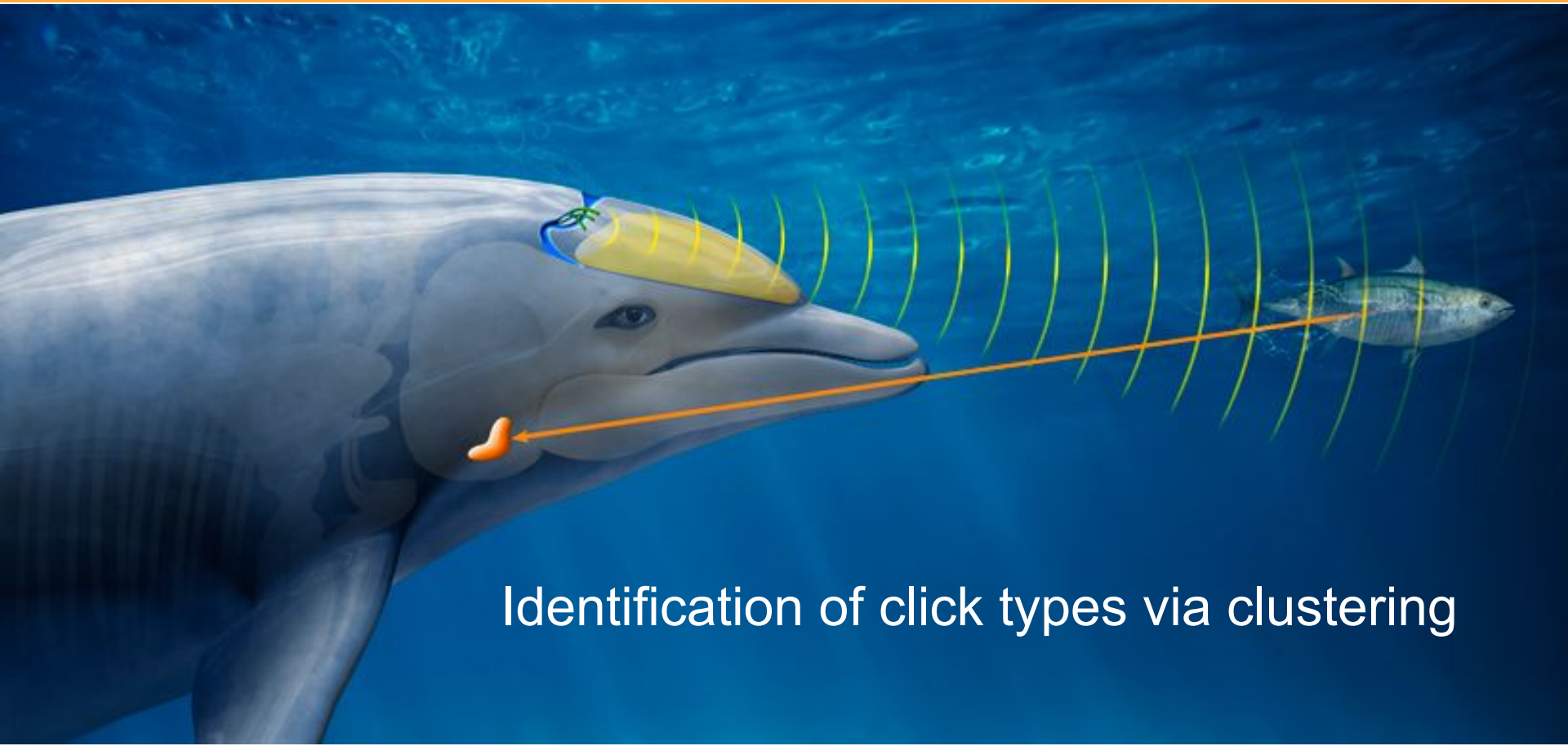
Dolphin echolocation behaviour



The Data

The dataset was obtained from a Phd candidate conducting an ongoing research on dolphin communities

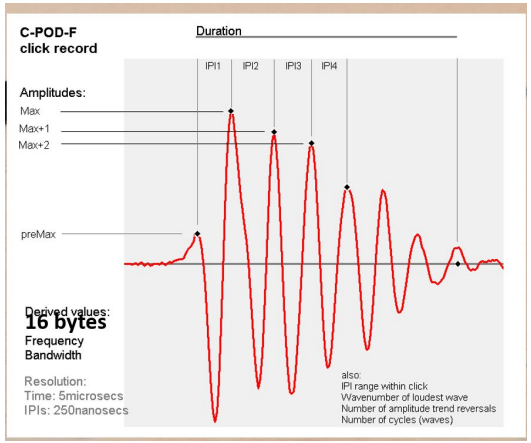
TrnID_1	ClksThisMin_1	NofClx_1	medianKHz	avEndF_1	nRisingIPIs	avSPL_1	TrDur_us_1	MedianPRF	nIClrising	MinICI_us_1	midpointICI	MaxICI_us_1	ClkNofMinICI	ClkNofMaxICI	NofClstrs_1	avClstrNx8	avclF0	avclF1	avPkIPI	BeforeIPIratio	PreIPIratio	Post1IPIratio	Post2IPIratio	Endl
16.0	2240.0	10.0	64.0	68.0	0.0	170.0	481481.0	21.0	4.0	8272.0	9440.0	10933.0	6.0	6.0	5.0	8.0	0.0	0.0	65.0	1.12	1.090	1.000	0.93	
15.0	2128.0	9.0	114.0	111.0	0.0	130.0	350000.0	27.0	4.0	6732.0	7333.0	8047.0	6.0	5.0	8.0	8.0	0.0	0.0	35.0	0.00	1.100	0.970	0.00	
16.0	1993.0	10.0	118.0	118.0	0.0	170.0	444444.0	22.0	4.0	8409.0	9149.0	9992.0	6.0	5.0	7.0	8.0	0.0	0.0	34.0	0.00	1.120	1.030	1.06	
16.0	2530.0	10.0	112.0	103.0	0.0	121.0	26262.0	400.0	3.0	461.0	483.0	549.5	5.0	5.0	0.0	0.0	0.0	0.0	36.0	0.00	1.140	0.000	0.00	

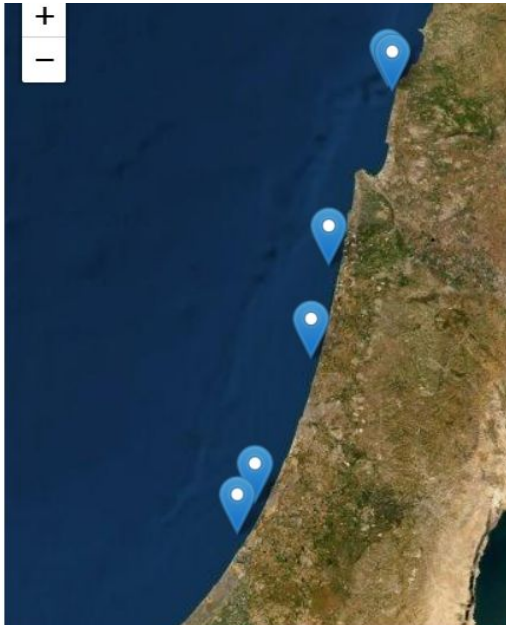
- Two different hydrophones C-POD, newer F-POD
- I am working only on F-pod for now due to size limitations

```
all_train.shape
```

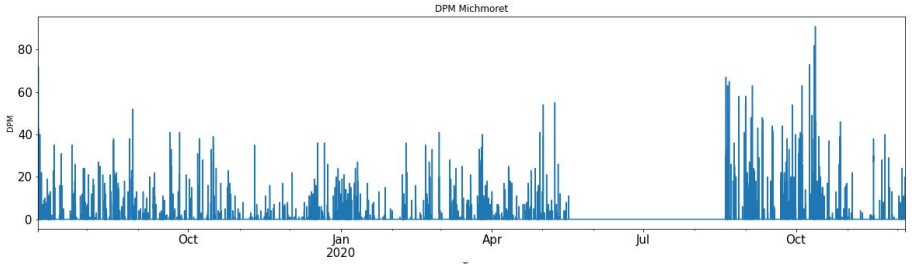
```
(614765, 76)
```

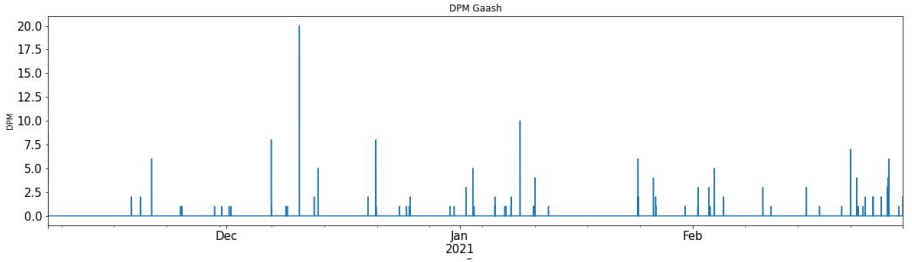
```
site
Achziv          410
Ashdod       529073
Ashkelon       6885
Dor             214
Gaash           277
Michmoret     71407
dtype: int64
```

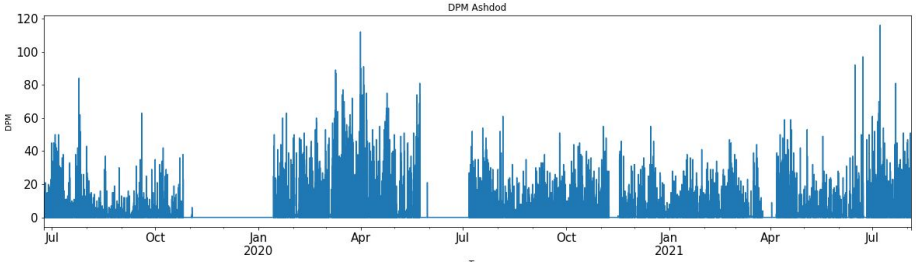


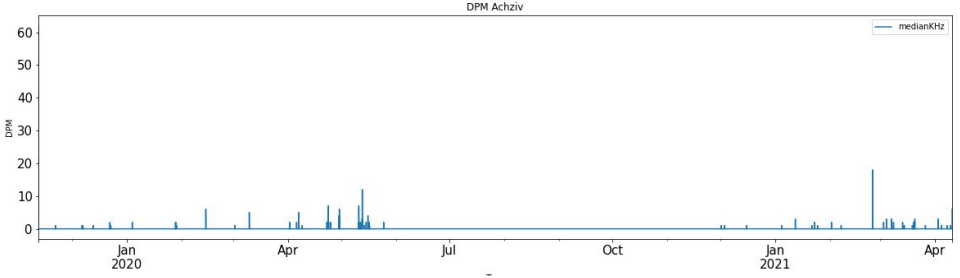


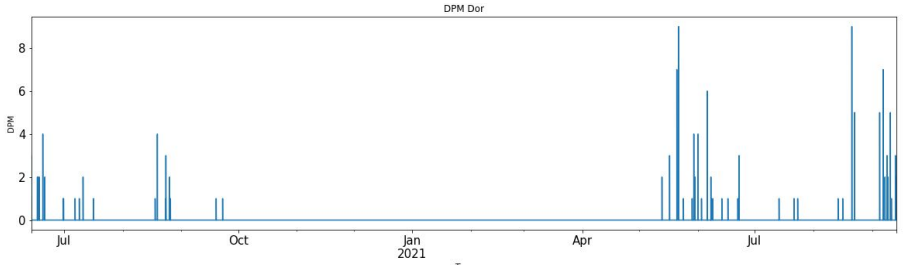
Typical information from the device - Present\Absent

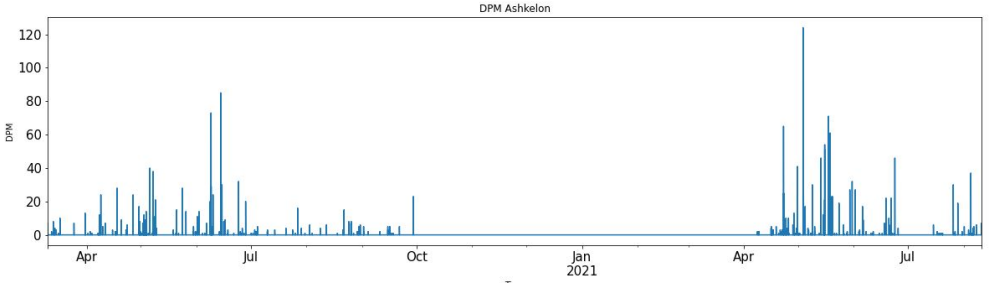




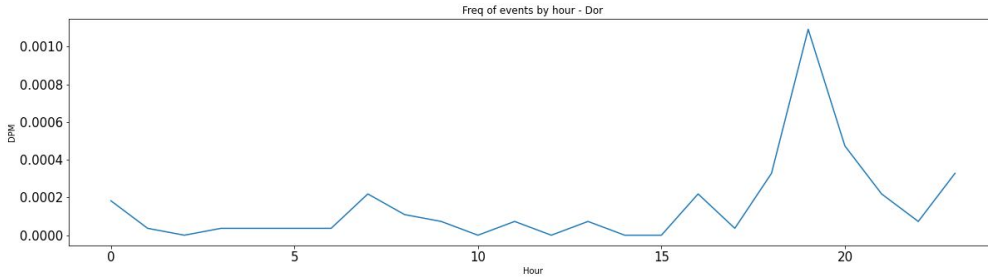


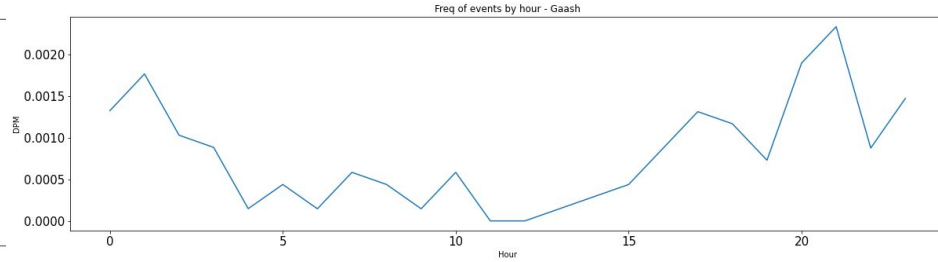


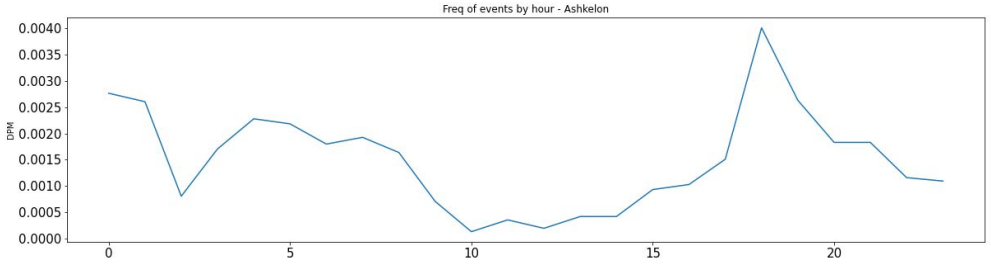


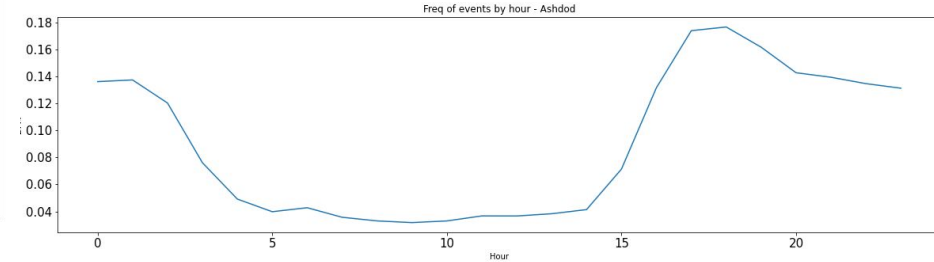


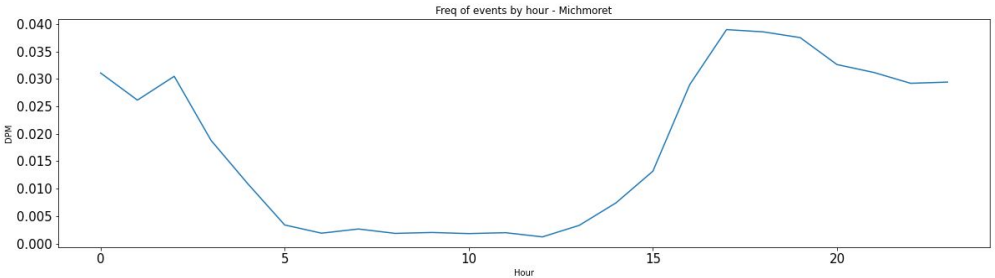
Hourly frequency of events - searching for daily behavior

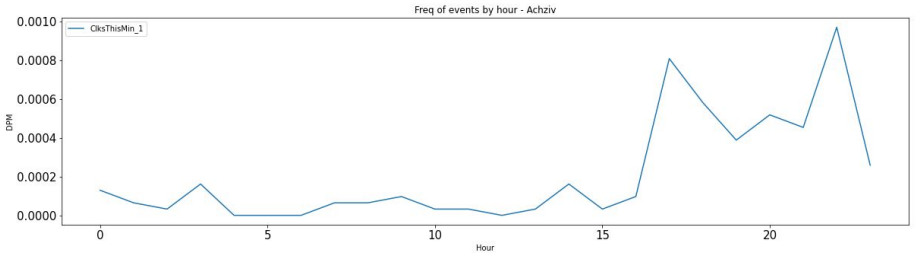




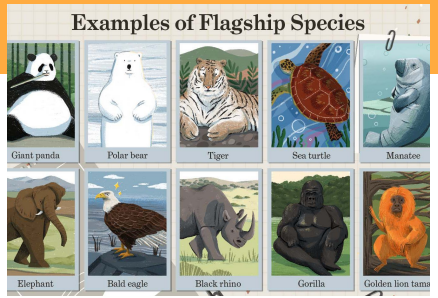








Outliers and Feature engineering



***Tursiops truncatus* - דולפינן מצוי**

The business question -" What is there???"

Why should you do it?

Better defined -

step 1 - finding different "behaviours" associated to the "train" types - Clustering - Initial stage

Stenella coeruleoalba - סטנלה מפוספסת

step 2 - Analysis of results - Future

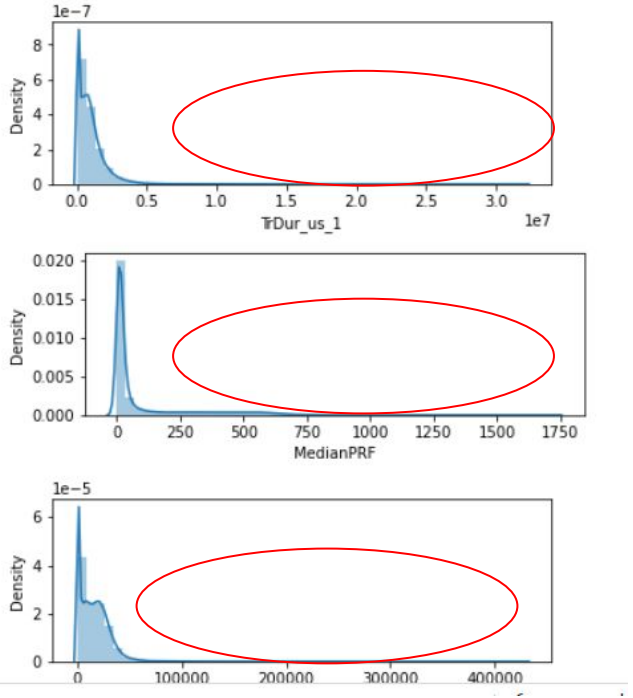
step 3 - Automated classification - Maybe?

Outliers and Feature engineering

I tried many different approaches

- 5 min bins
- Removing outliers with PCA
- I removed columns due to correlation
- bins for noisy columns of lesser importance
- extracting sites from other columns
- removing very low frequency values
- removing unreliable data
- I removed irrelevant columns

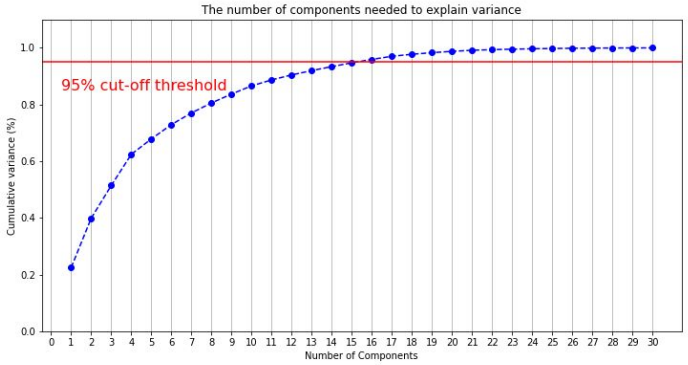
Most of the features has very long “tails” that can disturb later clustering



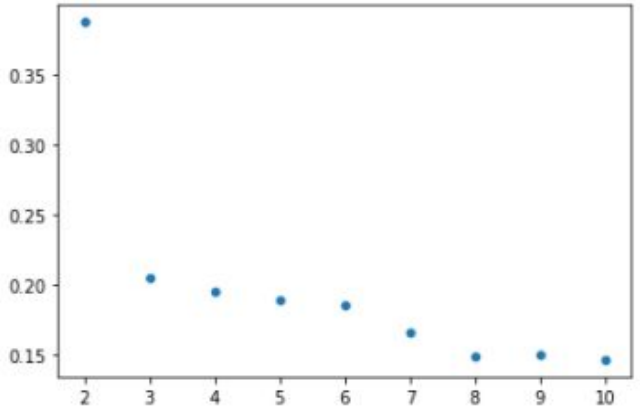
Clustering is a sea of uncertainty

K-Means - Standard Scaler, PCA, Silhouette and clustering results

PCA - I chose to reduce to 16 columns



Silhouette -I chose 8 clusters



Scaling

scikit learn's standard scaling.

standard scaling emphasize the mean at 0

I used dummies for the site column.

Cluster size

0	77241
1	6375
2	142764
3	31743
4	81416
5	67845
6	117955
7	34389

interesting…

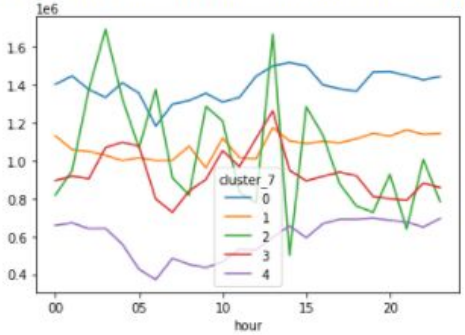
```
1  all_five.groupby(['cluster_7','site']).size()
```

```
cluster_7  site
0          ashd    12762
1          ashd    18627
2          ashk      579
3          Mich     5402
           achz      113
           dor        74
           gaa        99
4          ashd    11800
dtype: int64
```

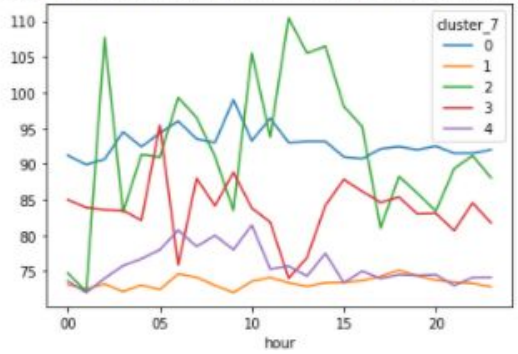
Between Cluster variability - testing results

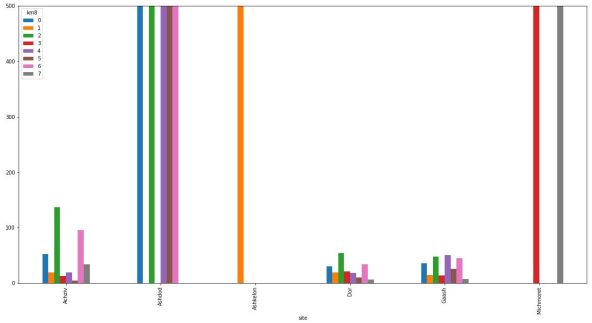
```
1  temp.groupby(['hour','cluster_7']).TrDur_us_1.median().
```

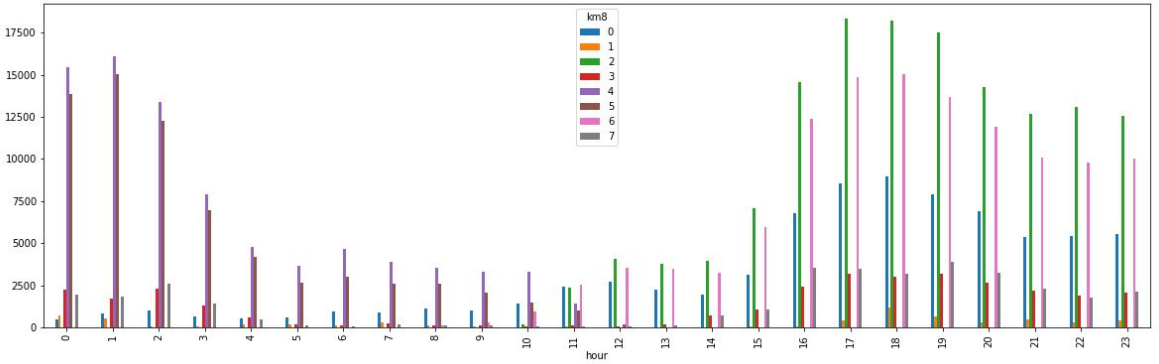
```
<matplotlib.axes._subplots.AxesSubplot at 0x7fd1a2e70490>
```



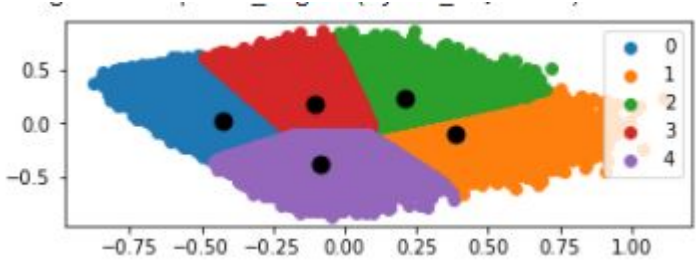

```
<matplotlib.axes._subplots.AxesSubplot at 0x7fd1a10cadd0>
```

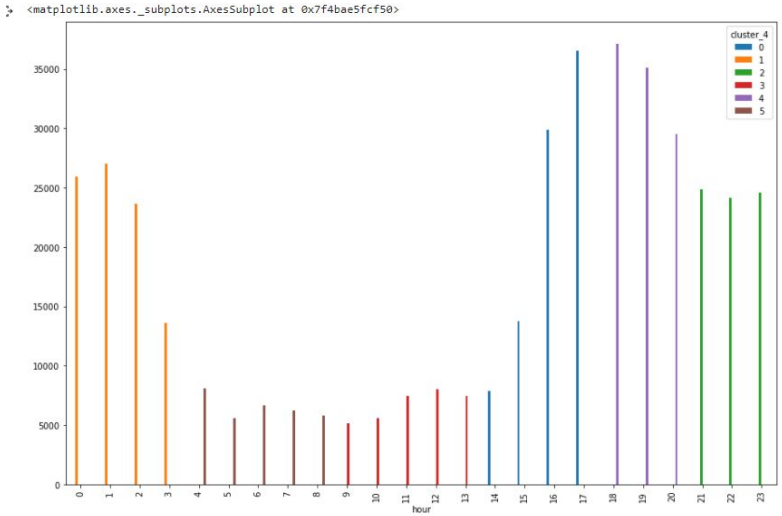






Biased results - weights of features control the results!






```
Requirement already satisfied: MarkupSafe>=0.23
```

Weight	Feature
102015.3407 ± 195.8733	site_Ashdod
101750.6633 ± 288.0505	site_Michmoret
30874.7928 ± 104.3634	avclF1
29215.5402 ± 130.1415	avclF0
18175.2166 ± 16.5591	medianKHz
14716.2647 ± 66.8436	avEndF_1
12484.0408 ± 102.4597	avSPL_1
9765.3351 ± 51.1398	TrDur_us_1
7853.9554 ± 23.6647	avPklPI
5460.5633 ± 23.0392	BeforeIPIratio
4187.6535 ± 22.6757	Post2IPIratio
3552.1910 ± 21.9569	MaxICI_us_1
3277.2151 ± 24.0953	MedianPRF
2217.6506 ± 21.8857	midpointICI
1855.5308 ± 12.3580	PreIPIratio
1783.7261 ± 11.8048	MinICI_us_1
1631.6979 ± 5.8911	NofClstrs_1
883.0130 ± 3.4632	avClstrNx8
810.0554 ± 14.8920	Post1IPIratio
554.8347 ± 10.0705	nRisingIPIs
… 10 more …	

Light in the darkness..

RESEARCH ARTICLE

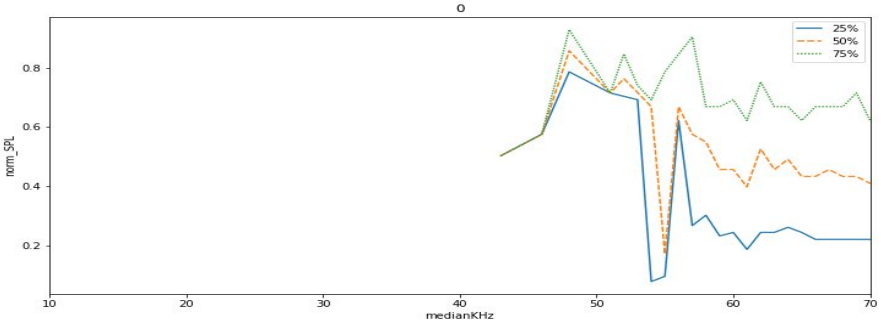
Automated classification of dolphin echolocation click types from the Gulf of Mexico

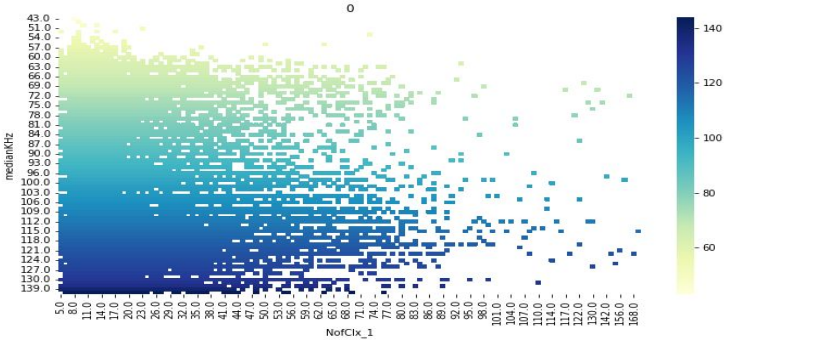
Kaitlin E. Frasier[1]*, Marie A. Roch[2], Melissa S. Soldevilla[3], Sean M. Wiggins[1], Lance P. Garrison[3], John A. Hildebrand[1]

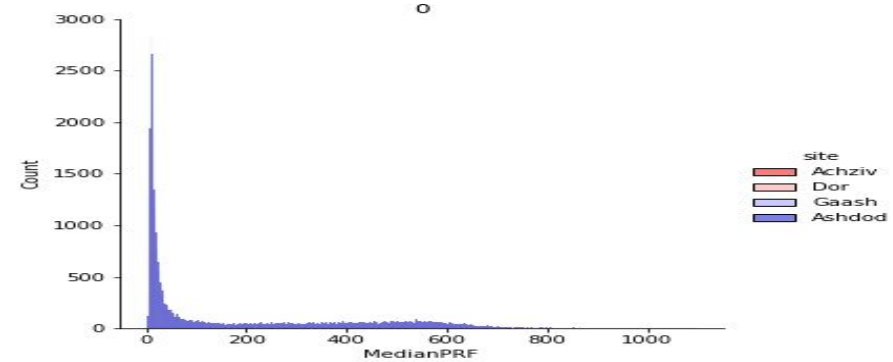
1 Scripps Institution of Oceanography, La Jolla, California, United States of America, **2** San Diego State University, San Diego, California, United States of America, **3** NOAA NMFS Southeast Fisheries Science Center, Protected Resources and Biodiversity Division, Miami, Florida, United States of America

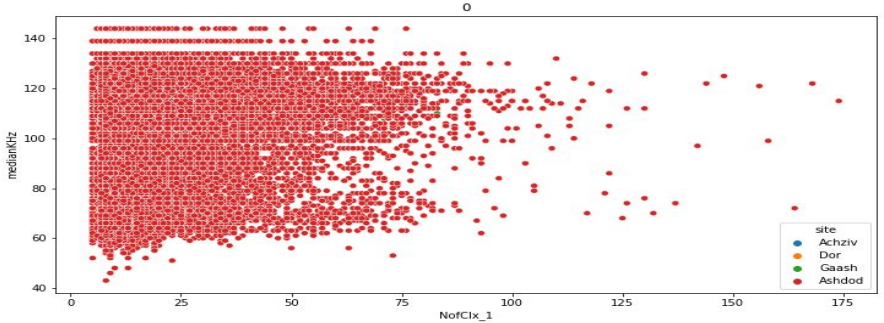
* kfrasier@ucsd.edu

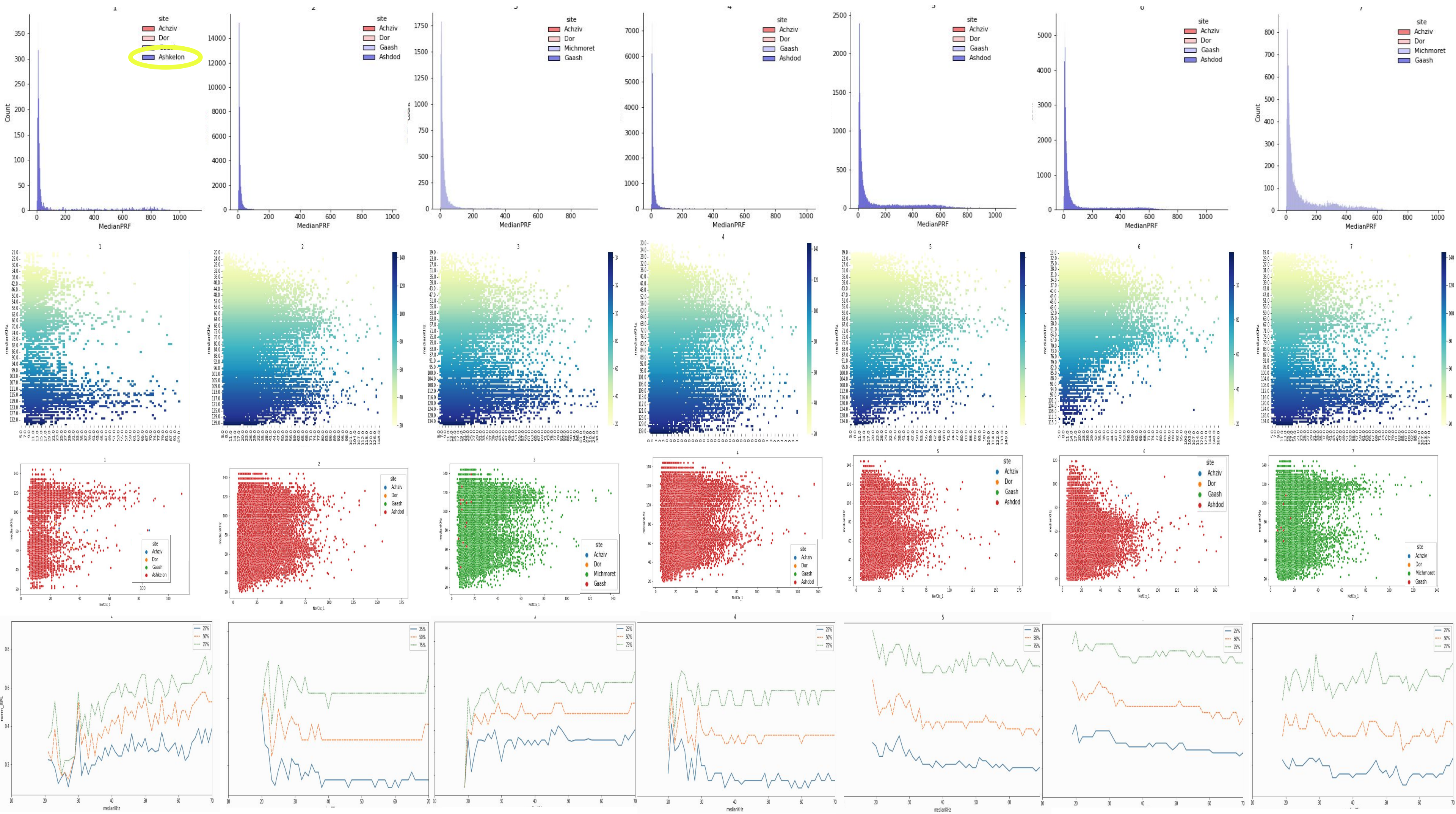
How to examine the clusters? Cluster Signature





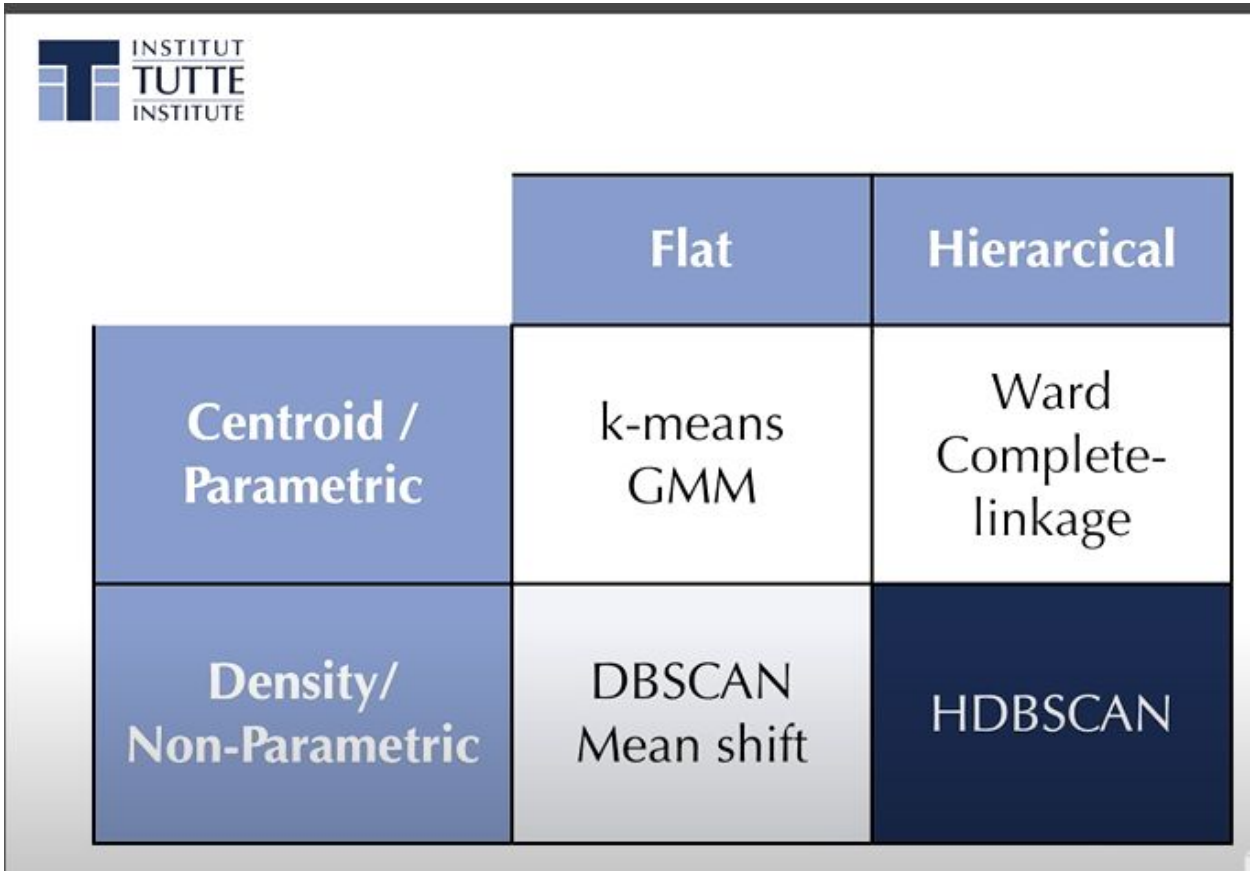






site
Achziv
Dor
Gaash
Ashkelon
Ashdod
Michmoret
Count
MedianPRF
medianKHz
NofClx_1
25%
50%
75%
norm_SPL

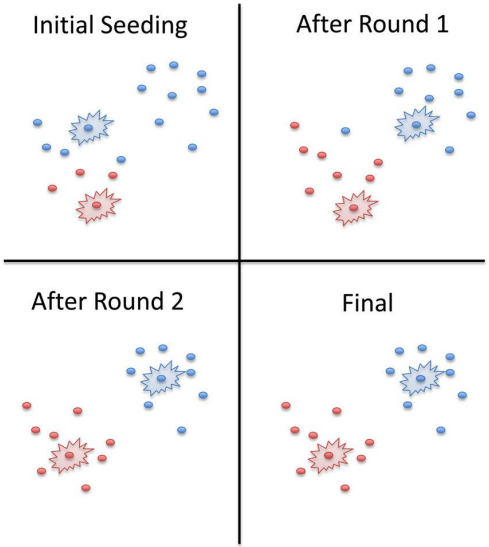
Different clustering algorithms



Different clustering algorithms

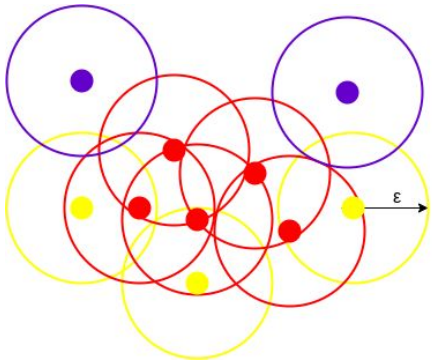
- you have to choose initial number of K
- Better for groups

K-Means

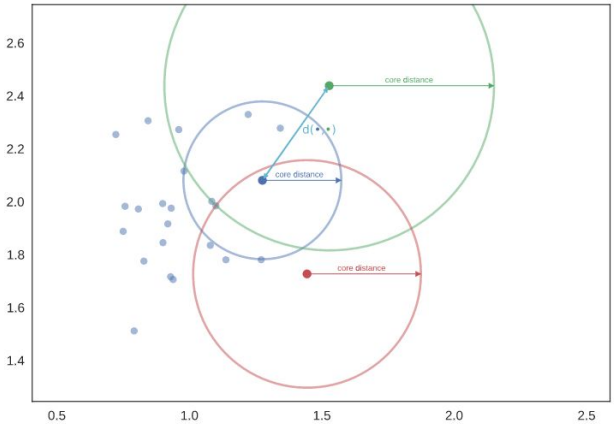


- same epsilon for all clusters
- very slow

DBSCAN

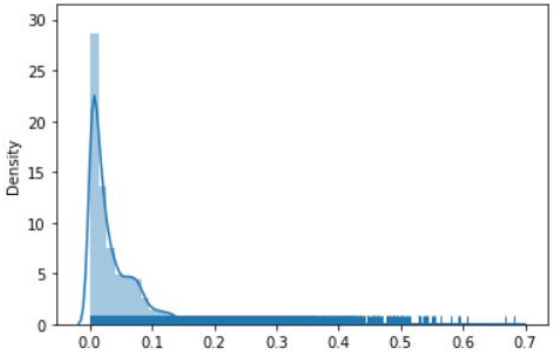


HDBSCAN

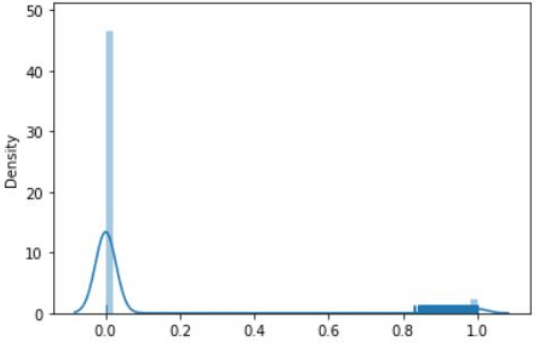


Why HDBSCAN is so nice?...

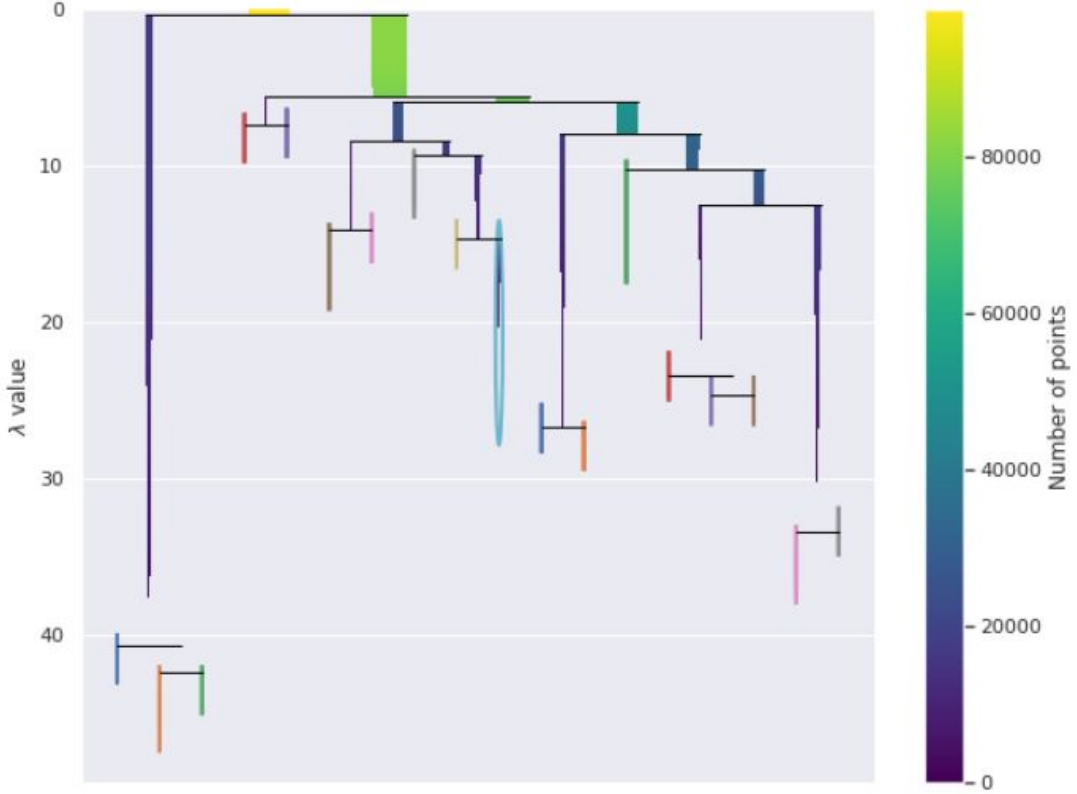
is it an outlier?



How close are points to their cluster center?

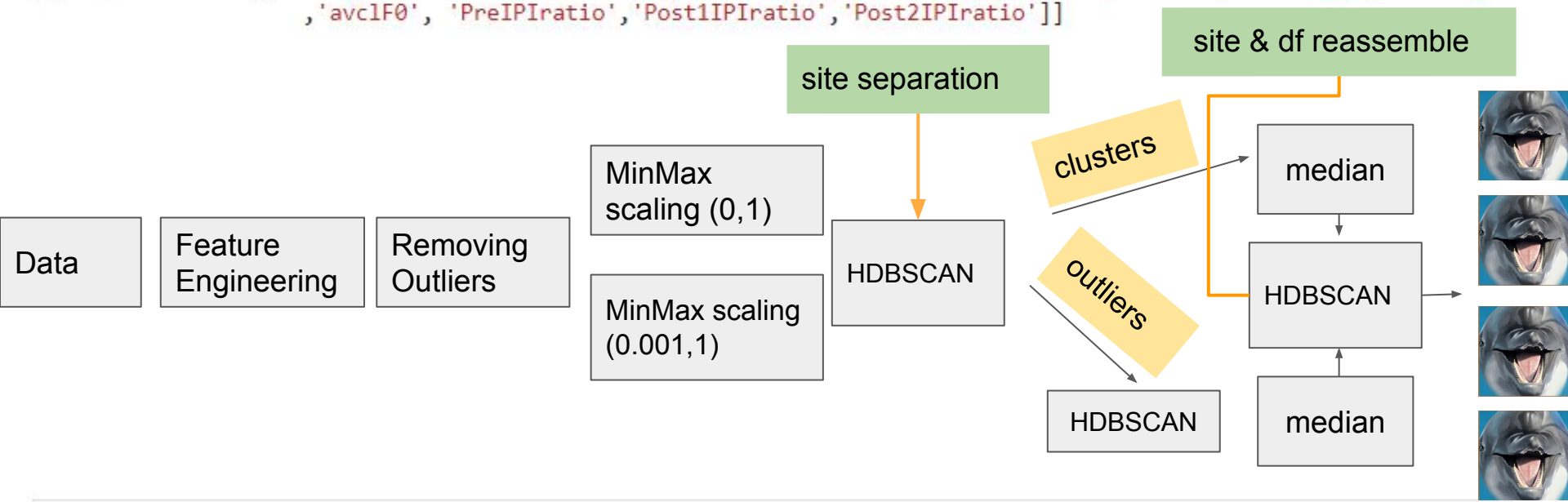


Decision Tree pruning



The final pipeline! (for now..)

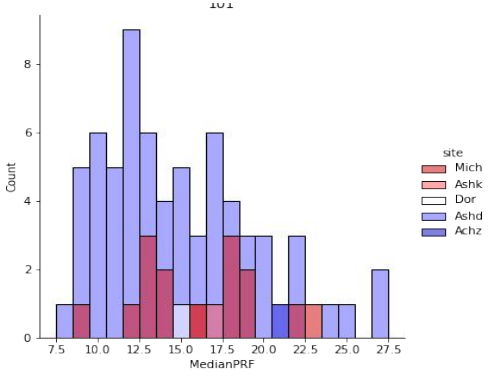
```
f_pod_model = all_median[['medianKHz','MedianPRF','MaxICI_us_1','MinICI_us_1','midpointICI','NofClx_1', 'avSPL_1'
                          ,'avclF0', 'PreIPIratio','Post1IPIratio','Post2IPIratio']]
```

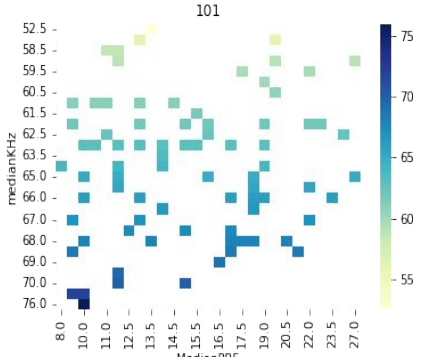


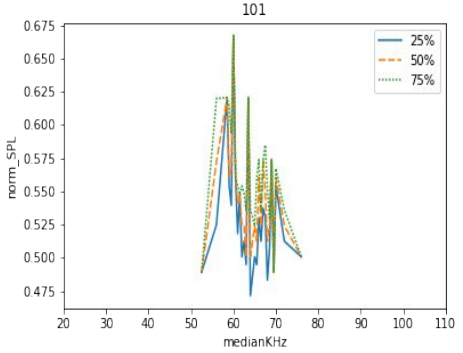

```
clusterer = hdbscan.HDBSCAN(algorithm='best',
                            approx_min_span_tree=True,
                           gen_min_span_tree=False, leaf_size=80,metric='euclidean',
                           min_samples=1,min_cluster_size=20,cluster_selection_method = 'leaf',alpha=1.0,cluster_selection_epsilon=0.01
                           ).fit(Clus)#, min_samples  = 2, p=None,min_cluster_size=30,,cluster_selection_epsilon=0.2
```

Cluster comparison

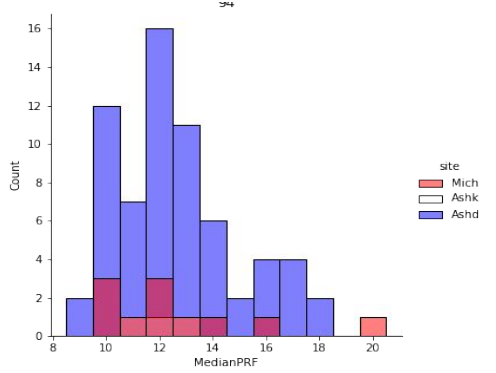
101

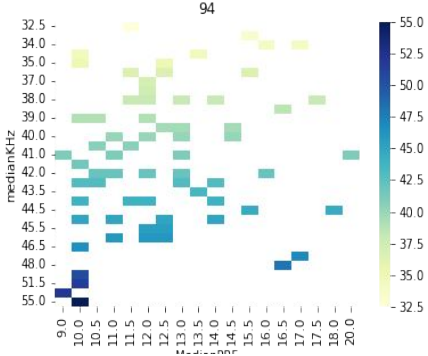


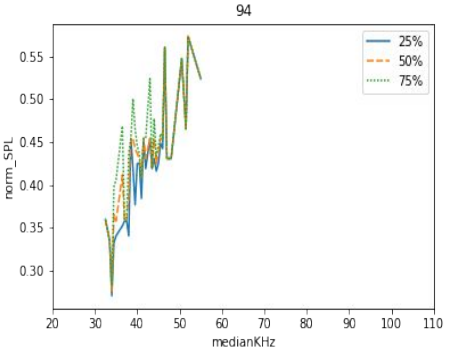




94







Whats next? Data is still collected

- Validation of promising clusters -density and difference from other clusters
- **cluster_persistence_:ndarray, shape (n_clusters,)**
- **prediction_data_:PredictionData object**
- **DBVC - both distance and density**

- Cluster analysis with experts
- Recluster as needed
- Analysis
- Classifier

The rest are in the notebook..