МИНОБРНАУКИ РОССИИ

САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ

ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ

«ЛЭТИ» ИМ. В. И. УЛЬЯНОВА (ЛЕНИНА)

Кафедра МО ЭВМ

ОТЧЕТ

по лабораторной работе №6

по дисциплине «Машинное обучение»

Студентка гр. 6307 _____ Кичерова А. Д.

Преподаватель _____ Жангиров Т. Р.

Санкт-Петербург

2020

Цель работы:

Ознакомиться с методами кластеризации модуля Sklearn.

Ход работы:

1. Был загружен датасет CC_General.csv

```
data = pd.read_csv('CC GENERAL.csv').iloc[:,1:].dropna()
data
```

executed in 54ms, finished 12:35:16 2020-12-21

	BALANCE	BALANCE_FREQUENCY	PURCHASES	ONEOFF_PURCHASES	INSTALLMENTS_PURCHASES	CASH_ADVANCE	PURCHASES_FREQUENCY	O
0	40.900749	0.818182	95.40	0.00	95.40	0.000000	0.166667	
1	3202.467416	0.909091	0.00	0.00	0.00	6442.945483	0.000000	
2	2495.148862	1.000000	773.17	773.17	0.00	0.000000	1.000000	
4	817.714335	1.000000	16.00	16.00	0.00	0.000000	0.083333	
5	1809.828751	1.000000	1333.28	0.00	1333.28	0.000000	0.666667	
...	...	...	...	...	...	...	...	...
8943	5.871712	0.500000	20.90	20.90	0.00	0.000000	0.166667	
8945	28.493517	1.000000	291.12	0.00	291.12	0.000000	1.000000	
8947	23.398673	0.833333	144.40	0.00	144.40	0.000000	0.833333	
8948	13.457564	0.833333	0.00	0.00	0.00	36.558778	0.000000	
8949	372.708075	0.666667	1093.25	1093.25	0.00	127.040008	0.666667	

8636 rows × 17 columns

2. Была проведена нормировка данных, так как разные признаки лежат в разных шкалах.

```
data = np.array(data, dtype='float')
min_max_scaler = preprocessing.StandardScaler()
scaled_data = min_max_scaler.fit_transform(data)
```

executed in 16ms, finished 12:35:42 2020-12-21

3. Была проведена кластеризация DBSCAN с параметрами по умолчанию:

```
clustering = DBSCAN().fit(scaled_data)

print(set(clustering.labels_))
print(len(set(clustering.labels_)) - 1)
print(list(clustering.labels_).count(-1) / len(list(clustering.labels_)))
```

executed in 4.06s, finished 11:13:01 2020-12-21

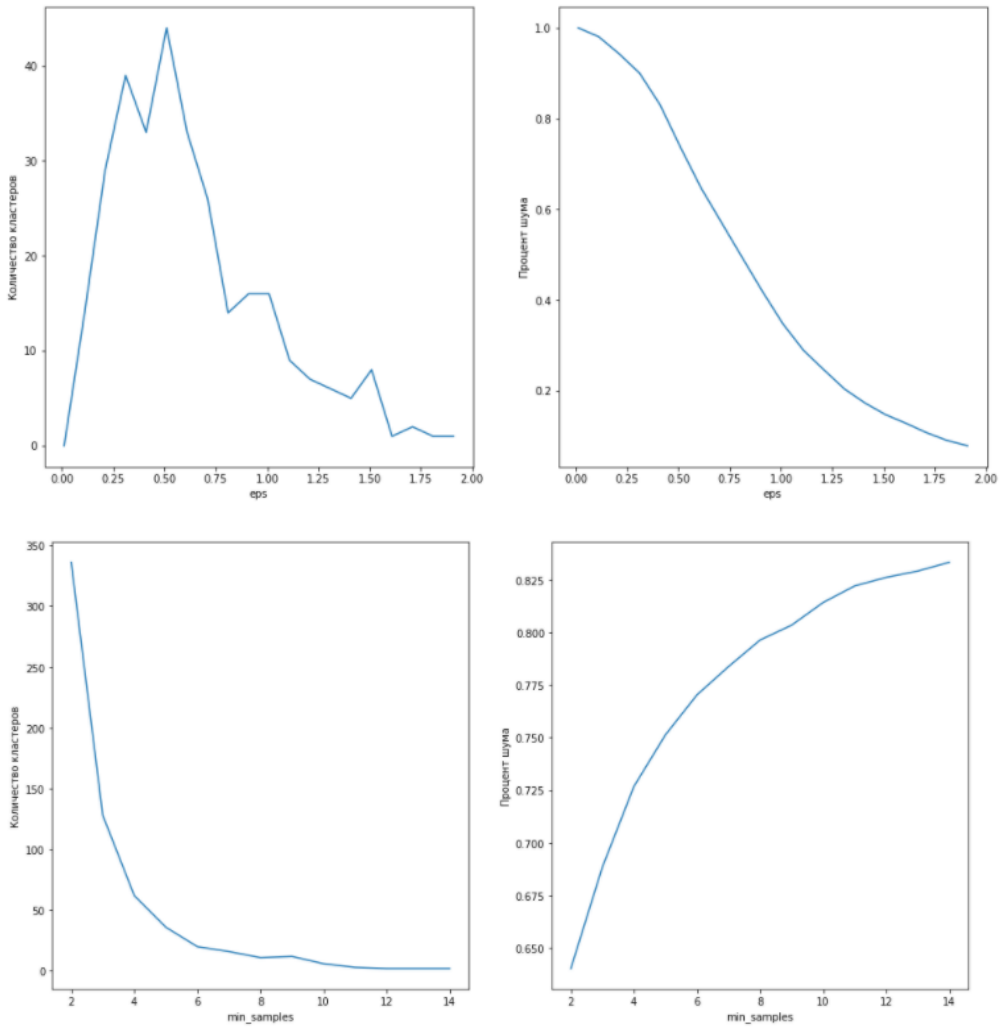
```
{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33,
34, 35, -1}
36
0.7512737378415933
```

Опишем параметры DBSCAN:

- eps: максимальное расстояние между двумя соседями;
- min_samples: число соседей в окрестности точки, необходимое для того, чтобы она считалась базовой (точка входит в подсчет);
- metric: метрика, которая используется для вычисления расстояния между соседями;

- metric_params: дополнительные аргументы ключевого слова для метрической функции (по умолчанию их нет);
- algorithm: алгоритм, который используется для нахождения ближних соседей для вычисления точечных расстояний;
- leaf_size: размер листа;
- p: степени метрики Минковского, которая будет использоваться для вычисления расстояния между точками;
- n_jobs: количество параллельных потоков для запуска.

4. Построены графики количества кластеров и процента не кластеризованных наблюдений в зависимости от максимальной рассматриваемой дистанции, а также от минимального значения точек для образования кластера.



5. Определили значения параметров для кластеризации, при котором количество кластеров получается от 5 до 7 и процент не кластеризованных наблюдений не превышает 12%.

```
eps_range = np.arange(0.1, 3, 0.4)
min_samples_range = np.arange(1, 15, 2)

dbscan_df = pd.DataFrame(columns=['min samples', 'eps', 'clusters', 'noise'])

for min_samples in min_samples_range:
    for eps in eps_range:
        clustering = DBSCAN(eps=eps, min_samples=min_samples).fit(scaled_data)
        n_clusters = len(set(clustering.labels_)) - 1
        percent_noises = list(clustering.labels_).count(-1) / len(list(clustering.labels_))
        dbscan_df = dbscan_df.append(
            {'min samples': min_samples, 'eps': eps, 'clusters': n_clusters, 'noise': percent_noises},
            ignore_index=True)


new_data = dbscan_df[(dbscan_df.clusters >= 5) & (dbscan_df.clusters <= 7) & (dbscan_df.noise <= 0.12)]
new_data
```

executed in 2m 46s, finished 12:42:43 2020-12-21

	min samples	eps	clusters	noise
15	3.0	2.9	5.0	0.022233

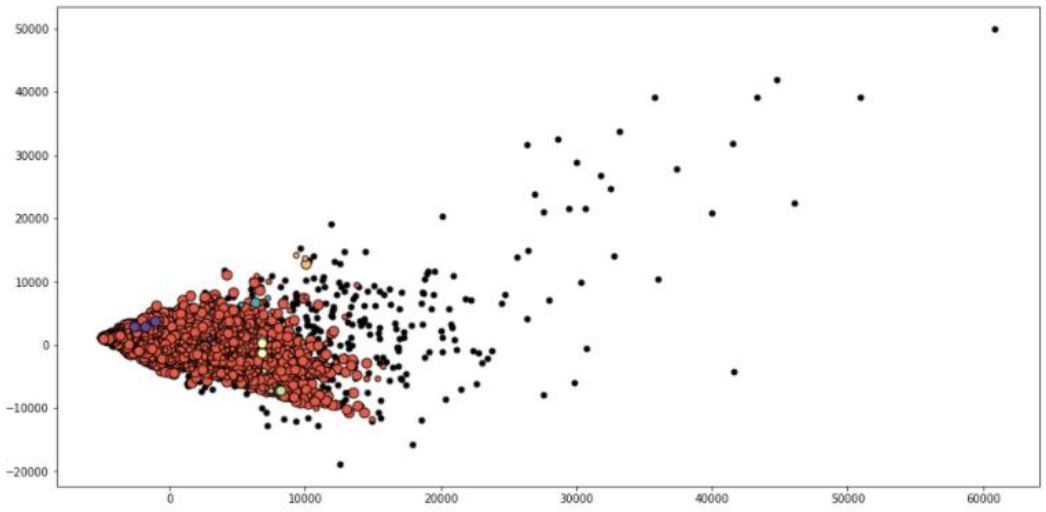
6. Понизили размерность данных до 2 используя PCA

```
pca = PCA(n_components=2)
reduced_data = pca.fit_transform(data)
pca.explained_variance_ratio_

clustering = DBSCAN(eps=2, min_samples=3, n_jobs=-1).fit(scaled_data)
```

executed in 1.20s, finished 13:19:24 2020-12-21

Визуализировали результаты кластеризации, проведенной в прошлом пункте



7. Изучили параметры OPTICS

- min_samples – минимально число точек в окрестности точек, при котором она считается основной;

- max_eps – максимальное расстояние, допускающее сходство между точками;
- metric – метрика для вычисления расстояния между точками;
- p – параметр метрики Минковского (при p = 1 равносильно использованию manhattan_distance, при p = 2 равносильно евклидовому расстоянию);
- metric_params – дополнительные параметры метрики;
- cluster_method – метод извлечения кластеров на основании вычислительной достижимости;
- eps – максимальная дистанция, при которой точки являются соседями;
- xi – минимальная крутизна на графике достижимости, показывающая границу кластера;
- predecessor_correction- коррекция кластеров по предшественникам;
- min_cluster_size – минимальное количество точек в кластере;
- algorithm – алгоритм поиска ближайших соседей;
- leaf_size – размер листа;
- n_jobs – число параллельно выполняемых потоков.

8. Нашли параметры OPTICS при которых получились результаты, схожие с результатами DBSCAN из пункта 5

```
clustering = OPTICS(max_eps=1.7, min_samples=4, cluster_method='dbscan').fit(scaled_data)

labels = clustering.labels_

n_clusters_ = len(set(labels)) - (1 if -1 in labels else 0)
n_noise_ = list(labels).count(-1)

print(n_clusters_)
```

executed in 11.8s, finished 13:24:01 2020-12-21

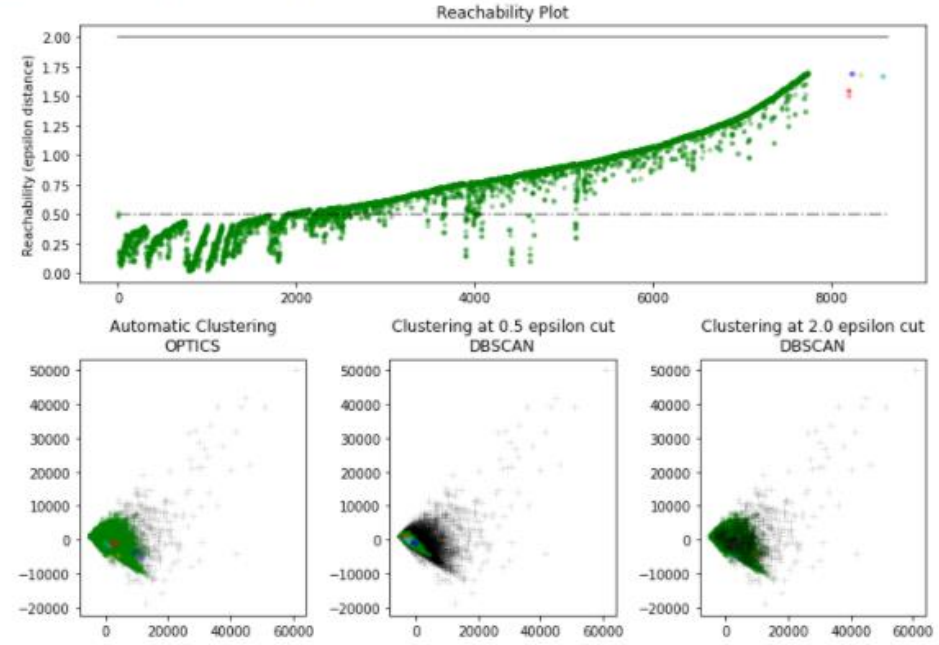
```
5
```

OPTICS в отличие от DBSCAN сохраняет иерархию кластеров для разных радиусов внутри окрестности. Также параметр eps не обязателен. Он может просто быть установлен в максимальное возможное значение. Однако при доступности пространственного индекса он влияет на сложность вычислений. OPTICS отличается от

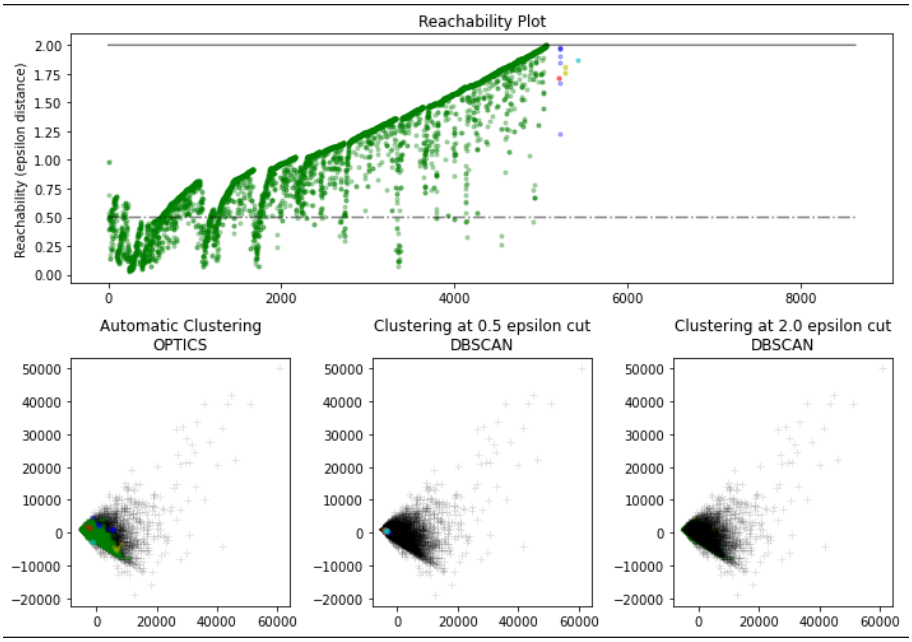
DBSCAN тем, что этот параметр не учитывается, если eps и может влиять, то только тем, что задаёт максимальное значение.

9. Были визуализирован результат и построен график достижимости

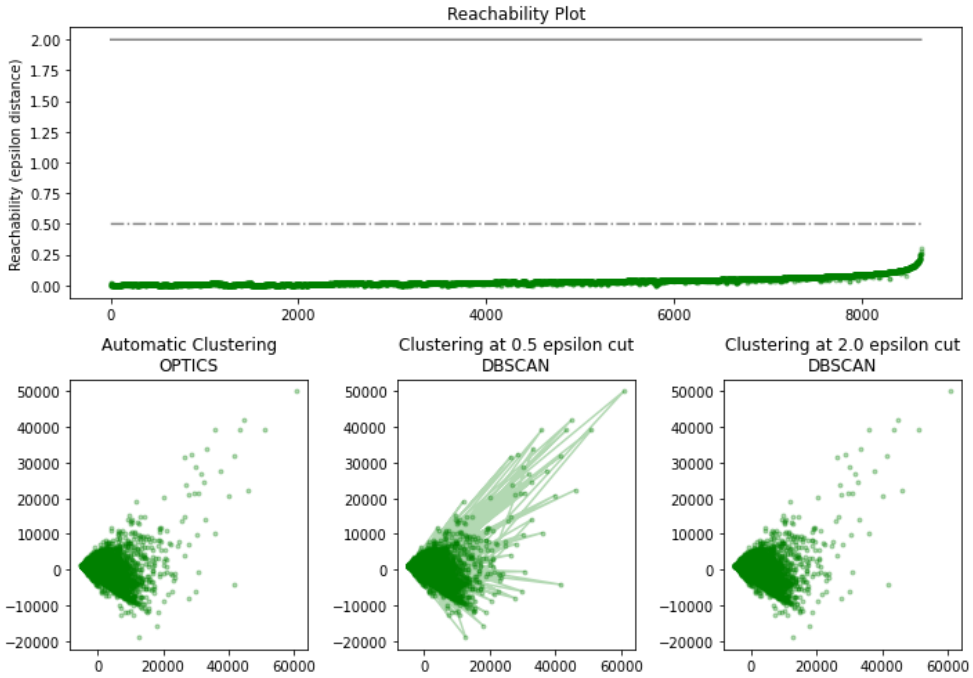
```
show_optics(clustering)
```
executed in 508ms, finished 13:26:58 2020-12-21



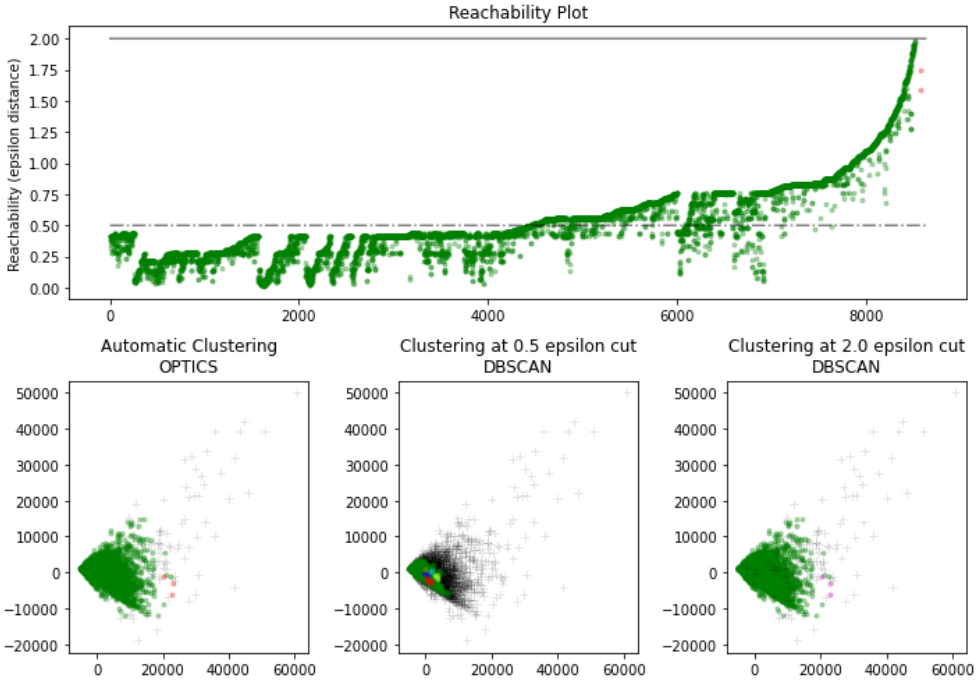
10. Исследовали работу OPTICS с использованием различных метрик cityblock



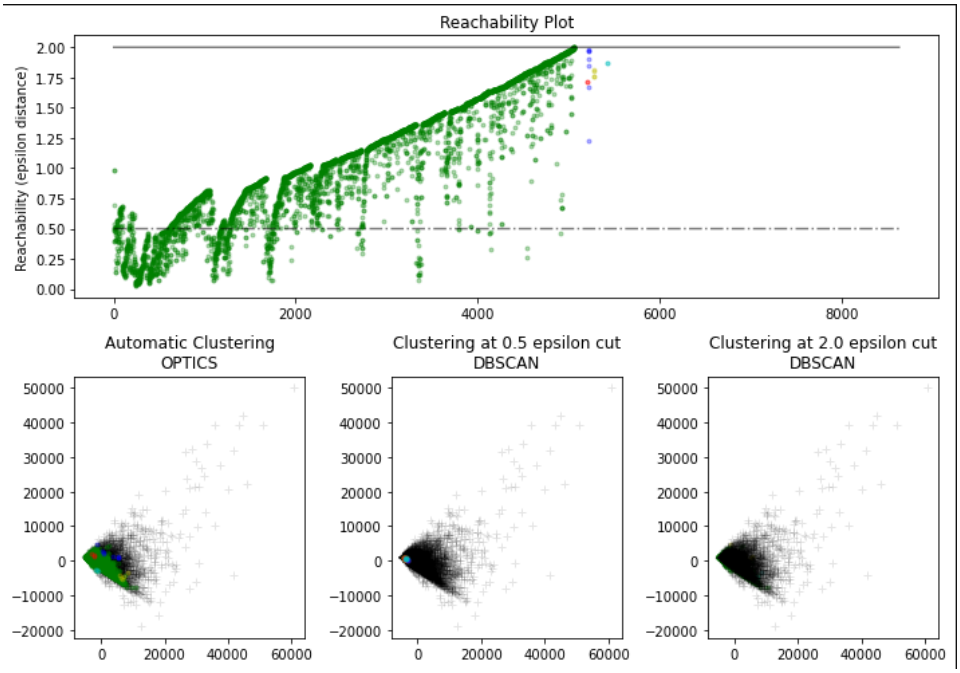
cosine



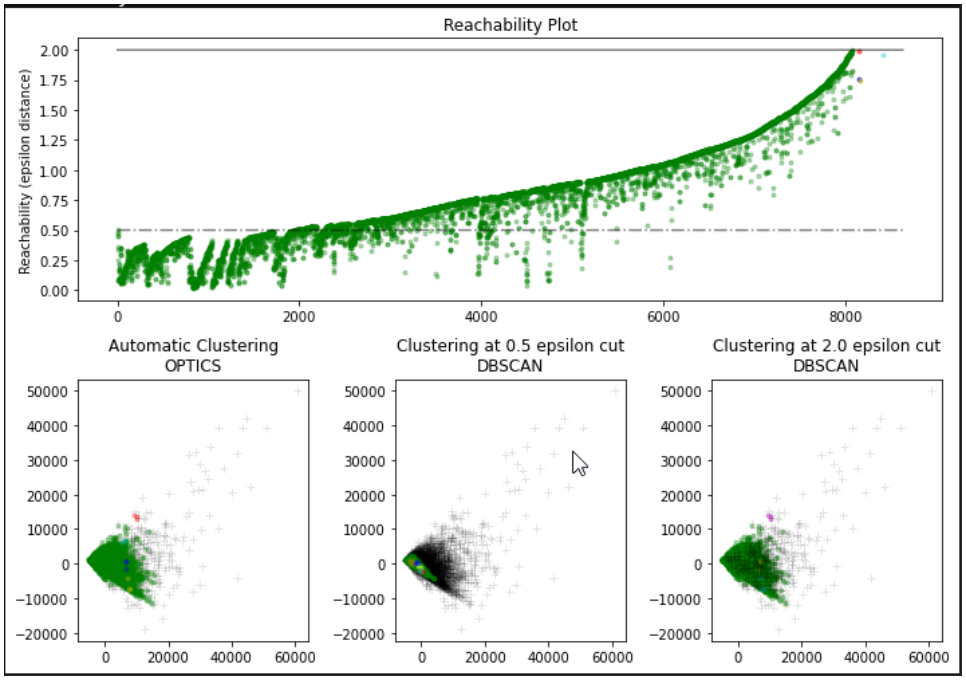
chebyshev



l1


Reachability Plot
Reachability (epsilon distance)
Automatic Clustering
OPTICS
Clustering at 0.5 epsilon cut
DBSCAN
Clustering at 2.0 epsilon cut
DBSCAN

l2


Reachability Plot
Reachability (epsilon distance)
Automatic Clustering
OPTICS
Clustering at 0.5 epsilon cut
DBSCAN
Clustering at 2.0 epsilon cut
DBSCAN