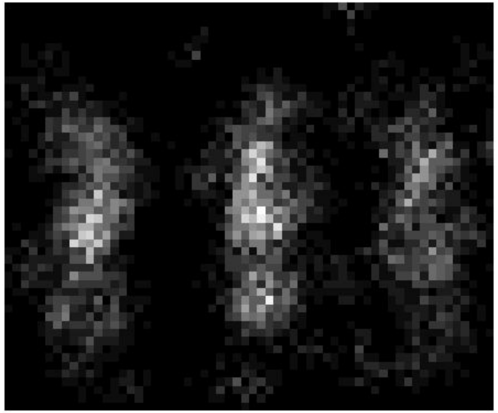
Capstone Project:
Heatmap Anomaly Detection

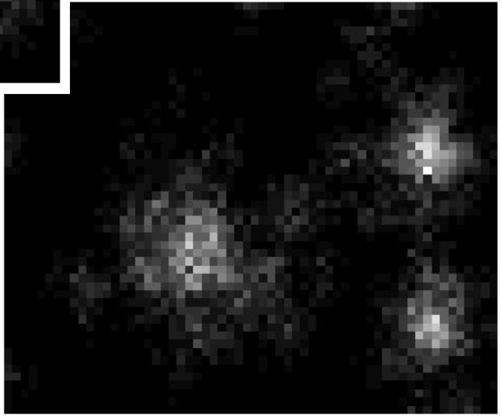
Week 12 Progress Report

This week:

1. More results from new data
   a. ResNet/ViT
   b. Clustering
2. “Clustering” broken banners
   a. Structure of broken banners → generate synthetic data
   b. Measure distance from characteristic broken banners
3. More results from Statistical Approach

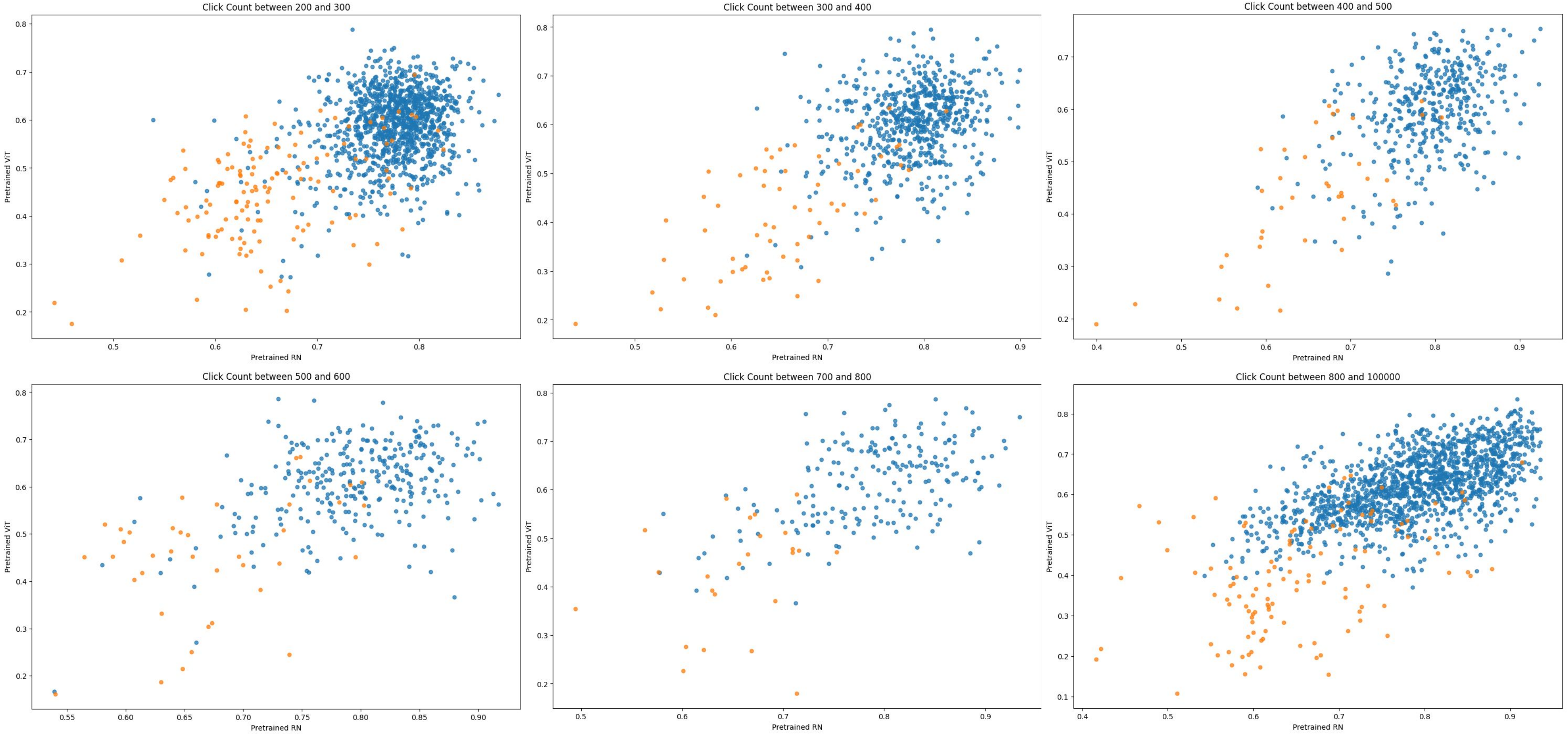
Pretrained ViT/ResNet (recap):



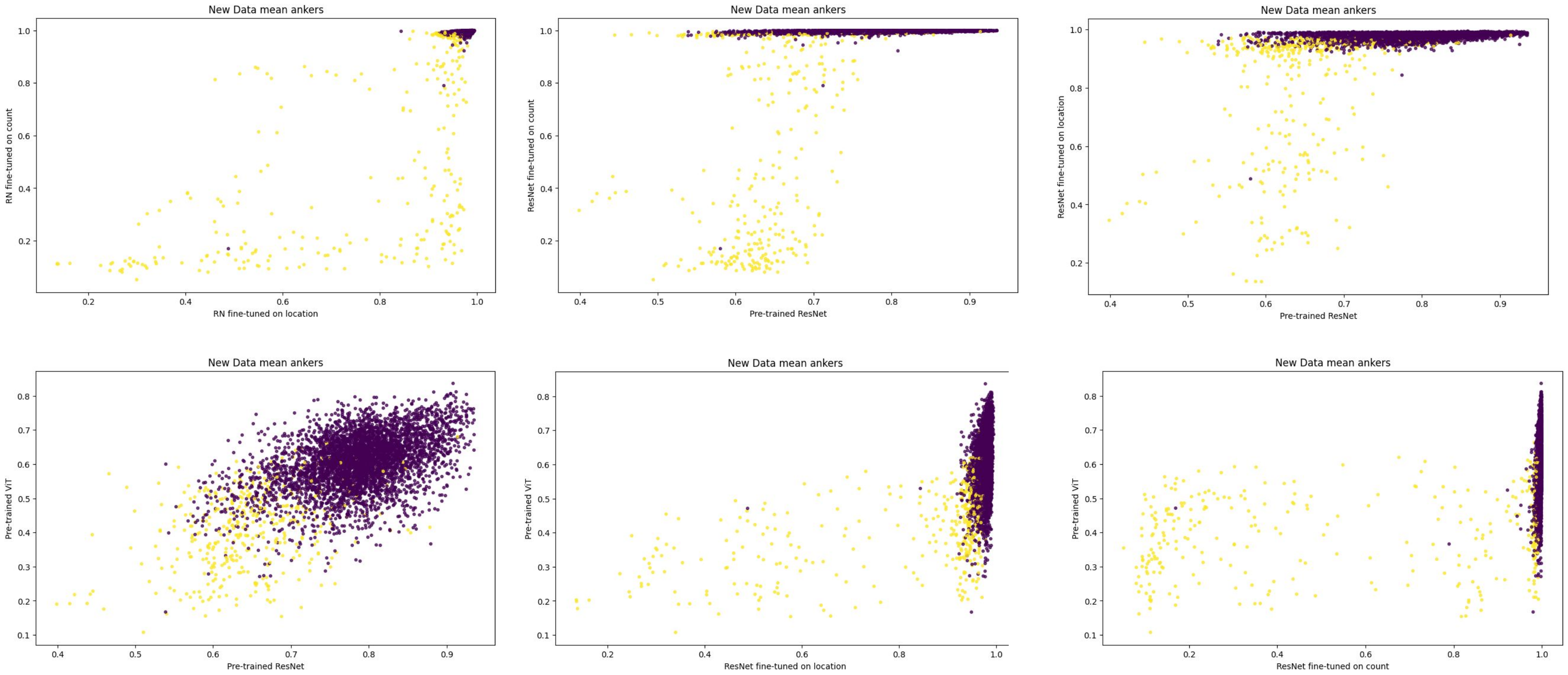


- Feed (transformed and binned) heatmaps into pre-trained ViT/ResNeT.
    - google/ViT: Transformer-based architecture, 14M images (224x224), 21k classes
    - Microsoft/ResNet-1k: trained on ImageNet-1k (224x224), 1k classes.
- Extract features (before classification head)
    - ViT → 151296 dim'l feature vector
    - ResNet → 2048 dim'l feature vector
- Play with upsampling (bootstrapping + noise)
- PCA and other dim'l reduction techniques
    - Apply clustering methods

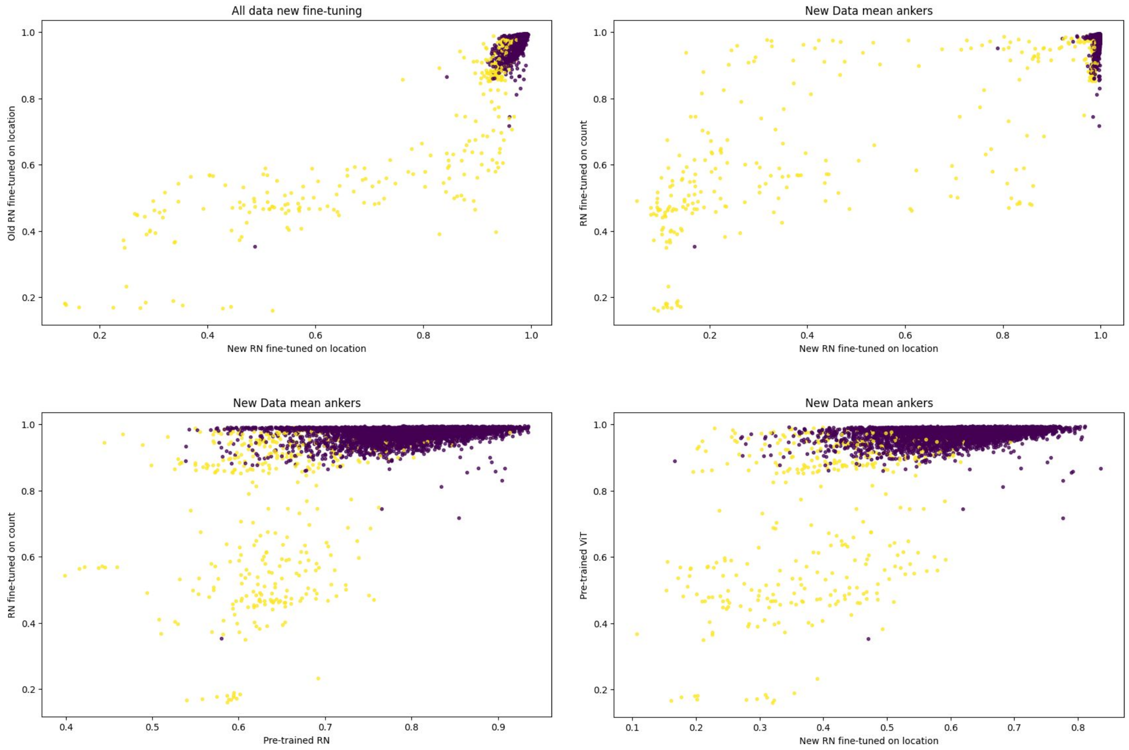
Pretrained ViT/ResNet (facet by click count):



All data cosine-simularities



All data: Better synthetic data



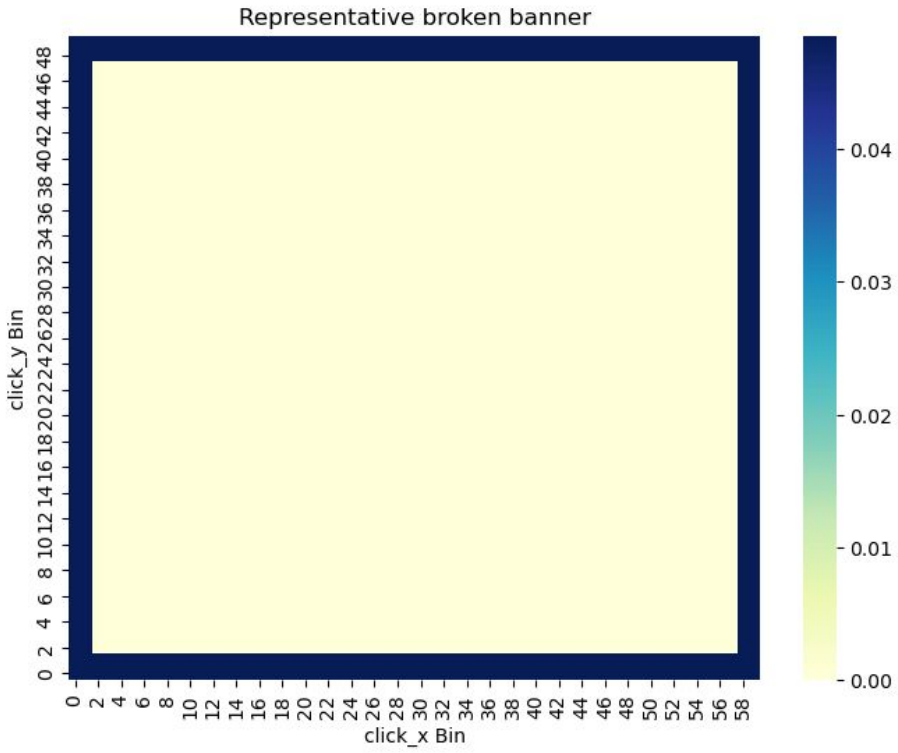
- New method for predicting locations of clusters.
    - Cleaner synthetic data
    - No overlaps
    - More noise

Summary DL models:

- Created 6 features:
  - ViT pretrained
  - RN pretrained
  - RN fine-tuned:
    - On cluster center location
    - On cluster center location II (better data/no overlaps/more noise)
    - On cluster number
    - Predicting noise or not-noise from synthetic data (symmetric cluster centers, etc) → abysmal

- Thresholding and majority voting → ~83% F1-score for 1-shot and worse for 0-shot (cosine similarity to broken mean banner)
- Comparatively slow inference speed

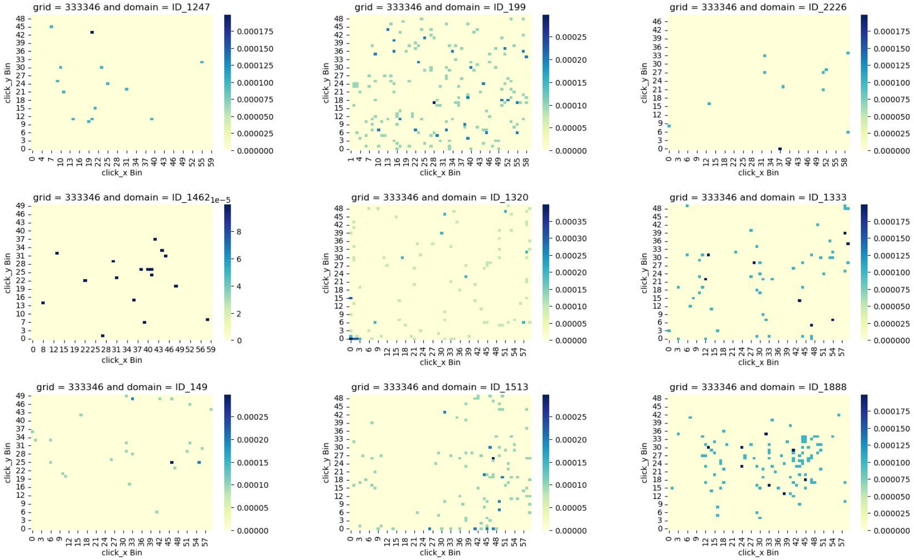
Toy example:



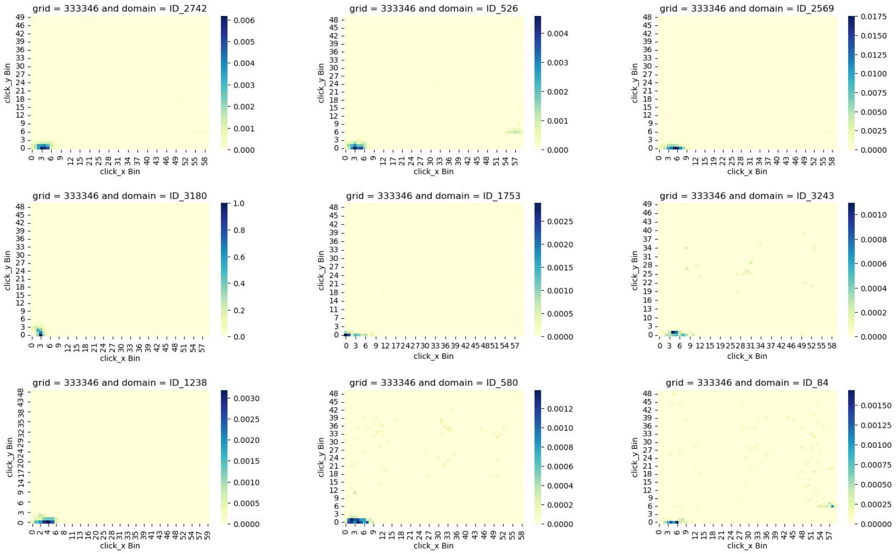
- Compute Euclidean distance (=cosine-similarity) between heatmaps and left banner:
  - Run over threshold distance:
  - New data:
    - **Threshold: 0.11**
    - ((TN,FN),(FP,TP)) = (2768,73),(36,153)
    - F1-score: 0.73735
  - Old data:
    - **Threshold: 0.11**
    - ((TN,FN),(FP,TP)) = (1493,70),(28,142)
    - F1-score: 0.74346

Clustering by broken heatmaps

211 cb in old data
227 cb in new data



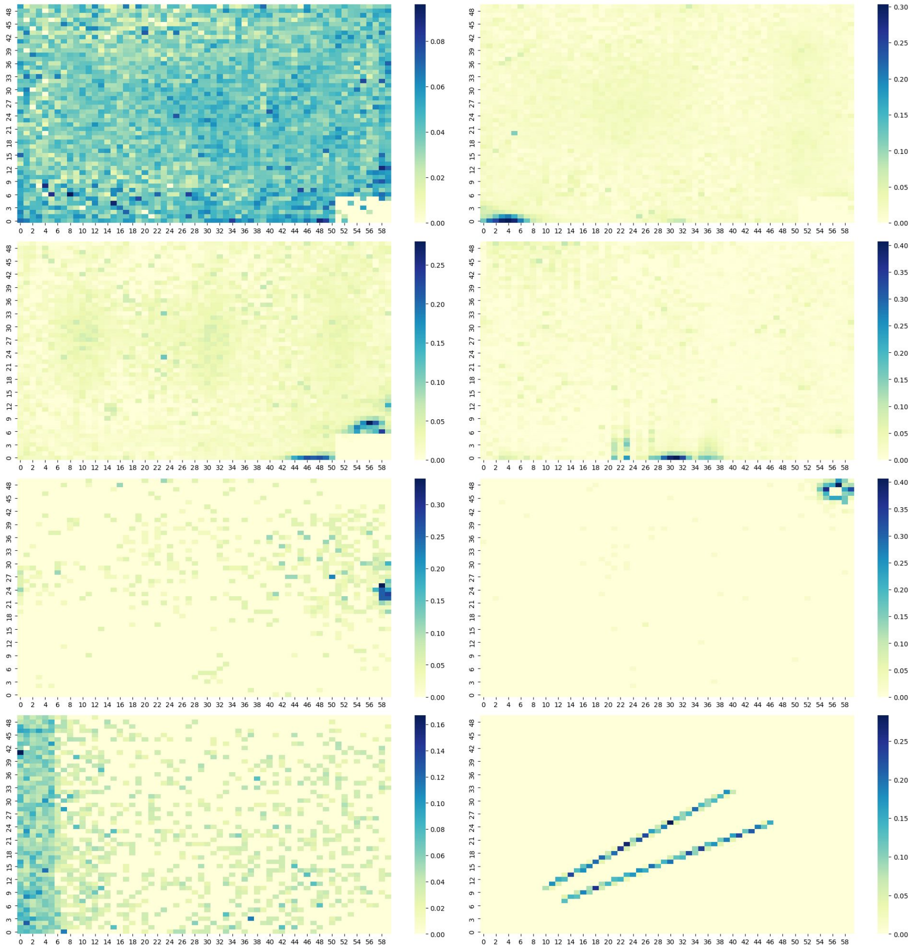
Cluster 1

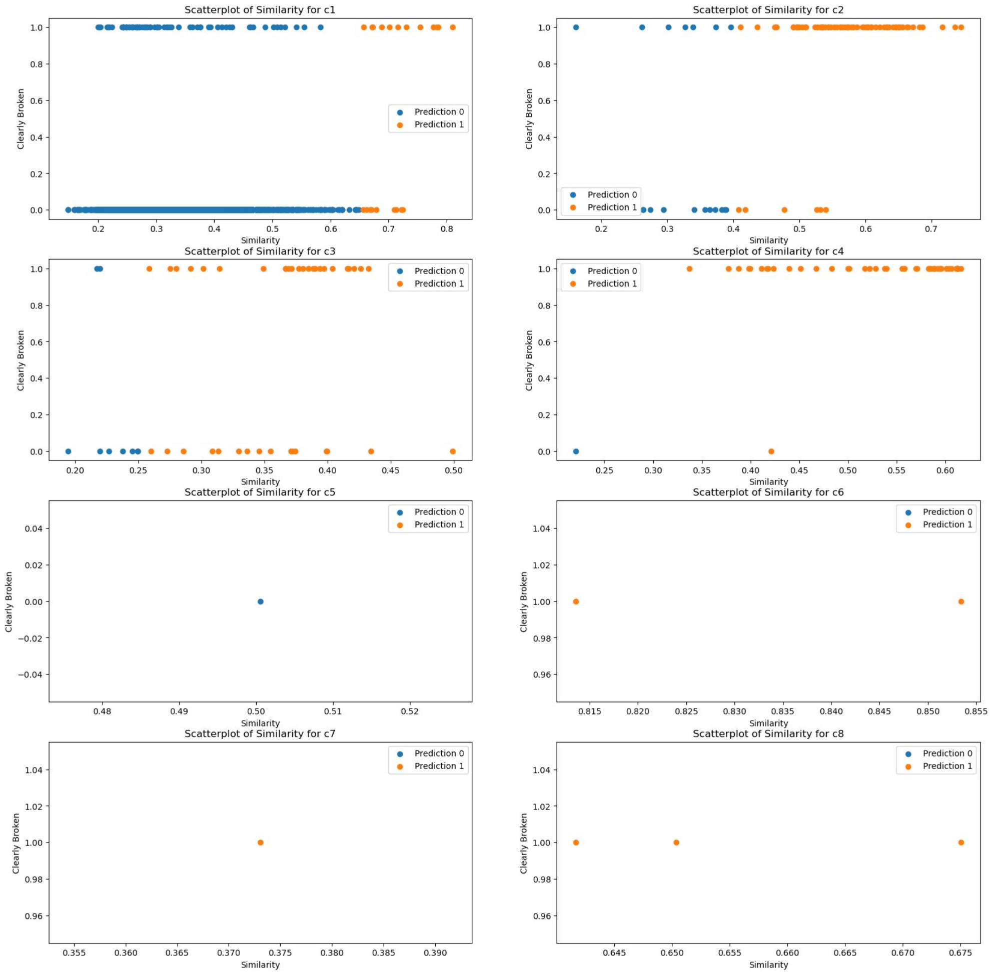


Cluster 2

Cluster_map

c1: all over the place
c2: left bottom corner
c3: right bottom corner
c4: bottom middle
c5: right middle
c6: top right corner
c7: all over left side
c8: diagonal



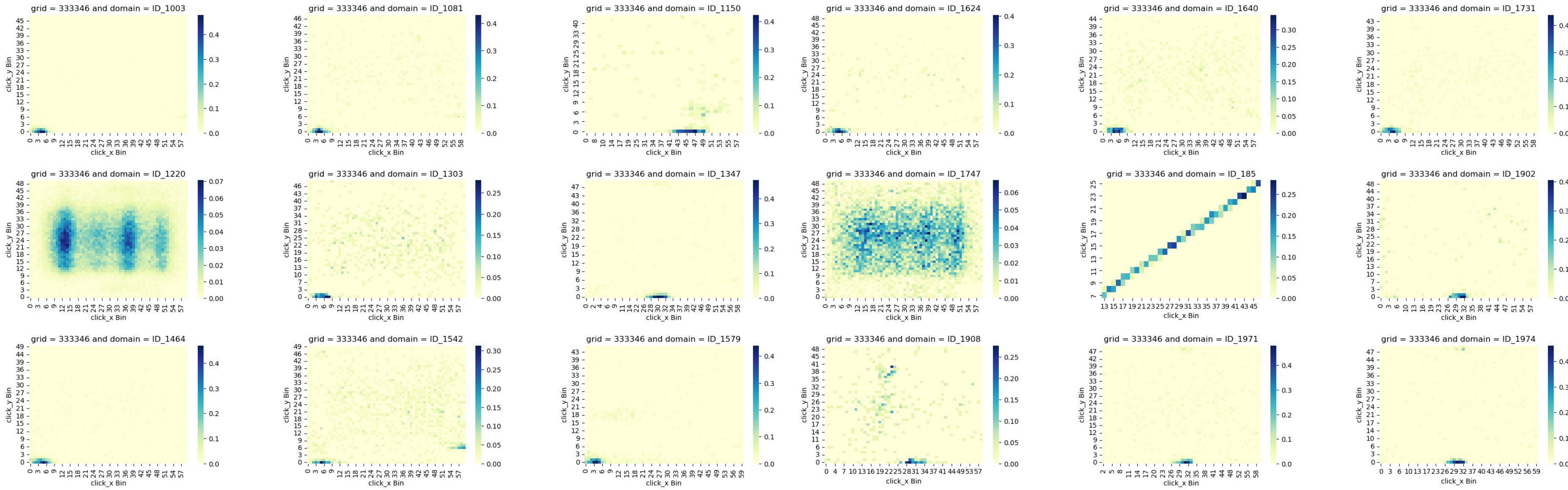


c1: all over the place
c2: left bottom corner
c3: right bottom corner
c4: bottom middle
c5: right middle
c6: top right corner
c7: all over left side
c8: diagonal

Thresholds
{'c1': 0.65,
'c2': 0.4,
'c3': 0.25,
'c4': 0.3,
'c5': 0.85,
'c6': 0.8,
'c7': 0.35,
'c8': 0.6}

F1 score = 0.74

Cosine Similarity (184 identified, 152 cb)

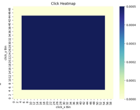


Test on New Data

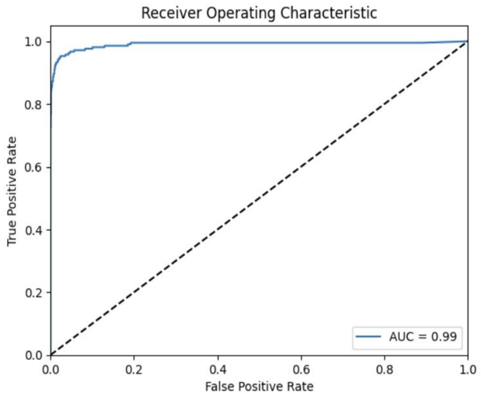
- Apply our trained model to full new data

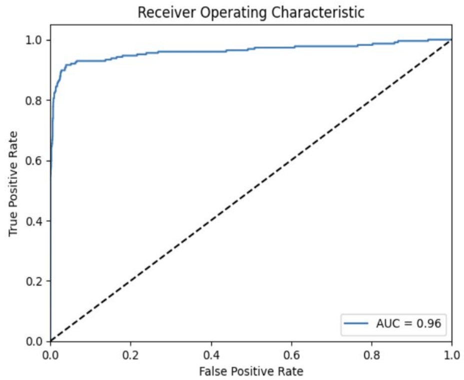
On new data	On previous data(test set)
SVM: F1 Score: 0.23	**SVM:** F1 Score: 0.25
KNN: F1 Score: 0.64	**KNN:** F1 Score: 0.83
K-Means: F1 score: 0.04	**K-Means:** F1 score: 0.24
DBScan: F1 Score: 0.82	**DBScan:** F1 Score: 0.72
Isolation Forest: F1 Score: 0.66	**Isolation Forest:** F1 Score: 0.85
Combined Model: F1 Score: 0.82	**Combined Model:** F1 Score: 0.87

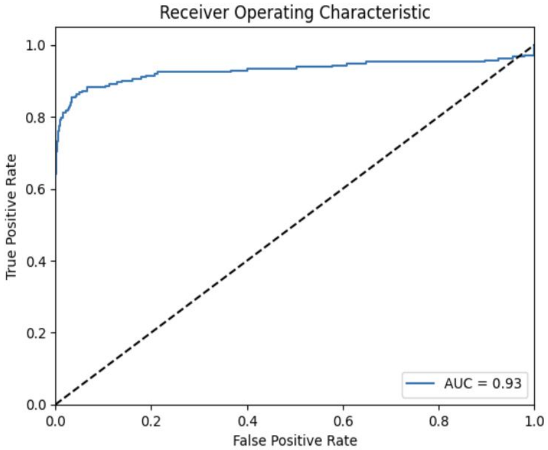
Old/New Statistical Approach

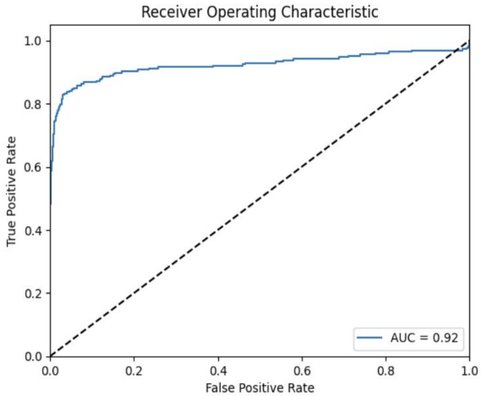
	"Good" banner	"Bad" banner	Learning	Results: old data (f1 macro (binary)/CM)	Results: new data (f1/CM)
1	**Landed clicks ratio > 0.8 + most no. clicks**	**Agg. bad data from other dataset**	**One shot**	**0.95 (0.92)** **[[1502 19]** **[15 197]]**	**0.92 (0.84)** **[[2782 22]** **[44 182]]**
2	Landed clicks ratio > 0.8 + most no. clicks	Uniform distribution	One shot	0.94 (0.9) [[1508 13] [29 183]]	0.9 (0.81) [[2769 35] [48 178]]
3	**Step uniform distribution** 	**Agg. bad data from other dataset**	**Zero shot**	**0.91 (0.84)** **[[1503 18]** **[44 168]]**	**0.89 (0.79)** **[[2774 30]** **[58 168]]**
4	Landed clicks ratio > 0.8 + most no. clicks	0.5* Agg. + 0.5* Uniform	One shot	0.95 (0.91) [[1507 14] [22 190]]	0.92 (0.83) [[2782 22] [44 182]]

First row: test 1, second row: test 3cis
First column: old data, second column: new data



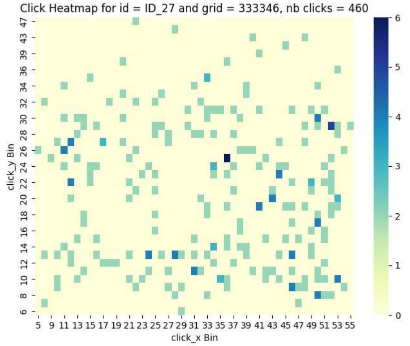


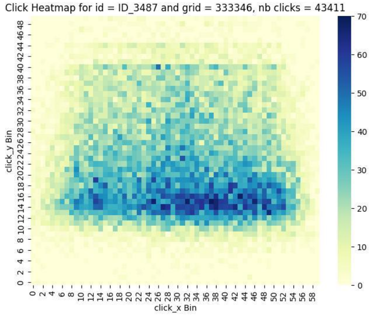




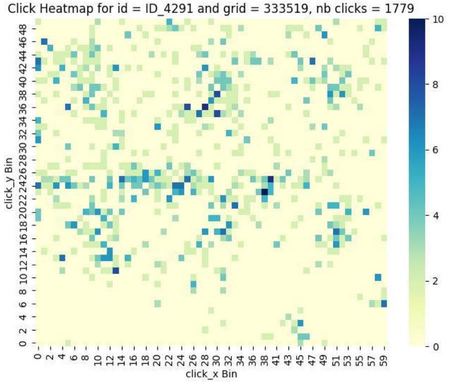
Test 1: FN (left) and FP (right) from new

Broken misclassified as working



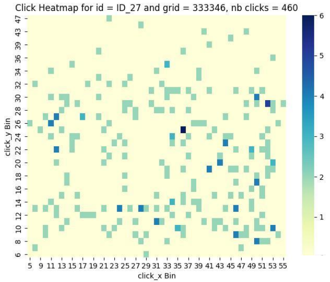


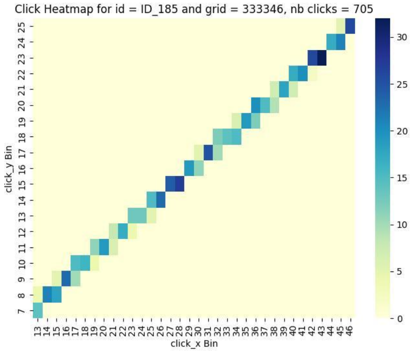
Working misclassified as broken



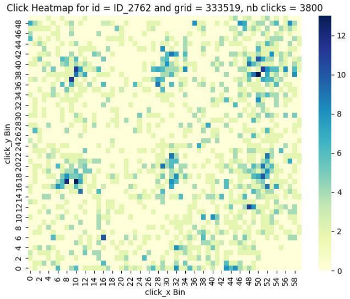
Test 3cis: FN (left) and FP (right)

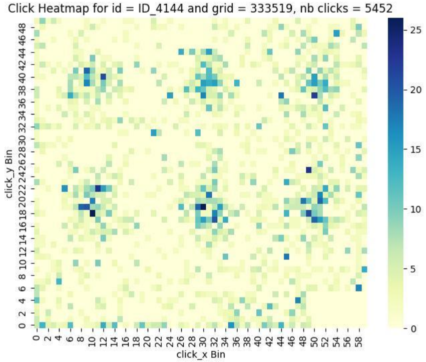
Broken misclassified as working





Working misclassified as broken





Test 1
Test 3 cis
Plot FP/FN and append number of clicks for each heatmap

Appendix

LRT method

Grid search: find optimal parameter
Threshold not that important, 1%?
Sd of 0 upsampling of 1k is OK
Change the loop for the threshold, save true positive in a dictionary, go through each domain once

Fix parameters based on three product and see how it works for six product -> how it works on test set

Ideas: upsample or downsample with noise
Look into misclassified cases

Grid search on hyperparameters: size of upsample, probability threshold,