# ЛЕКЦИЯ 8. КЛАСТЕРИЗАЦИЯ НА ОСНОВЕ ПЛОТНОСТИ. МЕТОД DBSCAN

ДИСЦИПЛИНА: «ИНТЕЛЛЕКТУАЛЬНЫЙ АНАЛИЗ ДАННЫХ»

# ЦЕЛЬ ЛЕКЦИИ

- Сформировать у обучающихся представление о кластеризации на основе плотности, изучить основные понятия (**ε-окрестность**, **MinPts**, ядровые, граничные точки и шум), алгоритм **DBSCAN**, способы подбора его параметров, а также научить применять метод на практике.

# ПЛАН ЛЕКЦИИ

- Ограничения центроидных методов кластеризации
- Идея кластеризации на основе плотности
- Основные понятия DBSCAN: ε, MinPts, типы точек, достижимость
- Алгоритм DBSCAN: схема и псевдокод
- Пошаговый пример
- Выбор параметров, сложность, достоинства и недостатки
- Варианты метода и реализация на Python

# ОГРАНИЧЕНИЯ ЦЕНТРОИДНЫХ МЕТОДОВ

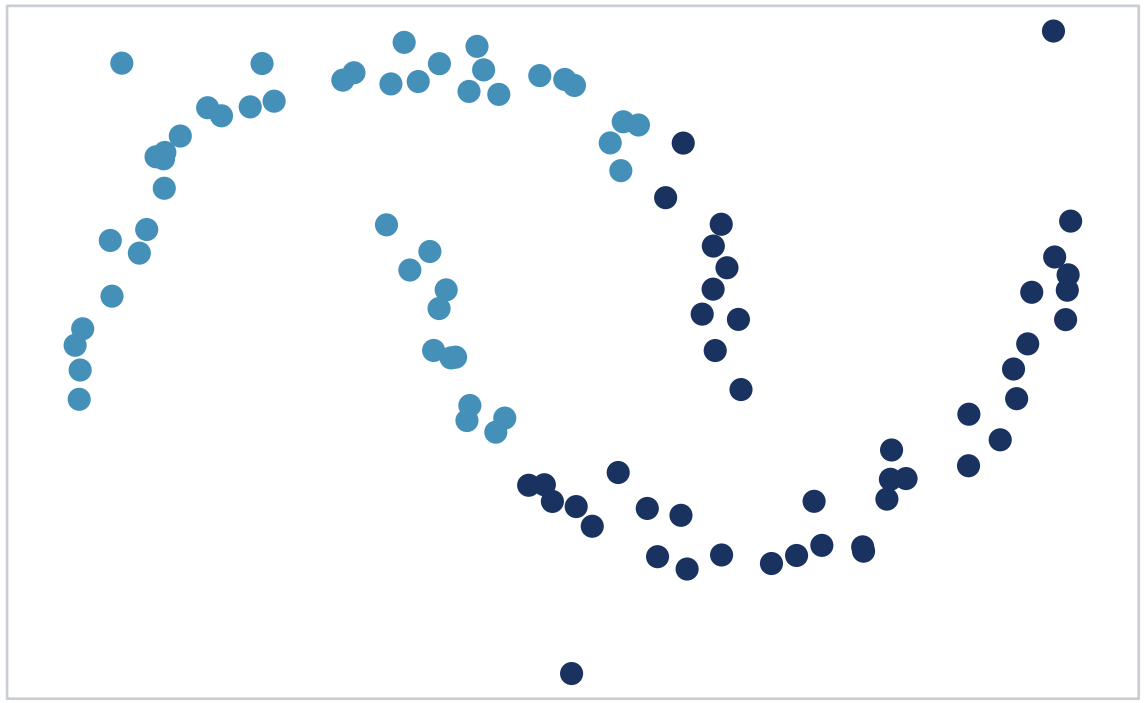
- **K-means** и другие центроидные методы ищут компактные кластеры шарообразной формы.
- Число кластеров k нужно задавать заранее.
- Каждая точка обязательно попадает в какой-либо кластер, поэтому выбросы искажают центры.
- Кластеры произвольной формы (полумесяцы, кольца, цепочки) такими методами находятся плохо.

# ИДЕЯ КЛАСТЕРИЗАЦИИ НА ОСНОВЕ ПЛОТНОСТИ

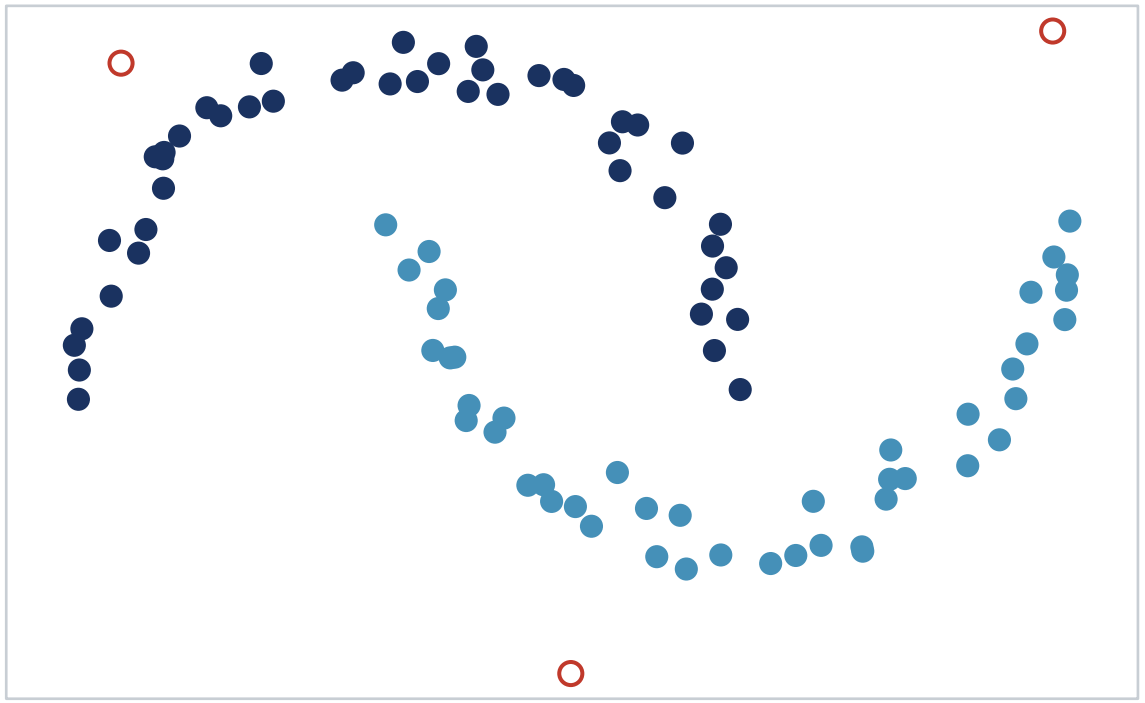
- **Кластер** — область пространства с высокой плотностью точек, отделённая от других кластеров областями низкой плотности.
- Плотность оценивается числом точек в окрестности заданного радиуса.
- Форма кластеров не ограничена, число кластеров определяется автоматически.
- Точки в областях низкой плотности считаются **шумом** (выбросами).

# K-MEANS И DBSCAN НА ДАННЫХ СЛОЖНОЙ ФОРМЫ

**K-means (k = 2): полумесяцы смешаны**



**DBSCAN (ε = 0,25; MinPts = 4): 2 кластера и шум**



Красные кружки без заливки — точки, помеченные как шум.

# АЛГОРИТМЫ НА ОСНОВЕ ПЛОТНОСТИ

| Алгоритм | Особенность |
|---|---|
| **DBSCAN** | Один глобальный порог плотности (ε, MinPts); основа семейства |
| **OPTICS** | Упорядочивает точки по достижимости; работает с разной плотностью |
| **HDBSCAN** | Иерархический вариант DBSCAN; не требует ε |
| **DENCLUE** | Плотность как сумма функций влияния точек |
| **Mean-shift** | Сдвиг к локальным максимумам оценки плотности |

# ОСНОВНЫЕ ПОНЯТИЯ: Ε И MINPTS

| Понятие | Определение |
|---|---|
| **ε (eps)** | Радиус окрестности точки |
| **ε-окрестность** | $N_\varepsilon(p) = \{ q \in D : dist(p, q) \leq \varepsilon \}$ — все точки в радиусе ε от p, включая саму p |
| **MinPts** | Минимальное число точек в ε-окрестности, чтобы область считалась плотной |
| **Плотная точка** | $\lvert N_\varepsilon(p) \rvert \geq MinPts$ |

# ТИПЫ ТОЧЕК

| Тип точки | Условие |
|---|---|
| **Ядровая (core)** | $\|N_\varepsilon(p)\| \geq MinPts$ — в окрестности достаточно точек |
| **Граничная (border)** | Не ядровая, но лежит в ε-окрестности хотя бы одной ядровой точки |
| **Шум (noise)** | Ни ядровая, ни граничная: в окрестности нет ядровых точек |

- Кластер строится вокруг ядровых точек; граничные точки лишь присоединяются к нему.

# ДОСТИЖИМОСТЬ И СВЯЗНОСТЬ ПО ПЛОТНОСТИ

| Отношение | Определение |
|---|---|
| **Прямо достижима по плотности** | q прямо достижима из p, если $q \in N_\varepsilon(p)$ и p — ядровая точка |
| **Достижима по плотности** | Существует цепочка $p = p_1, \ldots, p_n = q$, где каждая $p_{i+1}$ прямо достижима из $p_i$ |
| **Связана по плотности** | p и q связаны, если существует точка o, из которой достижимы и p, и q |

# ОПРЕДЕЛЕНИЕ КЛАСТЕРА И ШУМА

**Кластер C — непустое множество точек, для которого выполнены два условия**

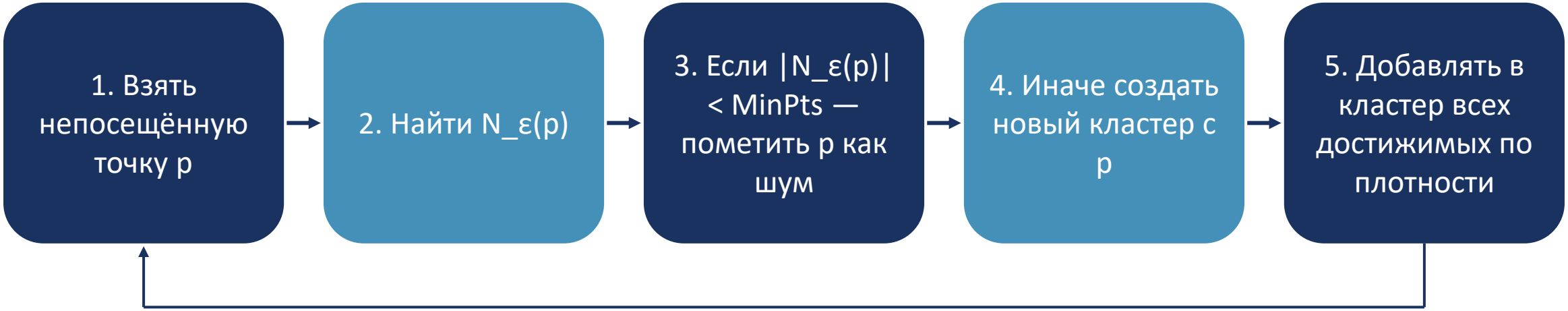
1. Максимальность: если $p \in C$ и $q$ достижима по плотности из $p$, то $q \in C$.
2. Связность: любые две точки $p, q \in C$ связаны по плотности.

- Шум — точки, не принадлежащие ни одному кластеру.
- Граничная точка, достижимая из ядер двух кластеров, достаётся тому, который был построен первым.

# АЛГОРИТМ DBSCAN: ИДЕЯ

- Предложен М. Эстером, Х.-П. Кригелем, Й. Сандером и К. Сюй в 1996 г. (**D**ensity-**B**ased **S**patial **C**lustering of **A**pplications with **N**oise).
- Параметры: радиус **ε** и порог **MinPts**; число кластеров не задаётся.
- Алгоритм перебирает точки, находит ядровую и «разращивает» кластер, добавляя все достижимые по плотности точки.
- Точки, не попавшие ни в один кластер, остаются шумом.

# СХЕМА РАБОТЫ DBSCAN



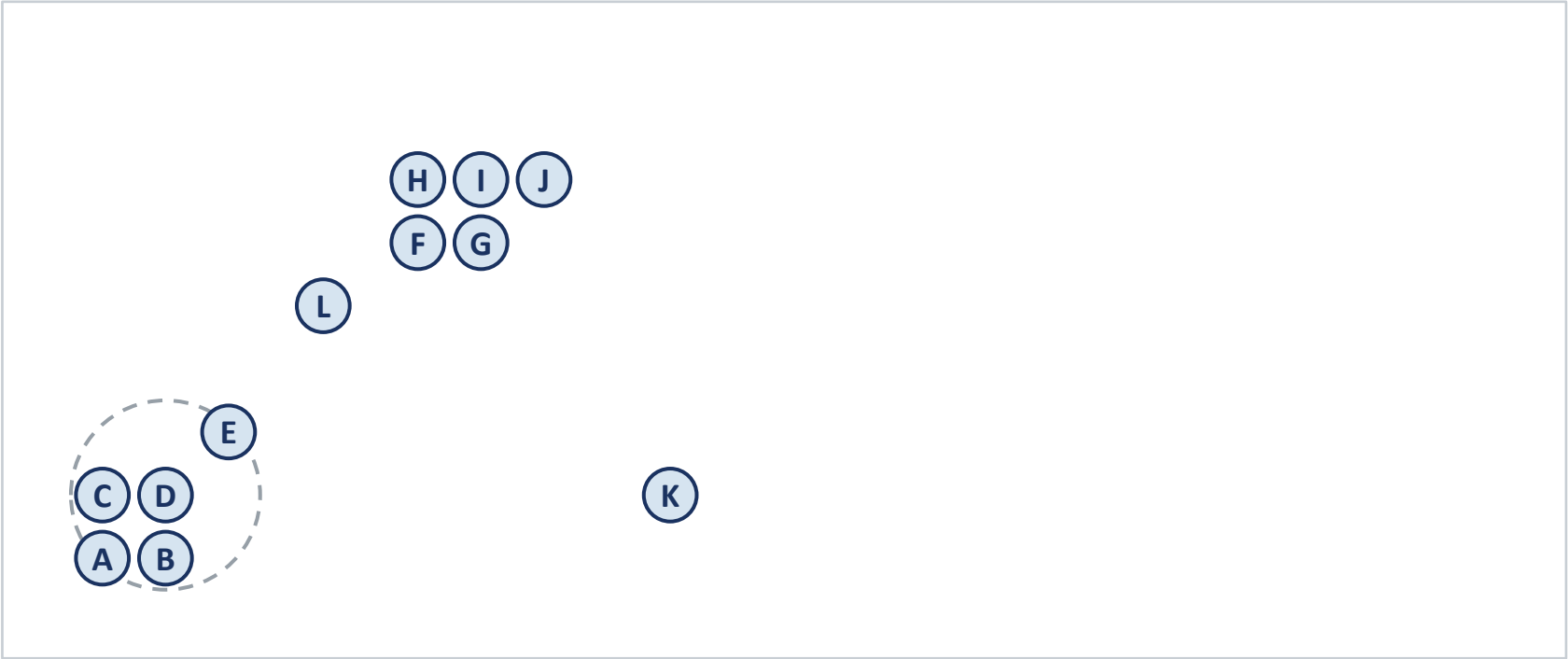

**Когда посещены все точки, остаются кластеры $C_1$, …, C_m и шум.**

# ПСЕВДОКОД АЛГОРИТМА DBSCAN

```
DBSCAN(D, eps, MinPts):
    C = 0
    для каждой точки p из D:
        если p уже посещена: пропустить
        пометить p как посещённую
        N = соседи(p, eps)
        если |N| < MinPts: пометить p как шум
        иначе:
            C = C + 1; добавить p в кластер C
            очередь Q = N
            пока Q не пуста:
                q = взять из Q
                если q шум: сделать q граничной точкой C
                если q уже в кластере: пропустить
                добавить q в кластер C; пометить q посещённой
                Nq = соседи(q, eps)
                если |Nq| >= MinPts: добавить Nq в Q
```

# ПРИМЕР: ИСХОДНЫЕ ДАННЫЕ

| Точка | x | y |
|---|---|---|
| A | 1 | 1 |
| B | 2 | 1 |
| C | 1 | 2 |
| D | 2 | 2 |
| E | 3 | 3 |
| F | 6 | 6 |
| G | 7 | 6 |
| H | 6 | 7 |
| I | 7 | 7 |
| J | 8 | 7 |
| K | 10 | 2 |
| L | 4,5 | 5 |




**Параметры: ε = 1,5; MinPts = 3 (считая саму точку). Пунктир — ε-окрестность точки D.**

# ШАГ 1: Ε-ОКРЕСТНОСТИ И ЯДРОВЫЕ ТОЧКИ

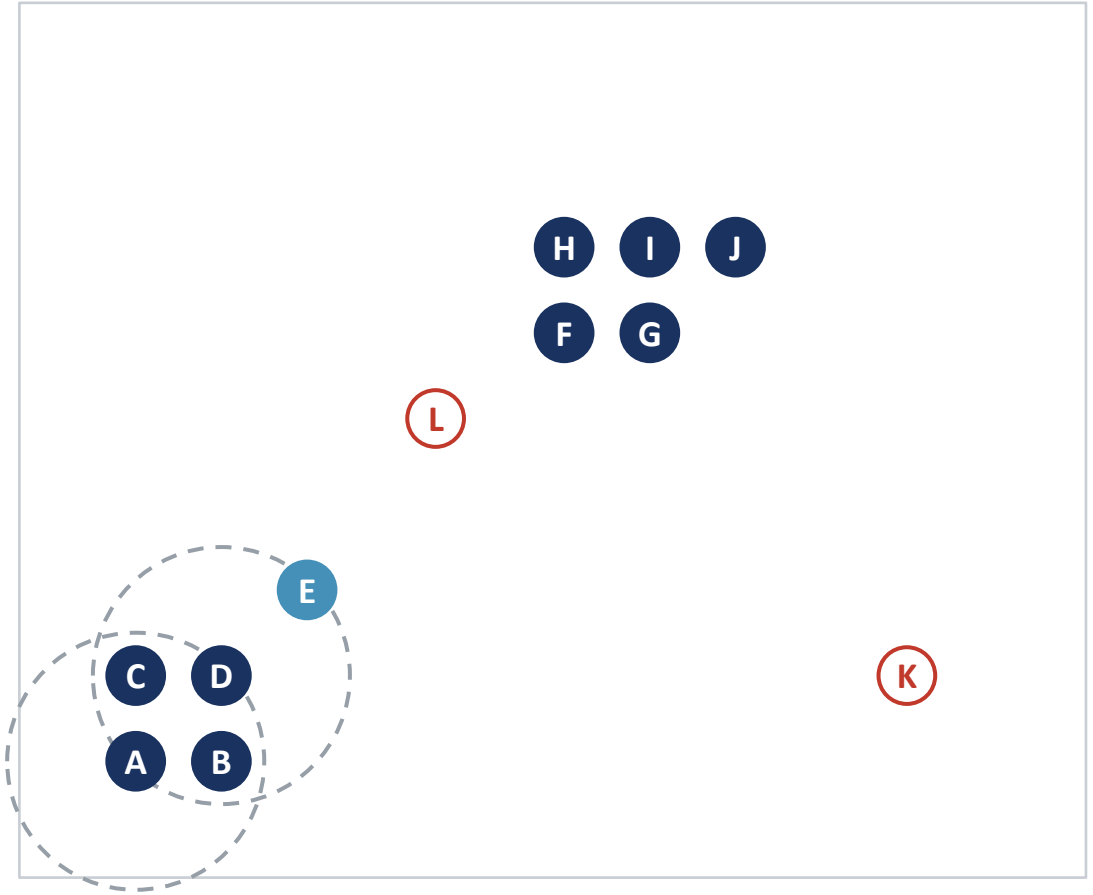
| Точка | N_ε(p) | \|N\| | Ядро? |
|---|---|---|---|
| **A (1; 1)** | A, B, C, D | 4 | да |
| **B (2; 1)** | A, B, C, D | 4 | да |
| **C (1; 2)** | A, B, C, D | 4 | да |
| **D (2; 2)** | A, B, C, D, E | 5 | да |
| **E (3; 3)** | D, E | 2 | нет |
| **F (6; 6)** | F, G, H, I | 4 | да |

| Точка | N_ε(p) | \|N\| | Ядро? |
|---|---|---|---|
| **G (7; 6)** | F, G, H, I, J | 5 | да |
| **H (6; 7)** | F, G, H, I | 4 | да |
| **I (7; 7)** | F, G, H, I, J | 5 | да |
| **J (8; 7)** | G, I, J | 3 | да |
| **K (10; 2)** | K | 1 | нет |
| **L (4,5; 5)** | L | 1 | нет |

- Ядровыми оказались 9 точек: A, B, C, D, F, G, H, I, J.
- E, K и L не ядровые; E лежит в окрестности ядра D, а K и L — одиноки.

# ШАГ 2: СТРОИМ КЛАСТЕР $C_1$

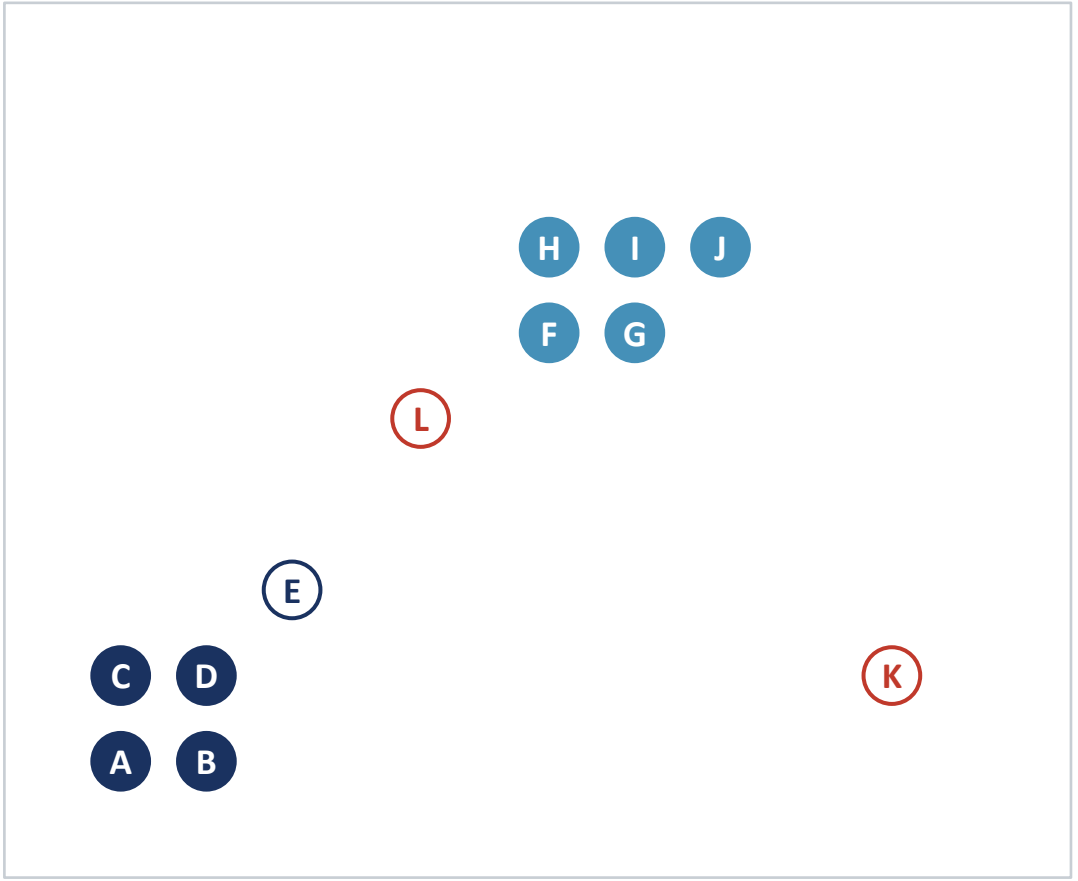
- Берём A: $|N| = 4 \geq 3$, A — ядро, создаём кластер $C_1$; очередь: B, C, D.
- B и C — ядра, их соседи уже в кластере.
- D — ядро, добавляет E; у E $|N| = 2 < 3$, поэтому E — граничная точка и кластер не расширяет.
- Очередь пуста: $\mathbf{C_1 = \{A, B, C, D, E\}}$.




Тёмные кружки — ядровые точки, голубой — граничная, красный контур — шум. Пунктир — ε-окрестность.

# ШАГ 3: КЛАСТЕР $C_2$ И ШУМ

- Следующая непосещённая точка F: |N| = 4, F — ядро, создаём **$C_2$**; очередь: G, H, I.
- G и I — ядра; через них достигается J.
- J — ядро (|N| = 3), её соседи уже в кластере: **$C_2$ = {F, G, H, I, J}**.
- K и L: |N| = 1 < 3, и ни одна ядровая точка их не достигает — **шум**.




Заливка — ядровые точки кластера, цветной контур — граничные, красный контур — шум.

# РЕЗУЛЬТАТ ПРИМЕРА

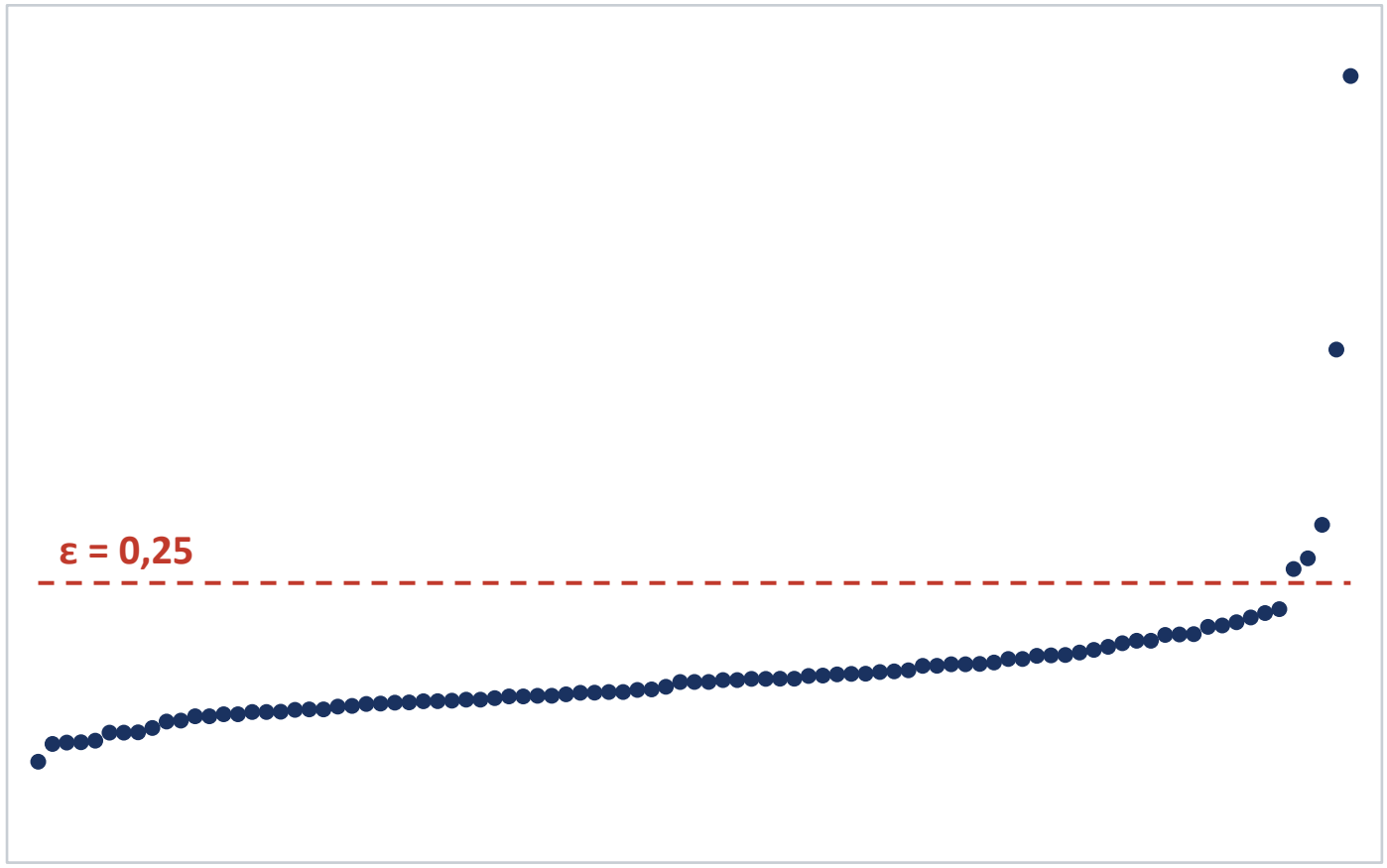
| Группа | Ядровые точки | Граничные точки |
|---|---|---|
| **Кластер $C_1$** | A, B, C, D | E |
| **Кластер $C_2$** | F, G, H, I, J | — |
| **Шум** | — | K, L (не принадлежат кластерам) |

Число кластеров (2) определилось автоматически; выбросы K и L не повлияли на кластеры.

# ВЫБОР ПАРАМЕТРОВ Е И MINPTS

- **MinPts**: не меньше D + 1, где D — число признаков; обычно берут 2·D (для 2D — 4). Для зашумлённых и больших данных — больше.
- **ε** подбирают по графику k-расстояний: для каждой точки считают расстояние до MinPts-го ближайшего соседа (считая саму точку) и сортируют.
- В качестве ε берут значение в «колене» графика: до него расстояния растут медленно, после — резко.
- Данные нужно **масштабировать**, иначе ε будет определяться признаком с наибольшим масштабом.

# ГРАФИК K-РАССТОЯНИЙ




## График k-расстояний

Данные: полумесяцы из слайда 6, MinPts = 4.

Резкий рост справа — выбросы. «Колено» около 0,25 даёт ε.

# ВЛИЯНИЕ ПАРАМЕТРОВ

| Параметр | Слишком малый | Слишком большой |
|---|---|---|
| **ε** | Много мелких кластеров, большая часть точек — шум | Кластеры сливаются в один |
| **MinPts** | Шум и случайные группы считаются кластерами | Малые кластеры исчезают, растёт доля шума |

- Результат чувствителен к параметрам; их подбирают по графику k-расстояний и проверяют визуально или метриками.

# ВЫЧИСЛИТЕЛЬНАЯ СЛОЖНОСТЬ

- Основная операция — поиск соседей в радиусе ε, выполняется для каждой точки один раз.
- Без индекса: $O(n^2)$ по времени. С пространственным индексом (k-d-дерево, R*-дерево): в среднем $O(n \log n)$ при небольшой размерности.
- Память: $O(n)$ без матрицы расстояний; с матрицей — $O(n^2)$.
- В пространствах высокой размерности индексы теряют эффективность, а понятие ε-окрестности — смысл.

# ДОСТОИНСТВА И НЕДОСТАТКИ DBSCAN

## Достоинства

- не нужно задавать число кластеров
- кластеры произвольной формы
- выделяет шум и выбросы
- один проход по точкам, поддержка индексов

## Недостатки

- чувствителен к ε и MinPts
- плохо работает при разной плотности кластеров
- проблемы в высоких размерностях
- граничные точки зависят от порядка обхода

# СРАВНЕНИЕ K-MEANS И DBSCAN

| Критерий | K-means | DBSCAN |
|---|---|---|
| **Число кластеров** | Задаётся заранее (k) | Определяется автоматически |
| **Форма кластеров** | Выпуклые, шарообразные | Произвольная |
| **Выбросы** | Искажают центры | Помечаются как шум |
| **Параметры** | k, начальные центры | ε, MinPts |
| **Разная плотность** | Допустима | Проблема (см. OPTICS, HDBSCAN) |

# ВАРИАНТЫ МЕТОДА

- **OPTICS** (1999): не фиксирует ε, а строит упорядочение точек по достижимости; кластеры разной плотности выделяются из графика достижимости.
- **HDBSCAN** (2013): строит иерархию DBSCAN-кластеров по всем ε и выбирает устойчивые; параметр ε не нужен.
- **ST-DBSCAN, GDBSCAN** и другие обобщения учитывают время, нечисловые признаки и произвольные отношения соседства.

# РЕАЛИЗАЦИЯ НА PYTHON (SCIKIT-LEARN)

```
import numpy as np
from sklearn.cluster import DBSCAN

X = np.array([[1, 1], [2, 1], [1, 2], [2, 2], [3, 3],
              [6, 6], [7, 6], [6, 7], [7, 7], [8, 7],
              [10, 2], [4.5, 5]])

db = DBSCAN(eps=1.5, min_samples=3).fit(X)
print(db.labels_)
print(db.core_sample_indices_)
```

## Результат

labels_:

```
0 0 0 0 0 1 1 1 1 1
-1 -1
```

core_sample_indices_:

```
0 1 2 3 5 6 7 8 9
```

Метка −1 — шум.
Совпадает с ручным расчётом.

# КОНТРОЛЬНЫЕ ВОПРОСЫ

- Чем кластеризация на основе плотности отличается от K-means?
- Что такое ε-окрестность и параметр MinPts?
- Чем различаются ядровые, граничные точки и шум?
- Что такое достижимость и связность по плотности?
- Опишите шаги алгоритма DBSCAN.
- Как подобрать ε по графику k-расстояний?
- В чём недостатки DBSCAN и чем их преодолевают?

# ВЫВОДЫ

- Кластеры — области высокой плотности, отделённые областями низкой плотности.
- DBSCAN использует два параметра: ε и MinPts; различает ядровые, граничные точки и шум.
- Метод находит кластеры произвольной формы, сам определяет их число и устойчив к выбросам.
- Качество зависит от ε и MinPts; ε подбирают по графику k-расстояний, данные масштабируют.
- При разной плотности кластеров применяют OPTICS и HDBSCAN.

# СПИСОК ЛИТЕРАТУРЫ


- Ester M., Kriegel H.-P., Sander J., Xu X. A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise // Proc. KDD, 1996. P. 226–231.
- Schubert E., Sander J., Ester M., Kriegel H.-P., Xu X. DBSCAN Revisited, Revisited // ACM Transactions on Database Systems, 2017, 42(3).
- Ankerst M., Breunig M., Kriegel H.-P., Sander J. OPTICS: Ordering Points To Identify the Clustering Structure // Proc. ACM SIGMOD, 1999. P. 49–60.
- Campello R., Moulavi D., Sander J. Density-Based Clustering Based on Hierarchical Density Estimates // PAKDD, 2013.
- Han J., Kamber M., Pei J. Data Mining: Concepts and Techniques. 3rd ed. Morgan Kaufmann, 2011.
- Документация scikit-learn: https://scikit-learn.org/stable/modules/clustering.html#dbscan