DBScan Clustering

- Density based approach – does not partition
- How it works.

  1 - Pick a point

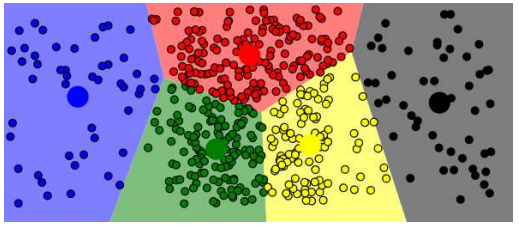
  2 - If ‘minPoints’ inside ‘eplison distance’ then point is in a cluster

  3 - Repeat for the next point

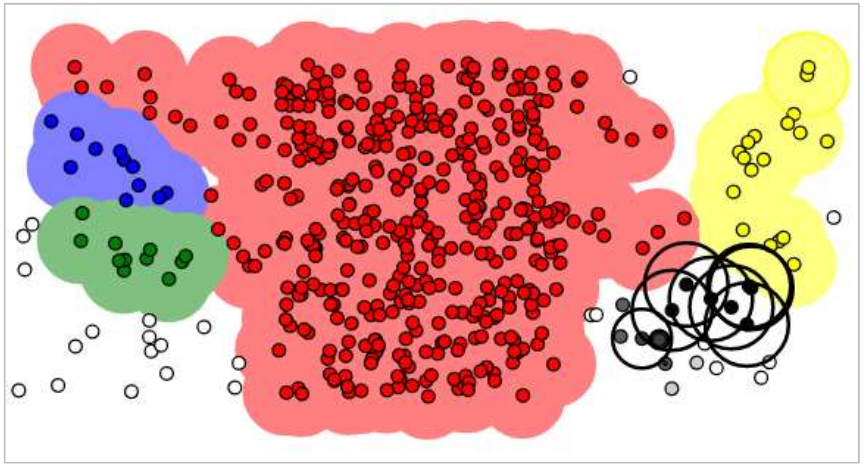
  4 – If less than ‘minPoints’ then it’s noise

- Demo: https://www.naftaliharris.com/blog/visualizing-dbscan-clustering/

K-Means



DBScan



- Advantages:

  Doesn’t assume globular clusters

  Stable – largely deterministic

  Fast on large datasets

- Challenges :

  Have to try different parameters to find best clusters (still difficult)

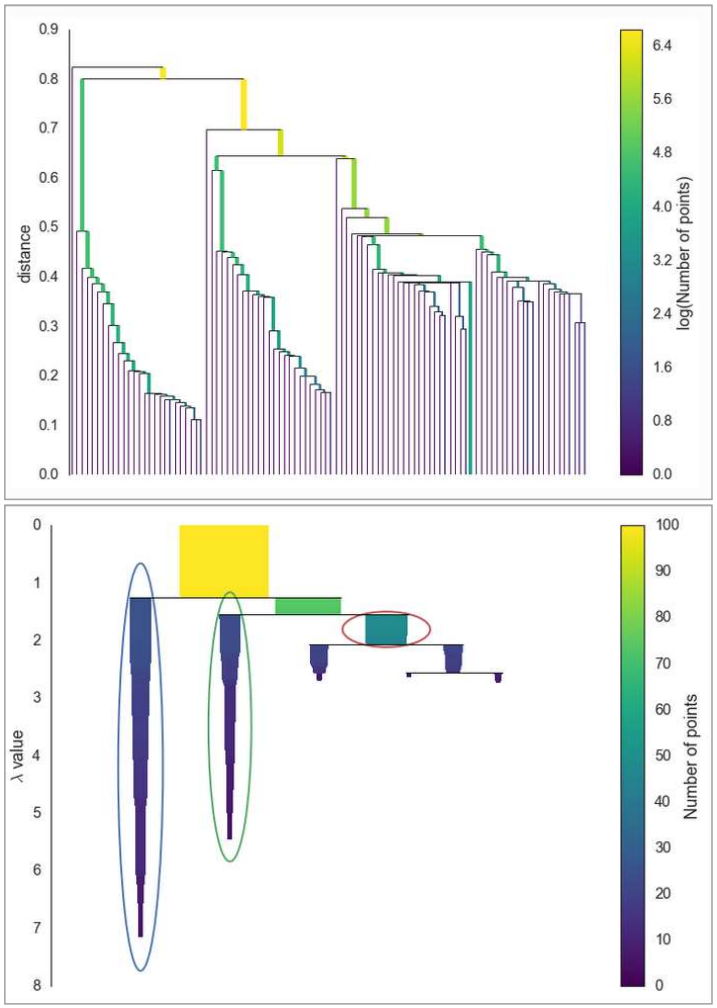
  Only searches for clusters of a certain density (not variable density)

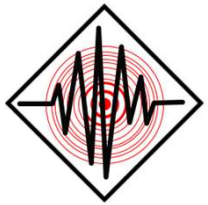
HDBScan Clustering

- Discovering clusters of varying density
- How it works.
  1. Transforms the space (increases distance in less dense areas)
  2. Performs ‘single linkage clustering’ on transformed space
  3. Builds a cluster hierarchy (dendrogram tree)
  4. Condenses the cluster tree down to min_cluster_size (cutting the tree at varying heights allows it to pick out varying density clusters based on cluster stability)
- Advantages: (DBScan advantages, plus…

  Variable density clusters

  One intuitive parameter min_cluster_size
  (What’s the minimum cluster size I care about?)





HDBScan In Action

```
import hdbscan

geo_cols = ["geo_level_1_id", "geo_level_2_id", "geo_level_3_id"]
xdata = df[geo_cols].values
print(xdata)

# create clusterer
clusterer = hdbscan.HDBSCAN(
    min_cluster_size=200,
    algorithm="best",
    alpha=1.0,
    approx_min_span_tree=True,
    gen_min_span_tree=False,
    leaf_size=40,
    # memory=Memory(cachedir=None),
    metric="euclidean",
)
clusterer.fit(xdata)
df["cluster"] = clusterer.labels_
print("Clusters: ", df["cluster"].value_counts())
df = pd.get_dummies(df, columns=["cluster"])
####
```

461 Clusters:

ID	POPULATION	
-1	145649	(-1 is noise)
40	2077	
137	1840	
235	1476	
182	1232	
210	1194	
...		
370	204	
185	203	
43	202	
260	201	
371	201	
59	200	

- LB increase .0006 (without re-tuning)