Задача кластеризации

Кластеризация относится к алгоритмам обучения без учителя. Исходные данные не имеют меток.

Кластер – это подмножество объектов в выборке, которые похожи друг на друга в пространстве. Метрику сходства необходимо выбирать под конкретную задачу, но чаще всего используется евклидово расстояние. Когда мы работаем с новыми данными, которые не размечены, кластеризация может быть хорошим способом получить некоторое представление о данных. Алгоритмы кластеризации используются для различных задач, например, обнаружение аномалий, выбросов в данных, сжатие данных, обнаружение структуры данных, восстановление недостающих данных и тд. Некоторые реальные приложения кластеризации обнаруживают мошенничества в страховании, сегментируют клиентов, проводят анализ землетрясений или городского планирования. Алгоритмы кластеризации используются в биологии, например, для описания и проведения пространственных и временных сравнений комплексов организмов.

Существуют множество алгоритмов кластеризации, но самые популярные из них – это k-means (k-средних), иерархический кластерный анализ и алгоритм кластеризации, основанной на плотности (DBSCAN). Рассмотрим данные алгоритмы подробнее.

**K-means** – является наиболее часто используемым алгоритмом кластеризации. Именно с него начинается знакомство с данным разделом машинного обучения.

K-means работает на основе центроидов. Центроид – это центр кластера. Алгоритм k-means:

1. Берутся случайные точки в пространстве, которые принимаются за центры кластеров (центроиды 𝐶𝑖).
2. Для каждого объекта в выборке находится ближайший к нему

центроид.

1. Каждому центроиду соответствует множество ближайших объектов. Для каждого образованного кластера находится его центр по формуле:

1

𝑐𝑖 = 𝑁

𝑁𝑖

∑ 𝑥𝑗,

𝑖 𝑗=1

где 𝑁𝑖 – количество объектов в кластере.

1. Центроиды переходят в найденный центр кластера. Далее алгоритм повторяется, пока центроиды не перестанут менять положение.

Функция стоимости алгоритма:

𝑘

𝑁𝑖

𝐿(𝐶1, 𝐶2,…, 𝐶𝑘) = ∑ ∑‖𝑥𝑗 − 𝐶𝑖‖ → 𝑚𝑖𝑛,

𝑗=1

𝑖=1

где 𝐶𝑖 – центры кластеров;

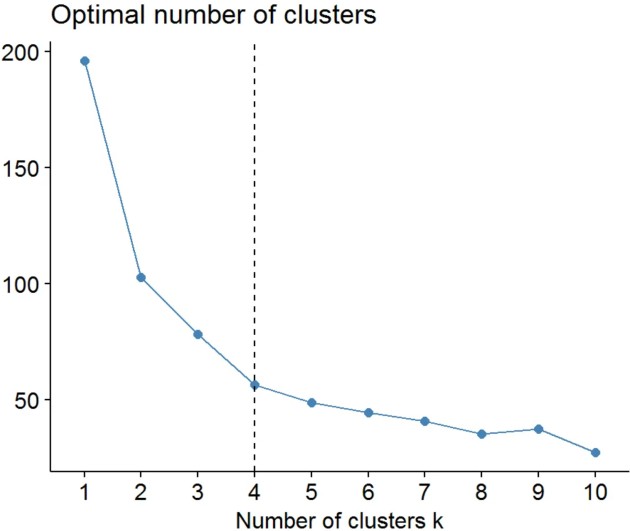
k – количество кластеров;

𝑁𝑖 – количество объектов в кластере;

𝑥𝑗 – объекты в кластере.

Целью алгоритма является минимизация дисперсии внутри кластера.

Одни и те же объекты можно по-разному разбить на кластеры, все зависит от начального положения центроидов. Чаще всего выбирается k случайных объектов из выборки, которые становятся центроидами. Можно запустить алгоритм несколько раз и посмотреть, какой результат будет лучше. Выбор количества кластеров k обусловлен решаемой задачей. Если неизвестно, какое количество кластеров использовать, то используется так называемое «правило сломанной трости» или «правило локтя». Строится график зависимости функции стоимости от числа кластеров. С ростом числа кластеров значение функции стоимости уменьшается. Количество кластеров,

где происходит «перелом» графика, берут в качестве k.

Но не всегда такой метод позволяет понять, какое же количество кластеров взять. Почему на предыдущем графике выбор пал на k=4? Почему не 5?

Существует еще способ определения количества кластеров – коэффициент силуэта. Он наиболее понятен и аргументирован, так как

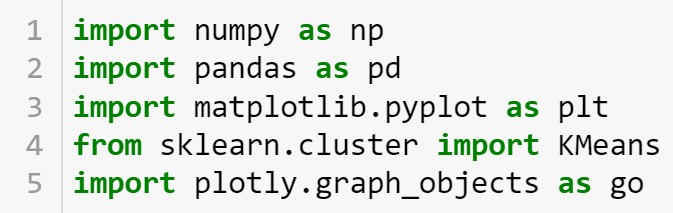
коэффициент силуэта демонстрирует пиковое значение, нежели плавный изгиб в методе локтя. Коэффициент силуэта рассчитывается с использованием среднего внутрикластерного расстояния a и среднего расстояния до ближайшего кластера b для каждой выборки. Коэффициент силуэта s для выборки по формуле:

𝑏 − 𝑎

𝑠 = 𝑚𝑎𝑥(𝑎, 𝑏)

Минусы алгоритма:

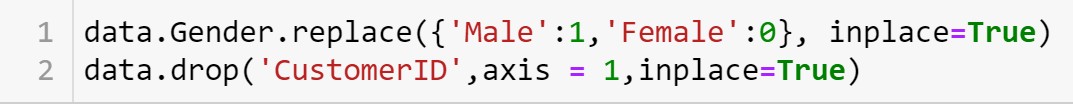
* необходимо выбирать количество кластеров;
* если объектов очень много, то требуется очень много расчетов, что занимает много времени. Именно поэтому лучше использовать k-means на небольших наборах.

Рассмотрим пример кластеризации для данных о клиентах магазина.



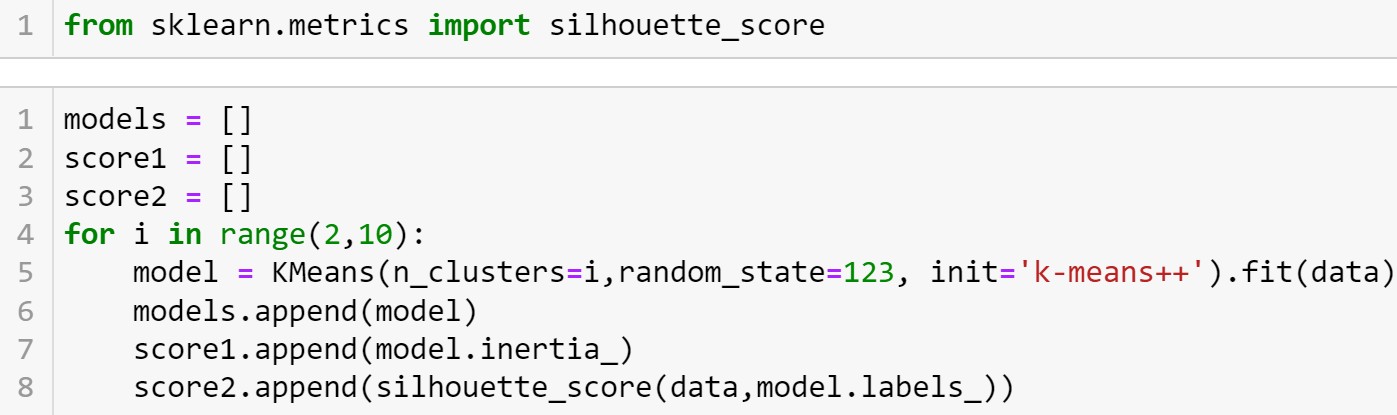
Датафрейм содержит id клиента, пол, возраст, годовой доход и оценку расходов.

Предварительно удалим столбец id клиента и приведем столбец Gender к числовым значениям.

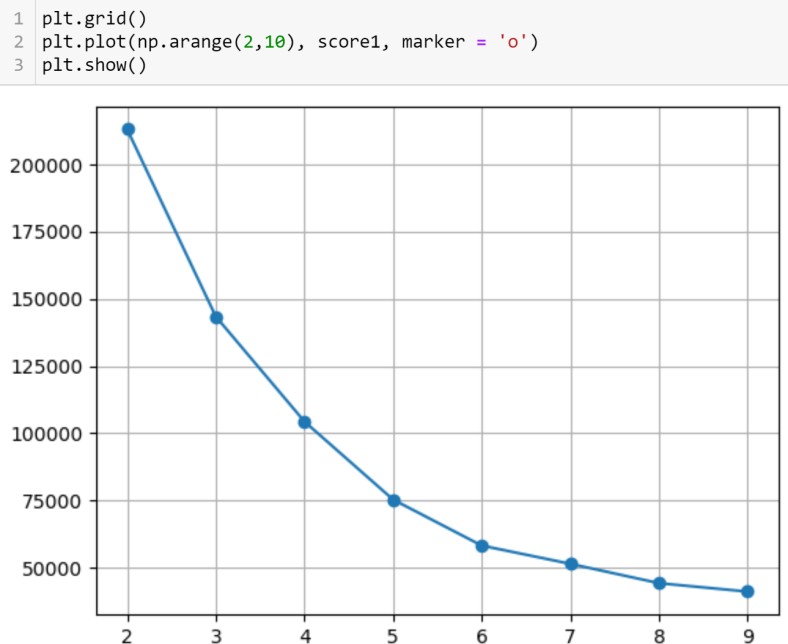


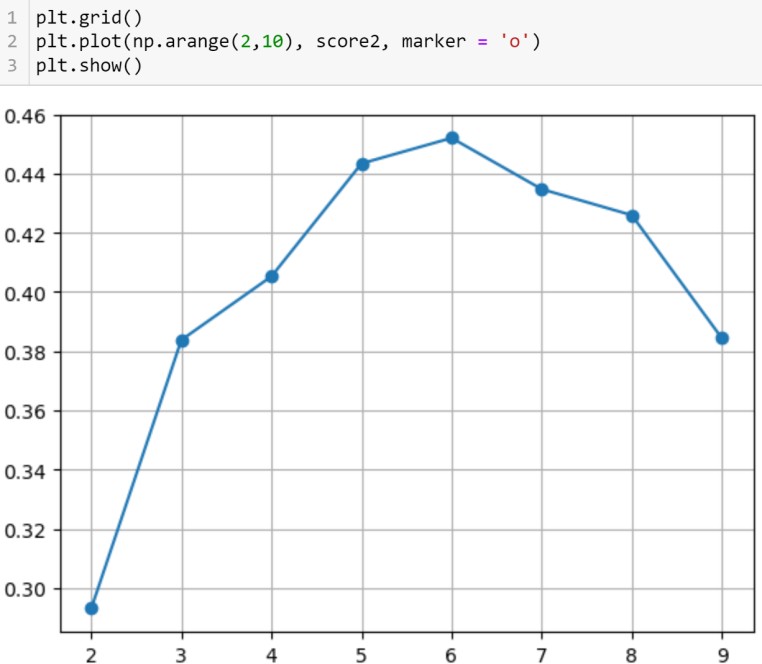
Используя библиотеку sklearn проведем кластеризацию данных. Обучим 10 моделей с разным количеством кластеров и построим график зависимости

функции стоимости от числа кластеров, и коэффициента силуэта от числа кластеров.

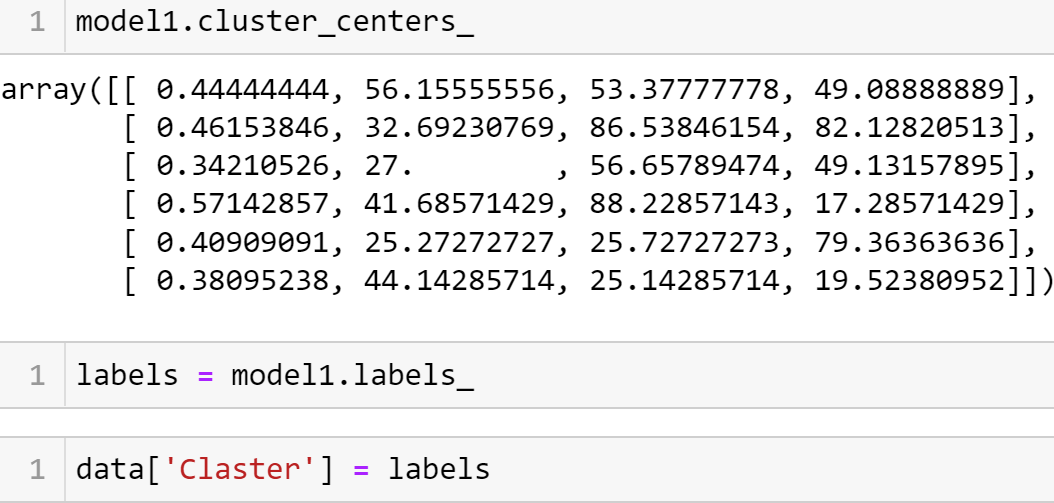
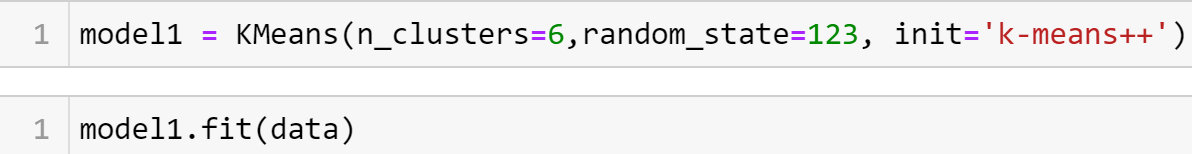
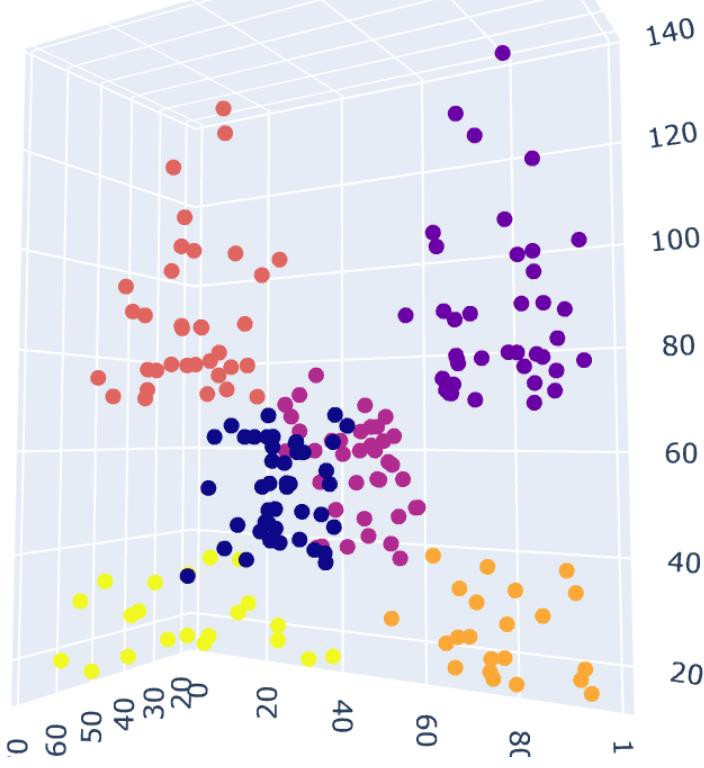
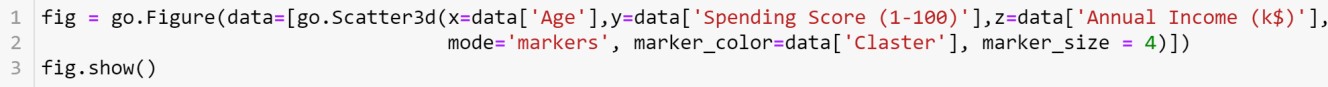


score1 – это значения функции стоимости, score2 – это значения коэффициента силуэта. model.labels\_ – это номера кластеров, которые присваиваются объектам в data.

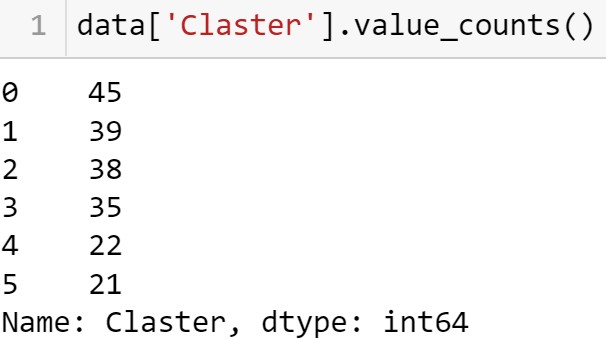




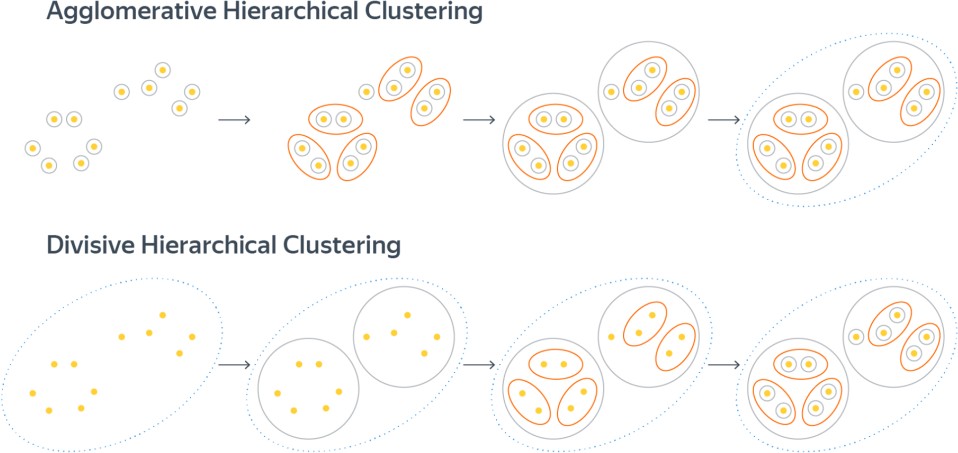
Коэффициента силуэта достигает максимума при k = 6. Результат кластеризации данных для 6 кластеров.



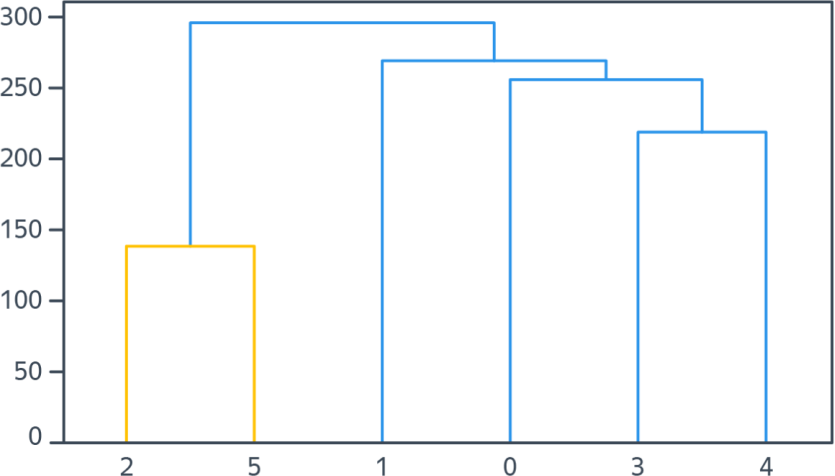
model1.cluster\_centers\_ – это координаты центров кластеров в пространстве. Количество объектов в каждом кластере:



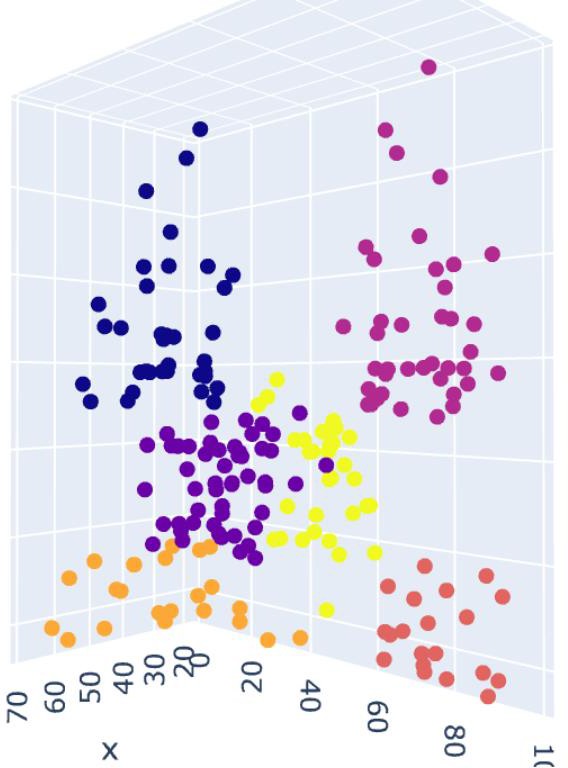
Следующий алгоритм кластеризации – это **иерархическая агломеративная кластеризация** (кластеры вложены друг в друга и образуют древовидную структуру).

Иерархическая кластеризация используются для того, чтобы определить взаимосвязи между объектами. Агломеративная означает, что алгоритм начинает работу с одного объекта и постепенно объединяет их в кластеры побольше. Второй вариант – это дивизионные алгоритмы. Такой алгоритм начинает с больших кластеров и постепенно их делит на кластеры поменьше.

Для начала вычисляется матрица расстояний между каждой парой объектов. Далее выбирается пара объектов с минимальным расстоянием и вычисляется среднее значение, после чего снова рассчитывается новая матрица расстояний, с учетом того, что пара объектов становится одним новым объектом. Таким образом алгоритм повторяется, пока не останется только одно значение.

В результате кластеризации строится дендрограмма, которая позволяется понять структуру исходного набора данных.

Ось абсцисс – номер объекта. Ось ординат – расстояние между объектами в момент слияния.

Пример:

**DBSCAN** (Density-based spatial clustering of applications with noise) основан на плотности. Алгоритм группирует вместе точки, которые тесно расположены, а выбросами помечает точки, которые находятся в областях с малой плотностью. Плотность в DBSCAN определяется в окрестности каждого объекта выборки 𝑥𝑖 как количество других точек выборки в шаре B(ε, 𝑥𝑖). В

качестве гиперпараметра кроме радиуса ε (окрестности шара) задается порог

N по количеству точек в окрестности.

Все объекты в пространстве делятся на 3 типа: основные точки, граничные точки и шумовые точки. Основные точки – это точки, в окрестности которых больше N точек. Граничные точки – это точки, в окрестности которых есть основные точки, но всего точек меньше N. И шумовые точки – это точки, в окрестности которых нет основных точек и всего точек меньше N.

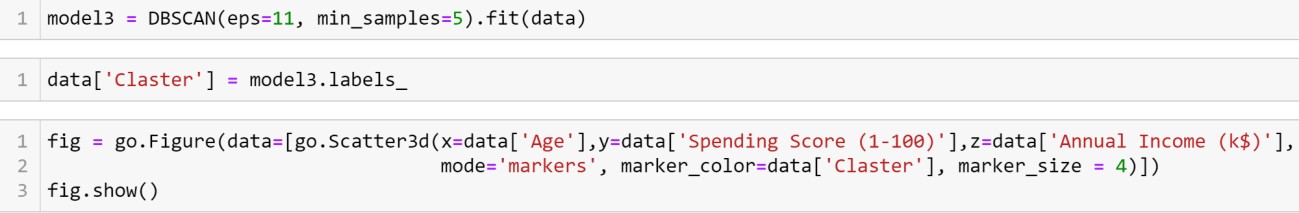
Шумовые точки убираются из рассмотрения и не приписываются ни к какому кластеру.

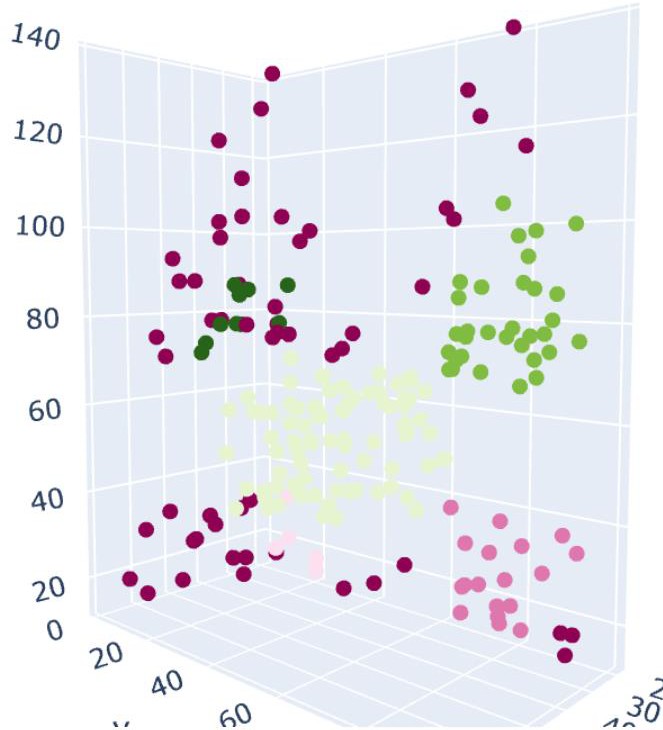
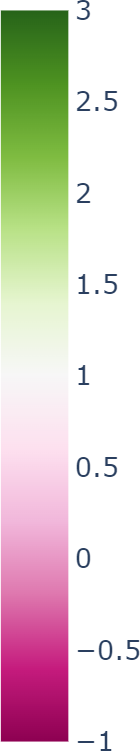
Основные точки, у которых есть общая окрестность, соединяются ребром. В полученном графе выделяются компоненты связности.

Каждая граничная точка относится к тому кластеру, в который попала ближайшая к ней основная точка.

Большим плюсом DBSCAN является то, что он сам выбирает количество кластеров, а также очень хорошо справляется со сложными формами кластеров. Это один из самых эффективных алгоритмов кластеризации, но работает он достаточно долго. К минусам алгоритма относят: неспособность соединять кластеры через проёмы, и, наоборот, способность связывать явно различные кластеры через плотно населённые перемычки.

Пример:



Номер кластера «-1» – это объекты, которые алгоритм выделил как шумовые.

**Практическое задание**

1. Найти данные для кластеризации. Данные в группе не должны повторяться. Если признаки в данных имеют очень сильно разные масштабы, то необходимо данные предварительно нормализовать.
2. Провести кластеризацию данных с помощью алгоритма k-means. Использовать «правило локтя» и коэффициент силуэта для поиска оптимального количества кластеров.
3. Провести кластеризацию данных с помощью алгоритма иерархической кластеризации.
4. Провести кластеризацию данных с помощью алгоритма DBSCAN.
5. Визуализировать кластеризованные данные с помощью t-SNE или UMAP, если необходимо. Если данные трехмерные, то можно использовать трехмерный точечный график.
6. Оформить отчет о проделанной работе. Сделать выводы.