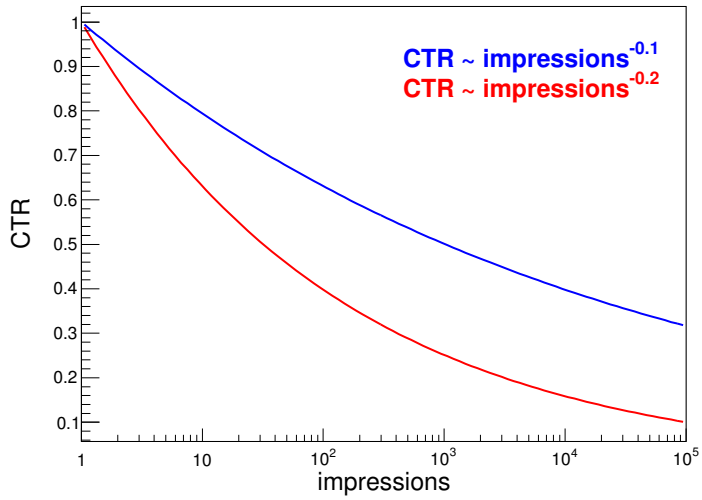

Figure 1: CTR as a function of number of impressions for $\beta = 0.8, 0.9$ and $\alpha = 1$ (see Eq. 3).

can be expressed as

$$predicted\ clicks = \sum_{i=1}^{K} p_i\ clicks_i = \sum_{i=1}^{K} p_i\ \alpha_i (imp)^{\beta_i}. \tag{4}$$

Uncertainties on the total number of clicks are coming from classification probabilities p_i and coefficients α_i, β_i for the individual fits for K clusters.

3.1 Validation, bias correction and uncertainty.

The predicted clicks are tested agains true clicks using a validation sample which was not used in the model training. This sample was created by taking 15% random events from the whole sample. Figure 3 compares predicted clicks versus true clicks for four cases, {Desktop, Mobile} x {Display, Video}. Due to a large variation in the number of clicks, we took $log10$ for both axes. Other input variables, including categorical, may take any value here. In Figure 4 we make a similar comparison for two cost types and cost subtypes with fixed platform and ad class as (Desktop, Display).

While we do see an approximate coherent behavior of true and predicted clicks, we have to estimate a size of a possible bias and uncertainty in our predictions. Figure 5 shows distributions of relative prediction accuracy defined as

$$Accuracy = (true - predicted)/true \tag{5}$$

The accuracy is estimated in 6 bins predicted clicks which serve as a scale for our estimates and shown in Fig. 5. We see that, on average, we overstimate number of clicks at a small scale and underestimate at high scale. One can use these intervals to parametrize the correction factor needed to corrected predicted clicks back to the true level (which is obtained from Eq. 5):

$$Correction = 1/(1 - \langle Accuracy \rangle) \tag{6}$$

Behavior of the correction factor versus predicted clicks is shown in Fig 7, top plot. We see that the required correction factor is within $\pm 20\%$ for the expected clicks $< 10,000$. The bottom plot

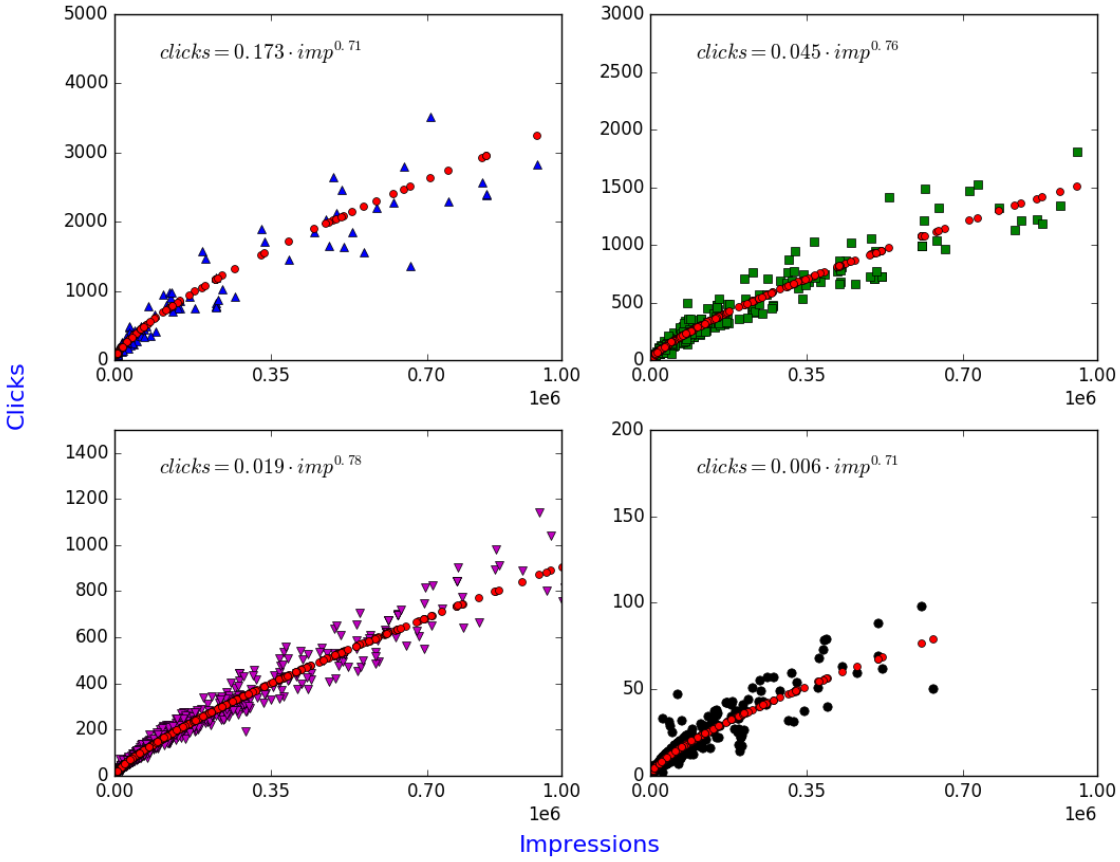

Figure 2: Clicks as a function of number of impressions for four different clusters. Shown data points versus fitting function for this cluster. Parametarization of clicks versus impressions are also presented (red circles)

shows uncertainty of the clicks prediction. For example, it is about 48% when the expected clicks is about 1,000 and drops to ≈30% for $> 10,000$ clicks.

Fig. 6 shows the accuracies after applying the bias correction. We see that mean value agrees with zero now within 3% in most cases (please also note that the number of events in the chosen 6 intervals has also changed due to events migration after applying the bias correction).

In Appendix II we discuss how big are the observed uncertainties as compared to a nature of underlying user clicking process.

3.2 Variable importance

Figure 8 shows importance of variables/features listed above for the click prediction. All the variable importances are normalized to the maximum importance corresponding to CPM. CPM is immediately followed by the number of impressions, and then the numbers of unique visitors and views per visitor.

In Fig. 9 we studied an effect of the removing the last two variables, unique visitors and views

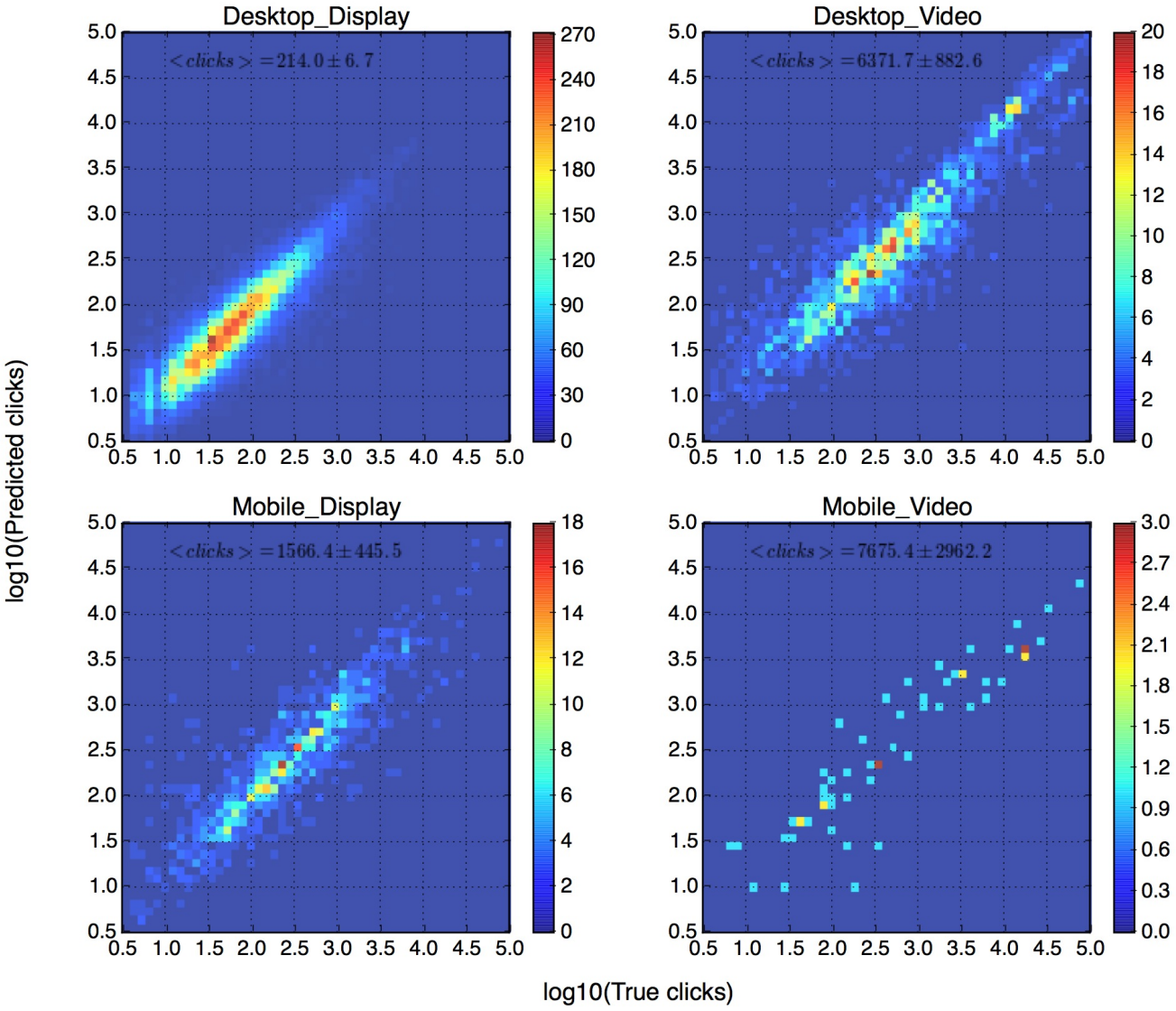

Figure 3: Predicted clicks versus true clicks for two platforms and two ad classes. On each plot we also show average number of true clicks for a given combination.

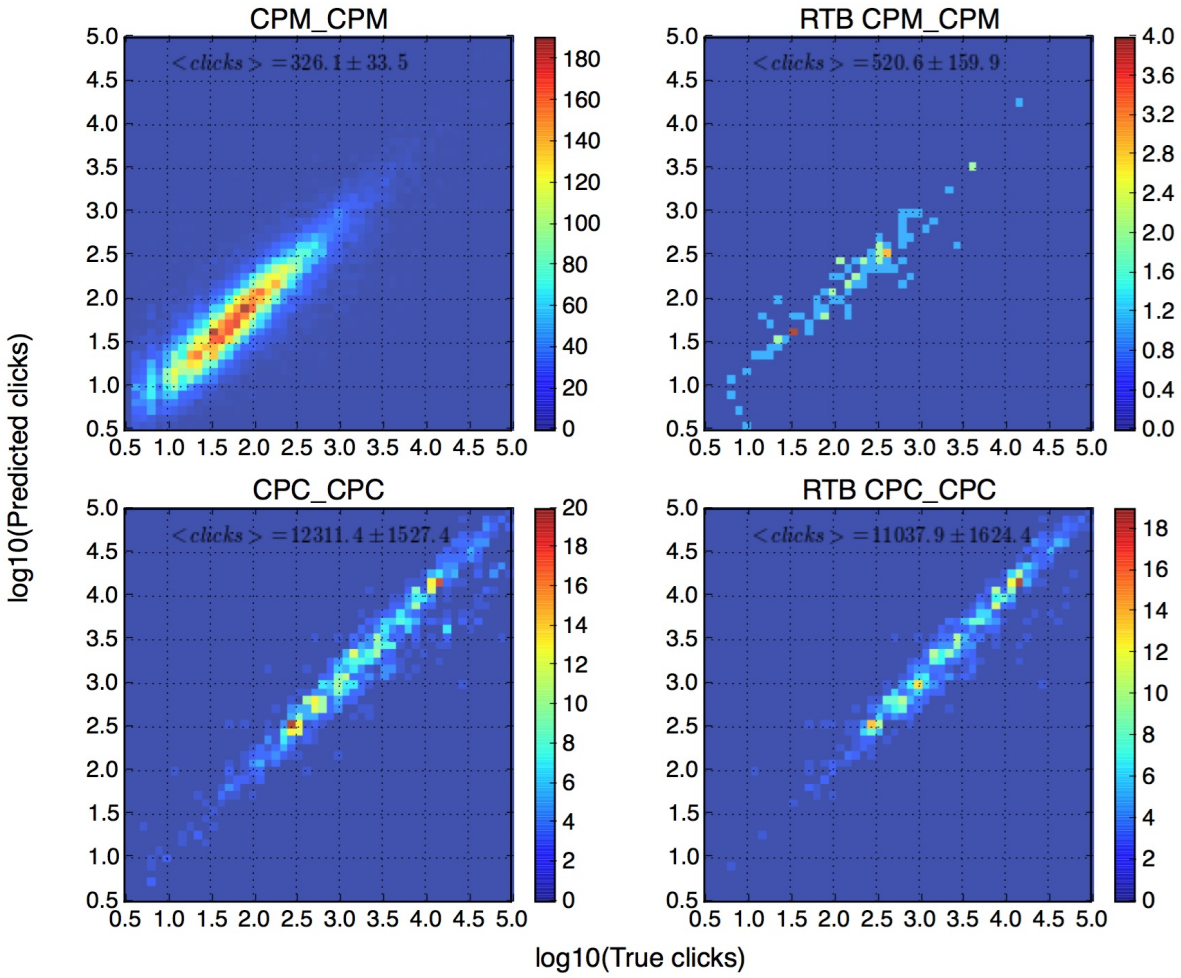

Figure 4: Predicted clicks versus true clicks for two platforms and two ad classes with fixed platform and ad class as (Desktop, Display). On each plot we also show average number of true clicks for a given combination.

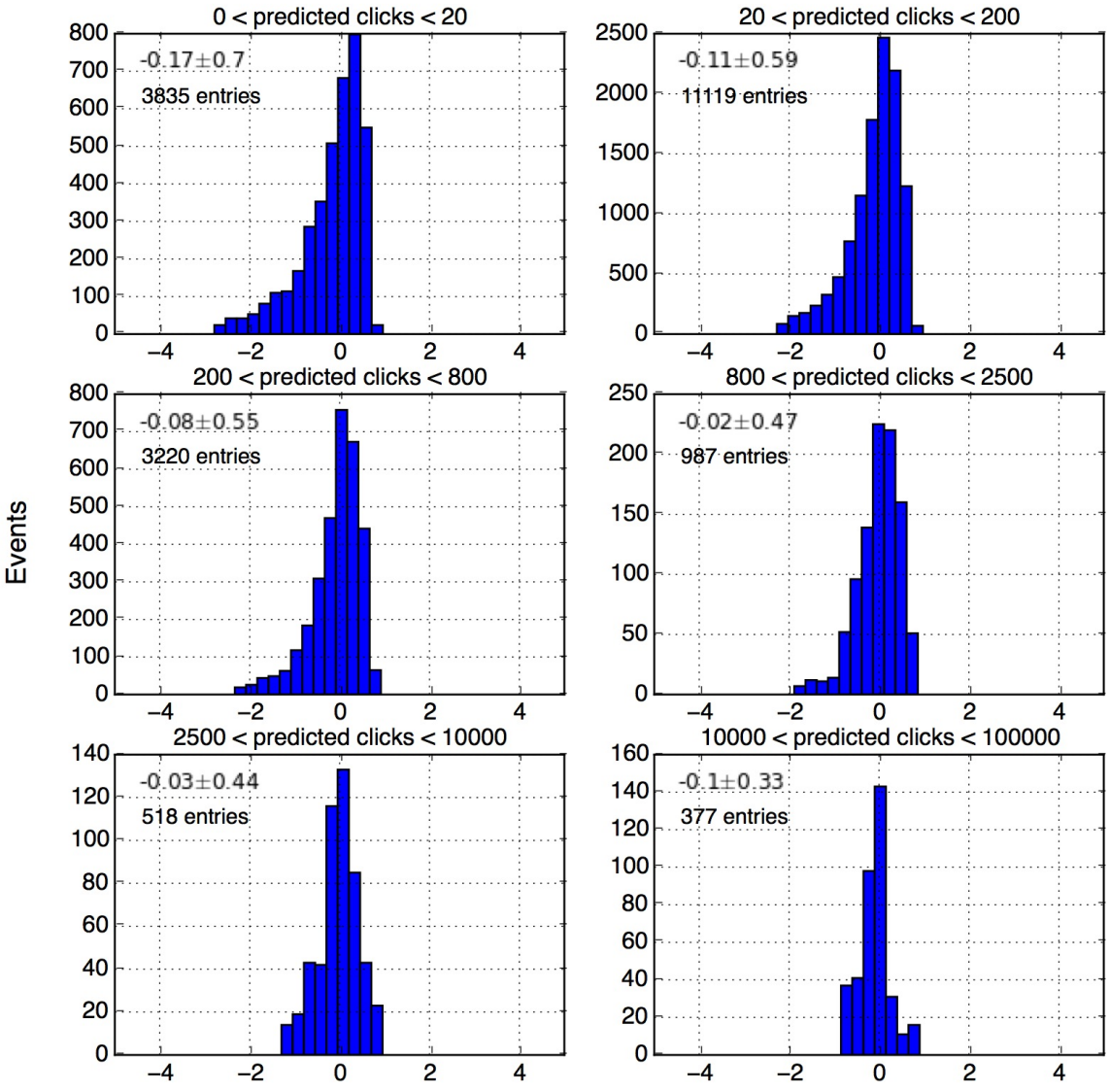

Figure 5: Relative accuracy of click prediction model in six bins of the predicted clicks. The accuracies are shown in the left top corner of each plot.

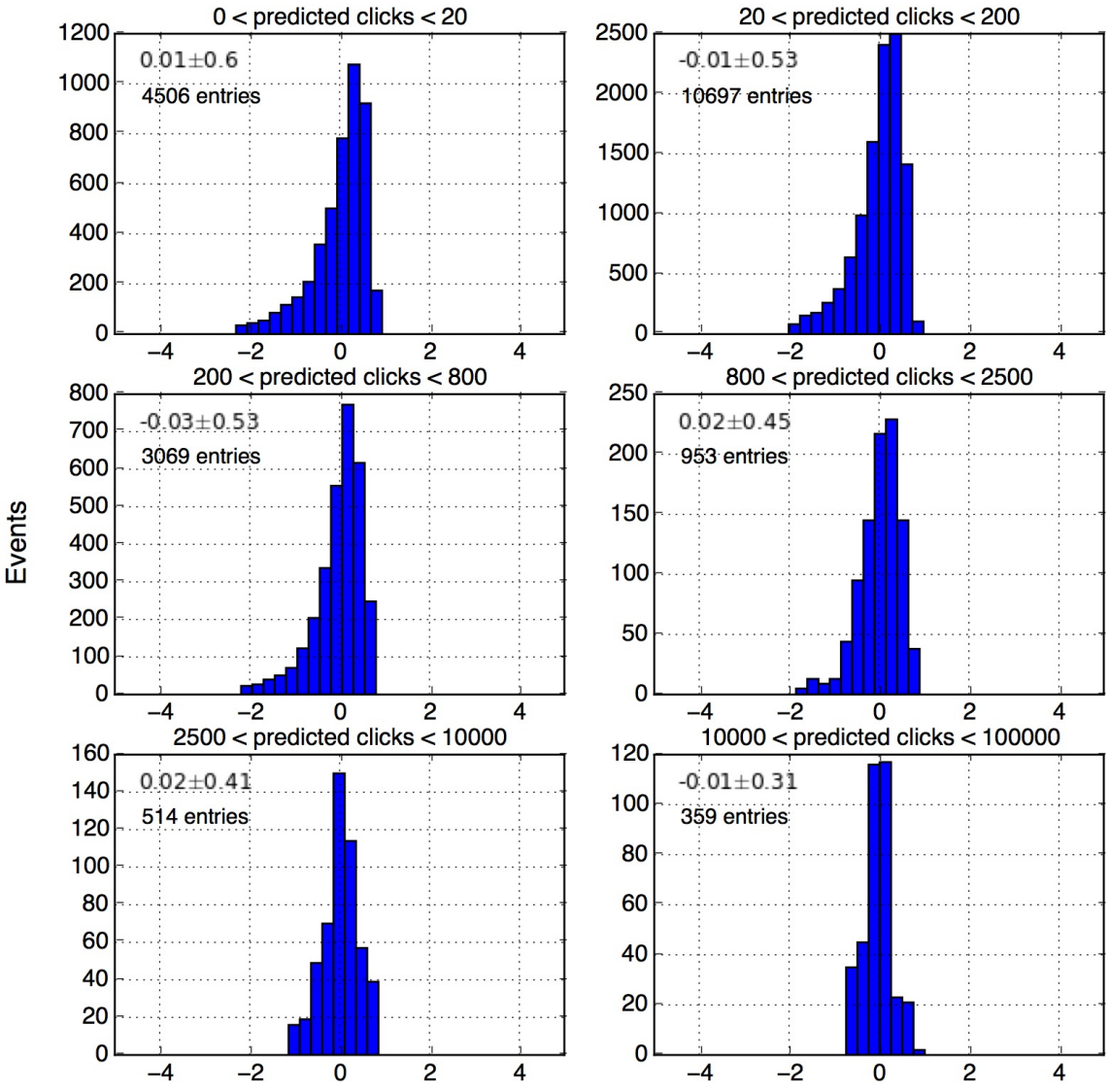

Figure 6: Relative accuracy of click prediction model in six bins of the predicted clicks *after the bias correction being applied*. The accuracies are shown in the left top corner of each plot.

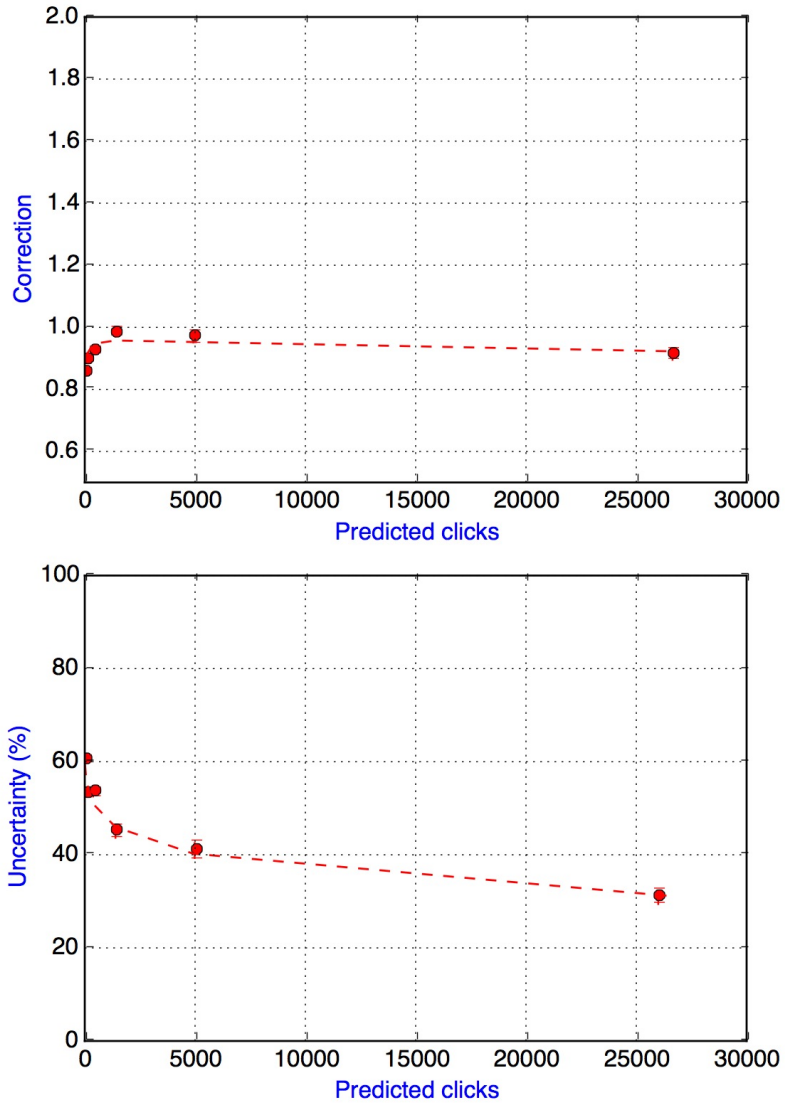

Figure 7: Correction factor for the average bias and uncertainty of the click prediction model.

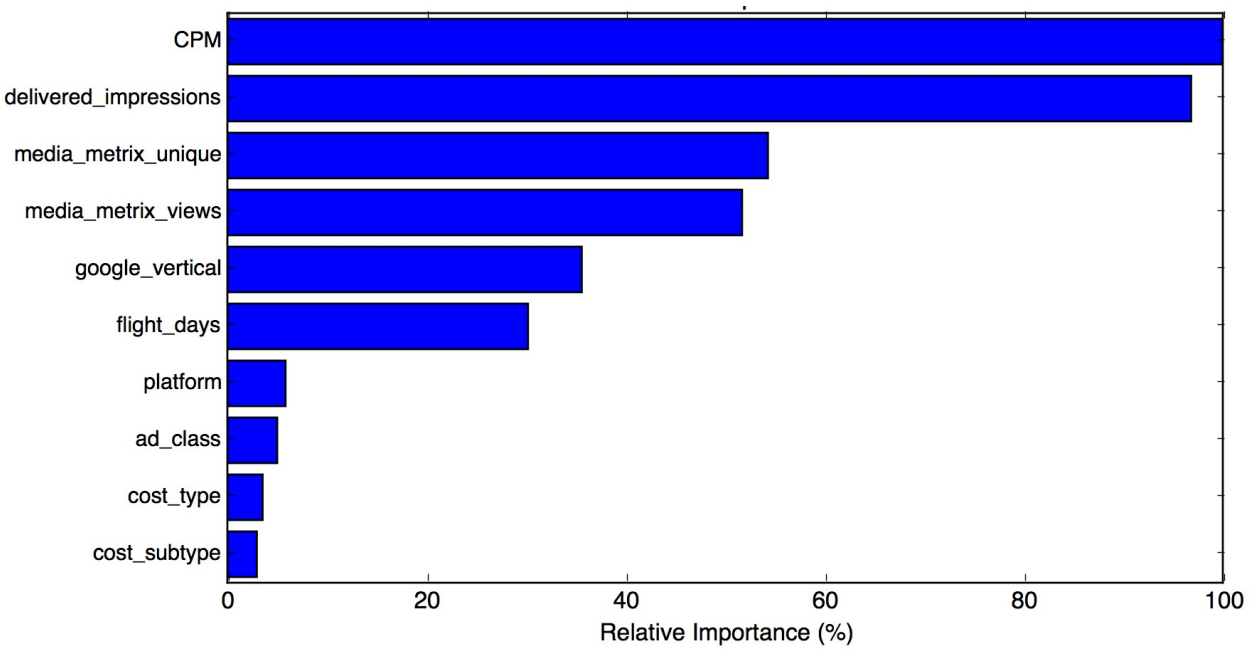

Figure 8: Importance of input variables for the click prediction.

per visitor from the feature vector. Being compared with Fig. 6, it is clear that relative accuracy degrades by about 20–25%.

4 Discussion

The built model can be used to compare expected number of clicks for different values of categorical variables with *same* other inputs. Table 4 compares expected number of clicks for different values of Platform, Ad class, Cost type and Cost subtype using $CPI = \$0.005, B = \$1000$, flight_days=30, and vertical ='Business & Industrial'. We see, for example, that we typically have more clicks for cost type 'CPC' than for 'CPM', and the largest amount of clicks corresponds to 'Mobile Video', followed by 'Desktop Video'.

Figure 10 shows an expected number of clicks versus flight days. We limited x-axis by 30 since we have built model using data from *monthly* fact tables. In Figure 11 we show an expected number of clicks versus page views per unique visitor (for half a million of unique visitors). As we might expect, one can see that the predicted number of clicks slowly increases in both cases.

The click prediction model is implemented in Python codes and available at Stash `https://stash.centro.net/projects/CENDS/repos/clickprediction/browse`.

We have also developed API for the click model. It may either predict a number of clicks with uncertainties for a given budget and other campaign inputs, or predict a number of required impressions and a budget. Please see examples in: `https://stash.centro.net/projects/CENDS/repos/clickprediction/browse/scripts/click_func_API.py`.

For example, (a) for a Desktop Video campaign with budget \$50000, CPM=\$5 and cost type CPM, we predict 4965 ± 1995 clicks; (b) for the similar campaign, to get 5000 clicks with 70% (90%)

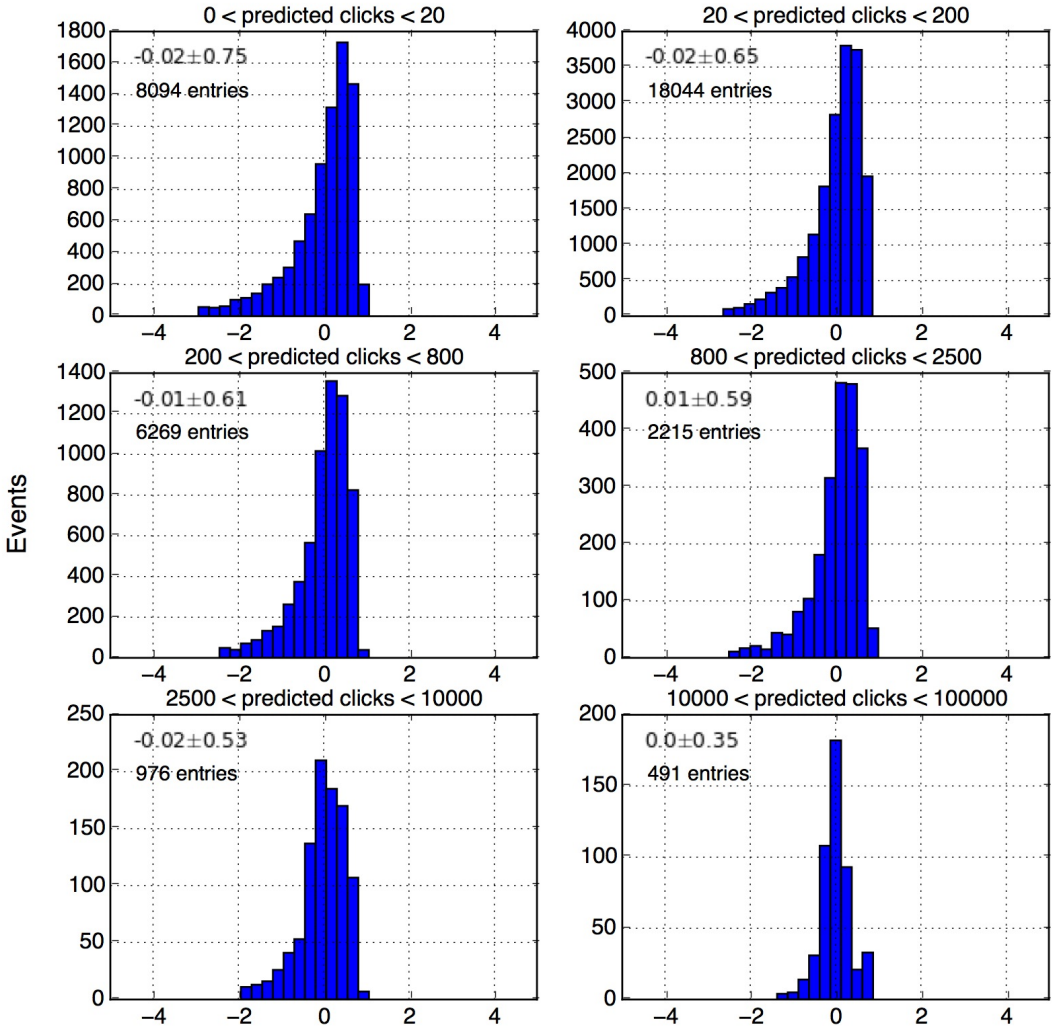

Figure 9: Similar to Fig. 6, but with removed unique visitors and views per visitor from the feature vector.

Table 4: Expected number of clicks for different values of Platform, Ad class, Cost type and Cost subtype for $CPI = \$0.005, B = \$1000$, flight_days=30, and vertical ='Business & Industrial'.

	Platform	AdClass	CostType	CostSubType	<Clicks>
0	Desktop	Display	CPC	CPC	335
1	Desktop	Display	CPC	RTB CPC	569
2	Desktop	Display	CPM	CPM	106
3	Desktop	Display	CPM	RTB CPM	139
4	Desktop	Video	CPC	CPC	1060
5	Desktop	Video	CPC	RTB CPC	3090
6	Desktop	Video	CPM	CPM	184
7	Desktop	Video	CPM	RTB CPM	518
8	Mobile	Display	CPC	CPC	318
9	Mobile	Display	CPC	RTB CPC	372
10	Mobile	Display	CPM	CPM	254
11	Mobile	Display	CPM	RTB CPM	243
12	Mobile	Video	CPC	CPC	2148
13	Mobile	Video	CPC	RTB CPC	3314
14	Mobile	Video	CPM	CPM	867
15	Mobile	Video	CPM	RTB CPM	1363

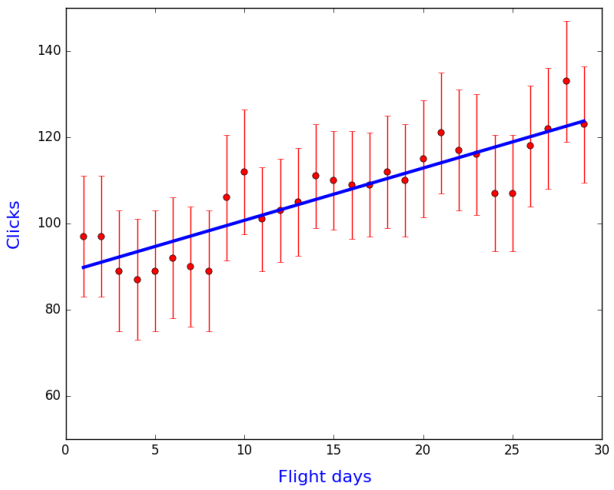

Figure 10: Number of expected clicks versus flight days for $CPI = \$0.005, B = \$1000$, Desktop, Display , cost type CPM.

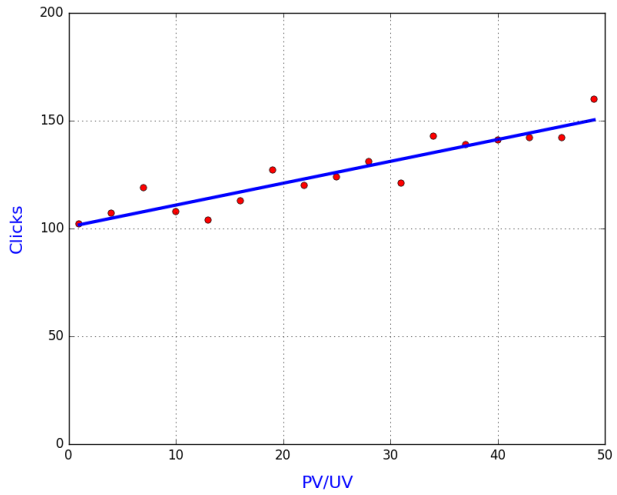

Figure 11: Number of expected clicks versus 'page views' /'unique visitor' for UV=0.5M, $CPI = \$0.005, B = \$1000$, Desktop, Display , cost type CPM.

confidence level we predict expected budget and number of impressions as \$67160 and 13432000 (\$112935 and 22587000), respectively. As expected, the required budget grows for the increased confidence level.

5 Appendix I.

In Section 3 while discussing the click model, we showed plot with a dependence of CTR vs impressions, Fig. 1. In this section we show some results from analysis of Sizmek pre-aggregated weekly data which are tightly related to the click probabilities discussed above. In Fig. 12 we show behaviour of CTR, defined as a ratio of 'user clicks' / 'served impressions', and a new quantity CTR′, defined as a ratio of 'user clicks' / 'unique impressions', versus frequency of ad serving. Here 'unique impressions' is the Sizmek's estimate of the number of unique visitors attending campaign. The frequency is defined as a ratio of 'served impressions'/'unique impressions' (or SI/UI). The analysis was for all direct (non-network) sites participating in Centro campaigns during one year July, 2014 – July, 2015.

From this plot we see that as the frequency increases, the CTR goes down. This behaviour can be understood from Fig. 13, where the left plot shows a dependence of SI and UI vs days. We see that SI grow much faster than UI, what leads to a growing frequency of ad serving (right plot of the same figure). Since the number of unique visitors does not grow that fast as SI, the clicking probability decreases with increasing frequency.

In the variable CTR′ we normalize 'user clicks' to the 'unique visitors', and in such a way it may serve a measure of the user clicking activity. It increases with the frequency, but saturates at frequency ≈ 5. It means that starting from this value, an additional increase of the frequency does not improve a users performance, and only linearly increases cost-per-click. If maximization of clicks is one of a campaign goals, one needs to keep in mind existence of the critical ad serving frequency.

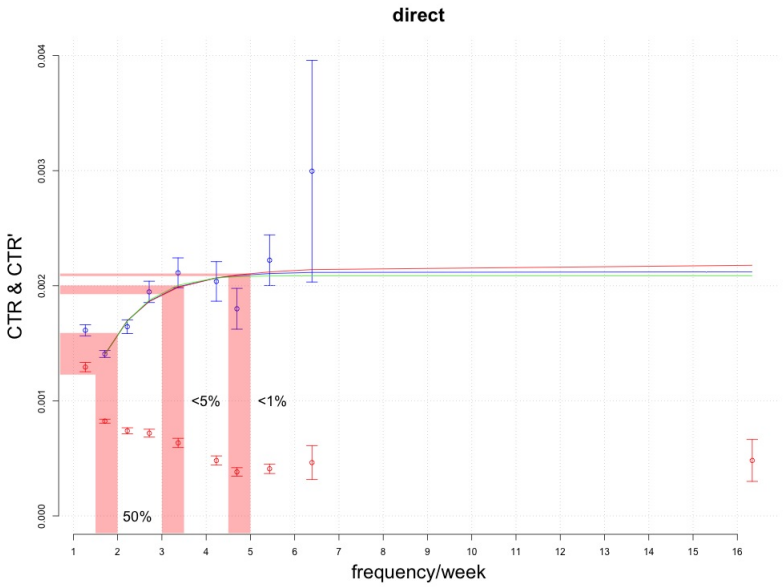

Figure 12: CTR and CTR′ versus ad serving frequency for direct sites (see the text above).

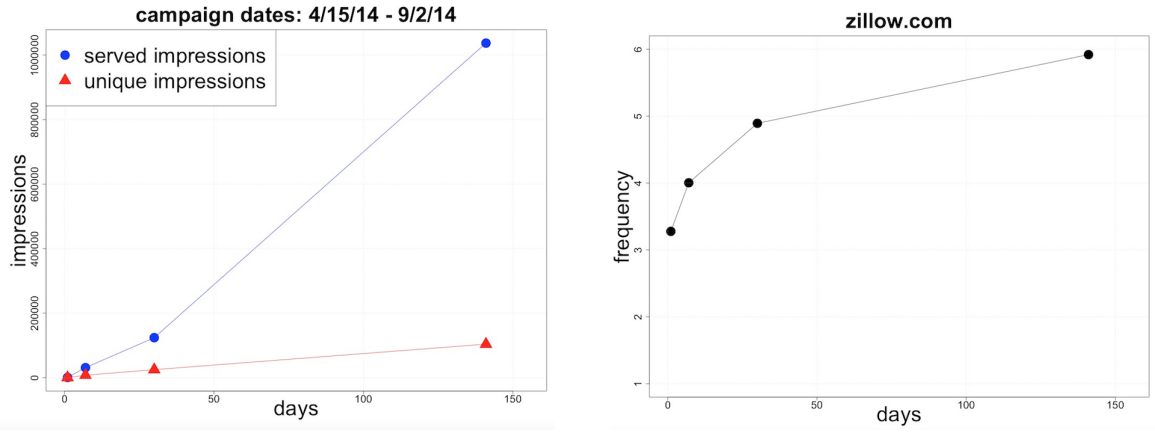

Figure 13: Served impressions, unique impressions (left) and ad serving frequency (right) vs campaign flight days for direct sites (see the text above).

6 Appendix II.

In this section, we estimate what best accuracy we may achieve predicting with our click model. For this purpose, we fix all our input variables at specific values and look at the true values in data.

- $10^5 <$ delivered_impressions $< 2 \cdot 10^5$;
- 4 <CPM< 6;
- 27 < flight days < 32;
- platform=Desktop, ad_class = Display, cost_type = CPM, cost_subtype = CPM, google_vertical = Autos & Vehicles

For obvious reasons, we cannot take a fixed value for the continuous variables, so we take a narrow intervals around their average values instead. Using these selections, we obtain distributions for CTR and delivered clicks as shown in Fig. 14. One can see that at $< deliveredclicks > \approx 100$ both distributions have a relative width of about 0.54–0.56. We refer to this value as "observed width" for the clicks. Along with the "intrinsic/true" click distribution (click resolution function), this width is also caused by the chosen range for impressions and partially (to the less extent) by CPM and flight days ranges. To find the true width of the click distribution we must subtract in quadrature the relative width related with the impressions:

$$\Big[\frac{\Delta clicks}{< clicks >}\Big]_{true} = \left(\Big[\frac{\Delta clicks}{< clicks >}\Big]^2_{obs} - \Big[\frac{\Delta imps}{< imps >}\Big]^2_{obs}\right)^{1/2} = (0.56^2 - 0.19^2)^{1/2} = 0.51 \qquad (7)$$

So, we get that the true relative width for the click resolution function is about 51%.

We don't know what variance in the number of clicks is related with the chosen (while narrow but nevertheless) CPM and flight days range. We know the may affect clicks to the less extent than the impressions. So, we assume their combined affect is as big as the impressions we would get a most pessimistic bottom limit for the true clicks:

$$\Big[\frac{\Delta clicks}{< clicks >}\Big]_{true} = (0.56^2 - 0.19^2 - 0.19^2)^{1/2} = 0.47, \qquad (8)$$

i.e. 47%. This resolution is caused by the nature of the underlying Poissonian process (how many time a given user will click our ad) which is also convoluted with other uncertainties related with fluctuating distribution of users in a chosen demographic group, finite (30-35%) efficiency to reach the demographic group of our interest, and other effects.

From Fig. 5 we see that the accuracy at 100 clicks obtained using the clicks prediction model is about 60%, which is not too much worse of the estimated bottom value.

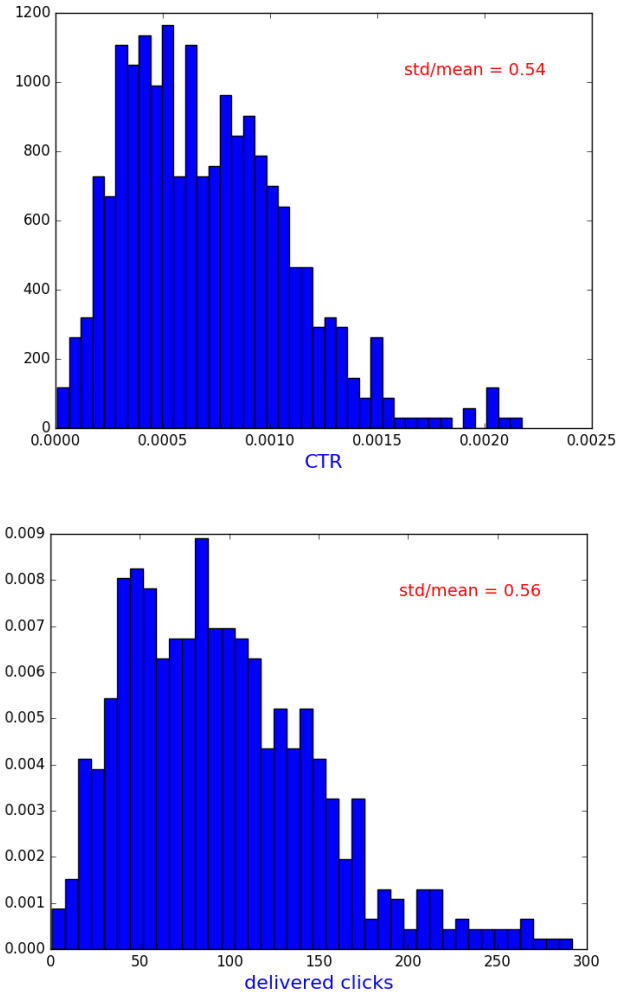

Figure 14: CTR and delivered clicks distributions for the fixed selections mentioned in the text.

References

[1] https://en.wikipedia.org/wiki/K-means_clustering.

[2] https://en.wikipedia.org/wiki/Random_forest.