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. 8304 |  | Сергеев А.Д. |
| Преподаватель |  | Жангиров Т.Р. |

Санкт-Петербург 2021

# Цель работы

Ознакомиться с методами кластеризации модуля *Sklearn*.

# Ход работы

**Загрузка данных**

Данные загружены в датафрейм.

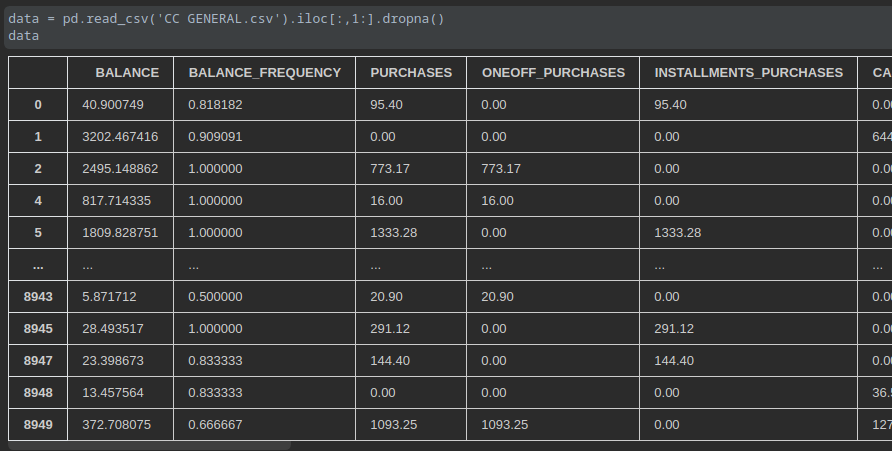
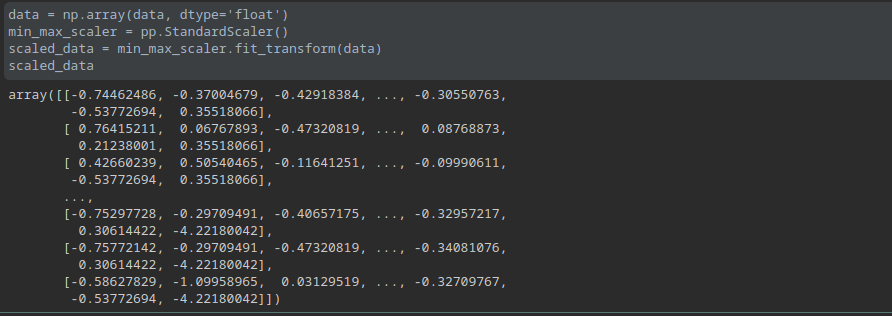


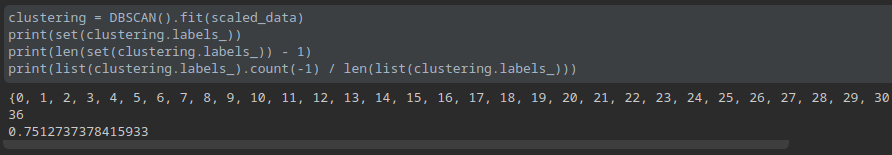
Рисунок 1 – Исходные данные

# DBSCAN

1. Проведена кластеризация методом k-средних. Данные стандартизированы, т.к. лежат в разных шкалах.

Рисунок 2 – Данные после стандартизации

1. Проведена кластеризация методом DBSCAN при параметрах по умолчанию.

Рисунок 3 – Метко кластеров, их количество и процент нераспределенных элементов

|  |  |
| --- | --- |
| Параметр | Описание |
| eps: float, default=0.5 | Максимальное расстояние между двумя наблюдениями, чтобы один считался соседним с другим (радиус окрестности наблюдения). |
| min\_samples: int,  default=5 | Минимальное количество наблюдений вокрестности точки, чтобы считать ее базовой (включая саму точку). |
| metric: string or  callable, default=’euclidean’ | Метрика для вычисления расстояния между экземплярами в массиве признаков. |
| **metric\_params:** *dict, default=None* | Дополнительные аргументы ключевых слов для функции метрики. |
| algorithm: {‘auto’, ‘ball\_tree’, ‘kd\_tree’, ‘brute’},  default = ‘auto’ | Алгоритм, который будет использоваться для вычисления точечных расстояний и поиска ближайших соседей. |
| **leaf\_size:** *int, default=30* | Размер листа. Это может повлиять на скорость построения и запроса, а также на память, необходимую для хранения дерева. Оптимальное значение зависит от задачи. |
| **P:** *float, default=None* | Мощность метрики, используемой для вычисления расстояния между точками. Если *None*, то эквивалентно евклидову расстоянию. |
| **n\_jobs:** *int, default=None* | Количество параллельных процессов для выполнения. |

1. Построен график зависимости количества кластеров и процента не кластеризованных наблюдений от максимальной рассматриваемой дистанции.

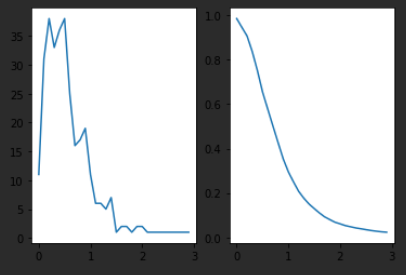
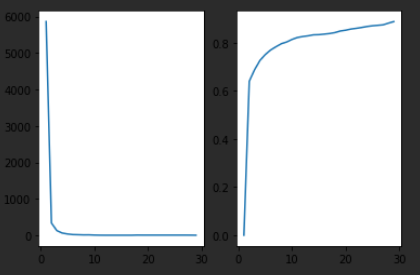


Рисунок 4 – Зависимост количества кластеров и процента выпавших наблюдений от eps

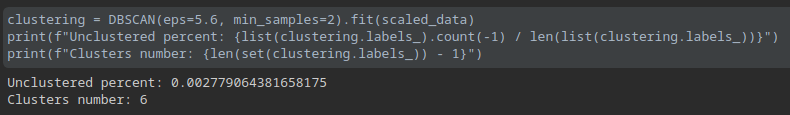
С увеличением eps все больше точек попадают в какой-либо кластер, в конце все точки будут находится в одном кластере.

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

Рисунок 5 – Зависимост количества кластеров и процента выпавших наблюдений от min\_samples

Увеличение min\_samples приводит к увеличению количества выпавших наблюдений и к уменьшению количества кластеров.

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

Рисунок 6 – значения eps и min\_samples, при которых достигается требуемый процент не кластеризованных измерений и количество кластеров.

1. Размерность данных понижена до 2 с помощью метода главных компонент.

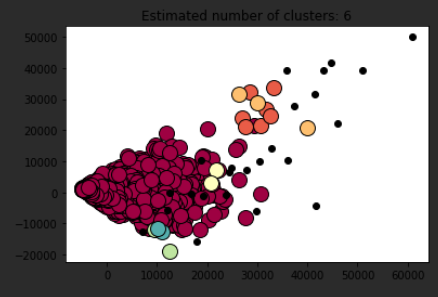
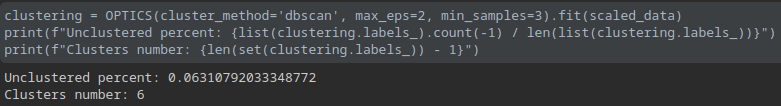


Рисунок 7 – Кластеризация данных пониженной размерности

# OPTICS

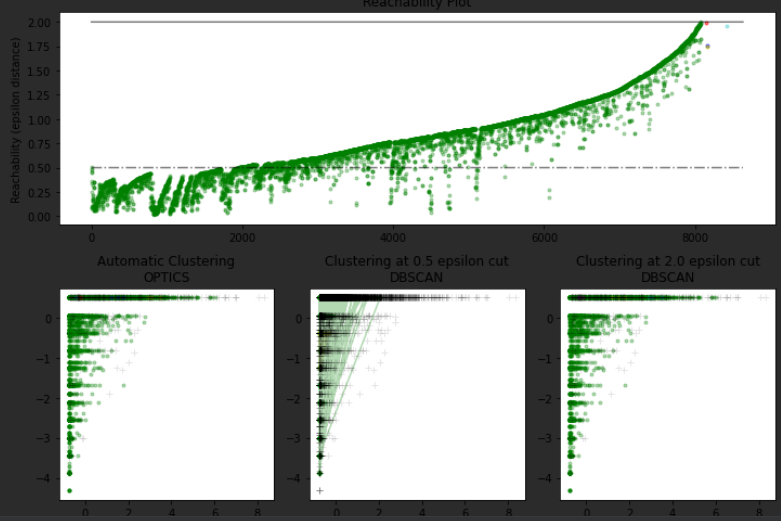
|  |  |
| --- | --- |
| Параметр | Описание |
| max\_eps: float, default=inf | Максимальное расстояние между двумя наблюдениями, чтобы одно считалось соседним с другим (радиус окрестности наблюдения). |
| min\_samples: int>1 or  float in [0, 1], default=5 | Количество наблюдений в окрестности точки, чтобы считать ее базовой. |
| metric: string or  callable, default=’minkowski’ | Метрика для вычисления расстояния. |
| p: int, default = 2 | Параметр для метрики Минковского. |
| cluster\_method: string,  default = ‘xi’ | Метод извлечения кластеров. Также можно поставить 'dbscan'. |
| eps | Максимальное расстояние между двумя наблюдениями, чтобы один считался соседним с другим (радиус окрестности наблюдения). Нужен только при cluster\_method=dbscan. |
| xi: float in [0,1], default=0.05 | Определяет минимальную крутизну на графике достижимости, который составляет границу кластера. |
| predecessor\_correction: bool, default=True | Коррекция кластеров в соответствии с предшественниками, рассчитанными OPTICS. Этот параметр оказывает минимальное влияние на большинство наборов данных. Используется только  когда cluster\_method = 'xi'. |
| min\_cluster\_size: int>1 or float in [0, 1], default=None | Минимальное количество выборок в кластере OPTICS, выраженное в виде абсолютного числа или доли от количества выборок (округленное до не менее 2). Если None, вместо этого используется значение min\_samples. Используется только когда cluster\_method = 'xi'. |
| algorithm: {‘auto’, ‘ball\_tree’, ‘kd\_tree’, ‘brute’}, default =  ‘auto’ | Алгоритм, который будет использоваться для вычисления точечных расстояний и поиска ближайших соседей. |
| **leaf\_size:** *int, default=30* | Размер листа. Это может повлиять на скорость построения и запроса, а также на память, необходимую для хранения дерева. Оптимальное значение зависит от задачи. |
| **Memory:** *str or object with the joblib.Memory interface, default=None* | Путь к каталогу кэширования. Используется для кэширования выходных данных вычисления дерева. По умолчанию кэширование не выполняется. |
| **n\_jobs:** *int, default=None* | Количество параллельных заданий, выполняемых для поиска соседей. |

1. Результаты кластеризации методом OPTICS близки к результатам DBSCAN при cluster\_method=dbscan, max\_eps=2, min\_samples=3.

Рисунок 7 – значения eps и min\_samples, при которых достигается требуемый процент не кластеризованных измерений и количество кластеров.

Процесс определения базовых точек в OPTICS идентичен DBSCAN, однако в OPTICS для точек вычисляются и сохраняются расстояния достижимости, на основе которых наблюдения выстраиваются в кластере, сохраняя при этом иерархическую структуру.

1. Построен график достижимости

Рисунок 4 – График достижимости

1. Метод OPTICS исследован с использованием различных метрик

|  |  |  |
| --- | --- | --- |
| Метрика | Количество кластеров | Процент выпавших наблюдений |
| canberra | 106 | 0.88722 |
| chebyshev | 143 | 0.86869 |
| manhattan | 99 | 0.90840 |
| euclidean | 111 | 0.89961 |
| sqeuclidean | 146 | 0.86962 |

# Выводы

В ходе лабораторной работы изучены такие методы кластеризации модуля *Sklearn*, как DBSCAN и OPTICS. При cluster\_method=“xi“ OPTICS разделяет данные на большое число кластеров, малое количество кластеров достигается только при большом количестве выпавших наблюдений.