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 _____ Медведев Е. Р.

Преподаватель _____ Жангиров Т. Р.

Санкт-Петербург

2020

Цель работы

Ознакомиться с методами кластеризации модуля Sklearn.

Ход работы

DBSCAN

Данные загружены из csv файла. Пропущенные значения выброшены.

[2]:

	BALANCE	BALANCE_FREQUENCY	PURCHASES	ONEOFF_PURCHASES	INSTALLMENTS_PURCHASES	CASH_ADVANCE	PURCHASES_FREQUENCY	ONEOFF_P
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	
...	...	...	...	...	...	...	...	...

Кластеризация методом K-Means:

```
[47]: unlabeled_data = data
      k_means = KMeans(init='k-means++', n_clusters=3, n_init=15)
      k_means.fit(unlabeled_data);
```

Стандартизация данных:

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

Кластеризация методом DBSCAN:

```
[85]: clustering = 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("Метки кластеров:", set(labels))
      print("Число кластеров: ", n_clusters_)
      print("% некластеризованных: ", n_noise_ / len(list(labels))*100)

      Метки кластеров: {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}
      Число кластеров:  36
      % некластеризованных:  75.12737378415933
```

Параметры DBSCAN:

1. eps: максимальное допустимое расстояние между двумя сэмплами одного кластера

2. min_samples: число соседей в окрестности точки, необходимое для того, чтобы она считалась базовой (точка входит в подсчет)

3. metric: метрика, которая используется для вычисления расстояния

4. metric_params: дополнительные аргументы для функции метрики

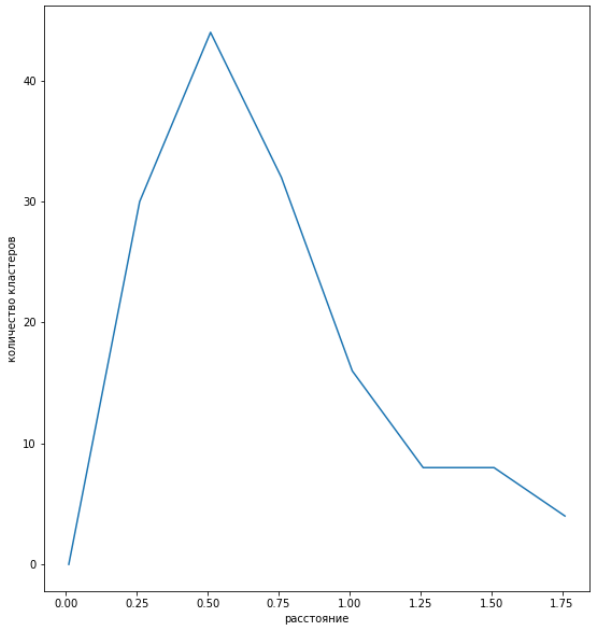
5. algorithm: алгоритм, который используется для нахождения ближних соседей для вычисления точечных расстояний

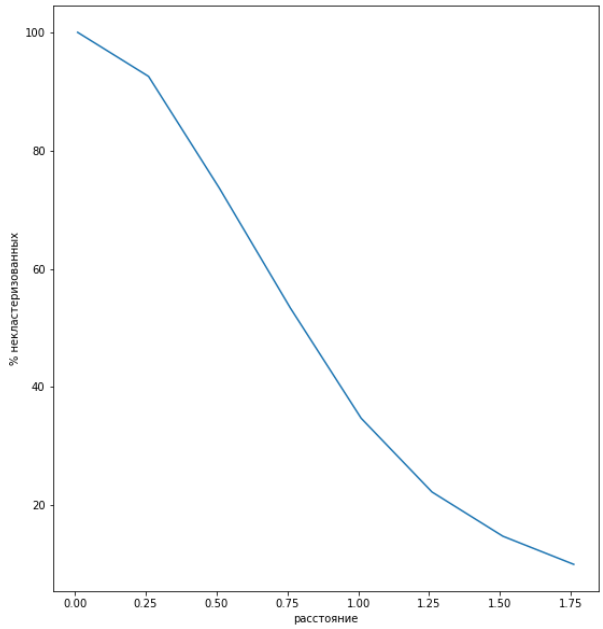
6. leaf_size: размер листа дерева в algorithm

7. p: степени метрики Минковского, которая будет использоваться для вычисления расстояния между точками

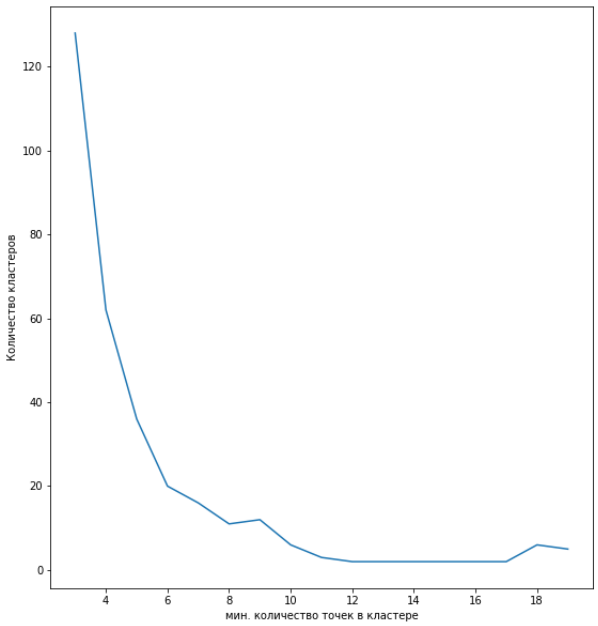
8. n_jobs: количество параллельных потоков

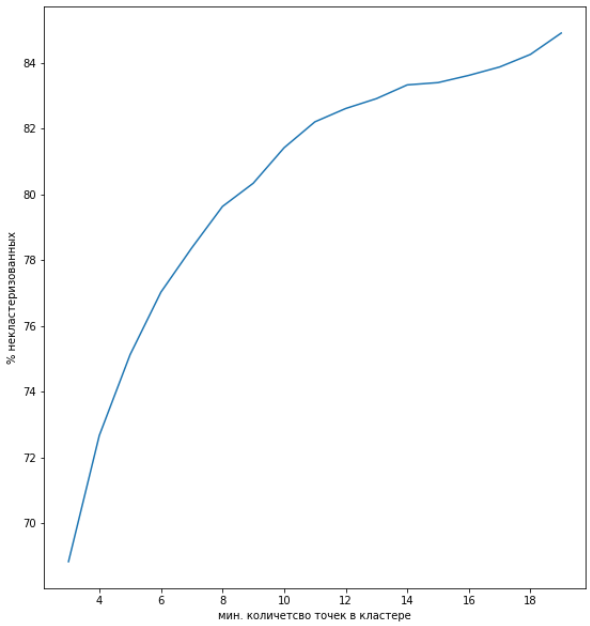
Построен график количества кластеров и процента не кластеризованных наблюдений в зависимости от максимальной рассматриваемой дистанции между наблюдениями:





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

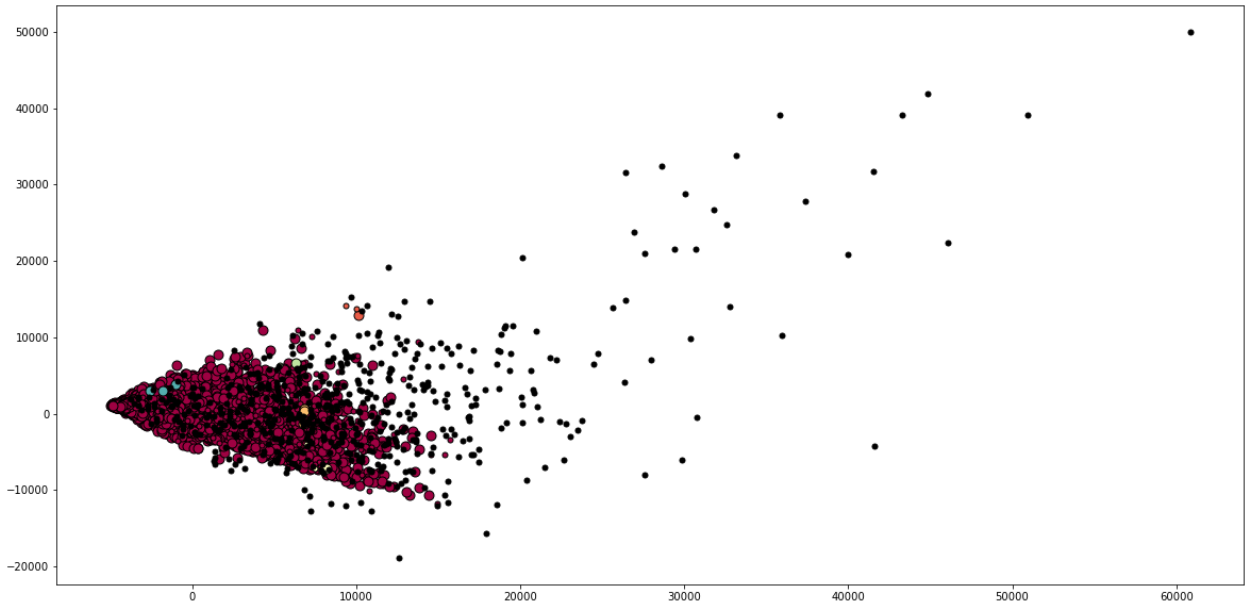




Определены значения параметров, при котором количество кластеров получается от 5 до 7, и процент не кластеризованных наблюдений не превышает 12%:

```
Число кластеров:  6
% некластеризованных 6.287633163501622
min_samples 3
eps 2.0000000000000004
```

Понижение размерности данных до 2 с помощью метод главных компонент и визуализация данных:



OPTICS

Параметры OPTICS:

1. min_samples: число соседей в окрестности точки, необходимое для того, чтобы она считалась базовой

2. max_eps: максимальное допустимое расстояние между двумя сэмплами одного кластера

3. metric: метрика, которая используется для вычисления расстояния

5. p: степени метрики Минковского, которая будет использоваться для вычисления расстояния между точками

6. metric_params: дополнительные аргументы для функции метрики

7. cluster_method: метод определения кластера

8. eps: то же самое, что в dbscan (используется, если cluster_method='dbscan')

9. xi: минимальное число сэмплов (используется, если cluster_method='xi')

10. predecessor_correction: корректировка кластеров на основе предшественников (используется, если cluster_method='xi')

11. min_cluster_size: минимальное число сэмплов в кластере

12. algorithm: алгоритм, который используется для нахождения ближних соседей для вычисления точечных расстояний

13. leaf_size: размер листа дерева в algorithm

14. n_jobs: количество параллельных потоков

Параметры OPTICS, при которых этот метод приближается к результатам DBSCAN из прошлого пункта:

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

В отличие от DBSCAN, сохраняет иерархию для разных радиусов окрестностей. Больше подходит для больших датасетов, чем текущая версия DBSCAN в sklearn.

```
Метки кластеров: {0, 1, 2, 3, 4, 5, -1}
Число кластеров:  6
% некластеризованных:  6.310792033348772
```

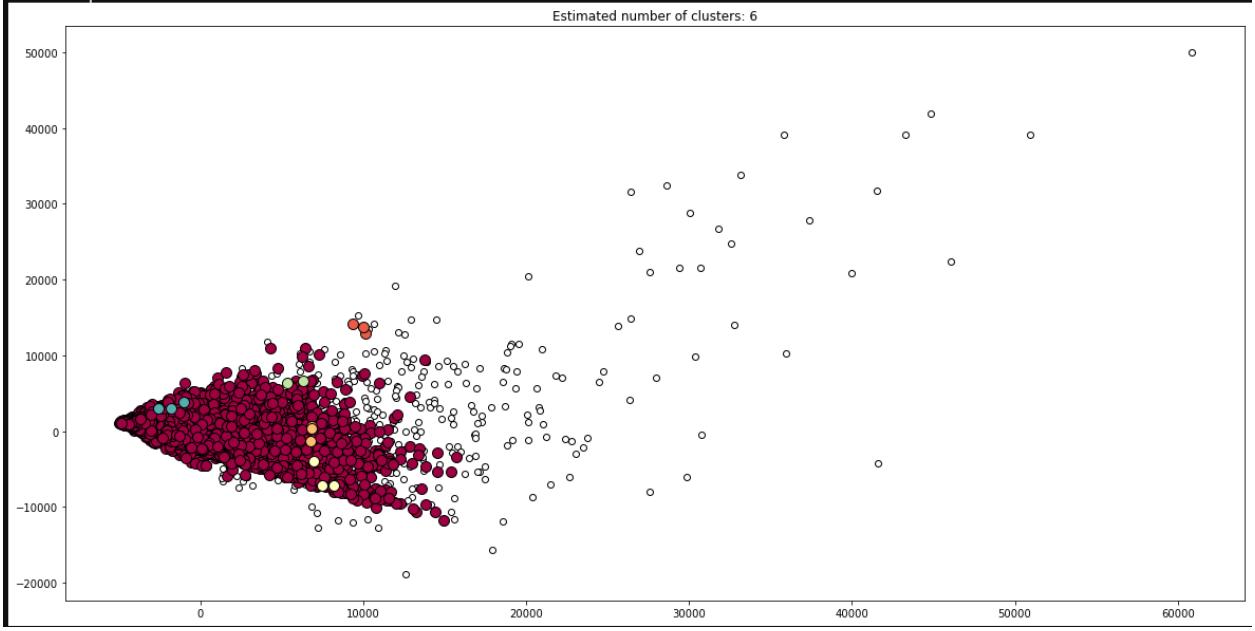
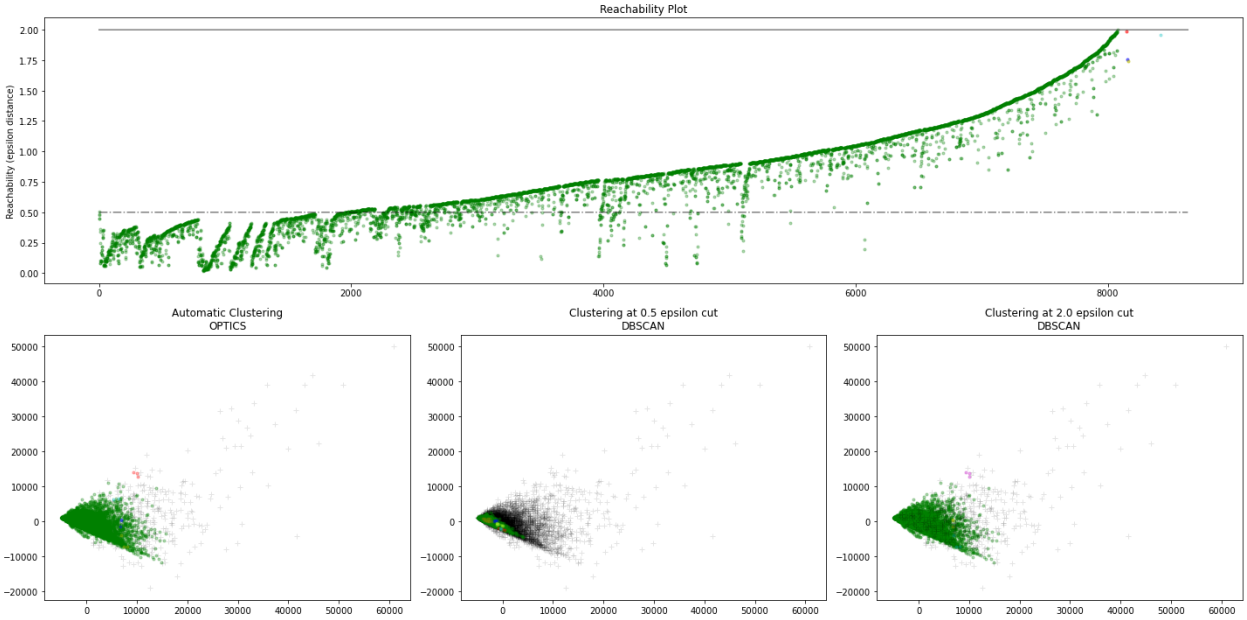
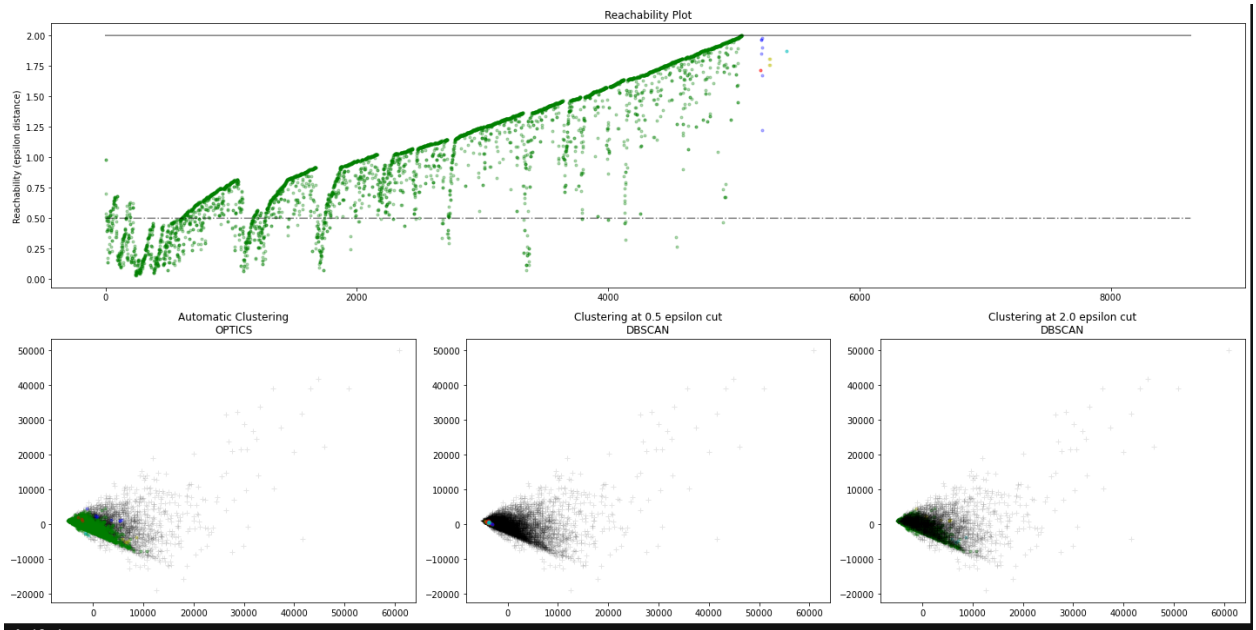


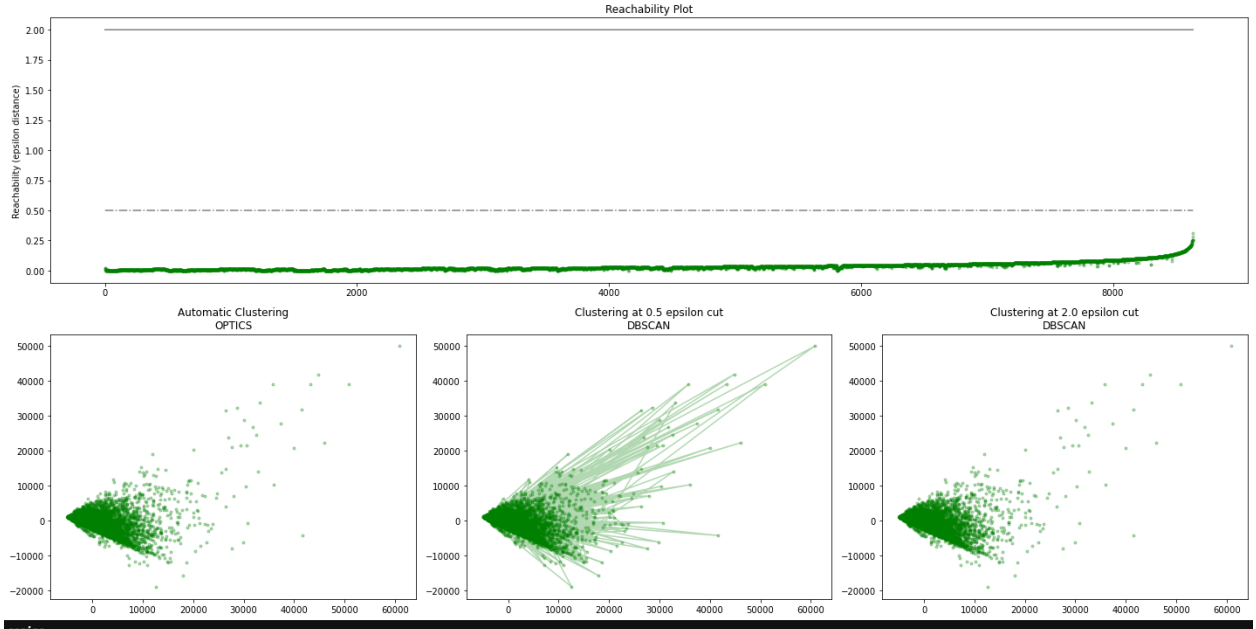
График достижимости для полученного результата:



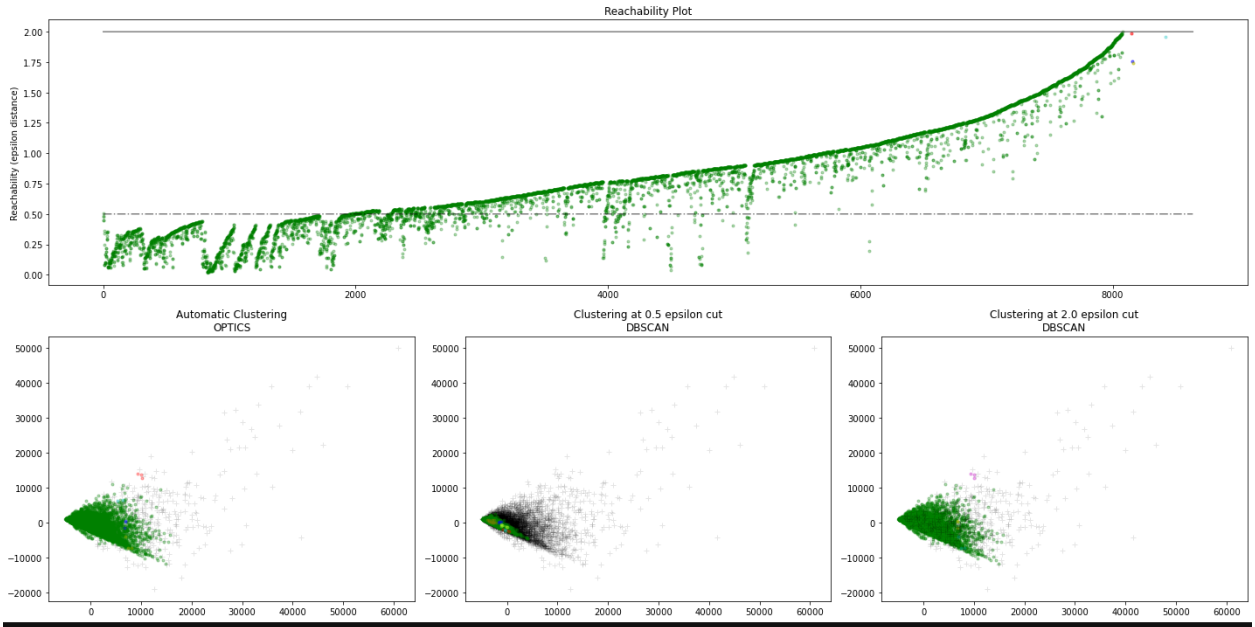
Исследована работа метода OPTICS при использовании других метрик:



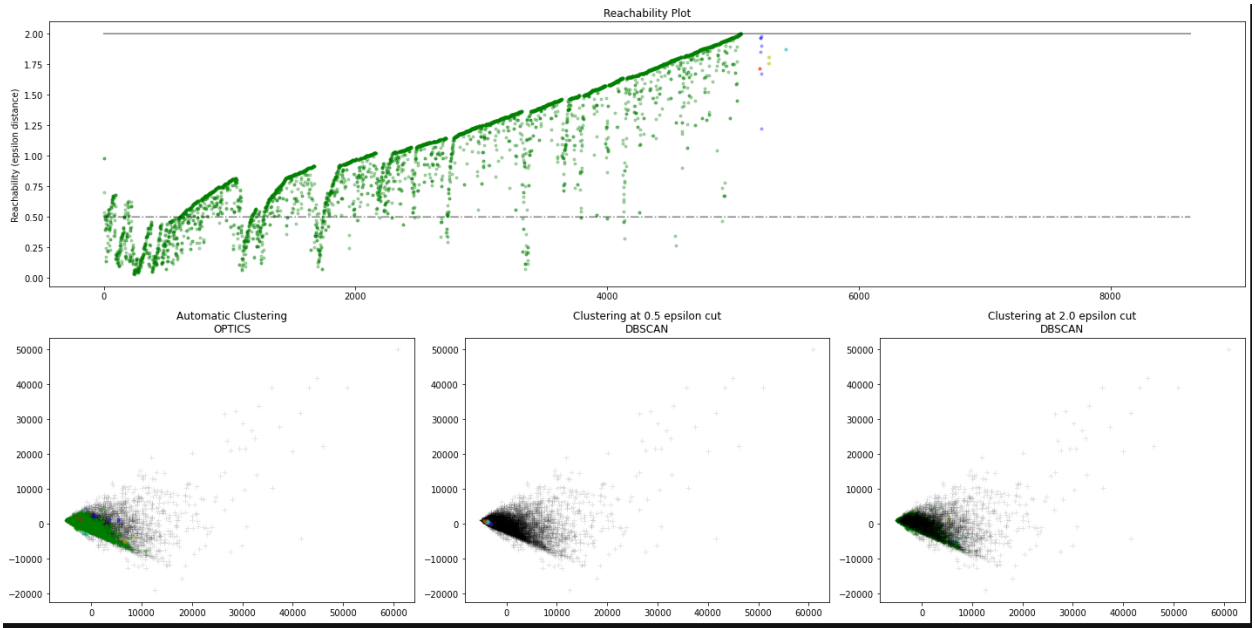

```
cityblock
Метки кластеров: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47,
48, 49, 50, 51, 52, 53, 54, -1}
Число кластеров:  55
% некластеризованных:  39.49745252431681
```



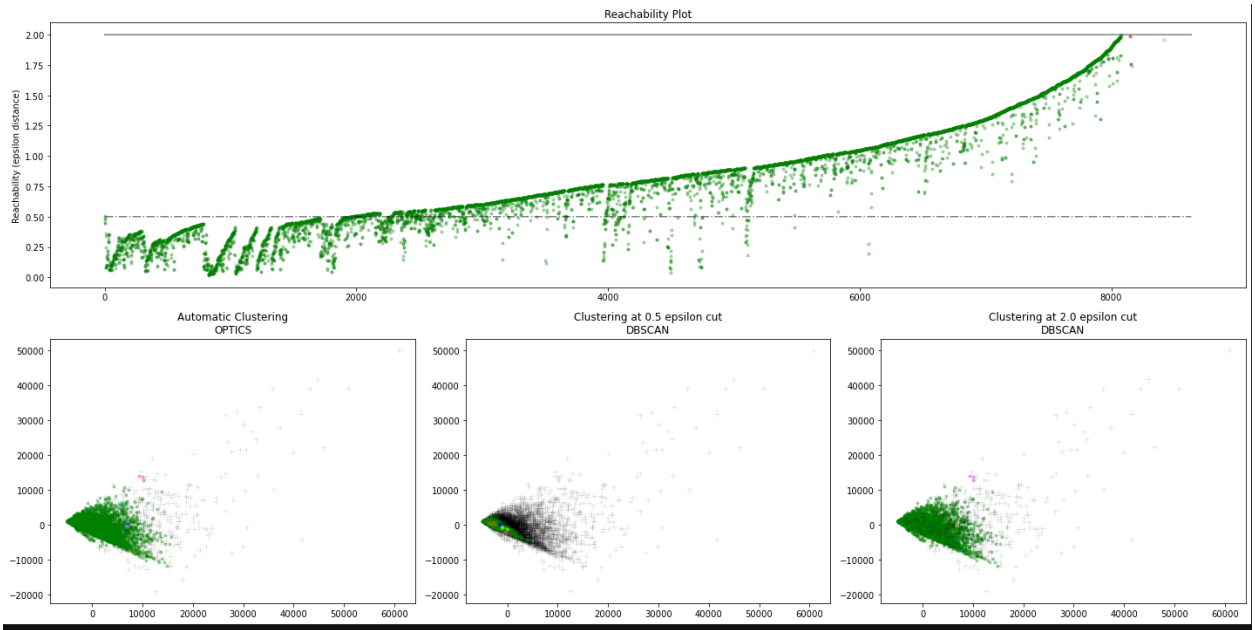

```
cosine
Метки кластеров: {0}
Число кластеров:  1
% некластеризованных:  0.0
```



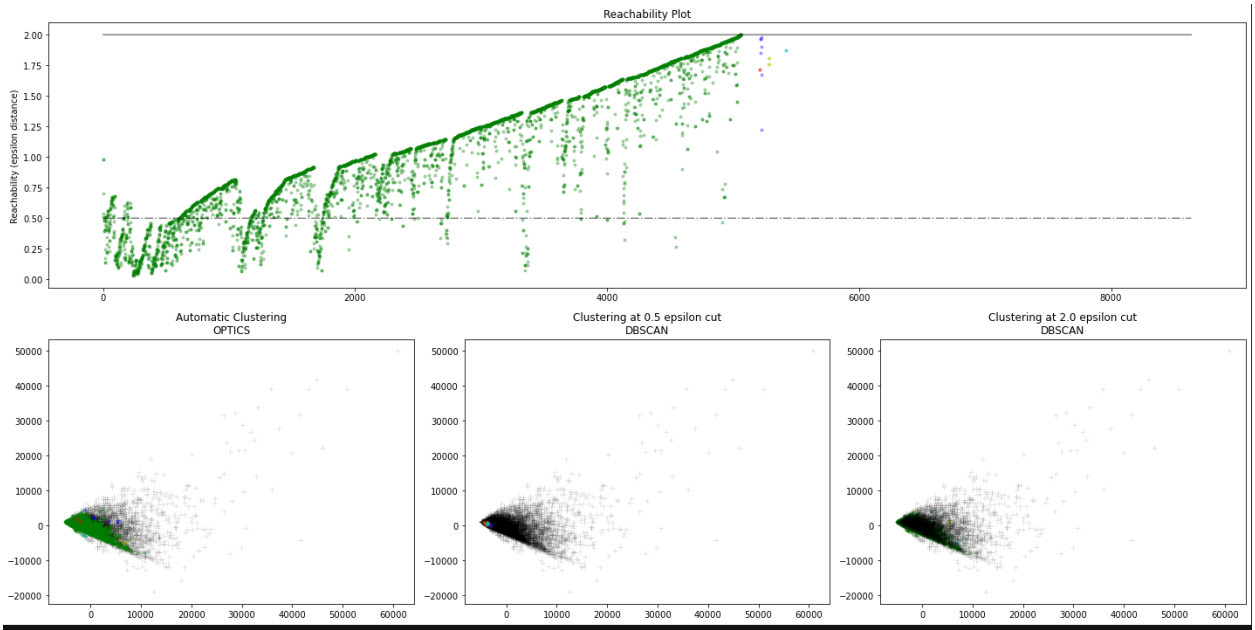

```
euclidean
Метки кластеров: {0, 1, 2, 3, 4, 5, -1}
Число кластеров:  6
% некластеризованных:  6.310792033348772
```



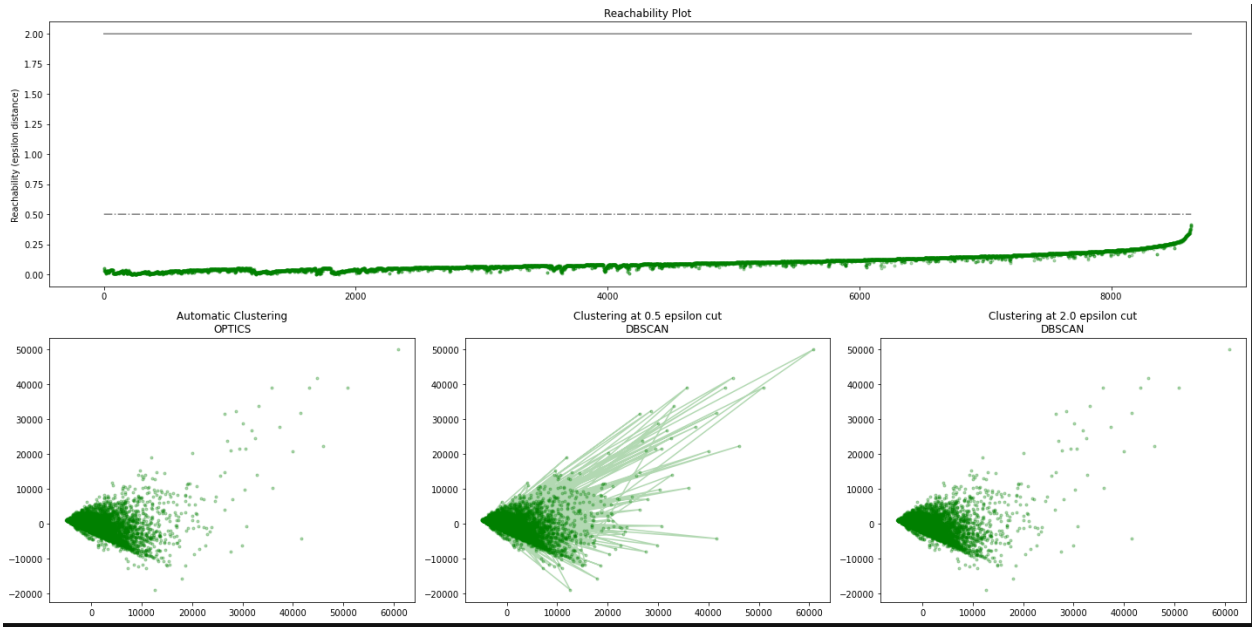

```
11
Метки кластеров: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47,
48, 49, 50, 51, 52, 53, 54, -1}
Число кластеров:  55
% некластеризованных:  39.49745252431681
```



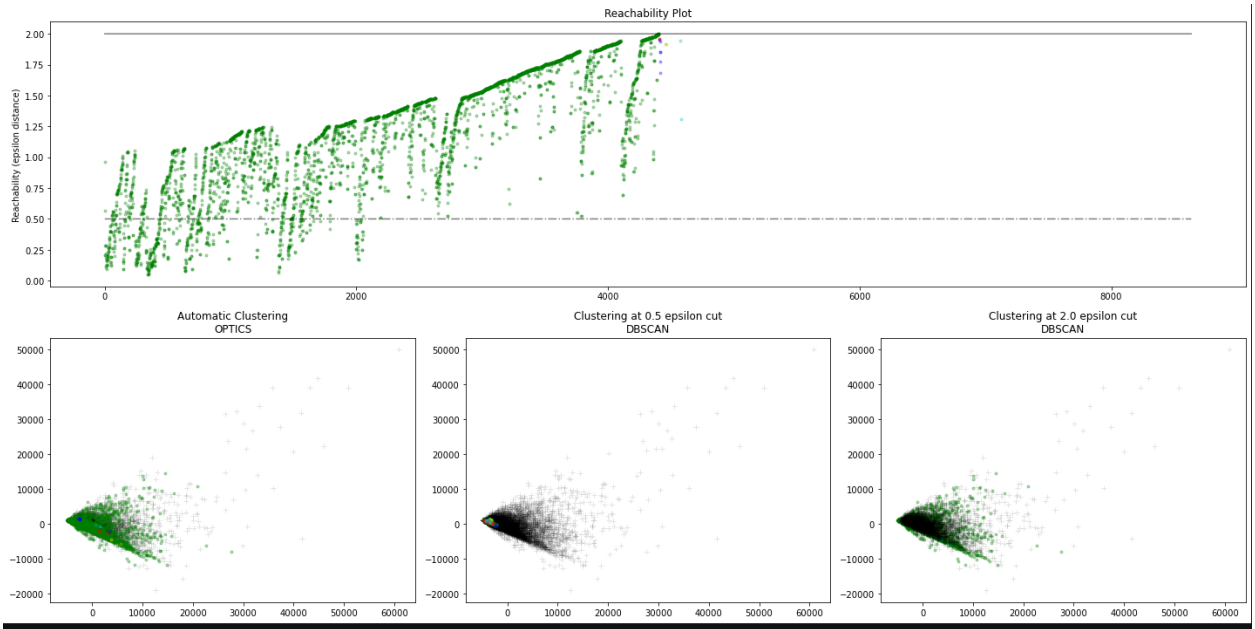

```
12
Метки кластеров: {0, 1, 2, 3, 4, 5, -1}
Число кластеров:  6
% некластеризованных:  6.310792033348772
```



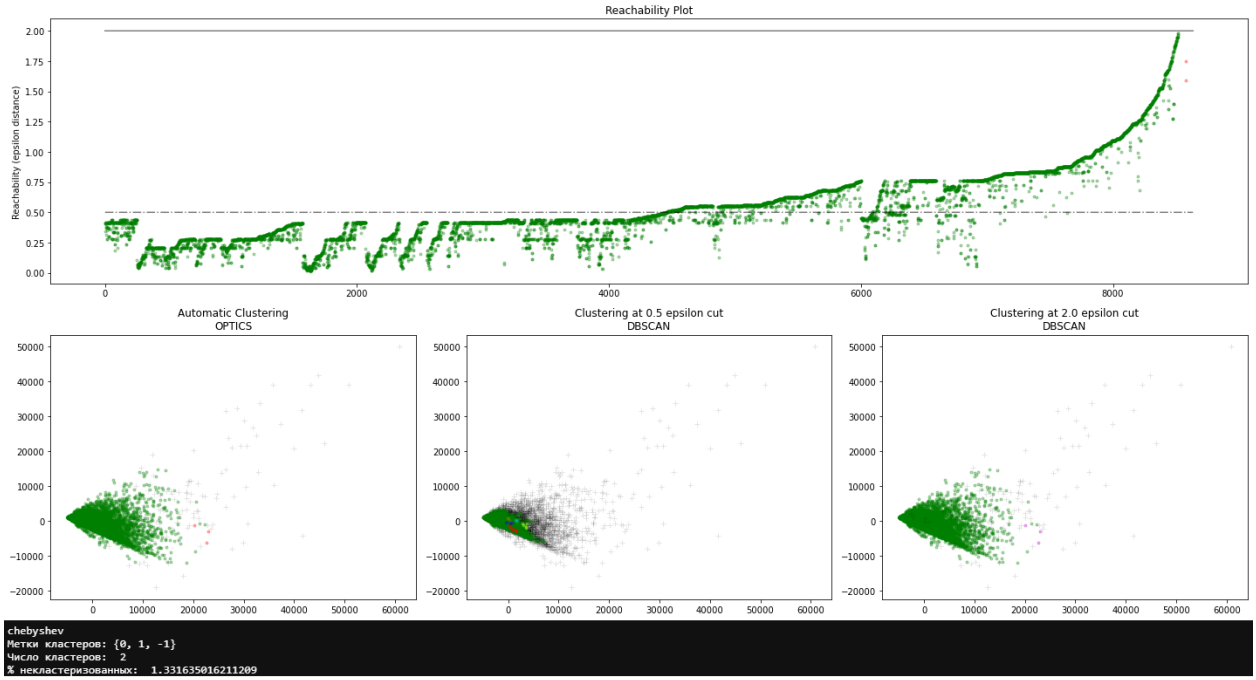

```
manhattan
Метки кластеров: {0, 1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20, 21, 22, 23, 24, 25, 26, 27, 28, 29, 30, 31, 32, 33, 34, 35, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47,
48, 49, 50, 51, 52, 53, 54, -1}
Число кластеров:  55
% некластеризованных:  39.49745252431681
```



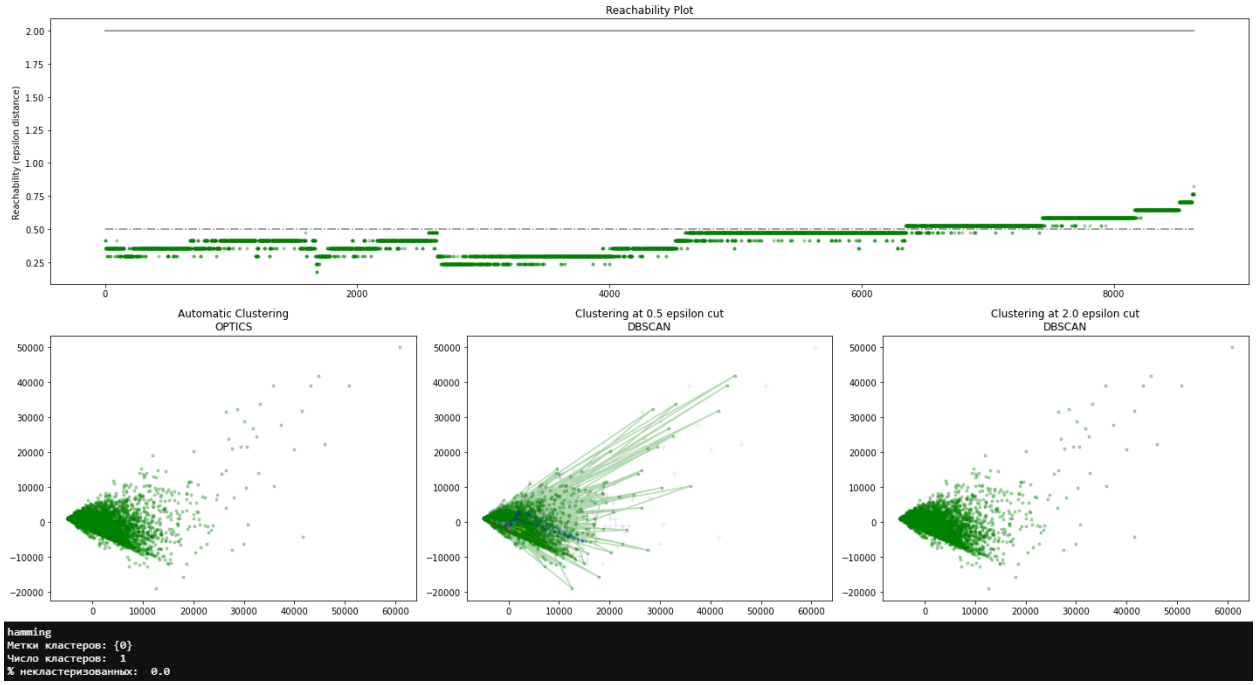

```
braycurtis
Метки кластеров: {0}
Число кластеров:  1
% некластеризованных:  0.0
```




```
canberra
Метки кластеров: {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, 36, 37, 38, 39, 40, 41, 42, 43, 44, 45, 46, 47,
48, 49, 50, 51, 52, 53, 54, 55, 56, 57, 58, 59, -1}
Число кластеров:  60
% некластеризованных:  46.41037517369153
```




```
chebyshev
Метки кластеров: {0, 1, -1}
Число кластеров:  2
% некластеризованных:  1.331635016211209
```




```
hamming
Метки кластеров: {0}
Число кластеров:  1
% некластеризованных:  0.0
```

Вывод

В результате выполнения лабораторной работы были изучены методы кластеризации данных из библиотеки Sklearn.