Capstone Project: Heatmap Anomaly Detection

Week 7 Progress Report

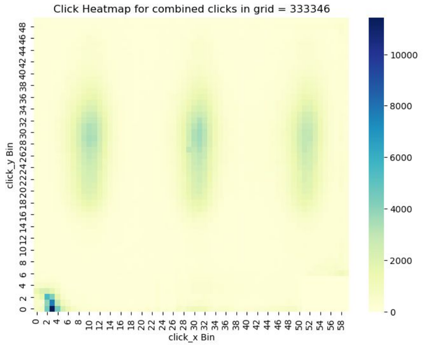
This week:

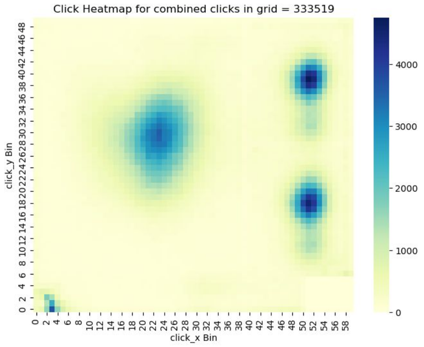
1. Understand heatmap PCA vectors
2. Finetune and combine heatmap clustering methods
3. Clustering on combined datasets (pca-ed heatmap + metrics)
4. Feature vectors of pretrained ViT and ResNet models.

Recap:

Definition of “clearly broken grid:”

Triangle/line structure not visible even with noisy bootstrap enhancement. We do not care about rest of heatmap as long as this pattern is clearly defined.

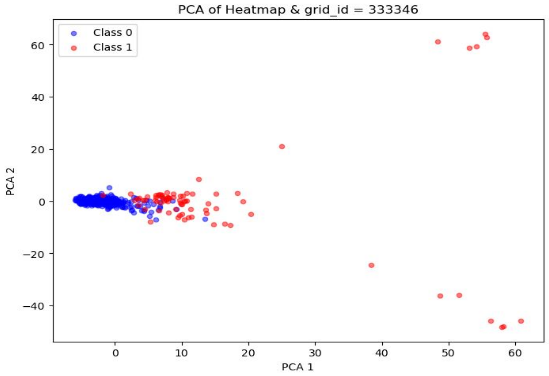


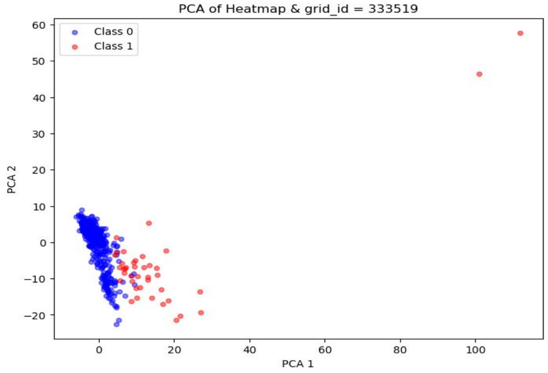


1. Created baseline of “clearly broken banners” for two grids:
    - “Triangle grid”: 57 clearly broken out of 872 → **~6.5%**
    - “Line grid”: 113 clearly broken out of 861 → **~13.5%**
    - Is it reasonable that there is such a discrepancy?

2. Added heatmap images for banners classified as “broken” to GitHub.

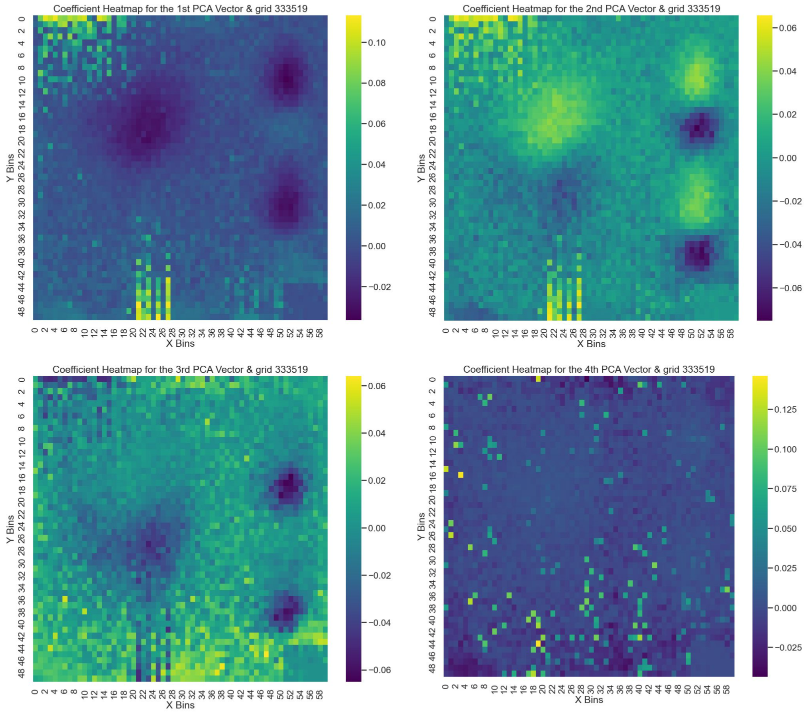
PCA (recap):





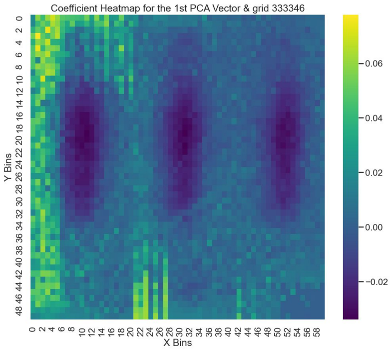
- PCA on 3000 dimensional binned heatmap vectors (EV ~ 4%, 3% resp.)
- Shows quite good pattern separating broken/non-broken banners.
- PCA vectors seem to be transferable between grid_id's (at least with 3 banners).

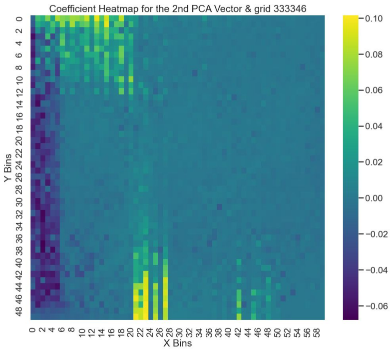
Analysis of PCA vectors (333519):

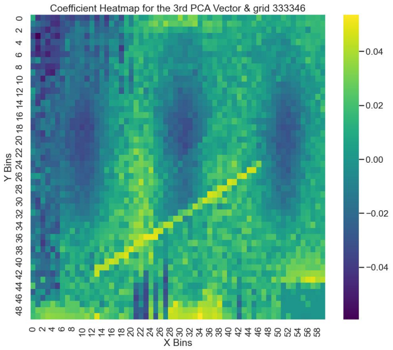


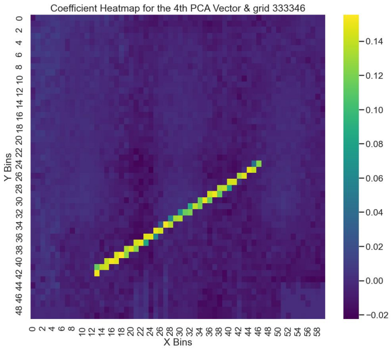
- Analysis of PCA-basis vectors:
  - PCA_1 = c_{ij} bucket_{ij}, etc.
  - Draw heatmap for c_{ij} coefficients.
- PCA1:
  - Captures “noise” outside of main clusters.
  - Especially focuses on two regions on the banner-border
- PCA2:
  - Focuses more on central clusters again together with similar outside regions.
- PCA3:
  - Similar to PCA1.
- PCA4:
  - Not really sure.

Analysis of PCA vectors (333346):









- PCA1:
  - Again, captures “noise” outside of main clusters.
  - Stronger focus on left border.
  - Similar structure on top and bottom.
- PCA2:
  - Again similar structure on top and bottom → not really sure why this is grid_id-independent
- PCA3:
  - Again similar to PCA1.
  - Including “bot-like” behavior
- PCA4:
  - Captures “bot-like” behavior.
-

Heatmap Dataset: K-NN

Comparison

- PCA dimensions: range from 2 to 200
- K-nn: 1, 2, 5, 10
- Threshold: 90, 95, 99

Grid id 333346

Best Results:

- Accuracy: 0.95
- F1 score: 0.8

Best Hyperparameter:

- PCA: 6
- K-NN: 10
- Threshold: 90

Predict on Test dataset:

- Confusion matrix: $\begin{bmatrix} 224 & 1 \\ 9 & 25 \end{bmatrix}$
- Accuracy: 0.96
- F1 Score: 0.83
- Predicted Knn labels

Grid id 333519

Best Results:

- Accuracy: 0.98
- F1 score: 0.85

Best Hyperparameter:

- PCA: 10
- K-NN: 5
- Threshold: 95

Predict on Test dataset:

- Confusion matrix: $\begin{bmatrix} 242 & 3 \\ 6 & 11 \end{bmatrix}$
- Accuracy: 0.97
- F1 Score: 0.71
- Predicted Knn labels

Heatmap Dataset: DBScan

Comparison

- PCA dimensions: range from 2 to 200
- Epsilons: 5,10,20,30,40,50,60,70,80,90,100
- Min # of data points: 2,3,4,5,10

Grid id 333346

Best Results:

- Accuracy: 0.99
- F1 score: 0.94

Best Hyperparameter:

- PCA: 12
- Epsilons: 10
- Min_sample: 10

Predict on Test dataset:

- Confusion matrix: [213 12]
[0 34]
- Accuracy: 0.95
- F1 Score: 0.85
- Predicted DBScan labels

Grid id 333519

Best Results:

- Accuracy: 0.99
- F1 score: 0.93

Best Hyperparameter:

- PCA: 8
- Epsilons: 10
- Min_sample: 10

Predict on Test dataset:

- Confusion matrix: [235 10]
[0 17]
- Accuracy: 0.96
- F1 Score: 0.77
- Predicted DBScan labels

Heatmap Dataset: OVM and Isolation Forest

Grid_ID 333346:

Best ***OVM*** model result:

```
Confusion matrix for 85 PCA dimensions:
 [[ 67 158]
 [  0  34]]
F1 Score for 85 PCA dimensions:  0.30
```

Best ***Isolation Forest*** result:

```
Confusion matrix for 6 PCA dimensions:
 [[223   2]
 [  1  33]]
F1 Score for 6 PCA dimensions with n_estimator = 50:  0.96
```

Grid_ID 333519:

Best ***OVM*** model result:

```
Confusion matrix for 163 PCA dimensions:
 [[ 56 189]
 [  0  17]]
F1 Score for 163 PCA dimensions:  0.15
```

Best ***Isolation Forest*** result:

```
Confusion matrix for 10 PCA dimensions:
 [[225  20]
 [  1  16]]
F1 Score for 10 PCA dimensions with n_estimator = 100:  0.60
```

For OVM, we select the best model from grid search for ***PCA dimensions range (from 2 to 200)***.
For IF, we select the best model from grid search over ***PCA dimensions (from 2 to 200)*** and ***number of estimators [5,10,50,100,150,200]***.

Heatmap Dataset: K-Means

Grid_ID 333346:
Best PCA dimension: 54

```
[[225   0]
 [ 31   3]]
Test accuracy: 0.88
Test precision: 1.00
Test recall: 0.09
Test f1-score: 0.16
```

Grid_ID 333519:
Best PCA dimension: 86

```
[[241   4]
 [  2  15]]
Test accuracy: 0.98
Test precision: 0.79
Test recall: 0.88
Test f1-score: 0.83
```

Heatmap Dataset: Combining all 5 models

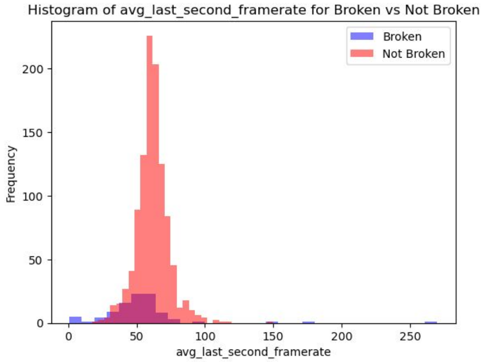
Grid_ID 333346:

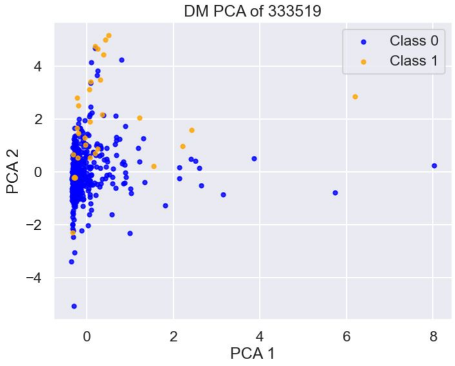
```
Confusion matrix for ((1, 1, 1, 1, 1)) included and 1:
  [[224   1]
 [ 10  24]]
Best f1 score: 0.97
```

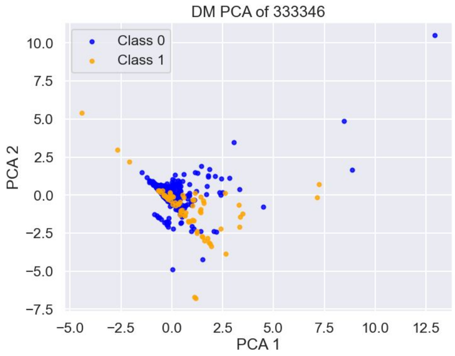
Grid_ID 333519:

```
Confusion matrix for ((1, 1, 1, 1, 1)) included and 1:
  [[241   4]
 [  1  16]]
Best f1 score: 0.87
```

Combine datasets (Recap)

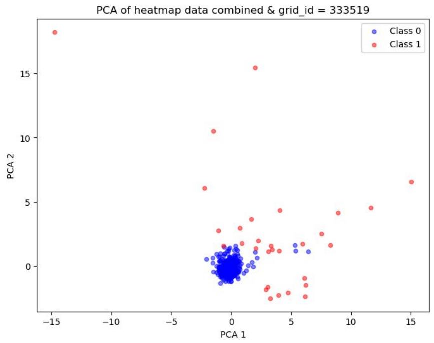


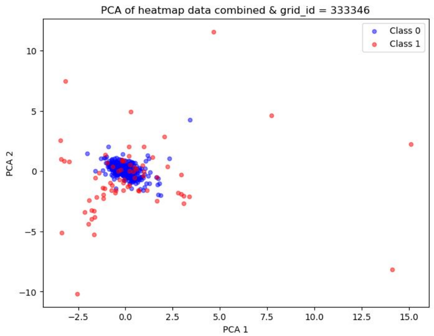


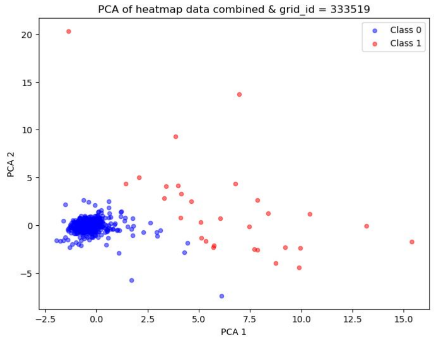


- Basic supervised classifiers (Decision Trees, Random Forest, Logistic Regression, XGBoost, etc) don't perform well
    - Problem with **unbalanced dataset** and overfitting.
        - **Upsample/downsample** → slight improvements but not competitive
    - **Overfitting** → Grid search over model complexity (regularization).
    - Add interactions/higher-order features
- **To Do** → Try to improve using SMOTE/other more sophisticated "upsamplying" strategies.
- PCA less powerful but shows some structure.

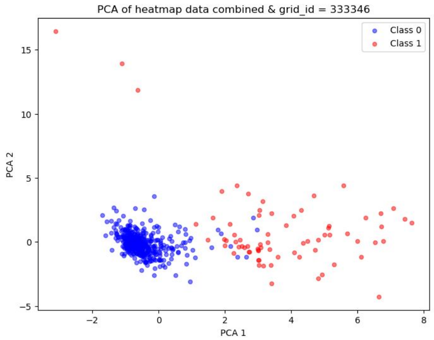
Combined Datasets







EV: 2.59%



EV: 2.2%

- Different methods of aggregating PCA's:
  - PCA on both individually → combine → scale → combined PCA (top)
  - PCA on HM only → combine → scale → combined PCA (bottom)
  - Some dependence on nr of heatmap-PCA components.
- Adding more components changes scores a little bit.

Combined Datasets clustering performance:

Grid: 333519
Training performance (KMeans & 70 PCA dim's):

```
---------
Confusion matrix for ((0, 0, 0, 1, 0)) included and 0:
  [[555   4]
 [  0  30]]
F1-score for ((0, 0, 0, 1, 0)) included and 0:  0.94
Accuracy for ((0, 0, 0, 1, 0)) included and 0:  0.99
Recall for ((0, 0, 0, 1, 0)) included and 0:  1.00
Precision for ((0, 0, 0, 1, 0)) included and 0:  0.88
---------
```

Test performance (KMeans & 70 PCA dim's):

```
---------
Confusion matrix for 70 PCA dimensions:
  [[243   0]
 [  0  15]]
F1-score for 70 PCA dimensions:  1.00
Accuracy for 70 PCA dimensions:  1.00
Recall for 70 PCA dimensions:  1.00
Precision for 70 PCA dimensions:  1.00
---------
```

- Run Gridsearch over different clustering methods (KNN, KMeans, DBScan, IsolationForest, OneClassSVM)
- Compare combined performance and select "best performing" model.
- DBScan on 2PCA dimension very good (but test performance worse.
- KMeans really powerful but very sensitive:

```
Confusion matrix for 30 PCA dimensions:
  [[  3 556]
 [ 28   2]]
F1-score for 30 PCA dimensions:  0.01
Accuracy for 30 PCA dimensions:  0.01
Recall for 30 PCA dimensions:  0.07
Precision for 30 PCA dimensions:  0.00
---------
Confusion matrix for 32 PCA dimensions:
  [[556   3]
 [  3  27]]
F1-score for 32 PCA dimensions:  0.90
Accuracy for 32 PCA dimensions:  0.99
Recall for 32 PCA dimensions:  0.90
Precision for 32 PCA dimensions:  0.90
---------
```

Combined Datasets clustering performance:

Grid: 333346
Training performance KMeans & 50 PCA dim’s:

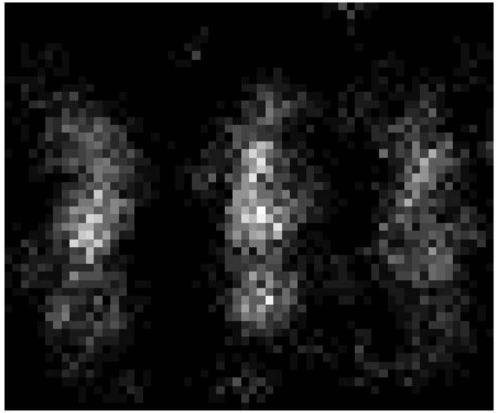
```
---------
Confusion matrix for ((0, 0, 0, 1, 0)) included and 0:
  [[506   2]
 [  0  72]]
F1-score for ((0, 0, 0, 1, 0)) included and 0:  0.99
Accuracy for ((0, 0, 0, 1, 0)) included and 0:  1.00
Recall for ((0, 0, 0, 1, 0)) included and 0:  1.00
Precision for ((0, 0, 0, 1, 0)) included and 0:  0.97
---------
```

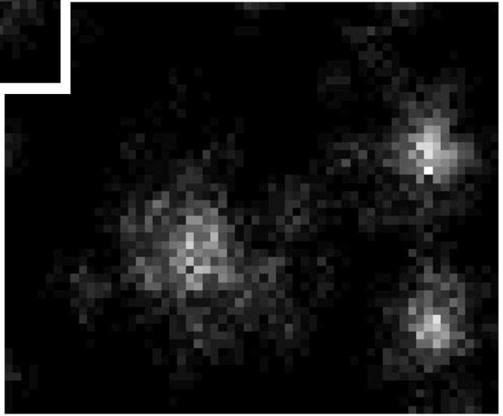
Test performance (KMeans & 50 PCA dim’s):

```
---------
Confusion matrix for 50 PCA dimensions:
  [[219   1]
 [ 30   1]]
F1-score for 50 PCA dimensions:  0.06
Accuracy for 50 PCA dimensions:  0.88
Recall for 50 PCA dimensions:  0.03
Precision for 50 PCA dimensions:  0.50
---------
```

- Run Gridsearch over different clustering methods (KNN, KMeans, DBScan, IsolationForest, OneClassSVM)
- Compare combined performance and select “best performing” model.
- KMeans again excellent performance
- Other clustering methods pretty poor

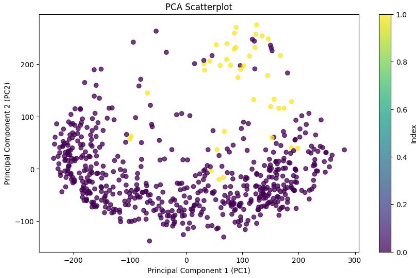
Pretrained ViT/ResNet:



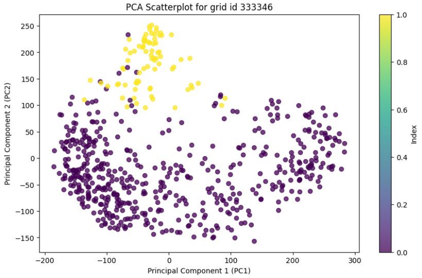


- Feed (transformed and binned) heatmaps into pre-trained ViT/ResNeT.
  - google/ViT: Transformer-based architecture, 14M images (224x224), 21k classes
  - Microsoft/ResNet-1k: trained on ImageNet-1k (224x224), 1k classes.
- Extract features (before classification head)
  - ViT → 151296 dim'l feature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

ViT + PCA:

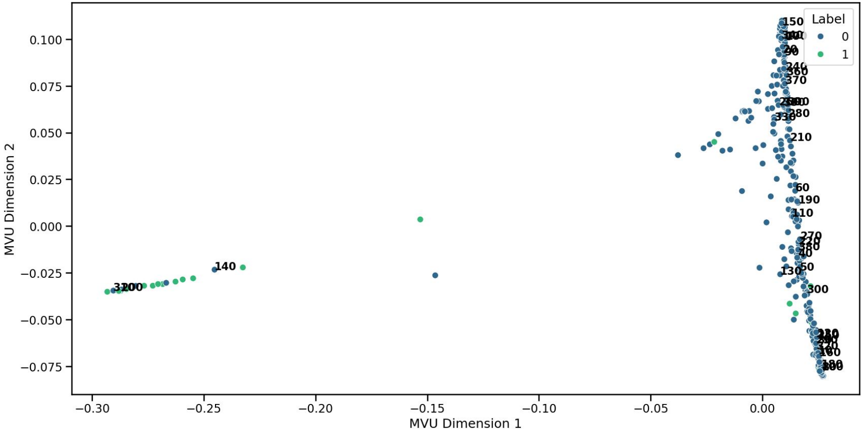


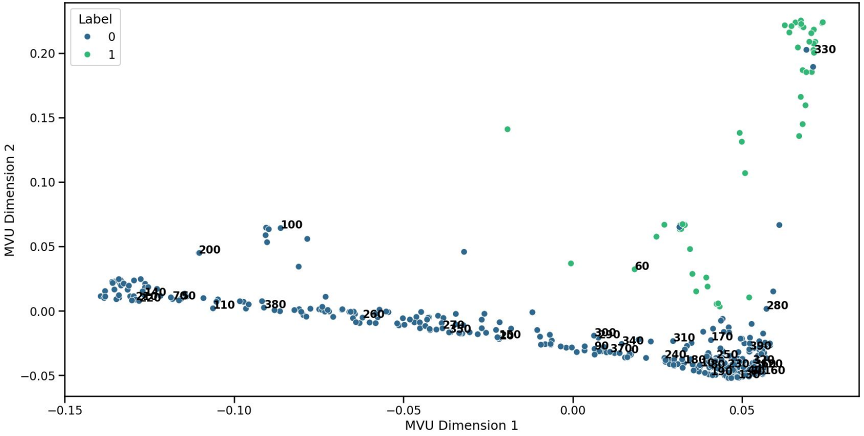
Grid 333510, EV ~ 18%



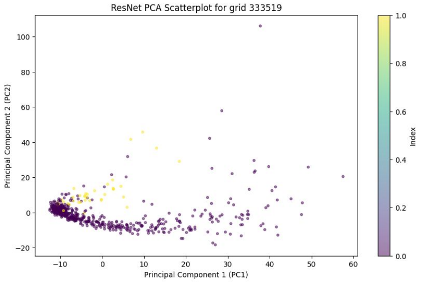
Grid 333346, EV ~ 17%

- PCA → some structure, but not as good as vanilla method.
- (nonlinear) MVU shows clear patterns however → clearly features contain important information.
- Clustering on PCA's not performant.
- → require fine-tuning? Contrastive learning on synthetic data?

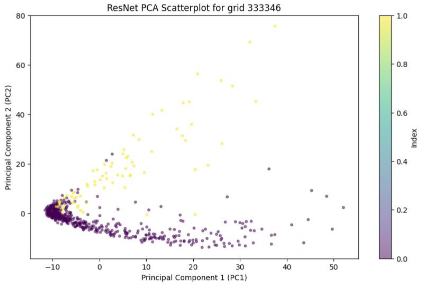




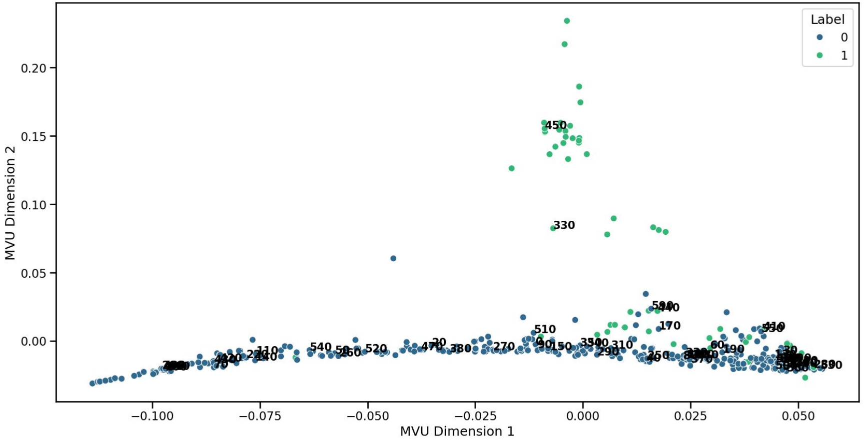
ResNet + PCA:



Grid 333510, EV ~ 18%

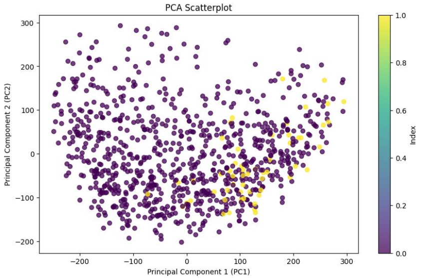


Grid 333346, EV ~ 17%

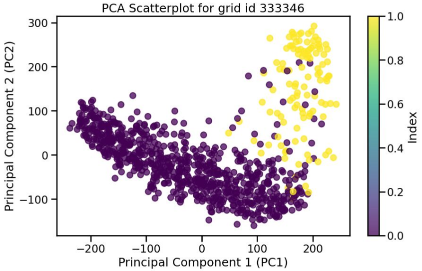


- PCA → some structure, but not as good as vanilla method.
- (nonlinear) MVU shows clear patterns however → clearly features contain important information.
- Clustering on PCA's not performant.
- → require fine-tuning? Contrastive learning on synthetic data?

ViT + PCA + data enhancement:

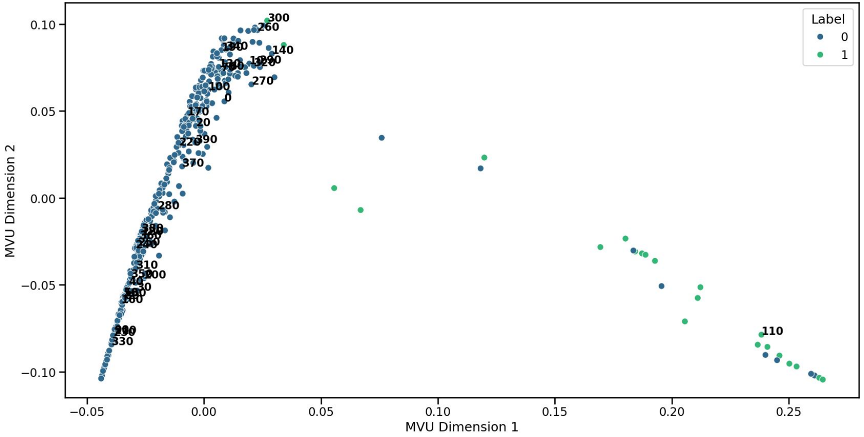


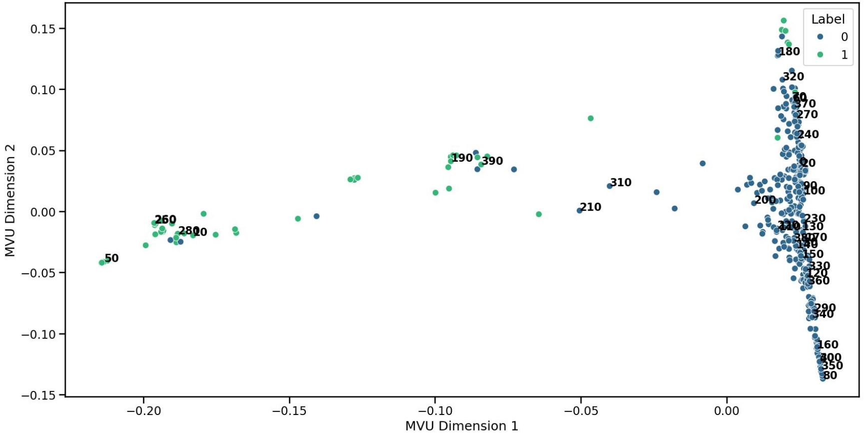
Grid 333519, EV ~ ??%



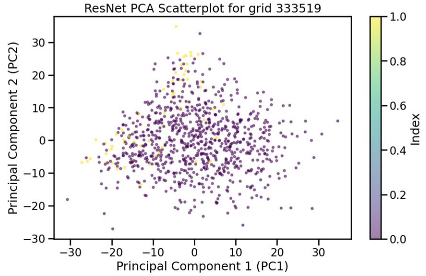
Grid 333346, EV ~ ??%

- PCA → worse especially in the 333519 grid
- (nonlinear) MVU shows clear patterns however → clearly features contain important information. Worse than without enhancement.
- Clustering on PCA's not performant.
- → require fine-tuning? Contrastive learning on synthetic data?

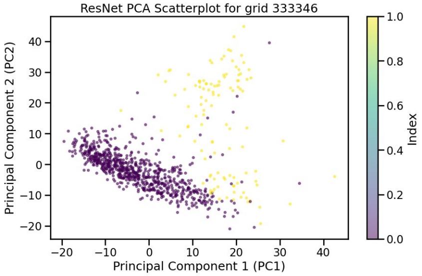




ResNet + PCA + data enhancement:

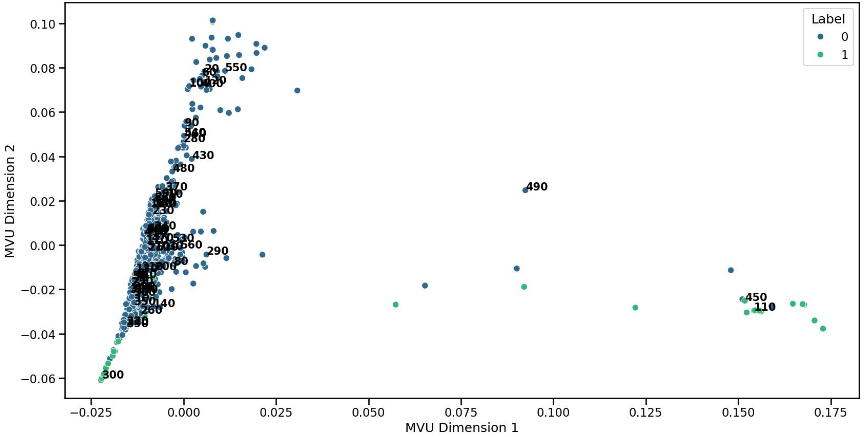


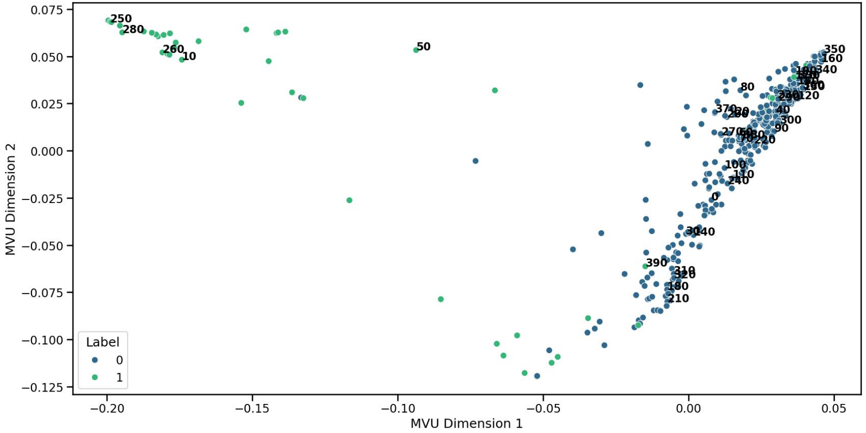
Grid 333519, EV ~ ??%



Grid 333346, EV ~ ??%

- PCA → worse especially in the 333519 grid
- (nonlinear) MVU shows clear patterns however → clearly features contain important information. Worse than without enhancement.
- Clustering on PCA's not performant.
- → require fine-tuning? Contrastive learning on synthetic data?



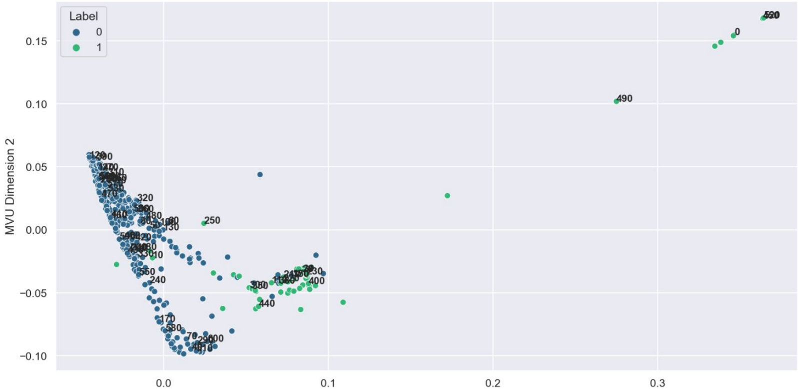


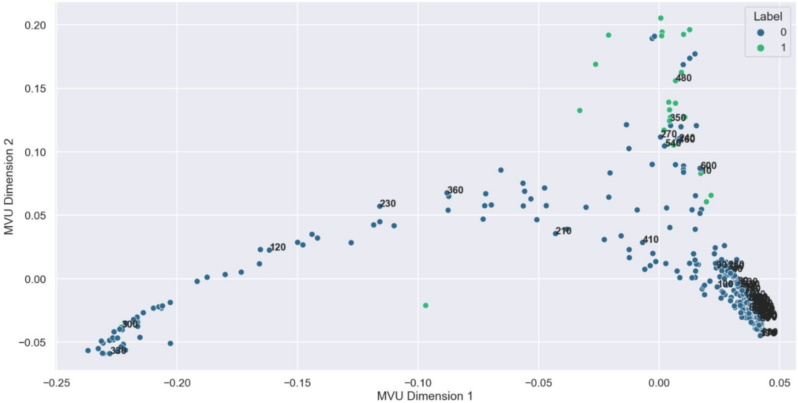
Next steps:

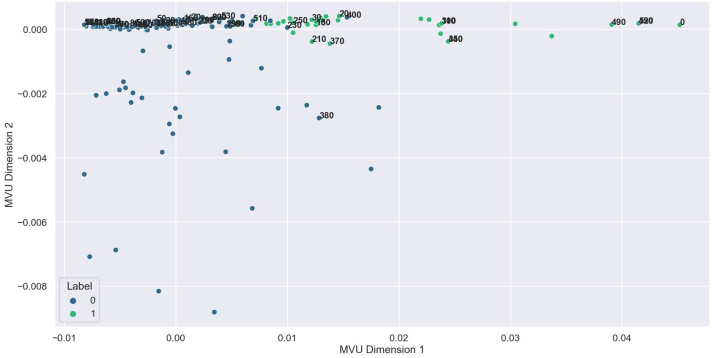
- Further exploration of "basic models":
  - Stability of models across grids.
    - Access to more grid's?
- "Productionize" current models into single pipeline:
  - Clustering on heatmap
  - Clustering on Metrics+heatmap
  - Click clustering methods
  - Clustering of pretrained features (?)
  - → combine into Majority voting pipeline → stability?
- Pre-trained models: Can we access "nonlinear" geometry?
  - Combine with vanilla features?
  - Re-add classification head
  - Other models trained on contrastive tasks?
  - Train Autoencoder on arbitrary synthetic clusterings (ask AE to recreate original image with discriminative loss).
  - Train/Fine-tune ResNet/ViT/... on synthetic data to count number of clusters → might lead to interesting feature vectors.

Appendix

MVU – study “data manifold”:

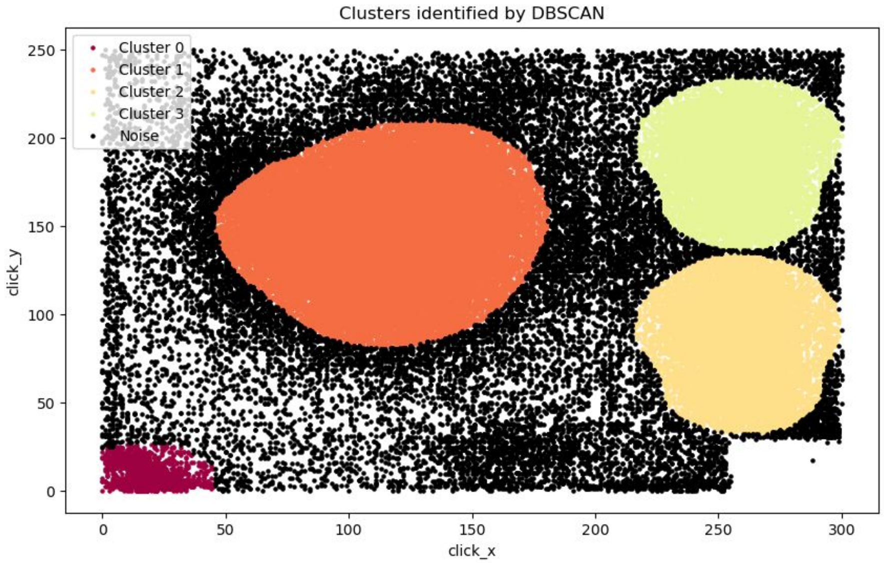






- Study data manifold using Maximum Variance Unfolding/Semidefinite Embedding (MVU)
  - Intuition: create graph of close points in high dimensional space with distances
  - Use convex optimization to maximize distance between disconnected points s.t. connected points being nearby.

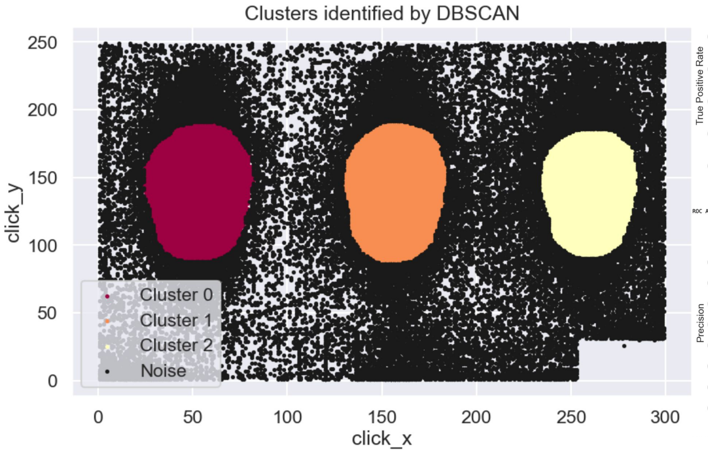
Click clustering method (recap):

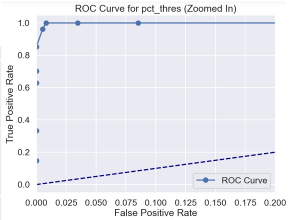


1. Bootstrap 100'000 clicks from fully aggregated dataset (filtered by grid_id).
2. Normalize (Standardize)
3. Run DBSCAN cluster with eps = .2 and min_samples = 1000
   → 4 clusters + noise.
4. For given (noisy bootstrap enhanced) domain, get 1-nn for each click in training data and select that label {0,1,2,3}
   a. If pct of points labelled as noise above a certain threshold → anomalous.
   b. Hypothesis testing: p_0 = pct of noise points in training data. H_0: p_0 < noise/total, H_A: p_0 >= noise/total → p-value larger than threshold (cannot reject null) → anomalous.

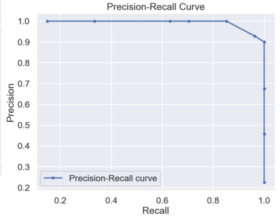
Click clustering method:

```
Epsilon: 0.13
Estimated number of clusters: 3
Estimated number of noise points: 37041
Pct of noise points: 0.370410
```





ROC AUC: 0.25

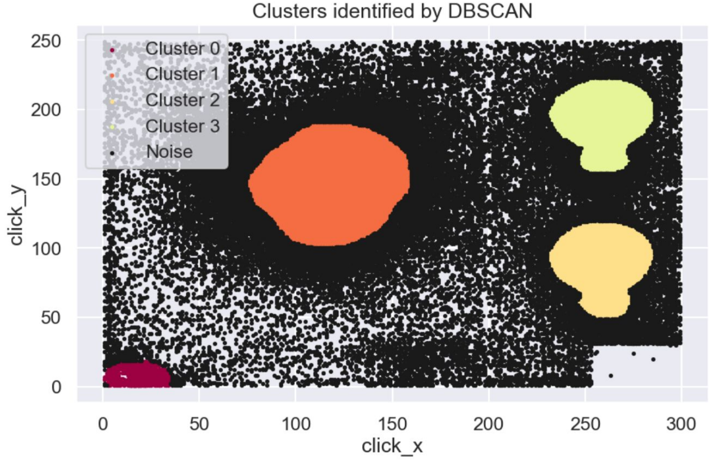


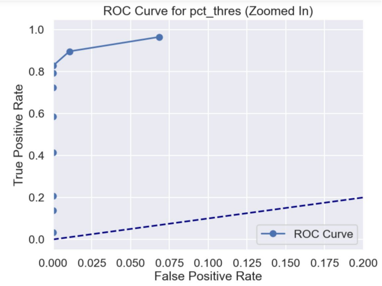

```
------------------------
Epsilon: 0.13 delta pct thres: 0.00 Pct_thres: 0.63

 Broken pct: 0.14, Total in CB: 27, Missed in CB: 1, Pct missed: 0.04, Not in CB: 2
confusion matrix (rate) ((TPR,FNR),(FPR,TNR)):
 ( 0.96 , 0.04 )
 ( 0.01 , 0.99 )
```

- **100k** bootstrapped samples to generate clusters.
- Use enhanced **5k** bootstrapped samples per domain.
  - The two grid's perform very well upon hyperparameter tuning:
    - Grid_id = 333346 performs best with eps = 0.13 and 1k min_sample

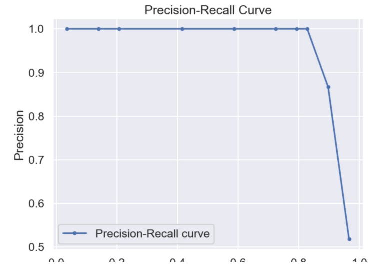
Click clustering method:

```
Epsilon: 0.13
Estimated number of clusters: 4
Estimated number of noise points: 38634
Pct of noise points: 0.521510
```






```
ROC AUC: 0.06
```




```
Epsilon: 0.15 delta pct thres: 0.00 Pct_thres: 0.57

 Broken pct: 0.06, Total in CB: 29, Missed in CB: 5, Pct missed: 0.17, Not in CB: 0
confusion matrix (rate) ((TPR,FNR),(FPR,TNR)):
 ( 0.83 , 0.17 )
 ( 0.00 , 1.00 )
```

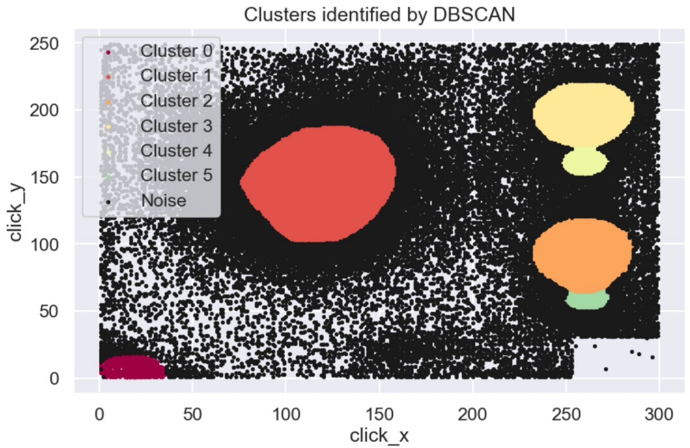
100k bootstrapped samples to generate clusters.

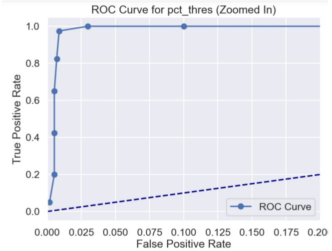
Use enhanced **5k** bootstrapped samples per domain.

- The two grid's perform very well upon hyperparameter tuning:
  - Grid_id = 333519 performs best with eps = 0.13 and 1k min_sample
  - We remove "corner clusters".

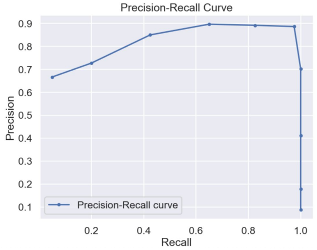
Performance across grid:

```
Epsilon: 0.13
Estimated number of clusters: 6
Estimated number of noise points: 39161
Pct of noise points: 0.391610
```





ROC AUC: 0.72




```
Epsilon: 0.13 Pct_thres: 0.64

 Broken pct: 0.07, Total in CB: 40, Missed in CB: 1, Pct missed: 0.03, Not in CB: 5
confusion matrix (rate) ((TPR,FNR),(FPR,TNR)):
 ( 0.97 , 0.03 )
 ( 0.01 , 0.99 )
```

- Correspondingly, translating the same parameters from one to the does perform well:
  - 333346 → 333519:
    - **Remove corner cluster**:
      - **TPR**: 0.97 (1/40 missed)
      - **FNR**: 0.03 (5/600-ish)
    - Include corner cluster:
      - TPR: 0.8 (8/40 missed)
      - FNR: 0.0 (0/600-ish)
  - 333519 → 333346:
    -