МИНОБРНАУКИ РОССИИ
САНКТ-ПЕТЕРБУРГСКИЙ ГОСУДАРСТВЕННЫЙ
ЭЛЕКТРОТЕХНИЧЕСКИЙ УНИВЕРСИТЕТ
«ЛЭТИ» ИМ. В.И. УЛЬЯНОВА (ЛЕНИНА)
Кафедра МОЭВМ

ОТЧЕТ

по лабораторной работе №6

по дисциплине «Машинное обучение»

Тема: Кластеризация (DBSCAN, OPTICS)

Студент гр. 6307 _____ Золотухин М. А.

Преподаватель _____ Жангиров Т. Р.

Санкт-Петербург

2020

Загрузка данных¶

```
import pandas as pd
import numpy as np

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

DBSCAN

Проведем кластеризацию методов k-средних

```
from sklearn.cluster import KMeans

k_means = KMeans(init='k-means++', n_clusters=3, n_init=15)
k_means.fit(data)
```

```
KMeans(n_clusters=3, n_init=15)
```

Так как разные признаки лежат в разных шкалах, то стандартизируем данные

```
from sklearn import preprocessing

data = np.array(data, dtype='float')

min_max_scaler = preprocessing.StandardScaler()
scaled_data = min_max_scaler.fit_transform(data)
```

Проведем кластеризацию методов DBSCAN при параметрах по умолчанию. Выведем метки кластеров, количество кластеров, а также процент наблюдений, которые кластеризовать не удалось

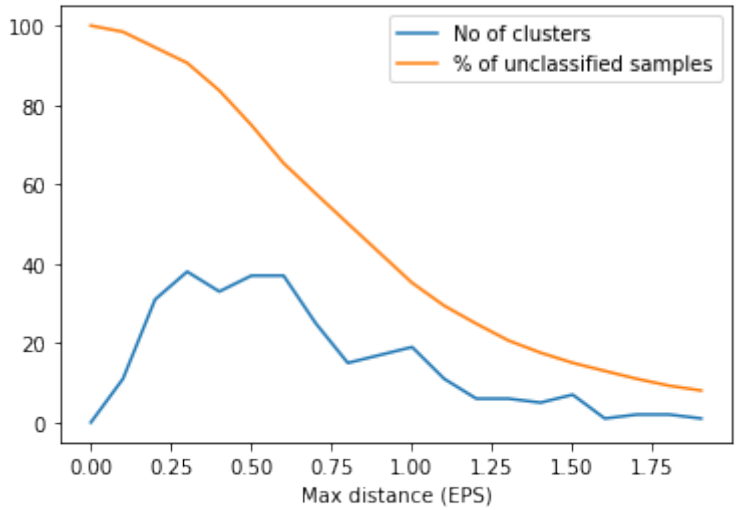
```
from sklearn.cluster import DBSCAN

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_)))
```

```
{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
```

Параметры принимаемые DBSCAN:

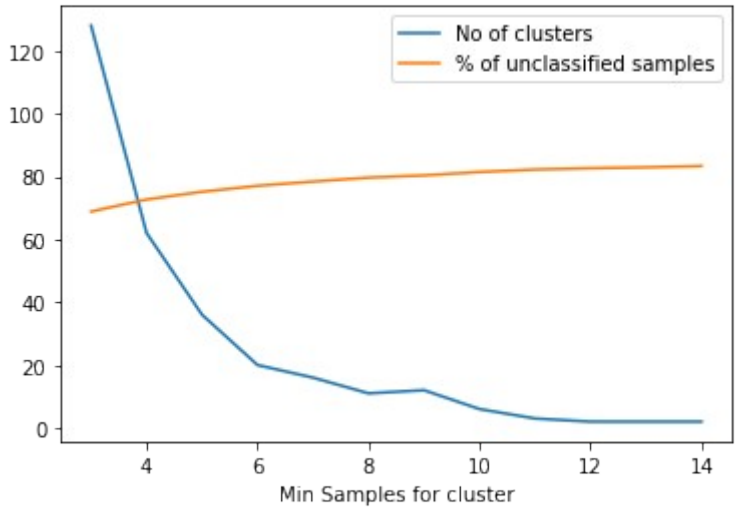
- eps - максимальная дистанция между двумя точками (сэмплами), допустимая для того, чтобы можно было бы сказать, что они соседи
- min_samples - минимальное количество точек соседей, включая её саму, необходимое для того, чтобы эту точку можно было бы назвать `core` (т.е. внутри кластера)
- metrics - метрика для вычисление расстояний между двумя точками (по-умолчанию, расстояние в Евклидовом пространстве)
- metric_params - параметры для метрики
- algorithm - алгоритм для вычисления ближайших соседей модулем NearestNeighbors
- leaf_size - количество листьев для некоторых алгоритмов в вышеказанном параметре
- p - мощность метрики Миковского (если таковая указана в metrics)
- n_jobs - количество паралельных работ для расчетов

Построим график количества кластеров и процента не кластеризованных наблюдений в зависимости от максимальной рассматриваемой дистанции между наблюдениями (EPS). Минимальное значение количества точек образующих кластер оставим по умолчанию



Построим график количества кластеров и процента не кластеризованных наблюдений в зависимости от минимального значения количества точек, образующих кластер.

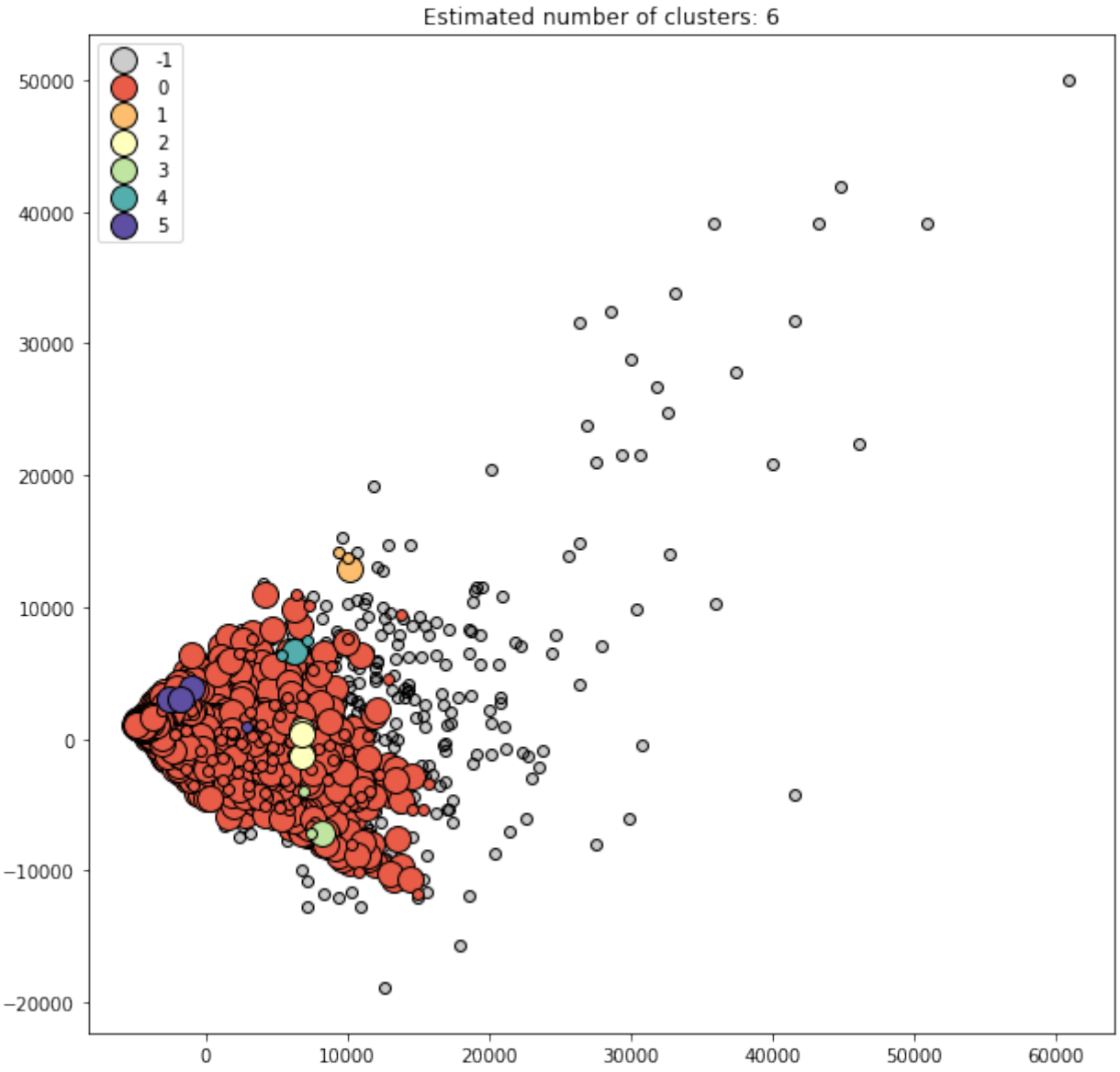
Максимальную рассматриваемую дистанцию между наблюдениями оставим по умолчанию



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

```
clustering = DBSCAN(eps=2, min_samples=3).fit(scaled_data)
print(f'Clusters: {len(set(clustering.labels_)) - 1}')
print(f'% of u.s.: {100 * list(clustering.labels_).count(-1)
/len(list(clustering.labels_))}')
Clusters: 6
% of u.s.: 6.287633163501621
```

Понизим размерность данных до 2 при используя метод главных компонент. Визуализируйте результаты кластеризации полученные в пункте 6 (метки должны быть получены на данных до уменьшения размерности).



OPTICS

Опишите параметры метода OPTICS, а также какими атрибутами он обладает

Параметры

- min_samples - минимальное количество точек в окрестности другой точки, чтобы считать её "core"-точкой;
- max_eps - максимальное расстояние между двумя точками для того, чтобы считать их в окрестности друг друга. По умолчанию - бесконечность;
- metric - метрика для вычисления расстояния;
- p - параметр для метрики Минковского;
- metric_params - дополнительные параметры метрики;

- cluster_method - метод для извлечения кластеров с импользованием вычисленных "reachability" и "ordereing". Может быть “xi” или “dbscan”;
- eps - (параметр dbscan) максимальная дистанция между точками, чтобы считать их в окресности друг друга. По-умолчанию такое же как и max_eps.
- xi - параметр для метода "xi". Опередеяет минимальную ступенчатость графика достижимости (reachability plot), которая определяет границу кластера.
- predecessor_correction - параметр для метода "xi", который позволяет корректировать кластеры в соответсвии с предшественниками.
- min_cluster_size - мимнимальный размер OPTICS кластера в количестве измерений.
- algorithm - алгоритм для вычисления ближайших соседей;
- leaf_size - размер листа дерева для BallTree или KDTree.
- n_jobs - количество параллельных работ для поиска соседей.

Атрибуты

- labels_ - метки кластеров
- reachability_ - reachability distances для каждой точки;
- ordering_ - индексы точке отсортированные в порядке кластеров;
- core*distances* - дистанции на которых каждая точка стоановится core-точкой. Для некорневых точек - бесконечность.
- predecessor_ - точка из которой была достигнута точка.
- cluster*hierarchy* - иерархия кластркров. Только для xi метода.

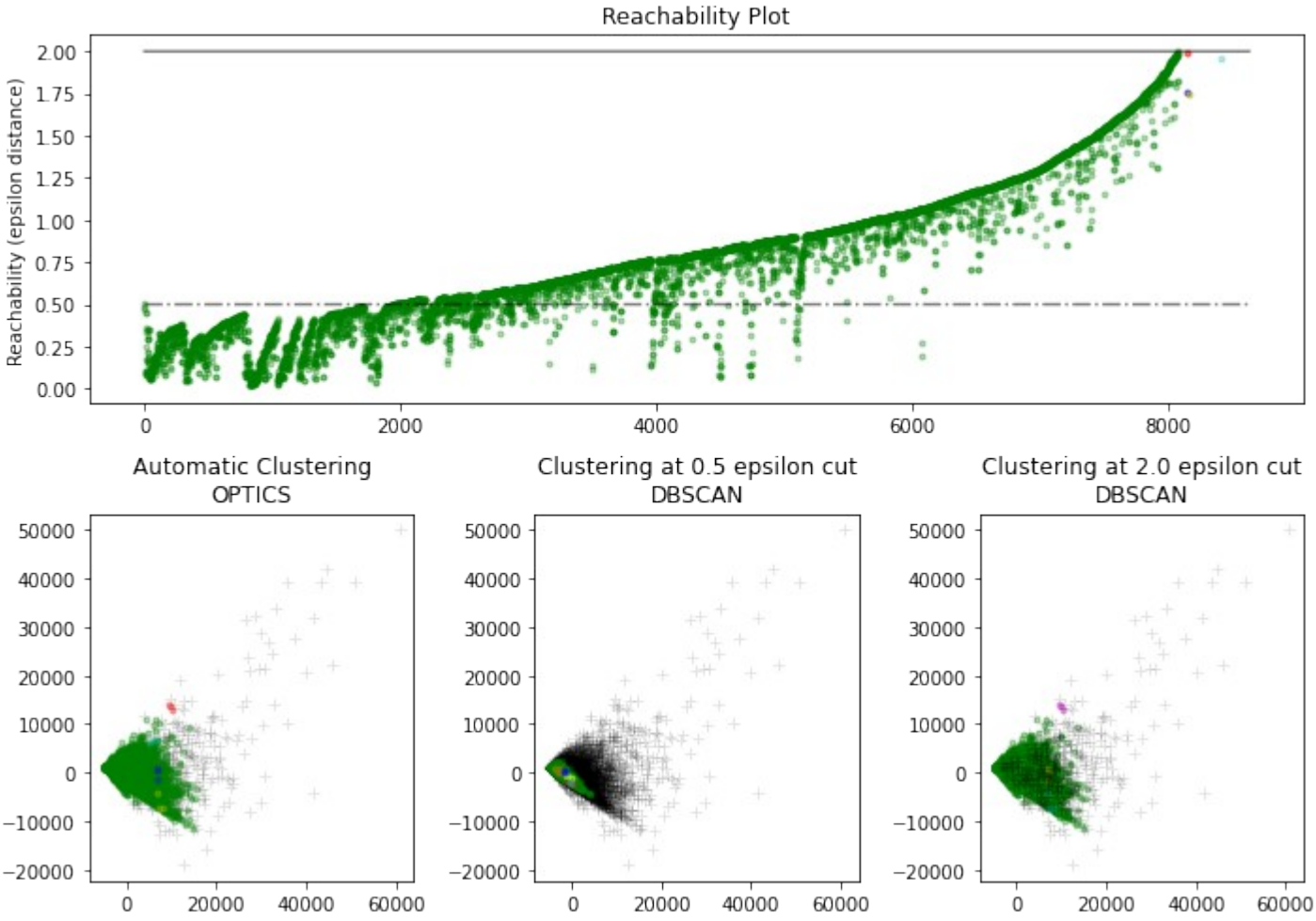
Найдите такие параметры метода OPTICS (*max_eps* и min_samples) при которых, чтобы получить результаты близкие к результатам DBSCAN из пункта 6

```
from sklearn.cluster import OPTICS

clustering = OPTICS(max_eps=2, min_samples=3, cluster_method='dbscan').fit(scaled_data)

print(set(clustering.labels_))
print(len(set(clustering.labels_)) -1)
print(100 * list(clustering.labels_).count(-1) / len(list(clustering.labels_)))
{0, 1, 2, 3, 4, 5, -1}
6
6.310792033348773
```

В чем отличия от метода OPTICS от метода DBSCAN

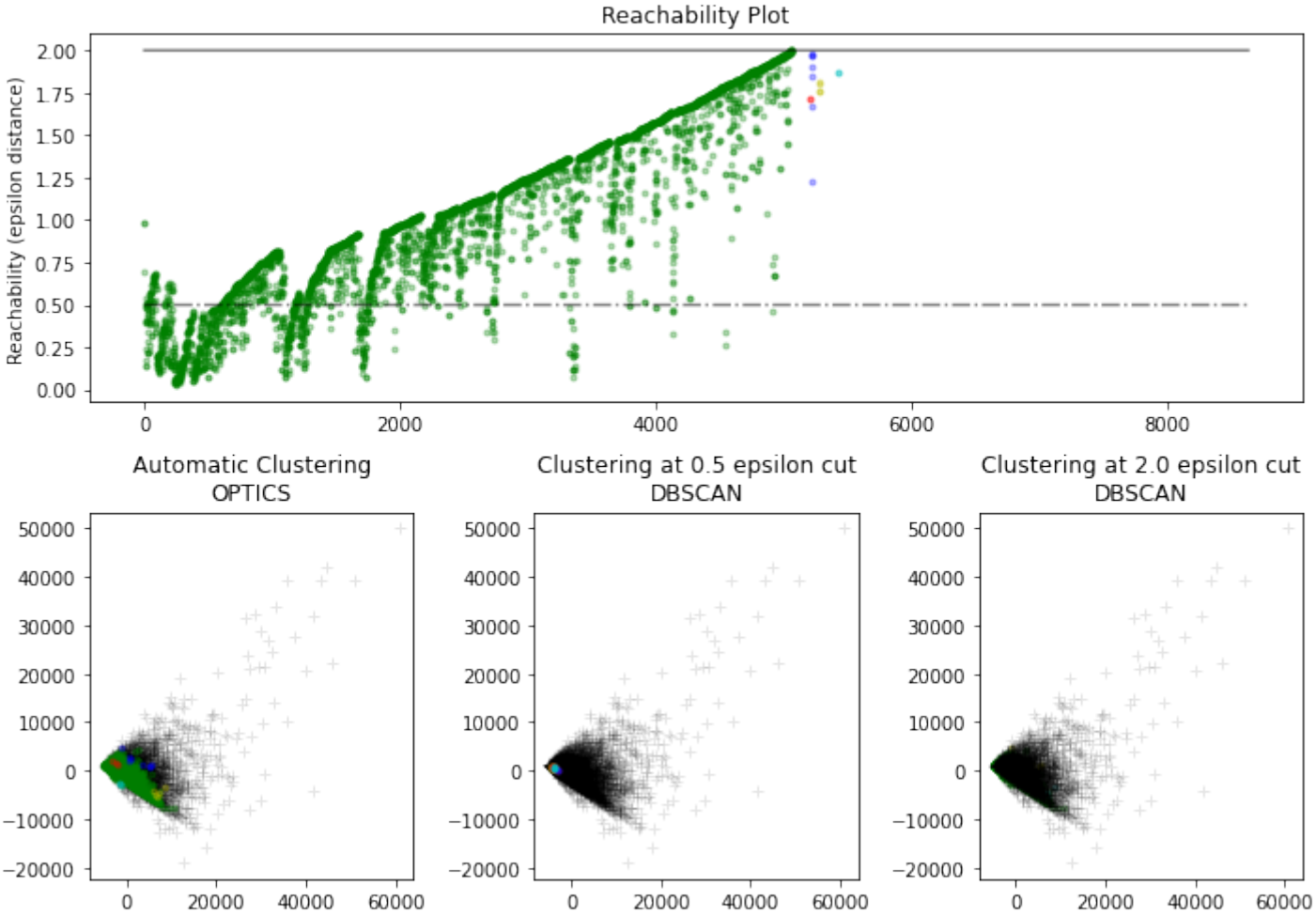
OPTICS - расширение метода DBSCAN, который использует переменную величину (eps) для окрестностей между точками. OPTICS строит график достижимости на основе показателей *reachability (*Reachability distance для двух точек - это максимум из двух: core-distance и расстояния между точками, но только в том случае, если первая точка - core. В противном случае r.d. неопределено*) и ordering*. Провалы на это графике показывают кластеры. OPTICS позволяет убрать недостаток DBSCAN - большую зависимость от параметра eps.

Визуализируйте полученный результат, а также постройте график достижимости (reachability plot).

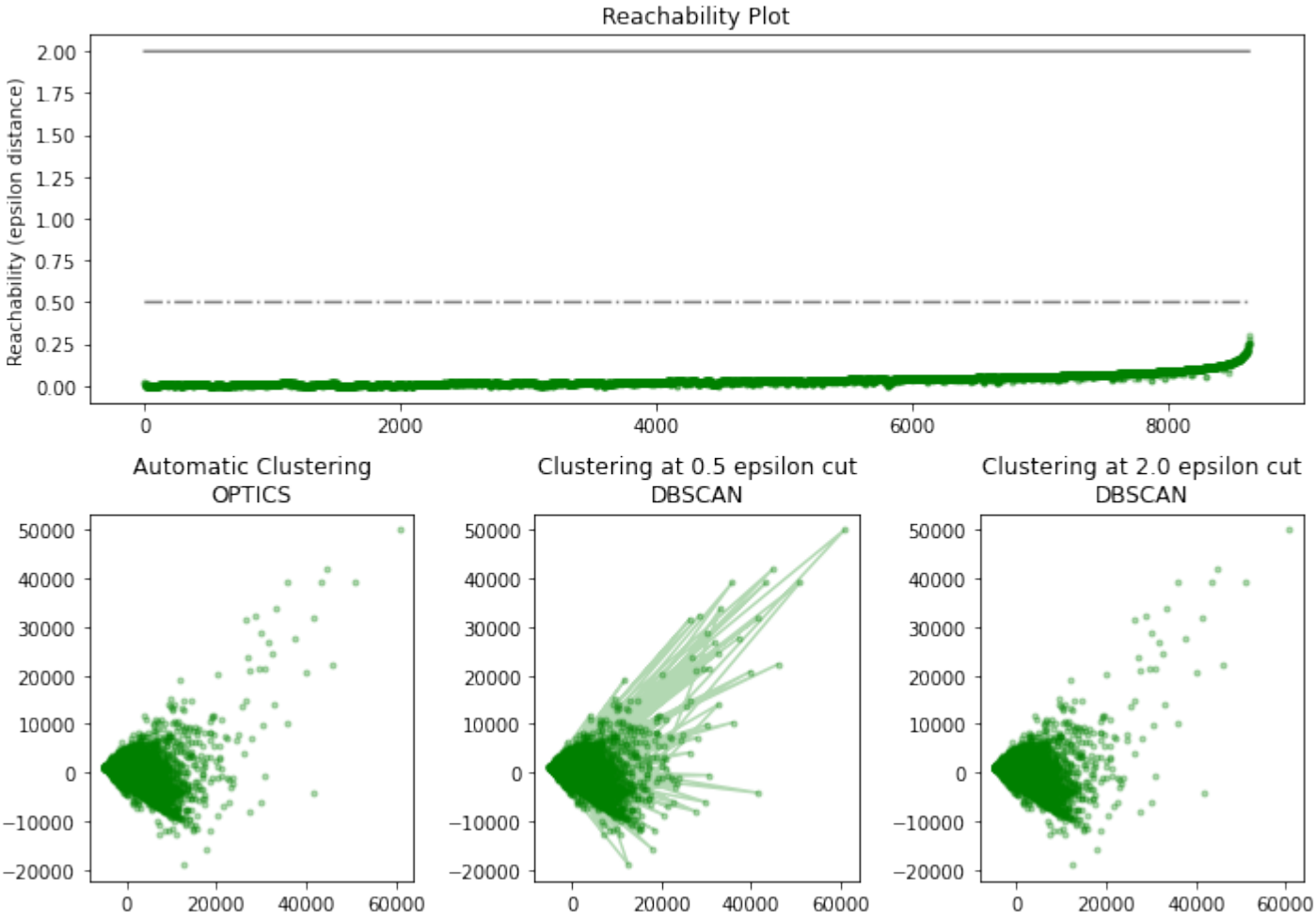


Исследуйте работу метода OPTICS с использованием различных метрик (выберите не менее 5 метрик).

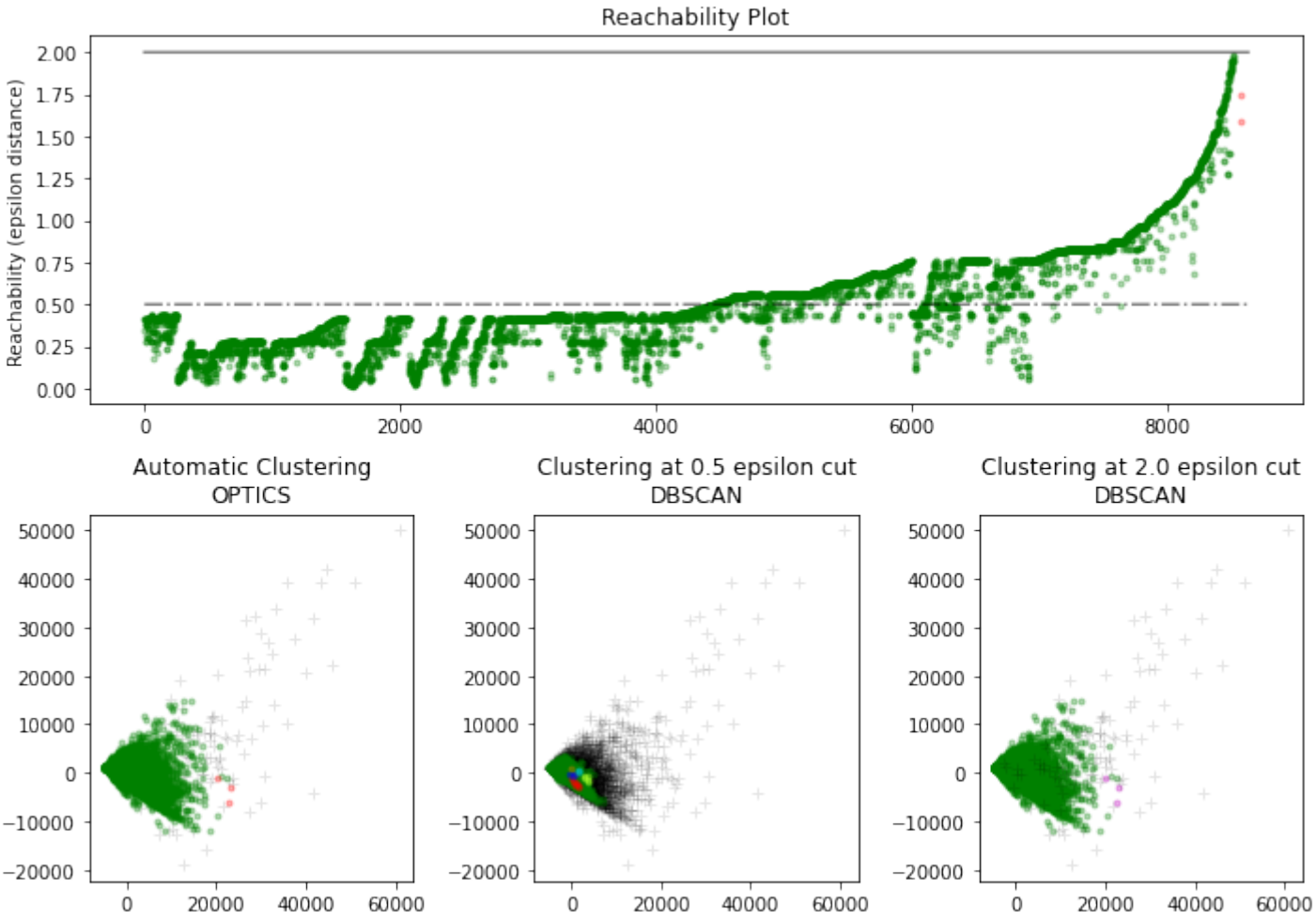
cityblock


Reachability Plot
Reachability (epsilon distance)
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
0
2000
4000
6000
8000
Automatic Clustering
OPTICS
Clustering at 0.5 epsilon cut
DBSCAN
Clustering at 2.0 epsilon cut
DBSCAN
50000
40000
30000
20000
10000
0
−10000
−20000
0
20000
40000
60000

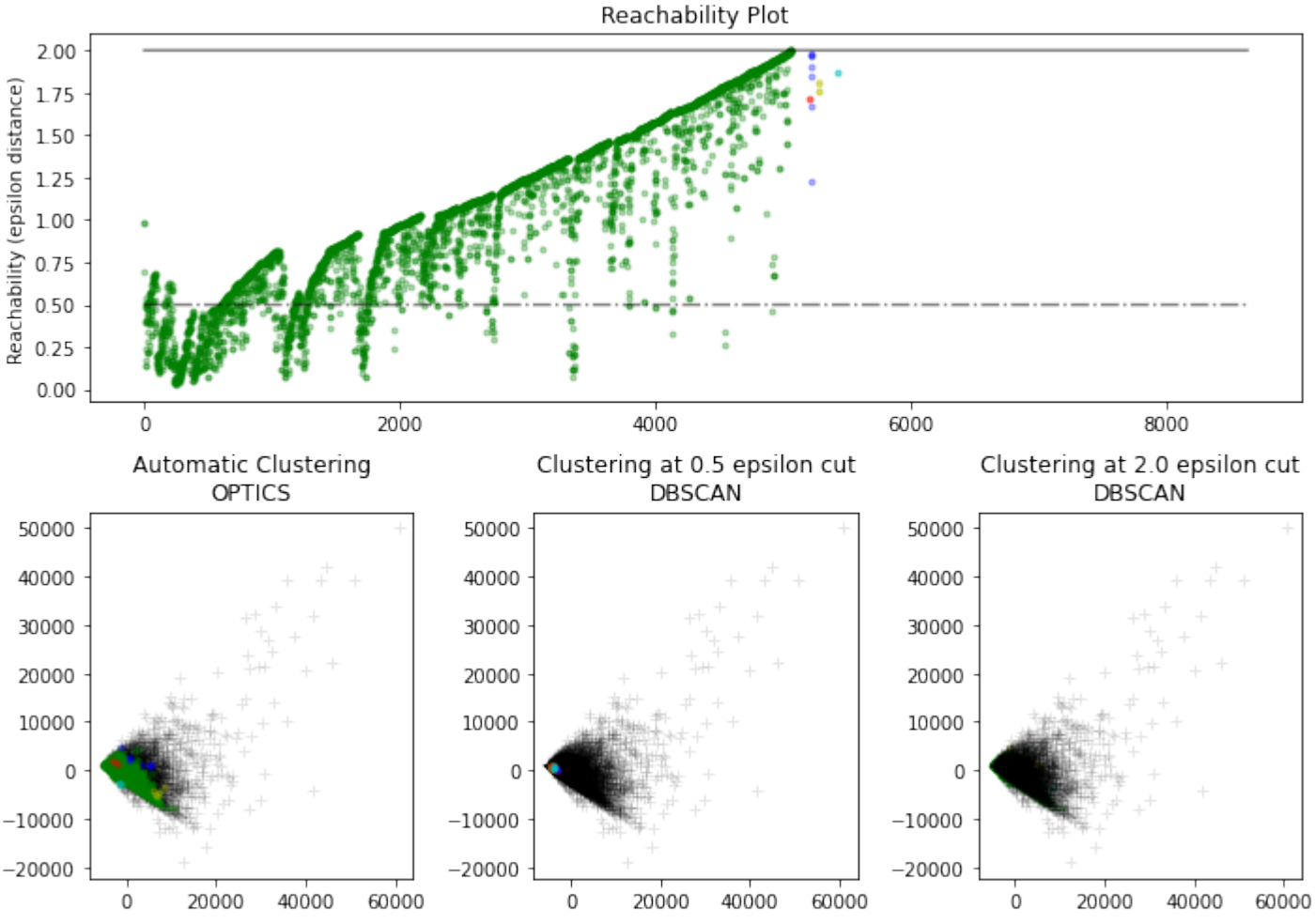
cosine


Reachability Plot
Reachability (epsilon distance)
2.00
1.75
1.50
1.25
1.00
0.75
0.50
0.25
0.00
0
2000
4000
6000
8000
Automatic Clustering
OPTICS
Clustering at 0.5 epsilon cut
DBSCAN
Clustering at 2.0 epsilon cut
DBSCAN
50000
40000
30000
20000
10000
0
−10000
−20000
0
20000
40000
60000

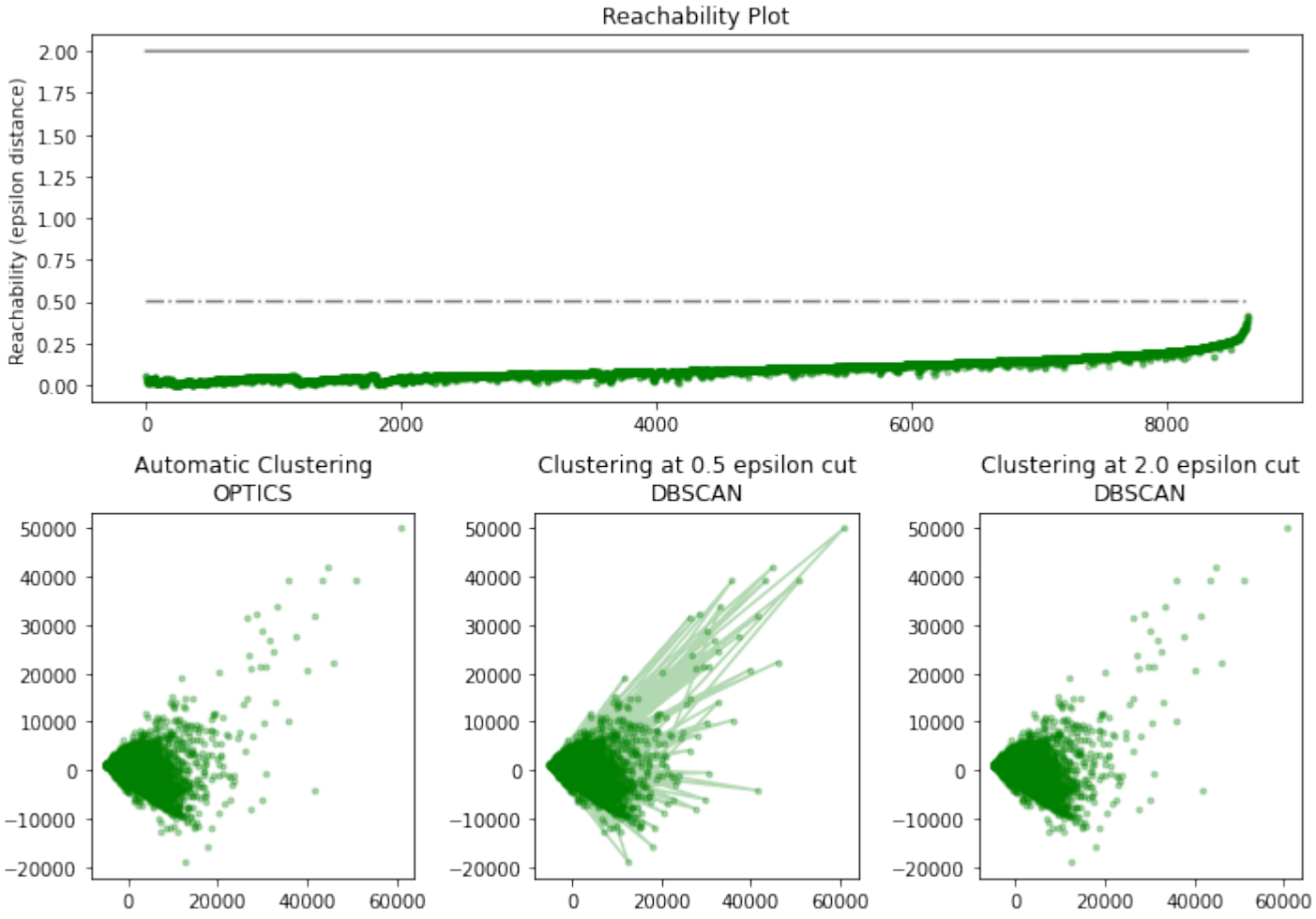
chebyshev



l1



Braycurtis



Видно, что метрики Cosine и Braycurtis — скорее всего бесполезные, так как с их использованием мы получаем один кластер со всеми данными. L1 потенциально полезная, так как образует много кластеров.