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 _____ Ястребков А. С.

Преподаватель _____ Жангиров Т. Р.

Санкт-Петербург

2020

Цель работы:

Ознакомиться с методами кластеризации модуля Sklearn.

Ход работы

Загрузка данных.

Был загружен датасет CC_GENERAL.csv (фрагмент исходного датасета показан на рис. 1).

	BALANCE	BALANCE_FREQUENCY	PURCHASES	ONEOFF_PURCHASES	INSTALLMENTS_PURCHASES	CASH_ADVANCE	PURCHASES_FREQUENCY	ON
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	
6	627.260806	1.000000	7091.01	6402.63	688.38	0.000000	1.000000	
7	1823.652743	1.000000	436.20	0.00	436.20	0.000000	1.000000	
8	1014.926473	1.000000	861.49	661.49	200.00	0.000000	0.333333	

Рис. 1. Фрагмент исходного датасета.

1. Для кластеризации DBSCAN необходимо нормировать данные, поскольку они имеют различный разброс. Для нормировки была использована функция StandardScaler пакета SKLearn, которая вычитает среднее значение и маштабирует данные до единичной дисперсии. После чего данные были кластеризованы алгоритмом DBSCAN, результаты:

Labels (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

Процент неклассифицированных данных (шум): 75.127%

Алгоритм DBSCAN принимает следующие параметры:

- eps — максимальное расстояние между элементами, при котором они всё ещё считаются соседними (при этом не является максимальным расстоянием точек в кластере);
- min_samples — число соседей точки, которое необходимо, чтобы считать её основной;

- metric — метрика для вычисления расстояния между точками;
- metric_params — дополнительные аргументы метрики;
- algorithm — алгоритм для вычисления межточечного расстояния и поиска ближайших соседей;
- leaf_size — размер листа, передаваемый в BallTree или cKDTree;
- p — степень метрики Минковского;
- n_jobs — число параллельно выполняемых потоков.

2. На рис. 2 показана зависимость количества формируемых DBSCAN кластеров и процента некластеризованных точек от максимальной рассматриваемой дистанции.

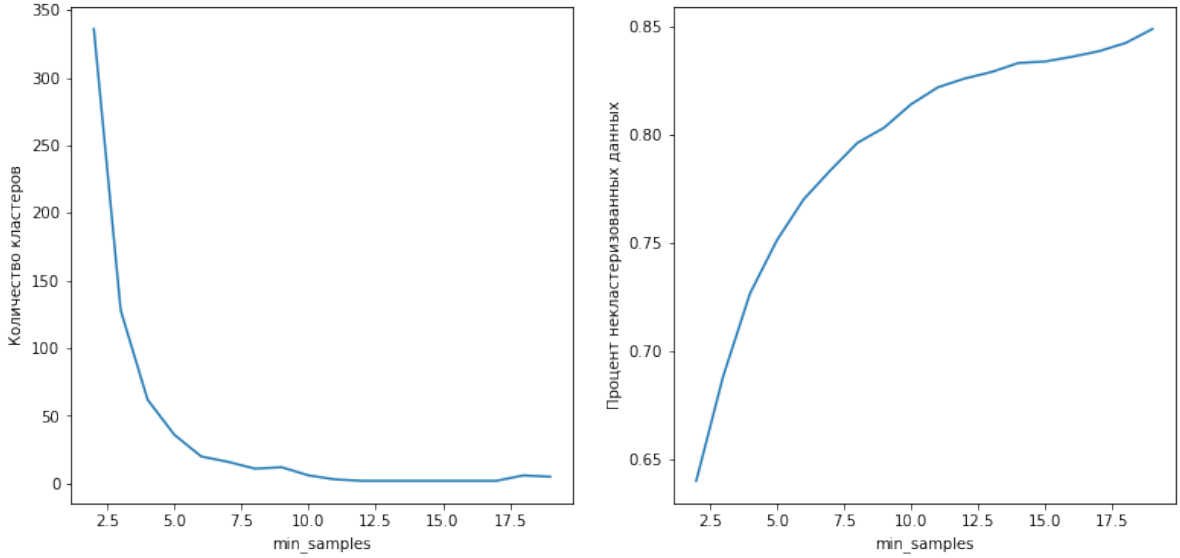

Рис. 2. Зависимость количества кластеров и процента некластеризованных данных от параметра min_samples.

3. На рис. 3 показана зависимость количества формируемых DBSCAN кластеров и процента некластеризованных точек от минимального значения количества точек, образующих кластер.

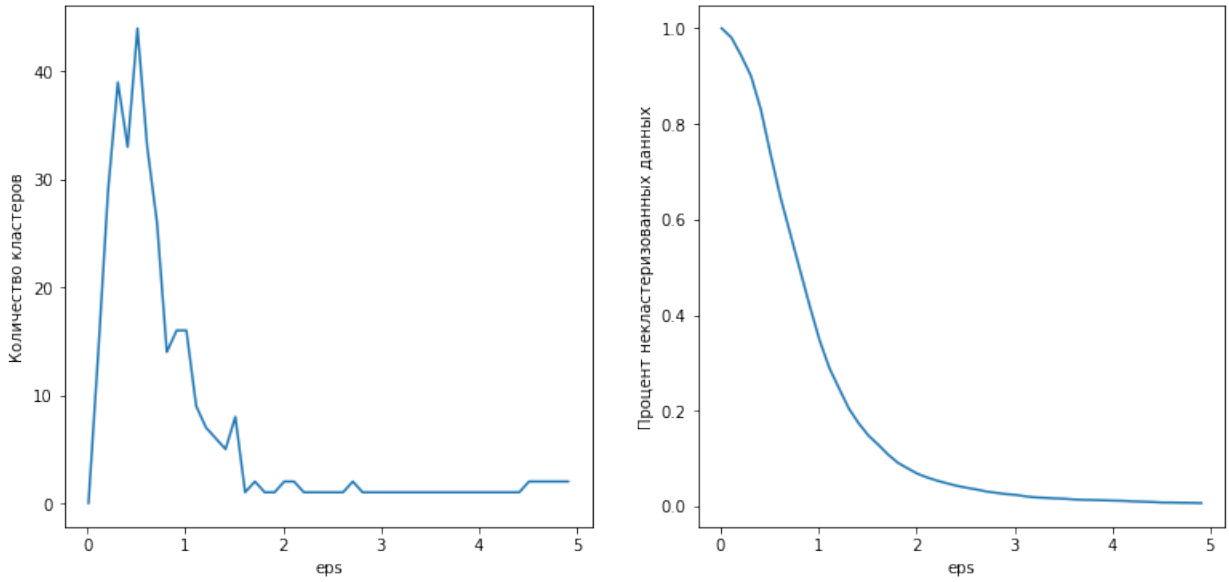

Рис. 3. Зависимость количества кластеров и процента некластеризованных данных от параметра eps.

4. Экспериментально были определены параметры, при которых алгоритм выдаёт 5-7 кластеров при объёме некластеризованных данных не более 12%, код для поиска и результат приведены в листинге 1.

Листинг 1 — Поиск параметров для получения 5-7 кластеров с процентом некластеризованных данных менее 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_ -> n_clusters, non_clustered')
for key, value in info.items():
    if value[0]>=5 and value[0]<=7 and value[1]<=0.12:
        print(key, value)

OUTPUT:
samples, eps_          -> n_clusters, non_clustered
(3, 2.0000000000000004)    [6, 0.06287633163501621]
```

5. При помощи алгоритма главных компонент (PCA) было выполнено понижение размерности пространства до двух и визуализирована кластеризация алгоиртмом с параметрами, рассчитанными в п. 5. Результат показан на рис. 4. Видно, что большая часть точек отнесена к одному кластеру.

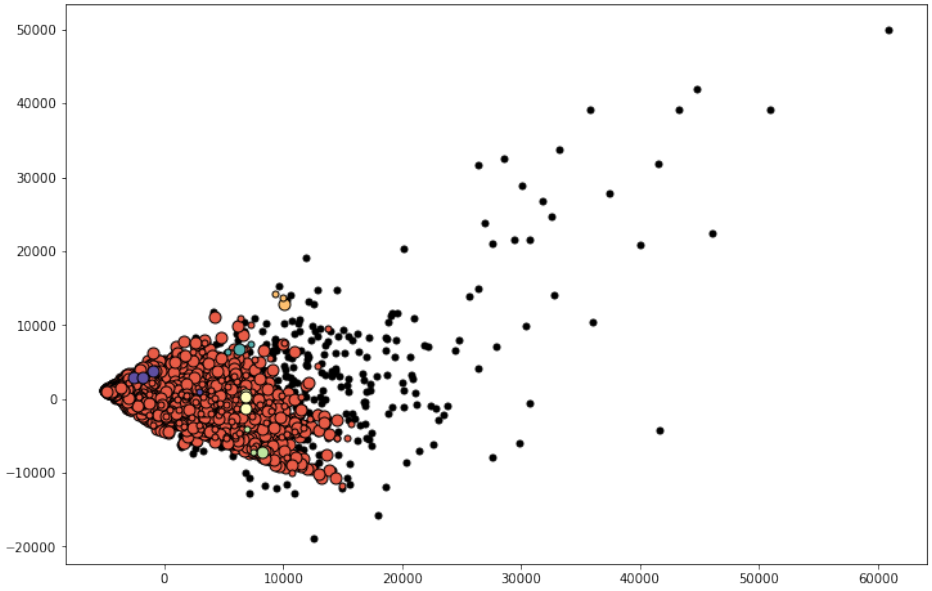

Рис. 4. Визуализация кластеризации алгоритмом DBSCAN.

6. Алгоритм кластеризации OPTICS в реализации библиотеки SKLearn принимает следующие параметры:

- min_samples — минимальное число точек в оскрестности точки, при котором она считается основной;
- max_eps — максимальное расстояние, допускающее сходство между точками;
- metric — метрика для вычисления расстояния между тчоками;
- p — параметр метрики Минковского (при p=1 равносильно использованию manhattan_distance, при p=2 — эвклидовому расстоянию);
- metric_params — дополнительные параметры метрики;
- cluster_method — метод извлечения кластеров на основании вычисленных достижимости и порядка;
- eps — максимальная дистанция между двумя сосдними точками;

- xi — минимальная крутизна на графике достижимости, обозначающая границу кластера, используется при cluster_method='xi’;
- predecessor_correction — коррекция кластеров по предшественникам, используется при cluster_method='xi’;
- min_cluster_size — минимальное количество точек в кластере OPTICS;
- algorithm — алгоритм поиска ближайших соседей;
- leaf_size — размер листа, передаваемый в BallTree или cKDTree;
- n_jobs — число параллельно выполняемых потоков.

7. Получить результаты, аналогичные п. 4, алгоритмом OPTICS удалось лишь при использовании метода кластеризации dbscan, при этом результаты:

Labels (метки кластеров):

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

Число кластеров: 6

Процент неклассифицированных данных (шум): 6.3108%

В отличие от алгоритма DBSCAN, OPTICS сохраняет иерархию кластеров с различным расстоянием соседства. Для этого, кроме основного расстояния от точки до её ближайших соседей, вычисляется достижимое расстояние, равное:

$$reachability-dist_{\epsilon,MinPts}(o,p)=\begin{cases} UNDEFINED & |N_\epsilon(p)|<MinPts \\ max(core-dist_{\epsilon MinPts}(p),dist(p,o)) & |N_\epsilon(p)|\geq MinPts \end{cases},$$

где

$$core-dist_{\epsilon,MinPts}(o,p)=\begin{cases} UNDEFINED & при\ |N_\epsilon(p)|<MinPts \\ MinPts-й\ наименьшее\ в\ N_\epsilon(p) & при\ |N_\epsilon(p)|\geq MinPts \end{cases}.$$

8. Была получена визуализация п. 7, показанная на рис. 5.

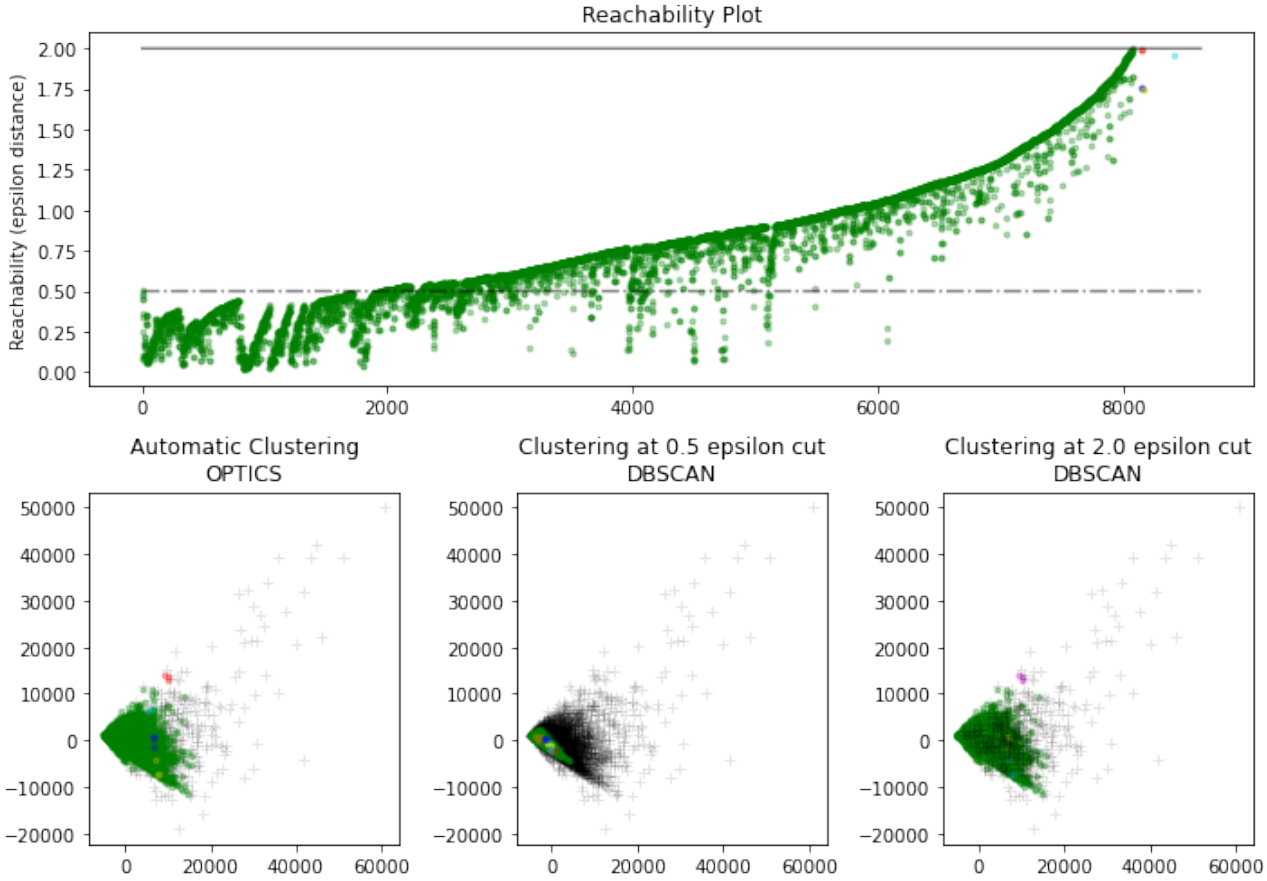

Рис. 5. Визуализация работы алгоритма OPTICS.

9. Были исследованы результаты кластеризации для различных метрик, результаты сведены в таблицу 1. Из таблицы видно, что ни одна метрика не позволила показать адекватные результаты кластеризации.

Таблица 1 — Сравнение различных метрик алгоритма OPTICS

метрика	**дополнительные параметры**	**кластеров**	**не кластеризовано, %**	**самый большой кластер, %**
l1	—	99	90.922	0.2894
	min_samples=5	99	90.922	0.2895
l2	—	112	90.111	0.2663
	min_samples=10	23	95.299	0.4516
	min_samples=50, max_eps=10	1	99.062	0.938
manhattan	—	99	90.922	0.2895
	min_samples=10	21	95.982	0.3126
	min_samples=15,	7	98.089	0.3821

	max_eps=1			
chebyshev	—	141	87.019	0.3011
	min_samples=10	28	91.477	0.7063
	min_samples=10, max_eps=1	140	87.147	0.3011
cosine	—	179	83.291	0.2895
	min_samples=25	8	93.446	1.4011
	min_samples=30, max_eps=3	11	91.535	1.4706

Вывод:

В результате выполнения лабораторной работы были изучены алгоритмы кластеризации DBSCAN и OPTICS модуля SKLearn. На выбранном датасете оба алгоритма показали неудовлетворительные результатыЮ либо выделяя большую часть транзакций в один кластер, либо не классифицируя большую часть данных. Предположительно, такое поведение связано со структурой датасета и его объёмом.