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. 6304 _____ Антонов С.А.

Преподаватель _____ Жангиров Т.Р.

Санкт-Петербург
2020

Цель работы:

Ознакомиться с методами кластеризации модуля Sklearn.

Ход работы:

Загрузка данных

1. На данном этапе был скачан и загружен датасет в датафрейм.

```
data = pd.read_csv('CC_GENERAL.csv', header=None)
print(data.head())
```

```
          BALANCE  BALANCE_FREQUENCY  PURCHASES  ONEOFF_PURCHASES  \
0   40.900749           0.818182      95.40              0.00
1  3202.467416           0.909091       0.00              0.00
2  2495.148862           1.000000     773.17            773.17
4   817.714335           1.000000      16.00             16.00
5  1809.828751           1.000000    1333.28              0.00

   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

   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
```

Рисунок 1 Загруженный датасет

DBSCAN:

1. Так как признаки в выборке соответствуют разным шкалам, была проведена стандартизация данных.

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

2. Была произведена кластеризация методом DBSCAN, выведены получившиеся метки кластеров, их количество, а также процент

наблюдений, который не удалось кластеризовать. Приведем полученные результаты:

Labels:

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

Num of clusters:

```
36
```

No classified, %:

```
0.7512737378415933
```

3. Параметры DBSCAN:
   - eps – Максимальное расстояние между двумя элементами, чтобы один считался соседним с другим.
   - min_samples – число элементов в окрестности точки, чтобы считать ее основной.
   - metric – метрика, используемая при вычислении расстояния между элементами.
   - metric_params – дополнительные ключевые аргументы для метрической функции.
   - algorithm – алгоритм, который будет использоваться для вычисления точечных расстояний и поиска ближайших соседей.
   - p – степень метрики Миньковского, которая будет использоваться для вычисления расстояния между точками.
4. Были построены графики зависимости параметра eps от количества кластеров и процента выбросов. Результаты приведены на рисунке 2.

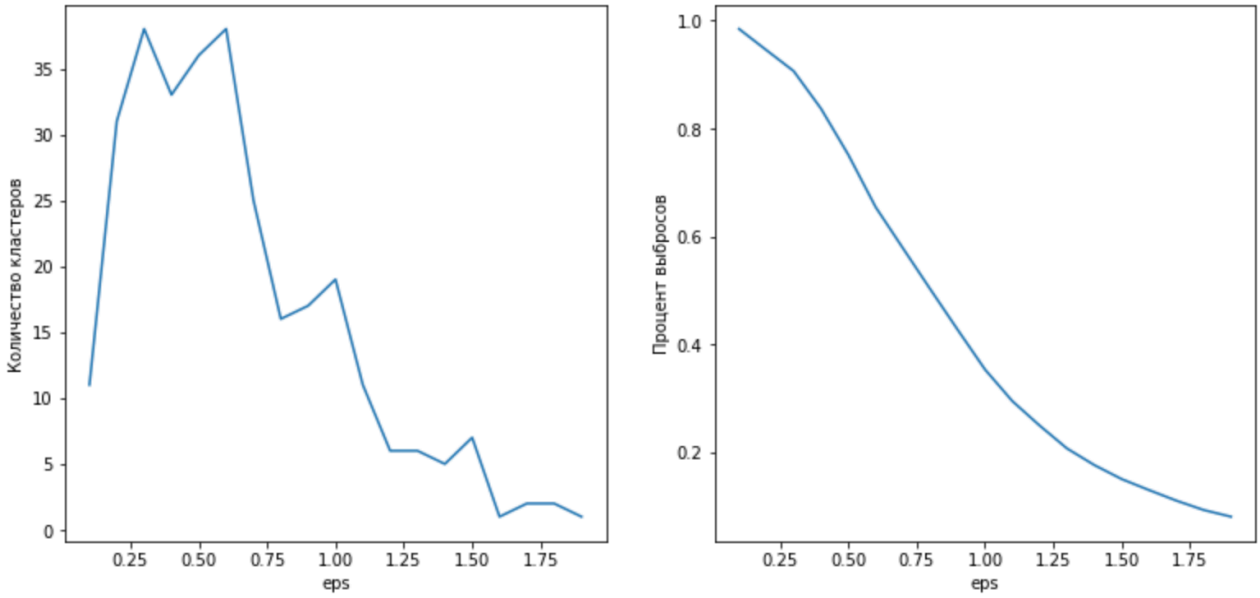

Рисунок 2 зависимость параметра eps от количества кластеров и процента выбросов.

5. Был построен график зависимости количества кластеров и количества не кластеризованных данных от параметра min_samples. График представлен на рисунке 3

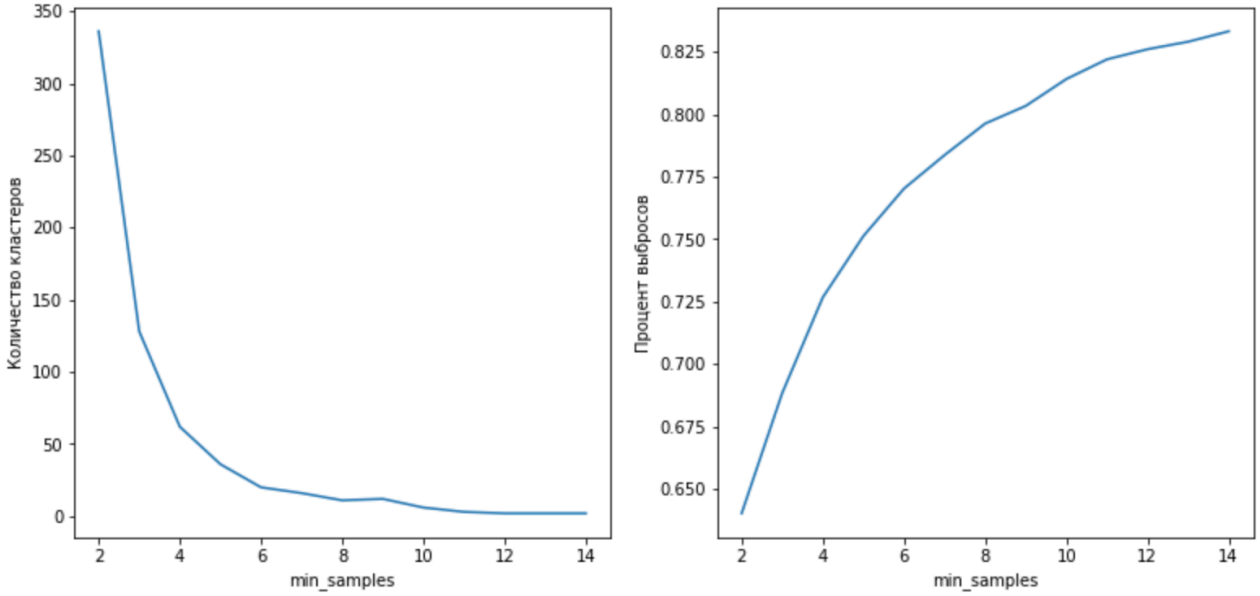

Рисунок 3 Зависимость количества кластеров и количества не кластеризованных данных от параметра min_samples

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

```
samples = np.arange(1, 4, 1)
```

```
eps_ = np.arange(1.5, 2.5, 0.1)
info = {}
for sample in samples:
    for eps in eps_:
        clustering = DBSCAN(eps=eps ,min_samples=sample, n_jobs=-
1).fit(scaled_data)
        labels_set = set(clustering.labels_)
        info[(sample, eps)]= [len(labels_set) - 1,
list(clustering.labels_).count(-1) / len(list(clustering.labels_))]

print('(samples, eps) -> [count of clusters, percent of ]')
for key, value in info.items():
    if value[0]>=5 and value[0]<=7 and value[1]<=0.12:
        print(key, value)
```

eps = 2, min_samples = 3

7. Размерность данных была понижена до 2 при помощи метода главных компонент, а также визуализированы результаты кластеризации, полученные в пункте 6. Результат показан на рисунке 4.

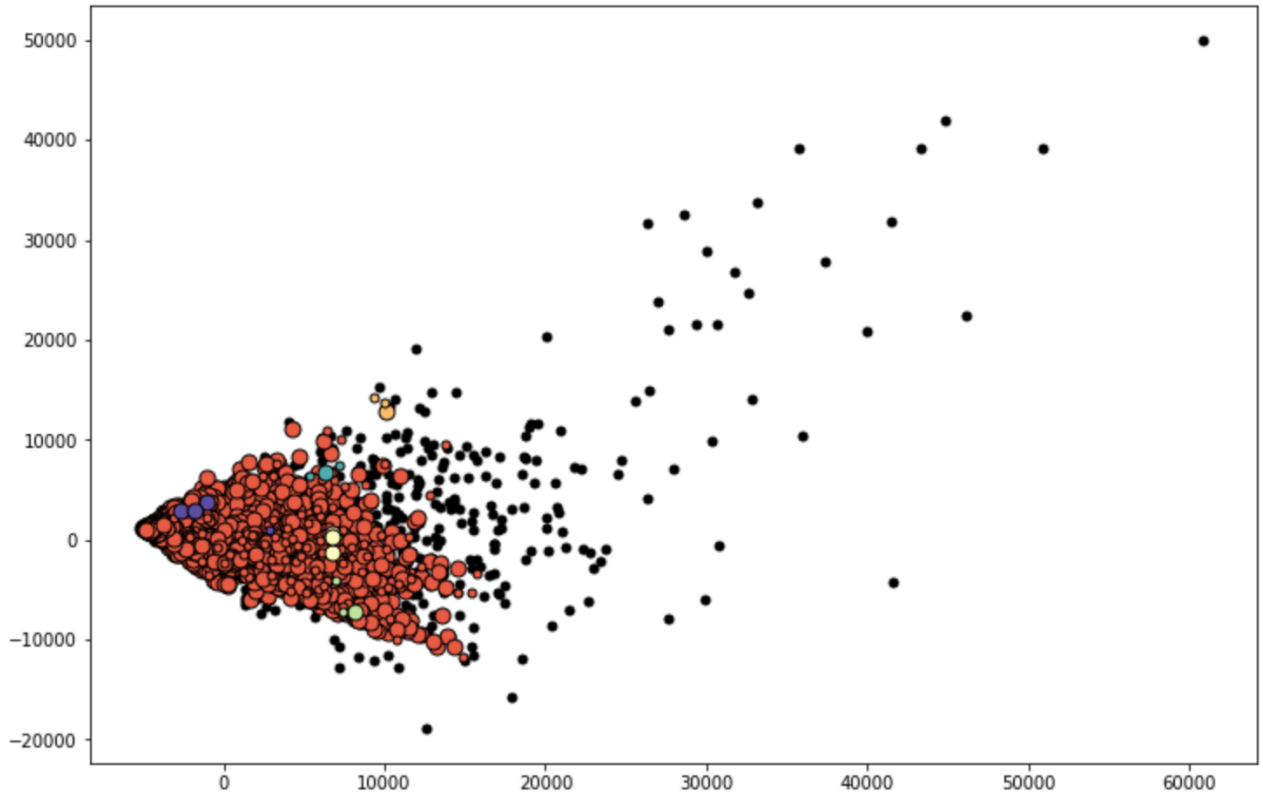

Рисунок 4 Результаты кластеризации

OPTICS

1. Параметры метода OPTICS:
   - min_samples – число элементов в окрестности точки, чтобы она считалась основной.
   - max_eps – максимально расстояние между элементами, допускающие их сходство
   - metric – метрика, используемая при вычислении расстояния между элементами
   - p – параметр для Минковского
   - metric_params – дополнительные ключевые аргументы для метрической функции
   - cluster_method – метод извлечения кластеров
   - eps – максимальное расстояние между двумя элементами, допускающее их соседство. По умолчанию соответствует max_eps, используется только для cluster_method = dbscan
   - xi – определяет максимальную крутизну на графике достижимости, который составляет границу кластера. Используется только при cluster_method = xi.
   - predecessor_correction – коррекция кластеров в соответствии с предшественниками. Используется только при cluster_method = xi.
   - min_cluster_size – минимальное количество элементов в кластере OPTICS
   - algorithm – алгоритм для поиска ближайший соседей.
2. Были найдены параметры метода OPTICS при которых, полученные результаты получились близкими к результатам DBSCAN из пункта 6

```
clust = OPTICS(min_samples=3, max_eps=2, cluster_method =
"dbscan").fit(scaled_data)
```

Labels:

```
{0, 1, 2, 3, 4, 5, -1}
```

Num of clusters:

```
6
```

No classified, %:

```
0.06
```

3. Полученные данные были визуализированы. График представлен на рисунке 5.

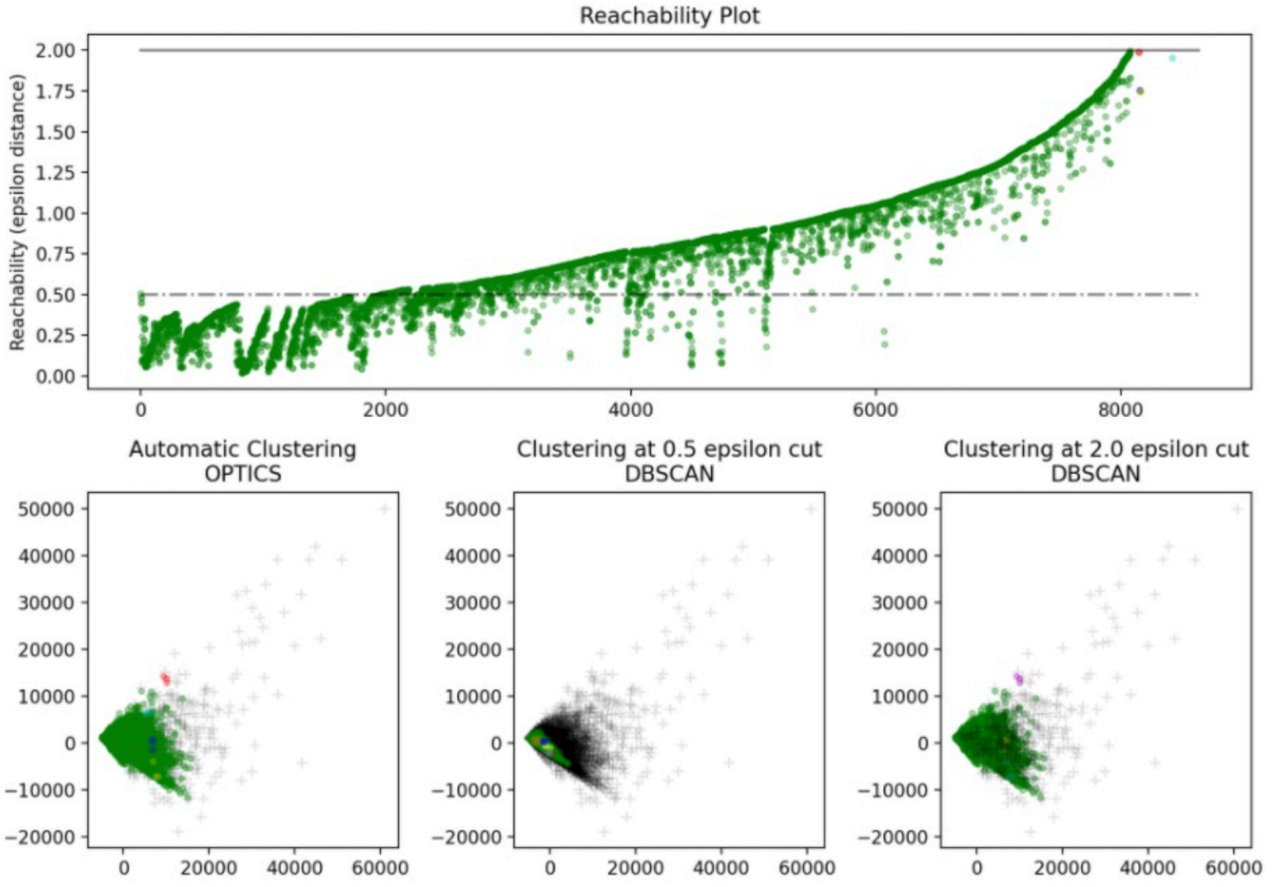

Рисунок 5 Результаты кластеризации OPTICS

Как и в случае с DBSCAN большинство данных принадлежат одному и тому же кластеру.

4. Было проведено исследование работы метода OPTICS с различными метриками.
   - cityblock – манхеттанское расстояние

Рисунок 6 Кластеризация с метрикой cityblock

Labels:

```
{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 2
1, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 4
0, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, -1}
```

Num of clusters:

```
55
```

No classified, %:

```
39.4
```

- cosine – косинусовое

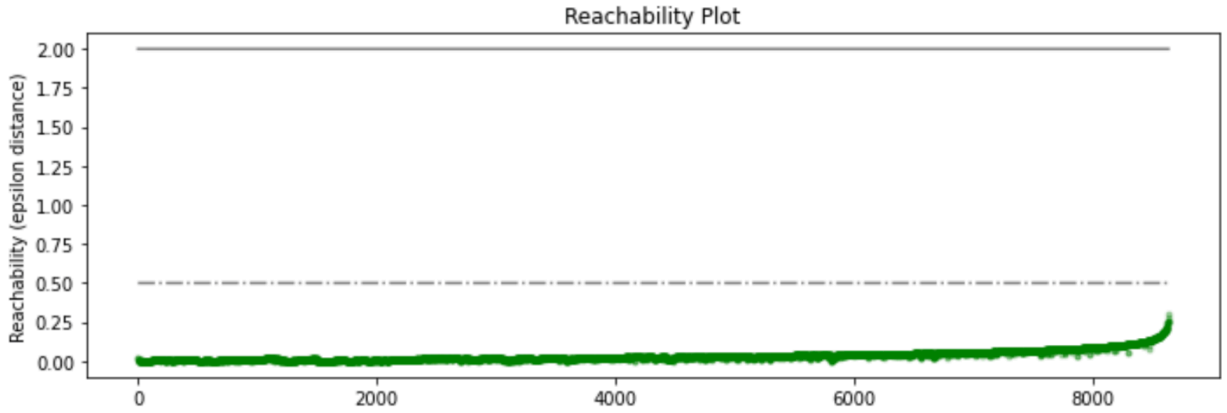

Рисунок 7 Кластеризация с метрикой cosine

Labels:

```
{0}
```

Num of clusters:

```
0
```

No classified, %:

```
0.0
```

- chebyshev – чебышева

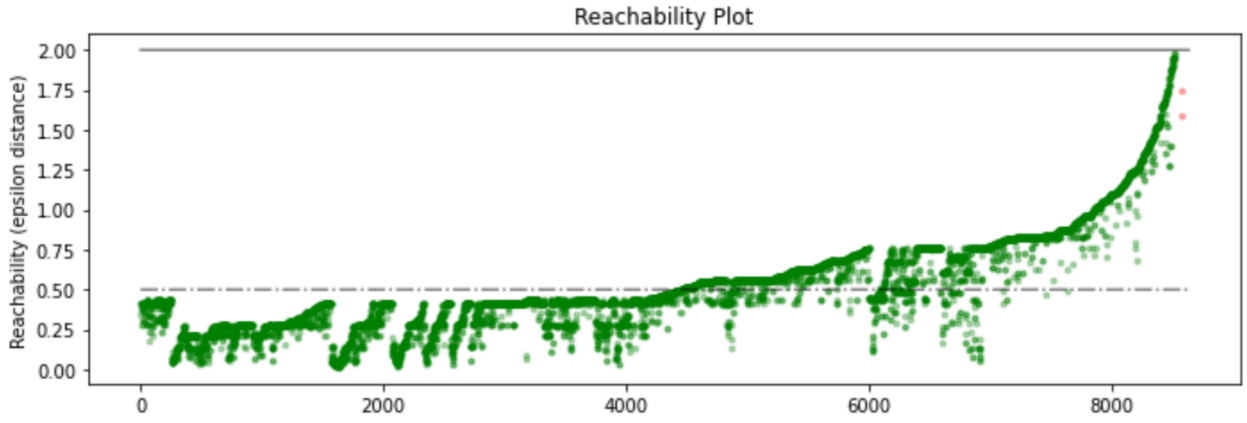

Рисунок 8 Кластеризация с метрикой Chebyshev

Labels:

```
{0, 1, -1}
```

Num of clusters:

```
2
```

No classified, %:

```
1.3
```

- l1

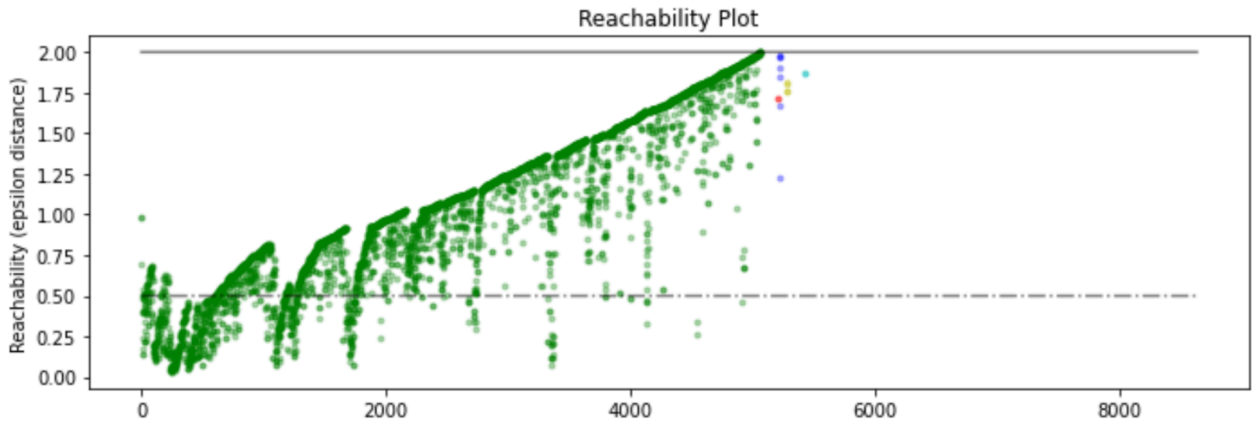

Рисунок 9 Кластеризация с метрикой l1

Labels:

```
{0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 2
1, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 4
0, 41, 42, 43, 44, 45, 46, 47, 48, 49, 50, 51, 52, 53, 54, -1}
```

Num of clusters:

```
55
```

No classified, %:

```
39.49
```

- braycurtis

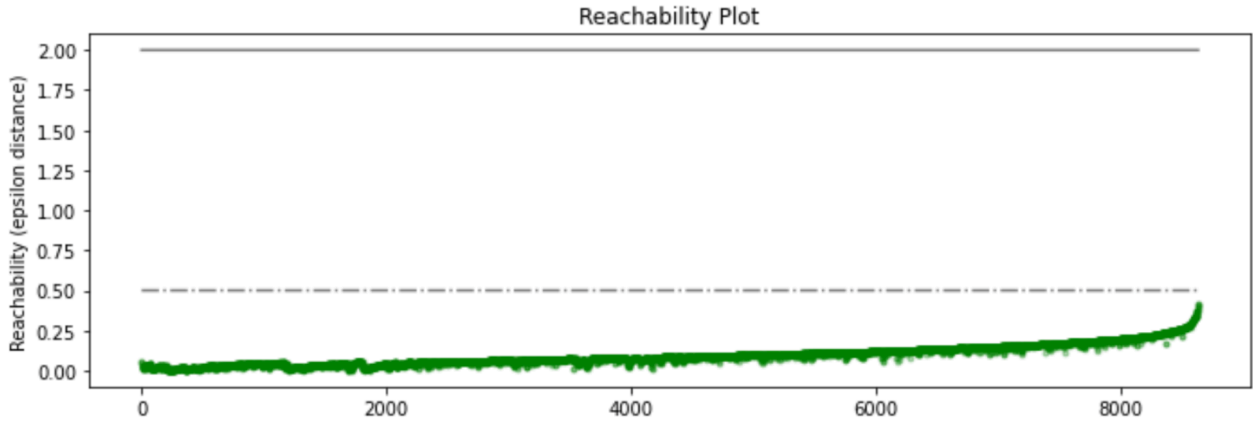

Рисунок 10 Кластеризация с метрикой braycurtis

Labels:

```
{0}
```

Num of clusters:

```
0
```

No classified, %:

```
0.0
```

Выводы:

В ходе выполнения лабораторной работы было произведено знакомство с кластеризацией методами DBSCAN и OPTICS из модуля Sklearn. Для исходного набора данных оба метода производят разбиение либо на большое количество кластеров, либо на один единственный.