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»

Тема: Кластеризация (DBSCAN, OPTICS)

Студент гр. 6307 _____ Михайлов И. Т.

Преподаватель _____ Жангиров Т. Р.

Санкт-Петербург

2020

Цель работы:

Ознакомиться с методами кластеризации модуля Sklearn.

Загрузка данных:

Загружен датасет CC General, убран столбец с метками и откинуты наблюдения с пропущенными значениями.

	CUST_ID	BALANCE	BALANCE_FREQUENCY	PURCHASES	ONEOFF_PURCHASES	\
0	C10001	40.900749	0.818182	95.40	0.00	
1	C10002	3202.467416	0.909091	0.00	0.00	
2	C10003	2495.148862	1.000000	773.17	773.17	
4	C10005	817.714335	1.000000	16.00	16.00	
5	C10006	1809.828751	1.000000	1333.28	0.00	
...	...	...	...	...	...	
8943	C19184	5.871712	0.500000	20.90	20.90	
8945	C19186	28.493517	1.000000	291.12	0.00	
8947	C19188	23.398673	0.833333	144.40	0.00	
8948	C19189	13.457564	0.833333	0.00	0.00	
8949	C19190	372.708075	0.666667	1093.25	1093.25	

	INSTALLMENTS_PURCHASES	CASH_ADVANCE	PURCHASES_FREQUENCY	\
0	95.40	0.000000	0.166667	
1	0.00	6442.945483	0.000000	
2	0.00	0.000000	1.000000	
4	0.00	0.000000	0.083333	
5	1333.28	0.000000	0.666667	
...	...	...	...	
8943	0.00	0.000000	0.166667	
8945	291.12	0.000000	1.000000	
8947	144.40	0.000000	0.833333	
8948	0.00	36.558778	0.000000	
8949	0.00	127.040008	0.666667	

	ONEOFF_PURCHASES_FREQUENCY	PURCHASES_INSTALLMENTS_FREQUENCY	\
0	0.000000	0.083333	
1	0.000000	0.000000	
2	1.000000	0.000000	
4	0.083333	0.000000	
5	0.000000	0.583333	
...	...	...	
8943	0.166667	0.000000	
8945	0.000000	0.833333	
8947	0.000000	0.666667	
8948	0.000000	0.000000	
8949	0.666667	0.000000	

	CASH_ADVANCE_FREQUENCY	CASH_ADVANCE_TRX	PURCHASES_TRX	CREDIT_LIMIT	\
0	0.000000	0	2	1000.0	
1	0.250000	4	0	7000.0	

DBSCAN

Исходные данные были стандартизированы.

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

Проведена кластеризация методом DBSCAN при параметрах по умолчанию.

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

Метки кластеров:

```
print(set(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, 3
0, 31, 32, 33, 34, 35, -1}
```

Количество кластеров:

```
print(len(set(clustering.labels_)) - 1)
```

```
36
```

Процент наблюдений, которые реализовать не удалось:

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

```
0.7512737378415933
```

Параметры, которые принимает DBSCAN:

class `sklearn.cluster.`**`DBSCAN`**(*eps=0.5*, ***, *min_samples=5*, *metric='euclidean'*, *metric_params=None*, *algorithm='auto'*, *leaf_size=30*, *p=None*, *n_jobs=None*)

eps: максимальное расстояние, на котором точки будут считаться соседними. Значение по умолчанию eps = 0.5.

min_samples: число соседних точек, необходимое, чтобы считать точку основной. Значение по умолчанию min_samples = 5.

metric: метрика для вычисления расстояния между экземплярами в массиве объектов.

metric_params: дополнительная метрика для вычисления расстояния.

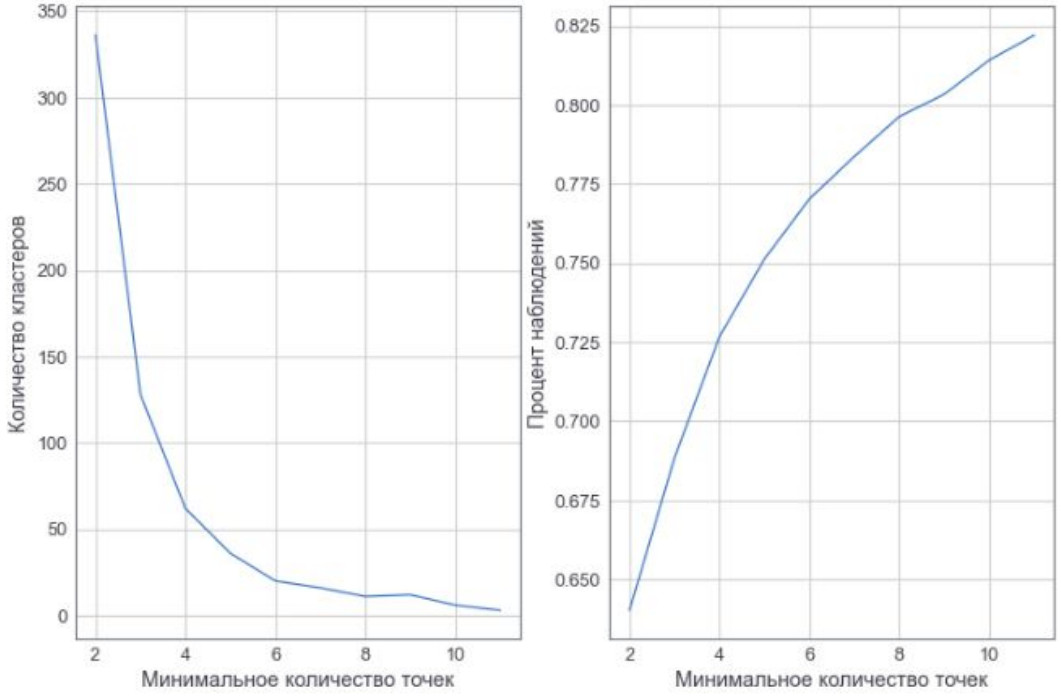
algorithm: алгоритм для поиска соседей. Может принимать следующие значения: {‘auto’, ‘ball_tree’, ‘kd_tree’, ‘brute’}

leaf_size: размер листа, передающийся в BallTree или cKDTree. Влияет на время выполнения и на объем памяти, необходимый для хранения дерева. Оптимальное значение зависит от типа задачи.

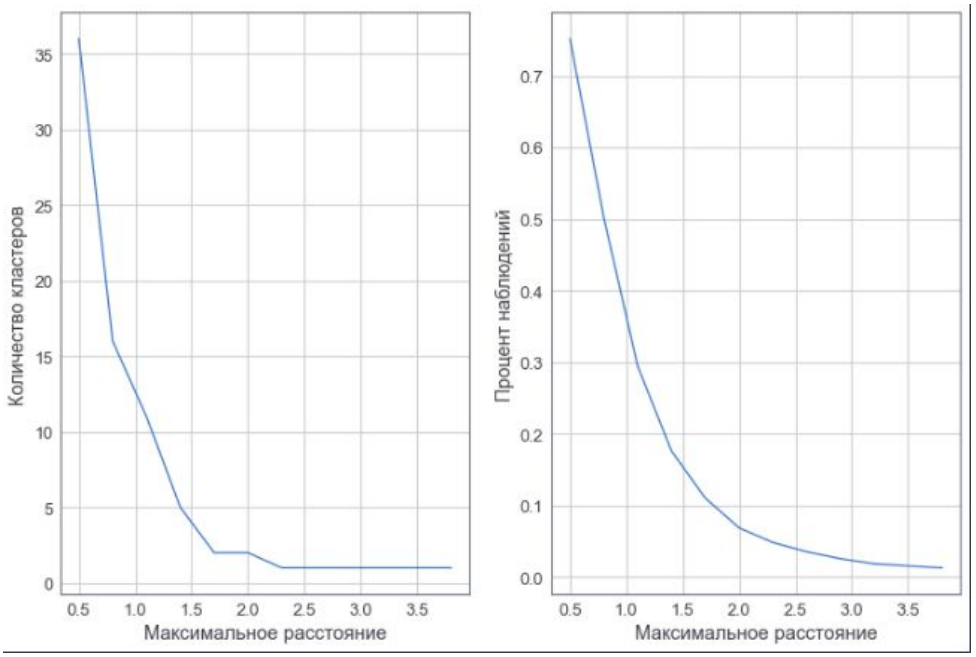
p: степень метрики Минковского. Используется для вычисления расстояния между точками. Значение по умолчанию p = 2 (эквивалент евклидова расстояния).

n_jobs: определяет количество используемых процессоров. По умолчанию n_jobs = 1. n_jobs = -1 значит использовать все процессоры.

Графики количества кластеров и процента не кластеризованных наблюдений в зависимости от минимального значения количества точек, образующих кластер:



Графики количества кластеров и процента не кластеризованных наблюдений в зависимости от максимальной рассматриваемой дистанции между наблюдениями:

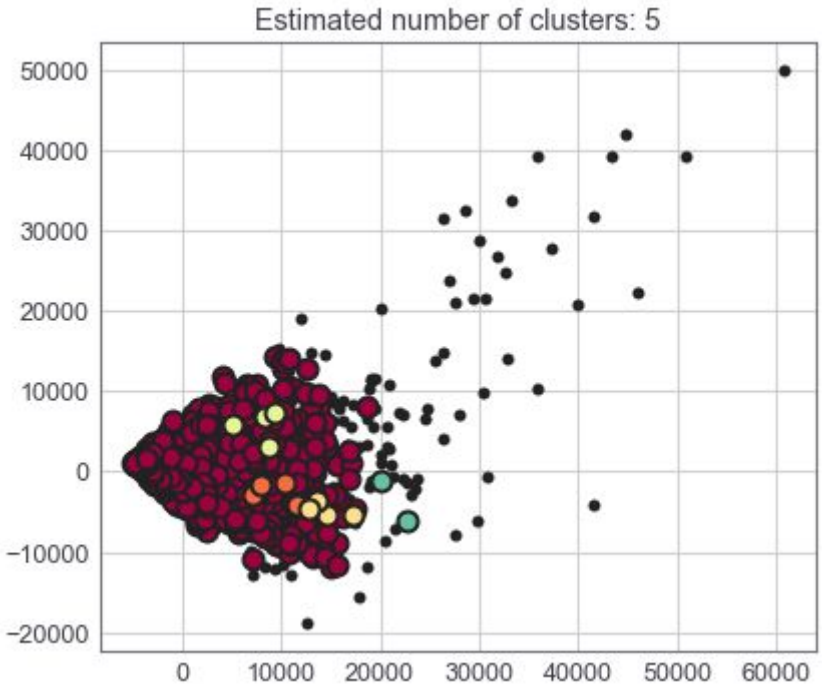


Значения параметров, при которых количество кластеров получается от 5 до 7, и процент не кластеризованных наблюдений не превышает 12%.

Было найдено значение параметров, при котором количество кластеров получается от 5 до 7 и процент не кластеризованных наблюдений не превышает 12%: минимальное количество точек 3, максимальное расстояние равно 2.9.

	min samples	eps	clusters	noise
15	3.0	2.9	5.0	0.022233

Понижена размерность данных до 2 с использованием метода главных компонент. Визуализация результатов кластеризации полученных в предыдущем пункте:



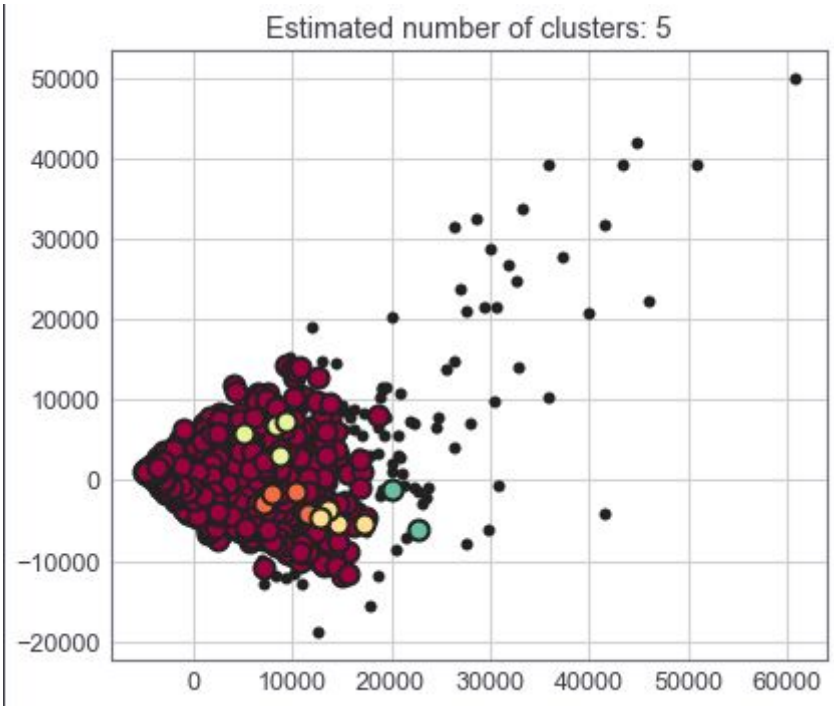
Параметры OPTICS:

class `sklearn.cluster.`**`OPTICS`**(**, min_samples=5, max_eps=inf, metric='minkowski', p=2, metric_params=None, cluster_method='xi', eps=None, xi=0.05, predecessor_correction=True, min_cluster_size=None, algorithm='auto', leaf_size=30, n_jobs=None*)

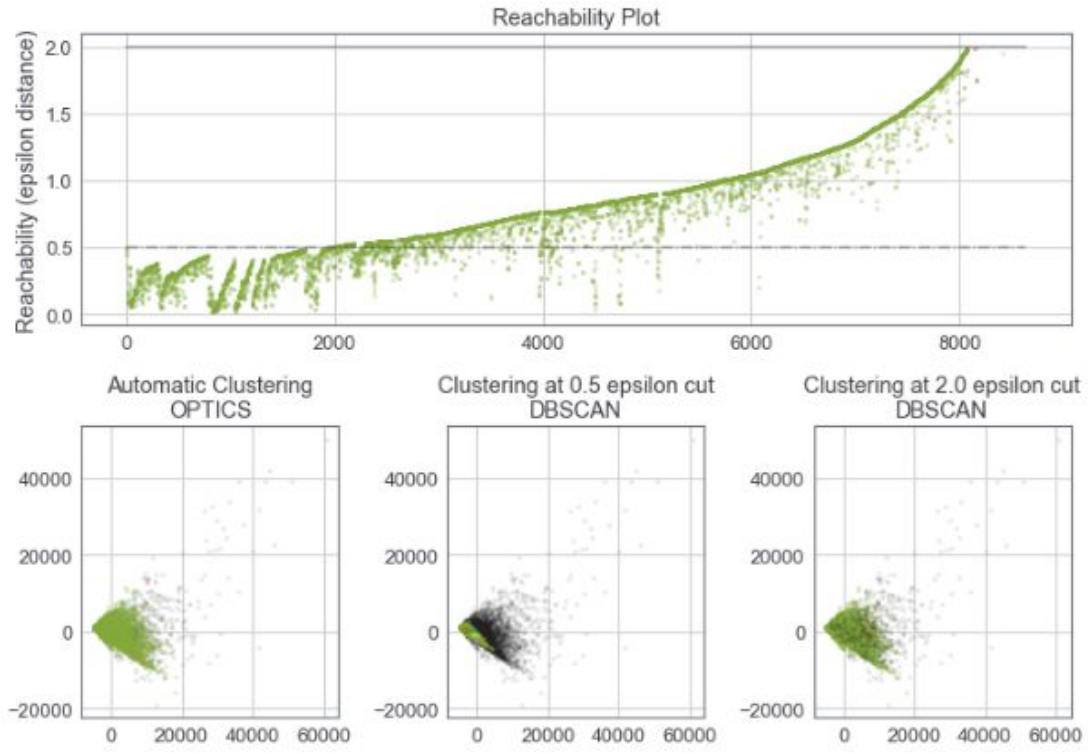
min_samples – минимально число точек в окрестности точек, при котором она считается основной;
max_eps – максимальное расстояние, допускающее сходство между точками;
metric – метрика для вычисления расстояния между точками;
p – параметр метрики Минковского (при p = 1 равносильно использованию manhattan_distance, 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.

При параметры метода OPTICS max_eps = 2.9 и min_samples = 3 и cluster_method='dbscan' получились результаты близкие к результатам DBSCAN из пункта 6:

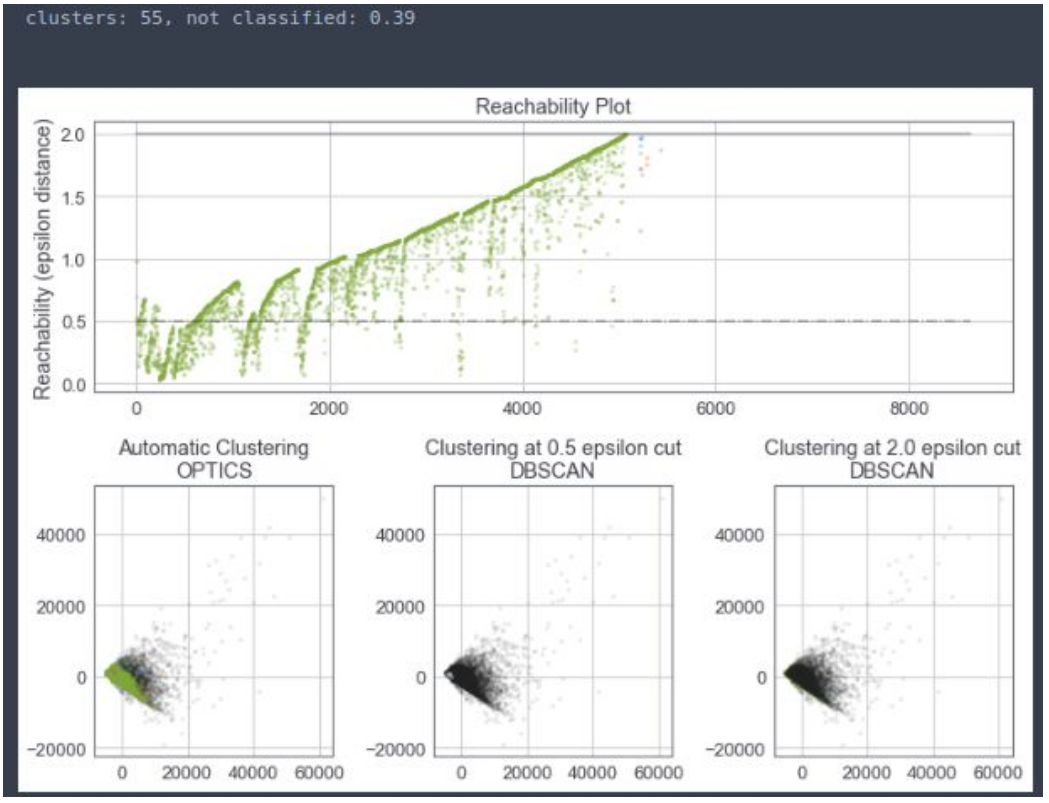


Визуализация результата и график достижимости:



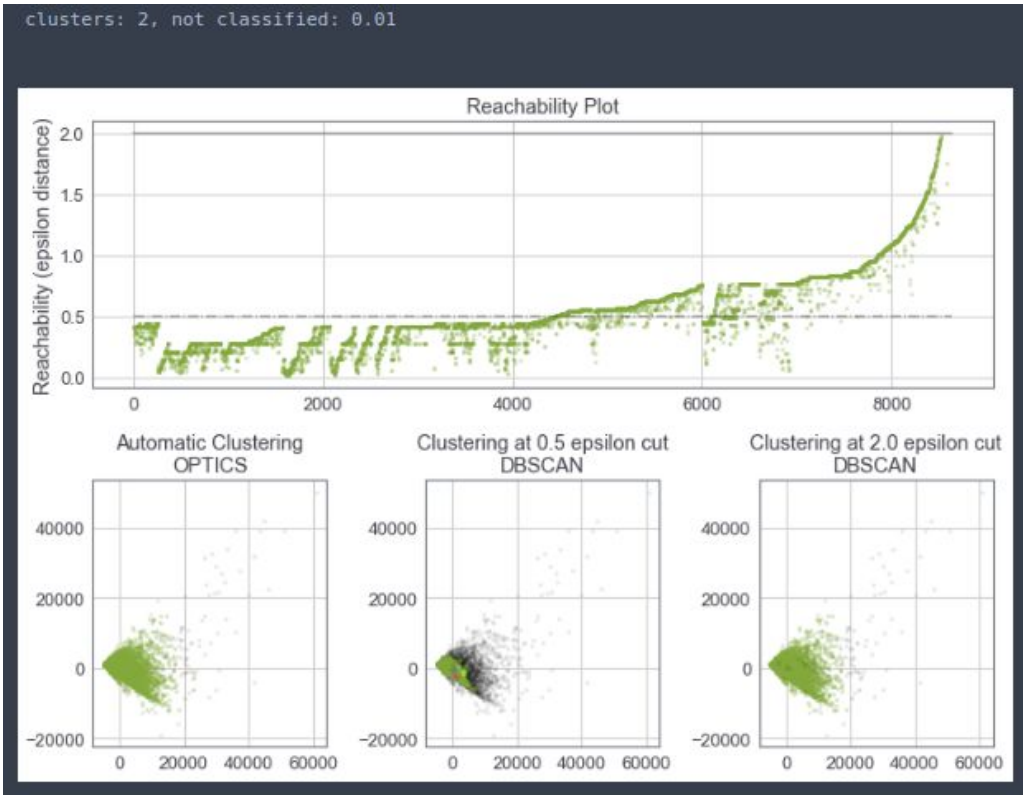
Исследование работы метода OPTICS с использованием различных метрик:

cityblock



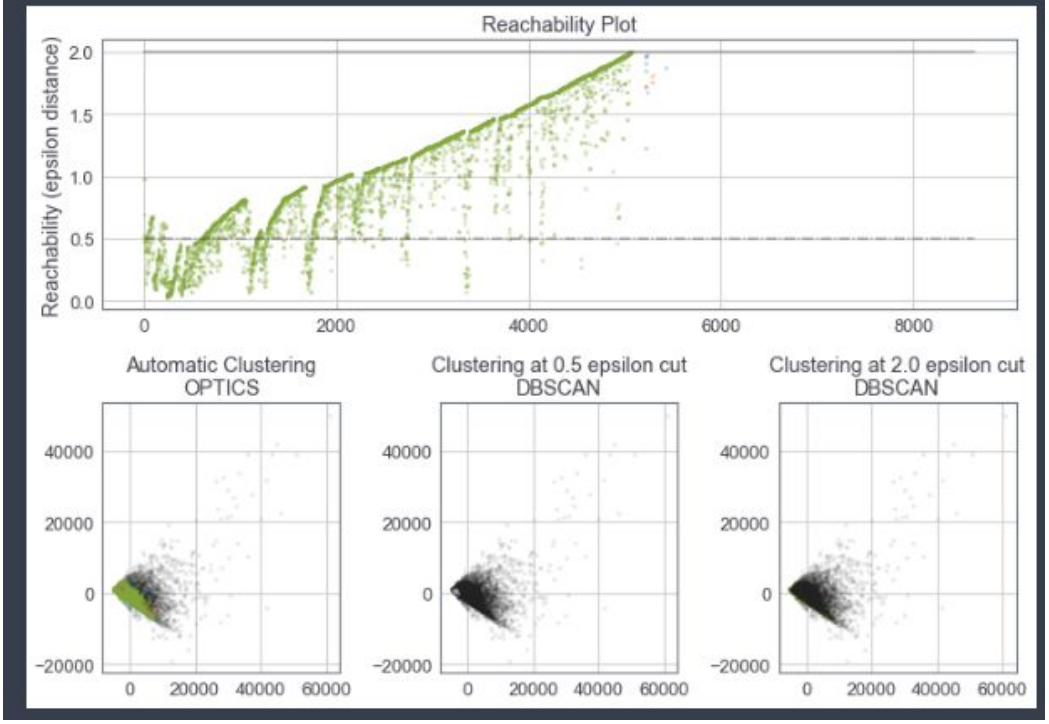
cosine

chebyshev



11

```
clusters: 55, not classified: 0.39
```



12

```
clusters: 6, not classified: 0.06
```