Capstone Project: Heatmap Anomaly Detection

Week 13 Progress Report

This week:

1. Metrics dataset with LRT result: Edge Case
2. Results on third dataset
3. Uniform good vs uniform bad: results, Summary of datasets used

Metrics Dataset with LRT result: Edge Case

- Add the LRT result as one feature to metrics dataset
- Add landing rate as a feature
- Perform Random Forest to Prediction on the new data

	Feature	Importance
0	preds	0.675318
5	landing_rate	0.175833
4	avg_last_second_framerate	0.051037
1	displays	0.040884
2	non_bounced_clicks	0.030177
3	closing_events	0.026752

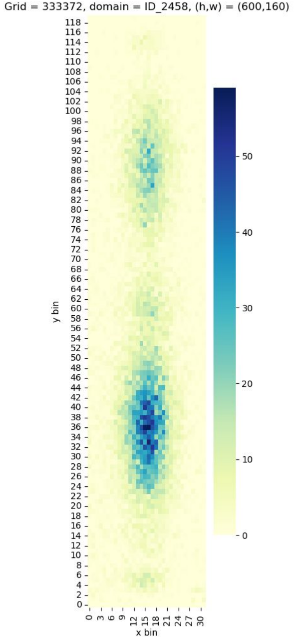
Old Test Data:

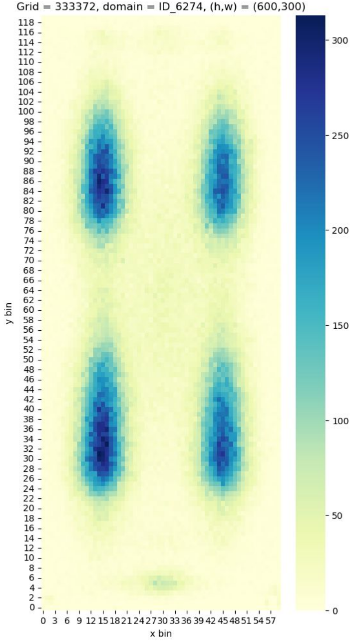
```
              precision    recall  f1-score   support

           0       0.98      0.99      0.99       303
           1       0.95      0.89      0.92        44

    accuracy                           0.98       347
   macro avg       0.97      0.94      0.95       347
weighted avg       0.98      0.98      0.98       347

[[301   2]
 [  5  39]]
0.9176470588235294
```

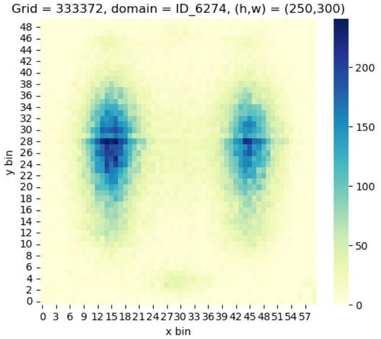
New Data:

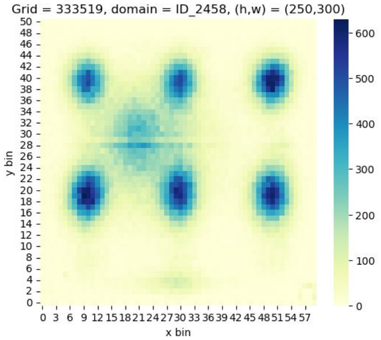
```
              precision    recall  f1-score   support

           0       0.49      0.92      0.64      2966
           1       0.40      0.05      0.09      2966

    accuracy                           0.49      5932
   macro avg       0.45      0.49      0.37      5932
weighted avg       0.45      0.49      0.37      5932

[[2732  234]
 [2809  157]]
0.09353589514447422
```

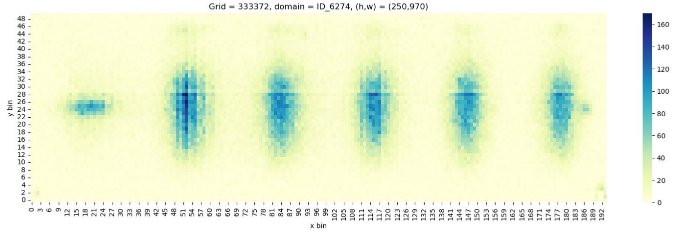
New “timestamped” Data:

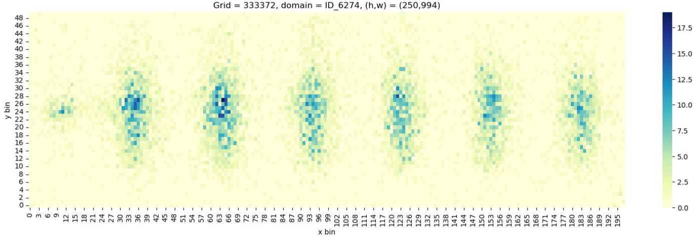


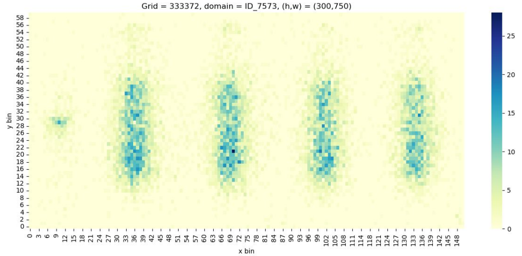




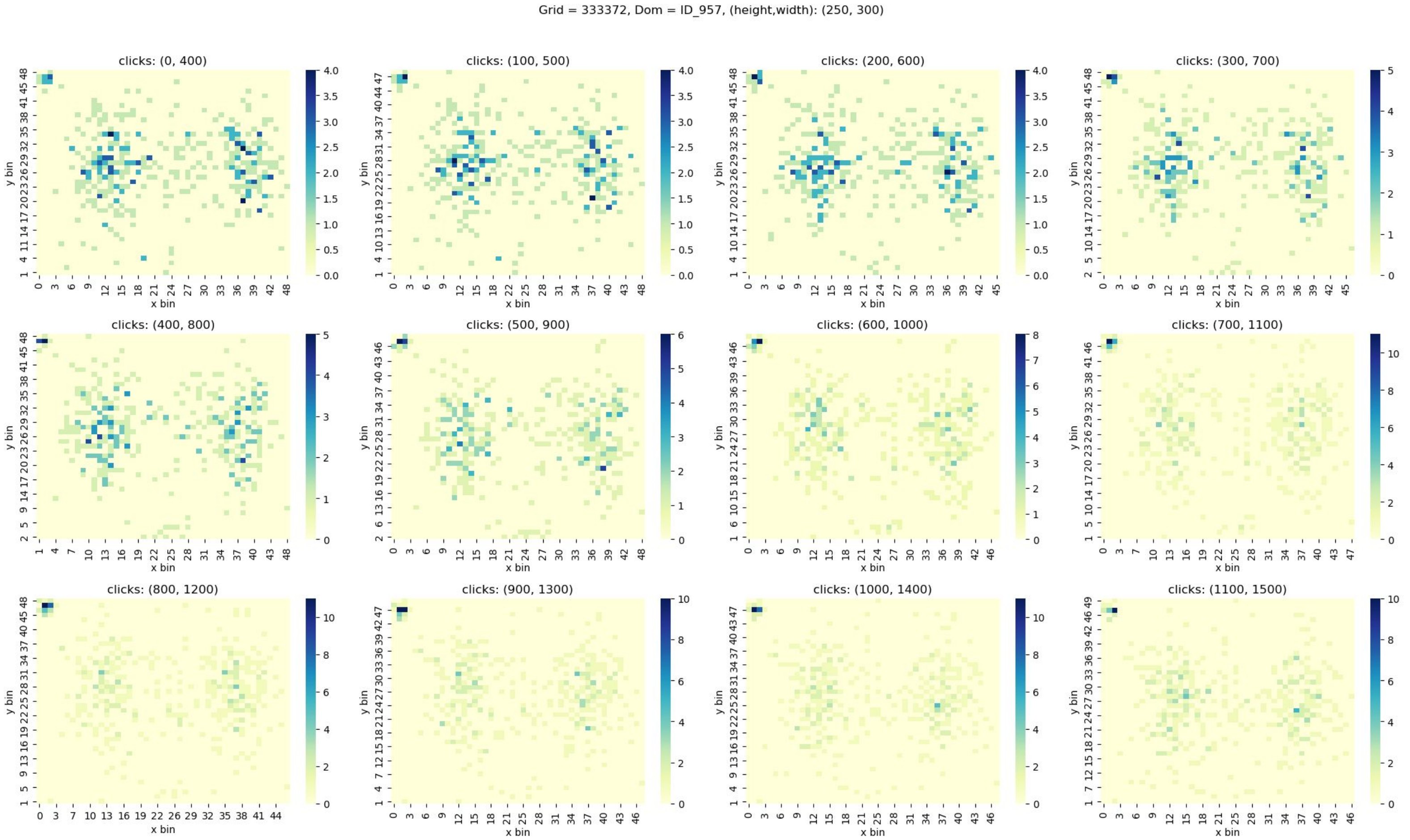




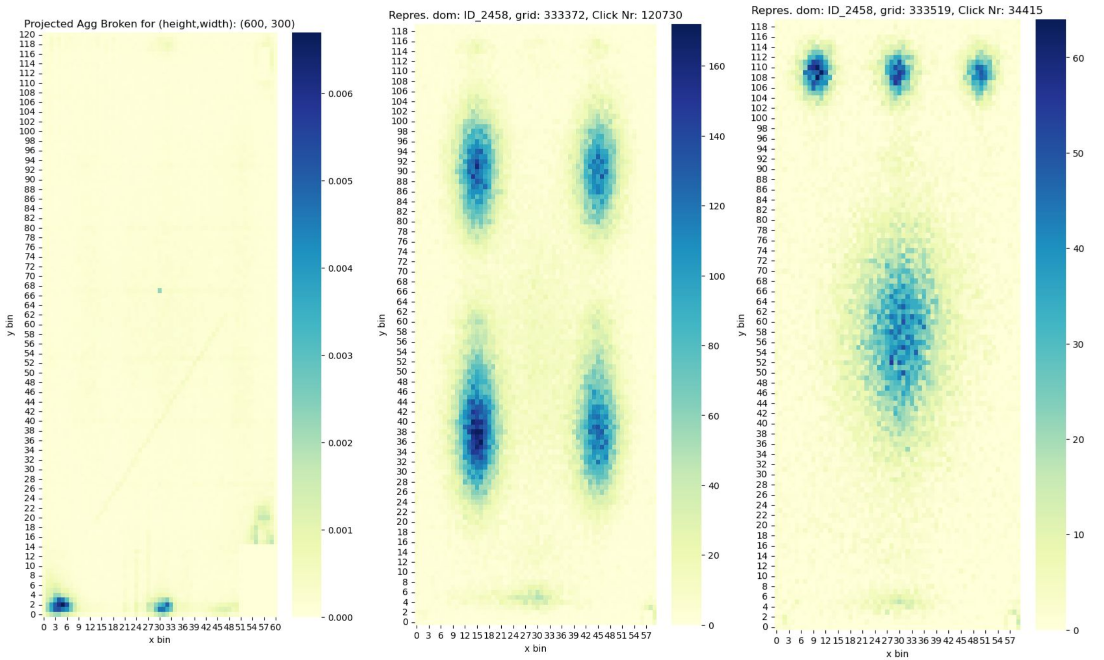




Example of “degradation”:

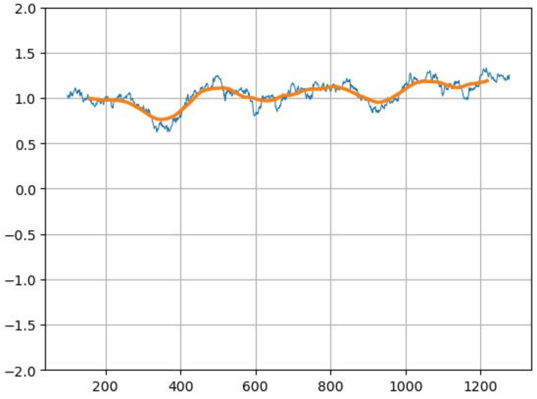


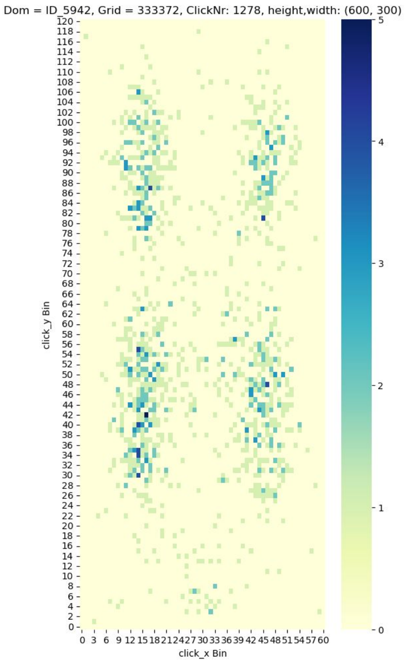
Time-series of LR (1-shot):



- Project aggregated broken banners to new size (eg. 600,300)
- Generate representative samples as before
- Compute LR across different window
- Plot (smoothed) LR over time
  - Careful to adjust epsilon to not be too large otherwise we get many outliers
- Study average derivative over window/smoothed curve and inspect for “degradation” of pattern.

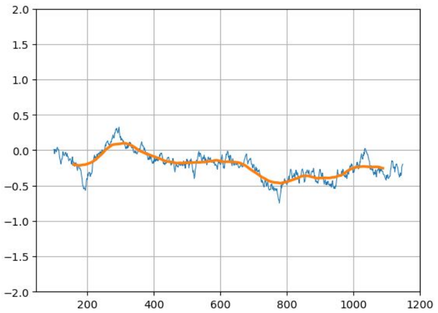
Time-series of LR (1-shot):

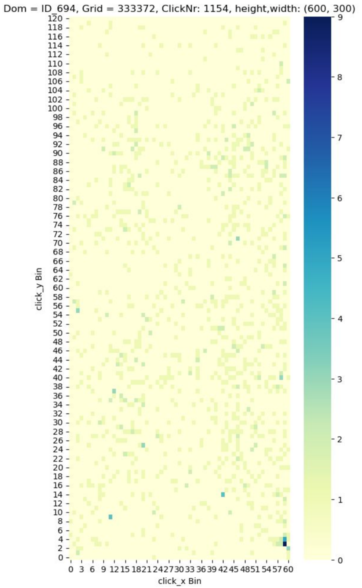




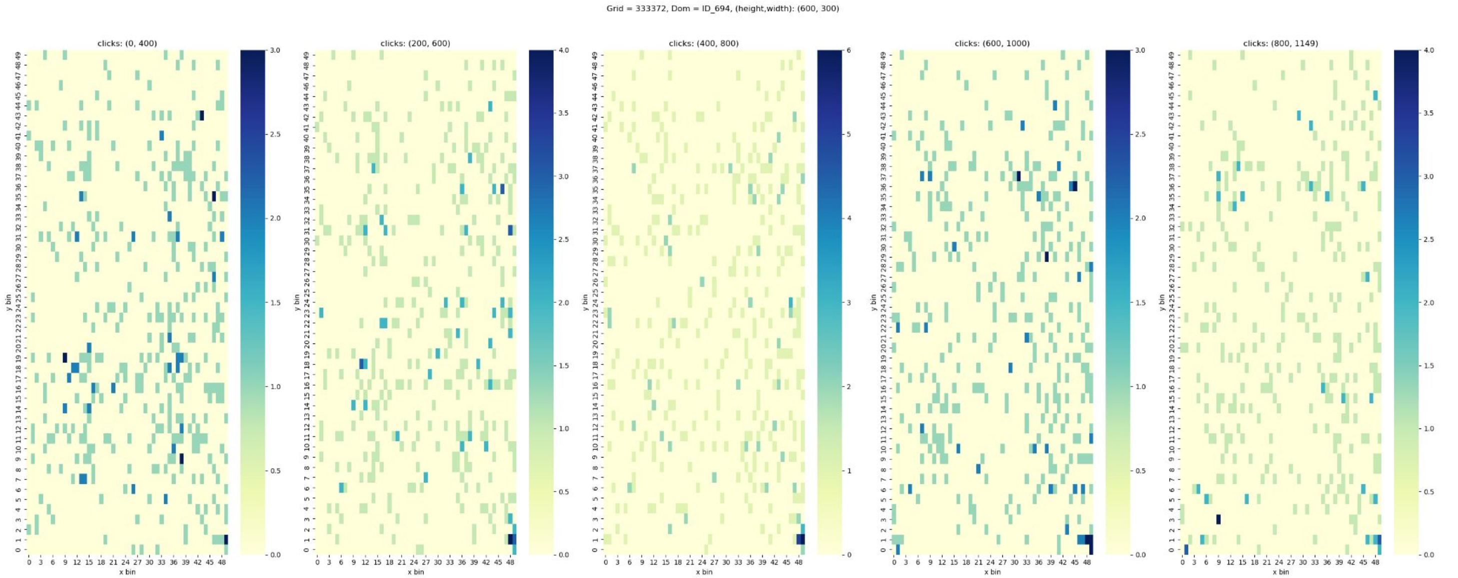
- Plot (smoothed) LR over time
  - Rolling window of 100 clicks in order.
  - Careful to adjust epsilon to not be too large otherwise we get many outliers
- Study average derivative over window/smoothed curve and inspect for “degradation” of pattern.

1st Example



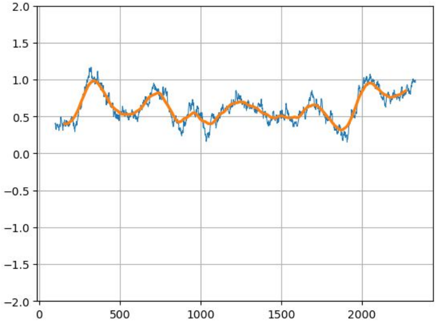


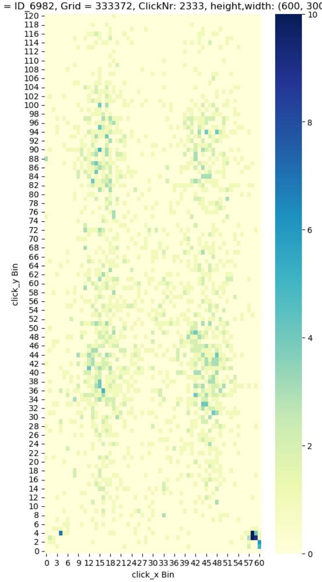
Dom = ID_694, Grid = 333372, ClickNr: 1154, height,width: (600, 300)
click_y Bin
click_x Bin

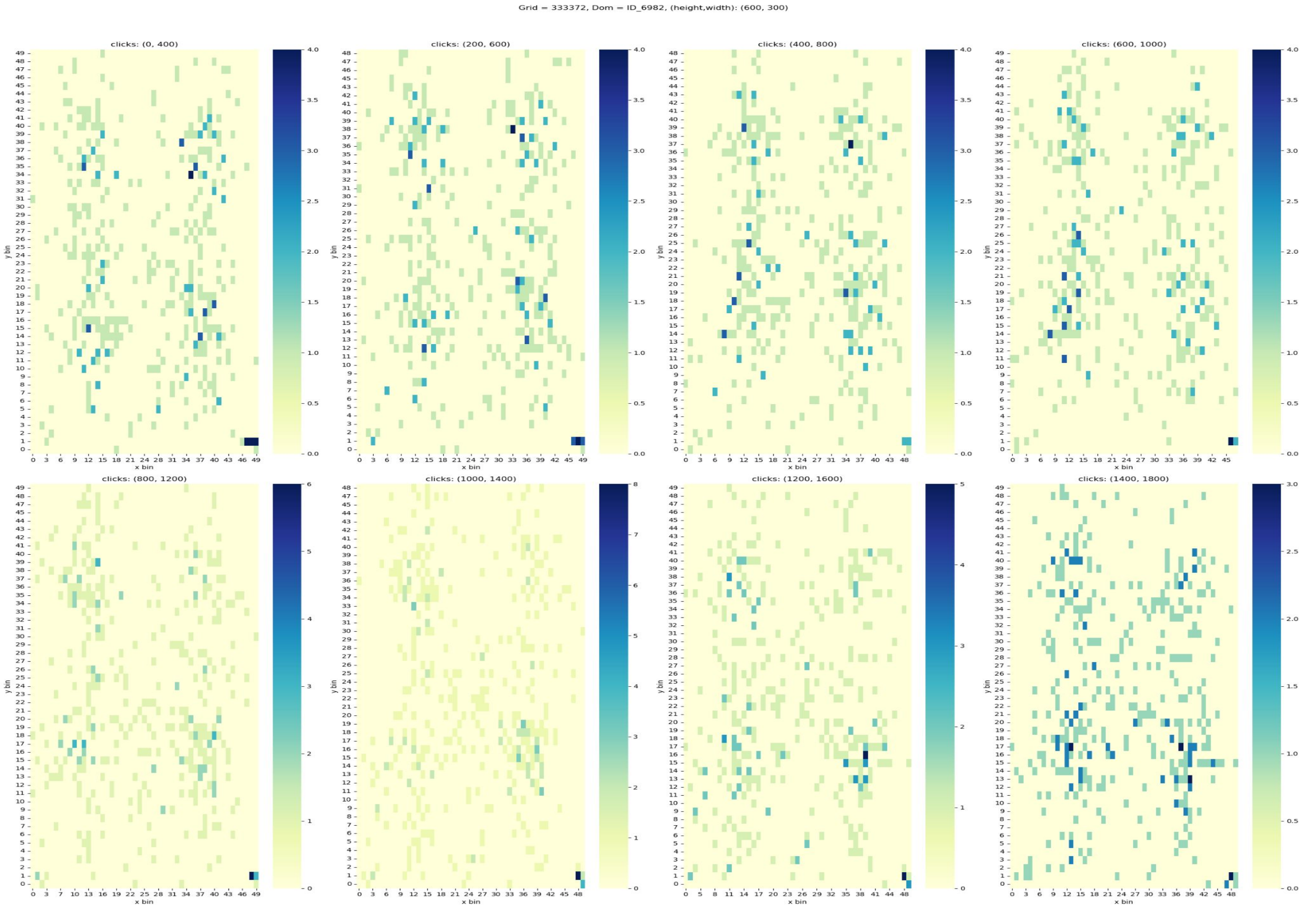


Grid = 333372, Dom = ID_694, (height,width): (600, 300)
clicks: (0, 400)
clicks: (200, 600)
clicks: (400, 800)
clicks: (600, 1000)
clicks: (800, 1149)
y bin
x bin

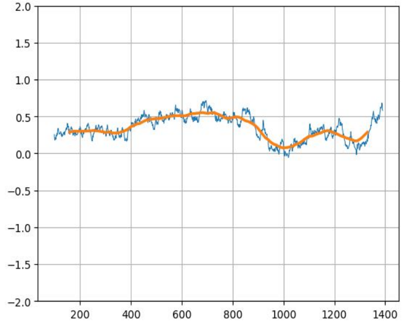
2nd Example

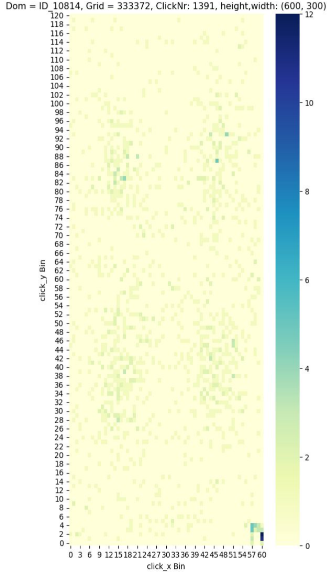


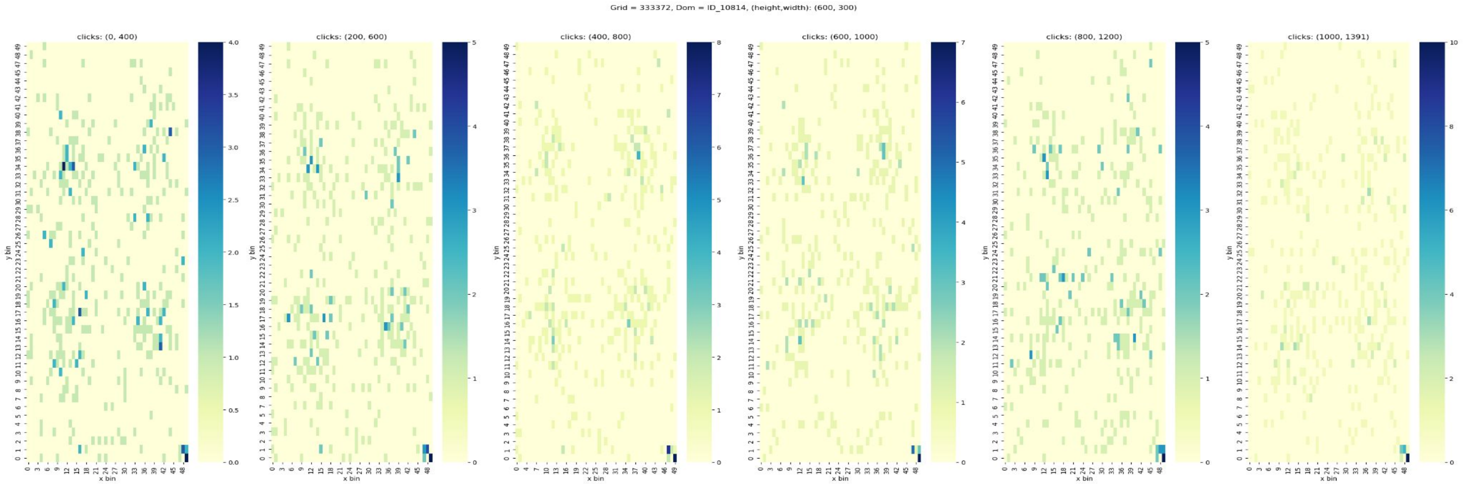

Dom = ID_6982, Grid = 333372, ClickNr: 2333, height,width: (600, 300)
click_y Bin
click_x Bin


Grid = 333372, Dom = ID_6982, (height,width): (600, 300)
clicks: (0, 400)
clicks: (200, 600)
clicks: (400, 800)
clicks: (600, 1000)
clicks: (800, 1200)
clicks: (1000, 1400)
clicks: (1200, 1600)
clicks: (1400, 1800)
y bin
x bin

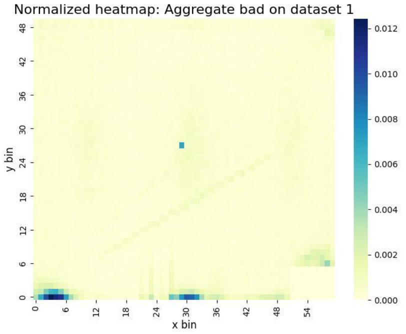
3rd Example

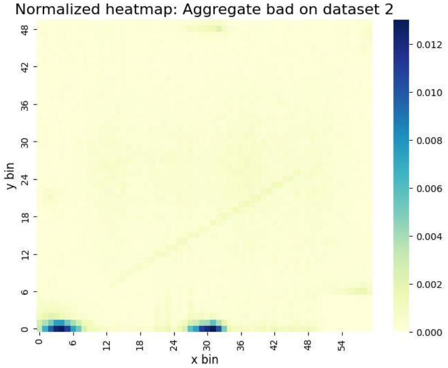


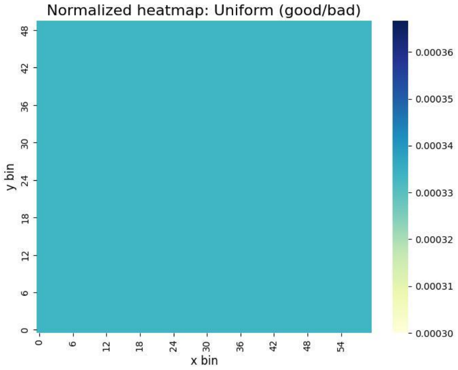

Dom = ID_10814, Grid = 333372, ClickNr: 1391, height,width: (600, 300)
click_y Bin
click_x Bin

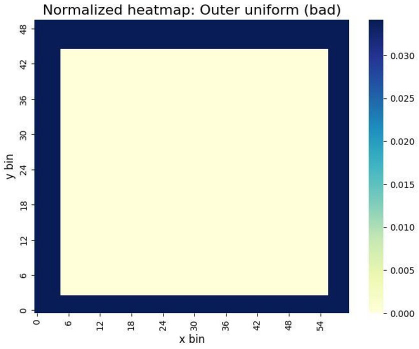

Grid = 333372, Dom = ID_10814, (height,width): (600, 300)
clicks: (0, 400)
clicks: (200, 600)
clicks: (400, 800)
clicks: (600, 1000)
clicks: (800, 1200)
clicks: (1000, 1391)
y bin
x bin

Visualization of bad representative banners

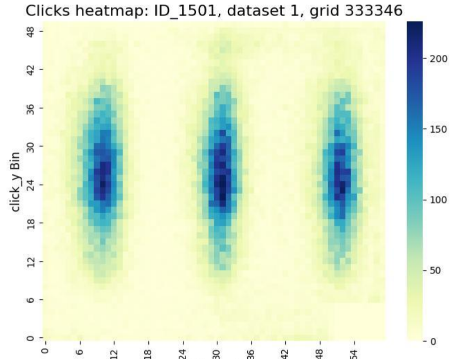


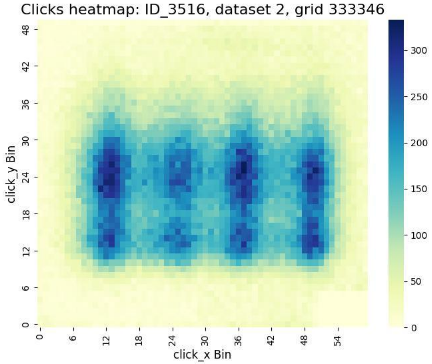


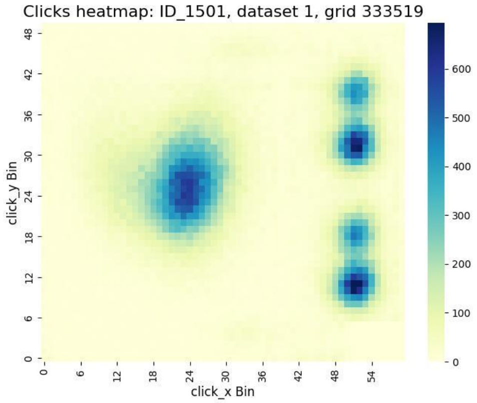


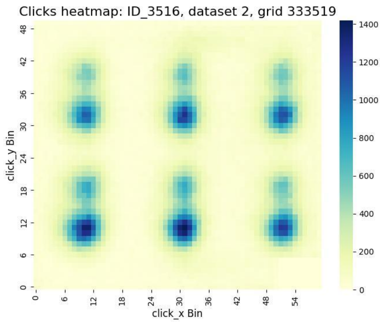


Visualization of good representative banners

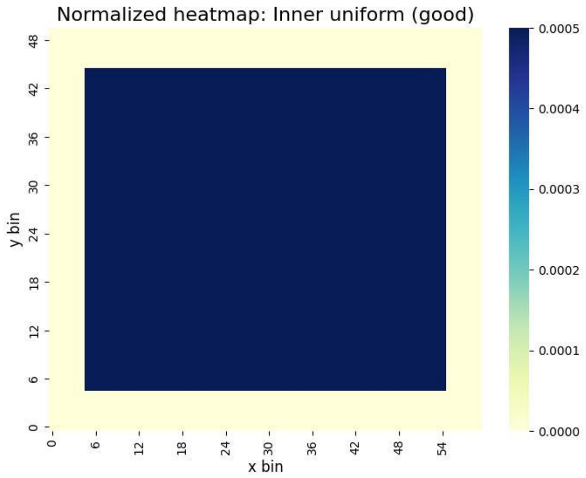


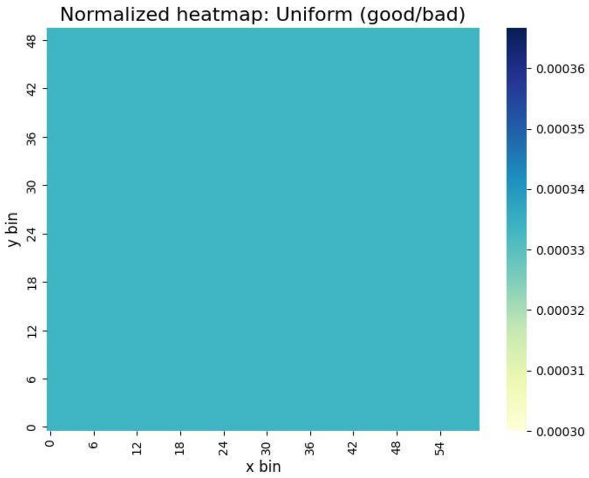


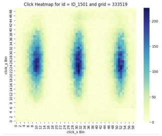
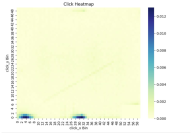
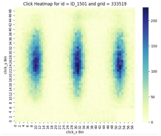
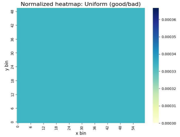
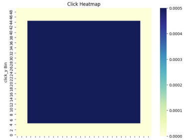
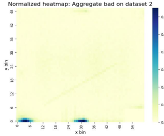
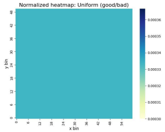
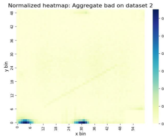
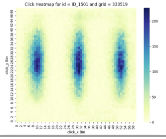
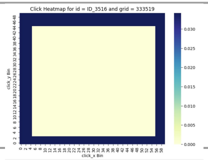


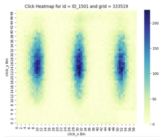
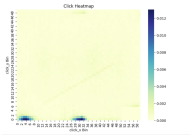
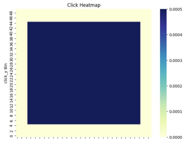
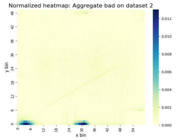


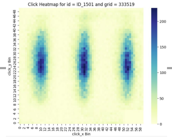
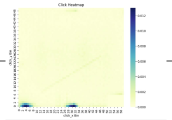
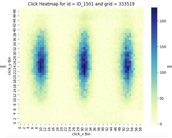
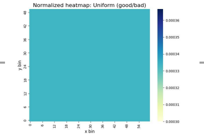
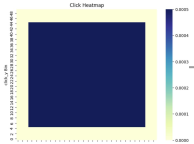
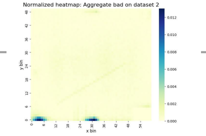
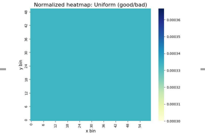
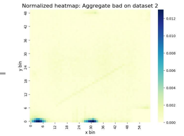
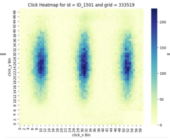
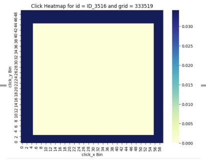
Visualization of good representative banners





"Good" banner	"Bad" banner	Learning	f1: 1st data (macro/macro)	f1: 2nd data (macro/macro)
		**One shot**	**0.95** 0.92	0.85 **0.92**
		One shot	**0.94** 0.73	0.79 **0.9**
		**Zero shot**	**0.91** 0.91	0.88 **0.89**
		Zero shot	0.84	0.89
		One shot	0.51	0.53

"Good" banner	"Bad" banner	Learning	Macro f1 (same dataset)	Macro f1 (different dataset)
		One shot	0.95	0.92
		Zero shot	0.91	0.91

"Good" banner	"Bad" banner	Learning	f1: 1st data (macro/binary)	f1: 2nd data (macro/binary)
		**One shot**	**0.95** 0.87	0.73 **0.92**
		One shot	**0.94** 0.58	0.61 **0.9**
		**Zero shot**	**0.91** 0.75	0.79 **0.89**
		Zero shot	0.84	0.89
		One shot	0.51	0.53

	"Good" banner	"Bad" banner	Learning	Results: old data (f1,CM, latency per banner)	Results: new data (f1/CM)
1	**Landed clicks ratio > 0.8 + most no. clicks**	**Agg. bad data from other dataset**	**One shot**	**0.95** **[[1502 19]** **[15 197]]**	**0.92** **[[2782 22]** **[44 182]]**
2	representative	uni	One shot	0.94 [[1508 13] [29 183]]	0.9 [[2769 35] [48 178]]
3	**Step uni**	**agg**	**Zero shot**	**0.91** **[[1503 18]** **[44 168]]**	**0.89** **[[2774 30]** **[58 168]]**
5	**uni**	**agg**	**One shot**	0.84 [[1517 4] [93 119]]	0.89 [[2768 24] [64 174]]
6	representative	step uni	One shot	0.51 [[1521 0] [202 10]]	0.53 [[2803 1] [215 11]]

More information on the experiment

Average latency: 0.05s / banner
CPU specs: Intel i7 processor, 4.7GHz, 16GB RAM

Number of banners processed:
Old: 872 with 75 broken (333519),861 with 137 broken (333346)
New: 2061 with 73 broken (333519), 969 with 153 broken (33346)

Types of good/bad banners

Good:

1. One shot: "Best" representative from the same dataset
2. ~~Zero shot: Normal uniform distribution~~
3. Zero shot: Step uniform distribution

Bad:

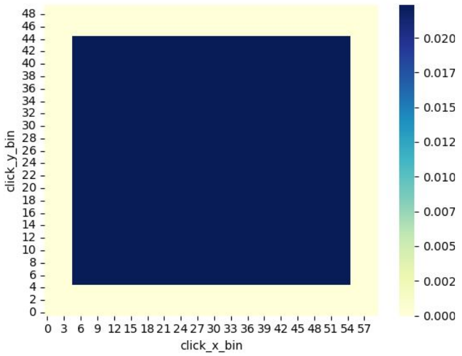
1. Zero shot: Aggregating all bad banners from other dataset
2. ~~Zero shot: Normal uniform distribution~~
3. Zero shot: Inverse step uniform distribution (toy banner)

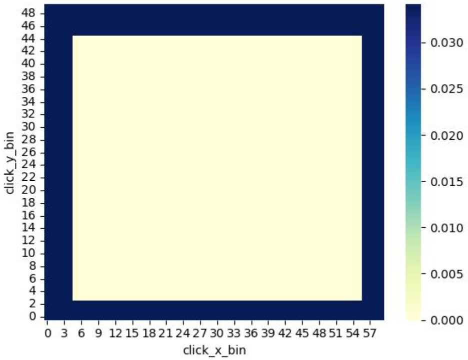
Note: One shot is not done here as the bad banners can look quite different

Question to answer: which distribution, good or broken, can be better modeled by a uniform distribution?

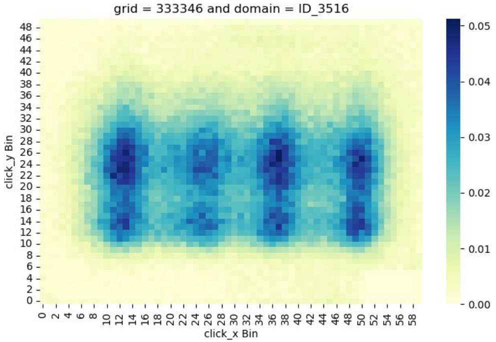
Uniform bad clusters + Good representative clusters

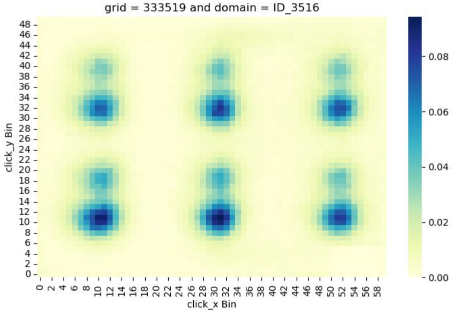
Bad Clusters





Good Clusters





Setting Thresholds

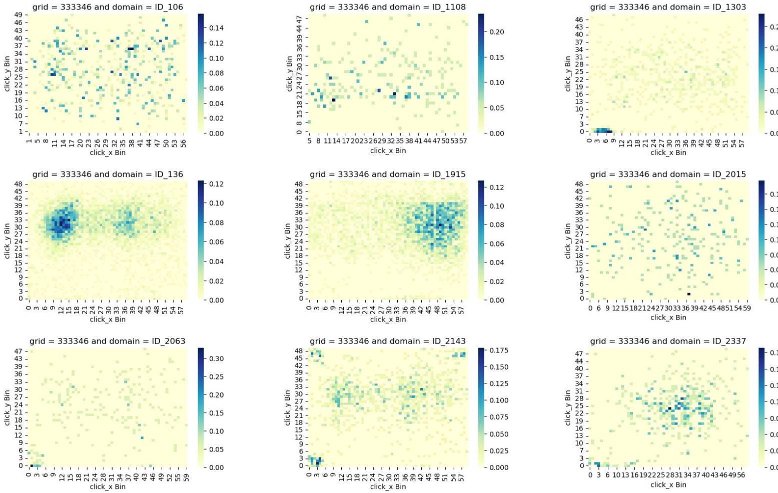
Using only cosine similarity with each cluster

max_f1 = 0.78
best_thresholds = [0.13, 0.85, 0.15, 0.11]

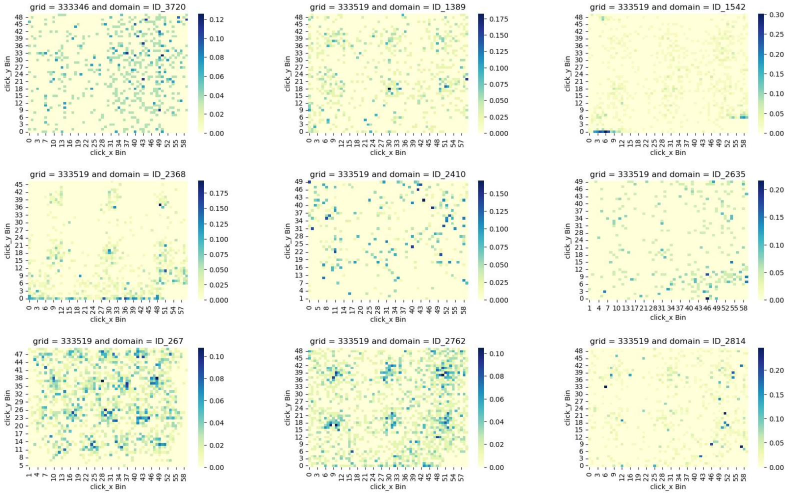
Using (sum of similarity to good clusters)/(sum of similarity to bad clusters) & log(product of similarities to good cluster)-log(product of similarities to bad cluster)

max_f1 = 0.83
best_thresholds = [1.23, 0.11]

FN


grid = 333346 and domain = ID_106
grid = 333346 and domain = ID_1108
grid = 333346 and domain = ID_1303
grid = 333346 and domain = ID_136
grid = 333346 and domain = ID_1915
grid = 333346 and domain = ID_2015
grid = 333346 and domain = ID_2063
grid = 333346 and domain = ID_2143
grid = 333346 and domain = ID_2337
click_x Bin
click_y Bin

FP


grid = 333346 and domain = ID_3720
grid = 333519 and domain = ID_1389
grid = 333519 and domain = ID_1542
grid = 333519 and domain = ID_2368
grid = 333519 and domain = ID_2410
grid = 333519 and domain = ID_2635
grid = 333519 and domain = ID_267
grid = 333519 and domain = ID_2762
grid = 333519 and domain = ID_2814
click_x Bin
click_y Bin