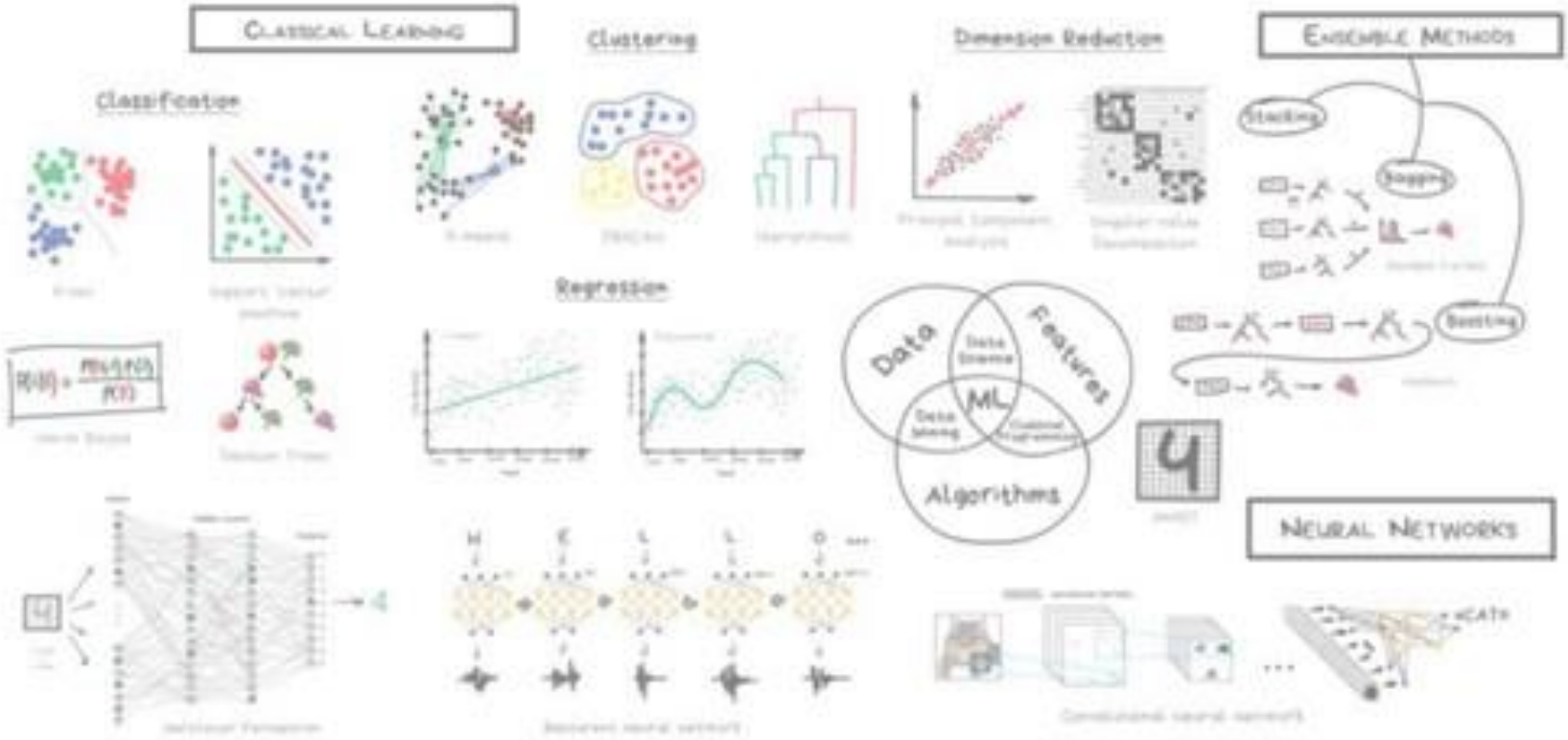
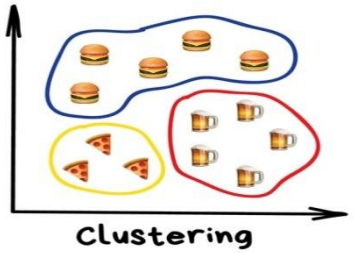
МАШИНЕ НАВЧАННЯ

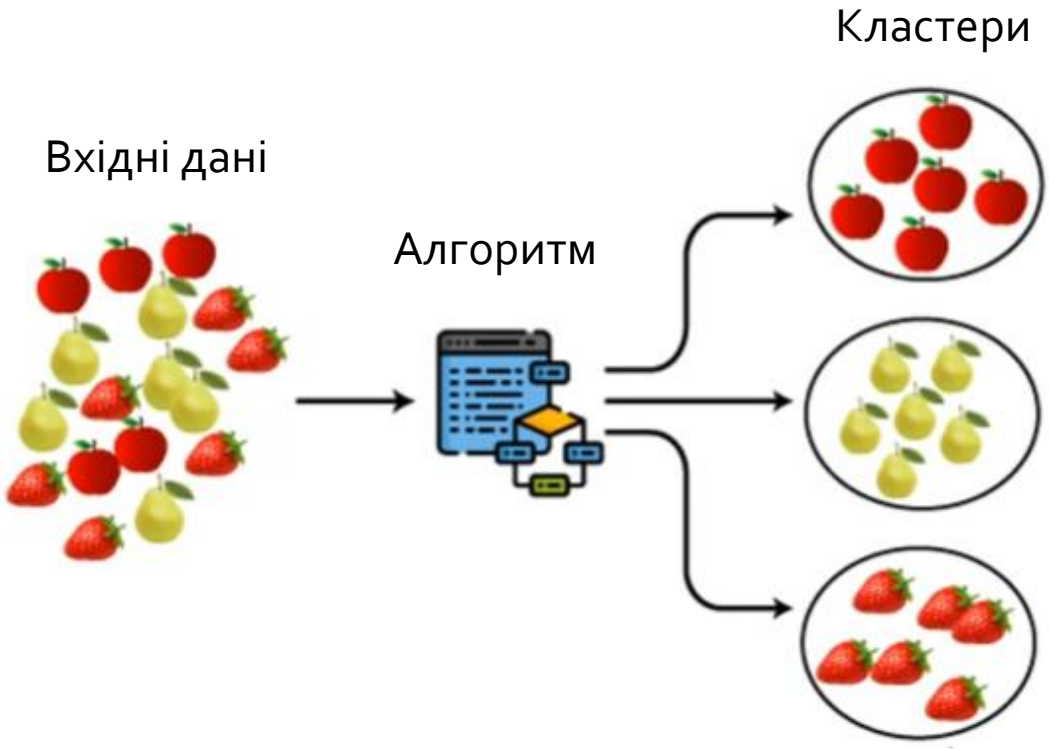
Навчання без вчителя. Методи кластеризації



Лекція №11

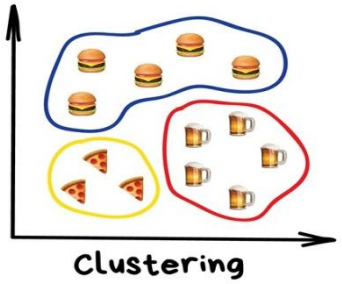
Кластеризація

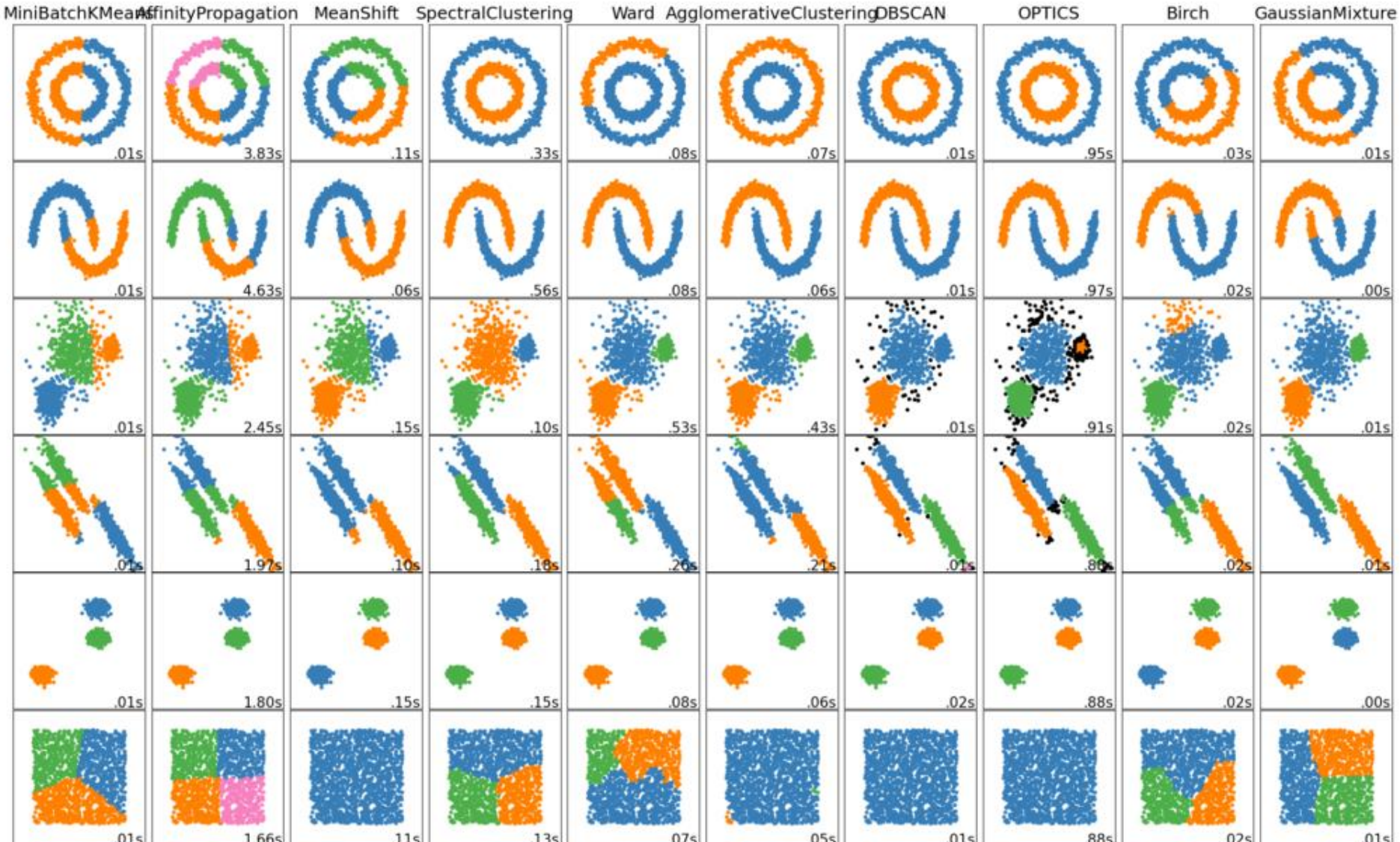




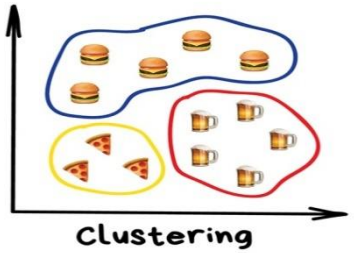
Кластеризація

Методи кластеризації





Кластеризація:
метод k-means

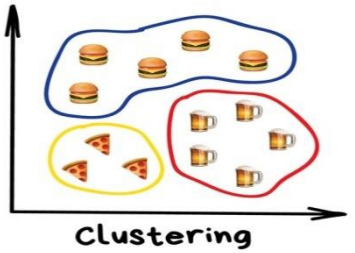


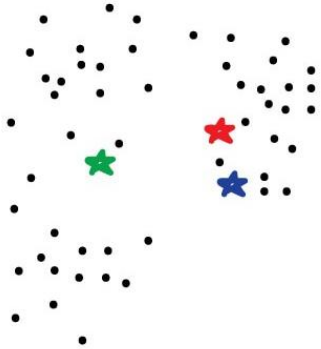
Алгоритм K-середніх, напевно, найпопулярніший і найпростіший алгоритм кластеризації і дуже легко представляється у вигляді простого псевдокоду:

- Вибрати кількість кластерів, яка нам здається оптимальною для наших даних.
- Висипати випадковим чином у простір наших даних точок (центроїдів).
- Для кожної точки нашого набору даних порахувати, до якого центроїду вона ближча.
- Перемістити кожен центроїд до центру вибірки, яку ми віднесли до цього центроїду.
- Повторювати останні два кроки фіксоване число разів, або до тих пір, поки центроїди перестануть змінювати свої положення (зазвичай це означає, що їх зміщення щодо попереднього положення не перевищує якогось заздалегідь заданого невеликого значення).

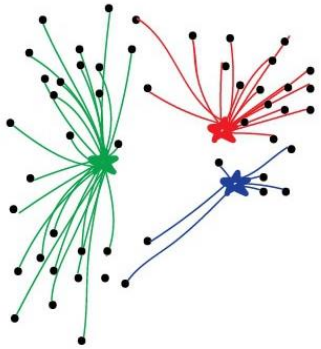
У разі звичайної евклідової метрики для точок, що лежать на площині, цей алгоритм дуже просто розписується аналітично і малюється.

Кластеризація:
метод k-means

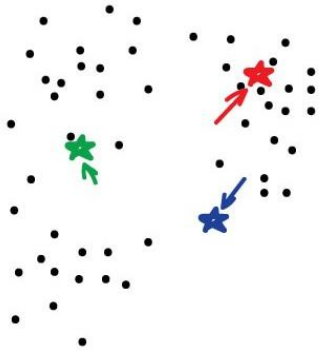




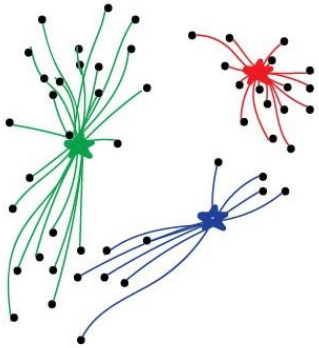
1. Вкидуємо центроїди



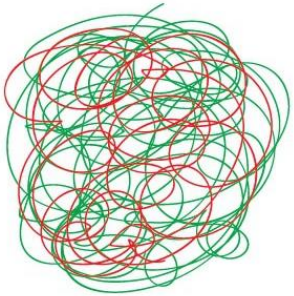
2. Визначаємо найближчі точки



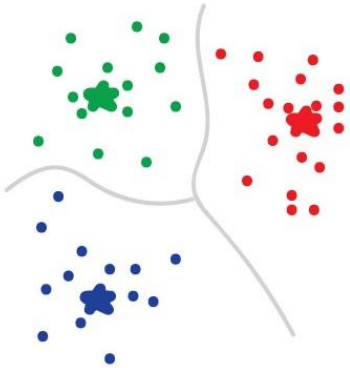
3. Рухаємо центроїди ближче до центрів



4. Знову дивимось та рухаємо

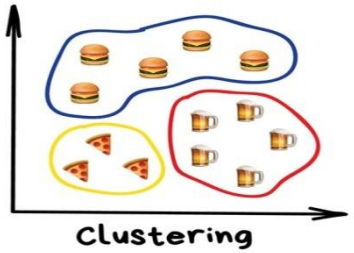


5. Повторюємо N разів



6. Готово !!!!

Кластеризація:
метод k-means



Мінімізація суми квадратів внутрішньо-кластерних відстаней

$$\sum_{i=1}^{\ell} \|x_i - \mu_{ai}\|^2 \to \min_{\{a_i\},\{\mu_a\}}, \quad \|x_i - \mu_{ai}\|^2 = \sum_{j=1}^{n} \big(f_j(x_j) - \mu_{aj}\big)^2$$

Алгоритм Лойда

вхід: вибірка $X^\ell = \{x_1, \dots, x_\ell\}$, $K = |Y|$;

вихід: центри кластерів μ_a, $a \in Y$;

фіксуємо початкове положення центрів всіх кластерів μ_a для всіх $a \in Y$

повторюємо

віднести кожен x_i до найближчого центру:

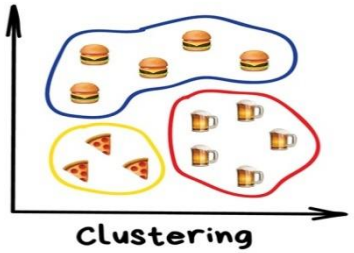
$$a_i := \arg\min_{a \in Y} \|x_i - \mu_{ai}\|, \qquad i = 1, .., \ell;$$

розрахувати нові положення центрів кластерів:

$$\mu_a := \frac{\sum_{i=1}^{\ell} [a_i = a] x_i}{\sum_{i=1}^{\ell} [a_i = a]}, \qquad a \in Y;$$

поки a_i перестануть змінюватись.

Кластеризація:
метод k-means



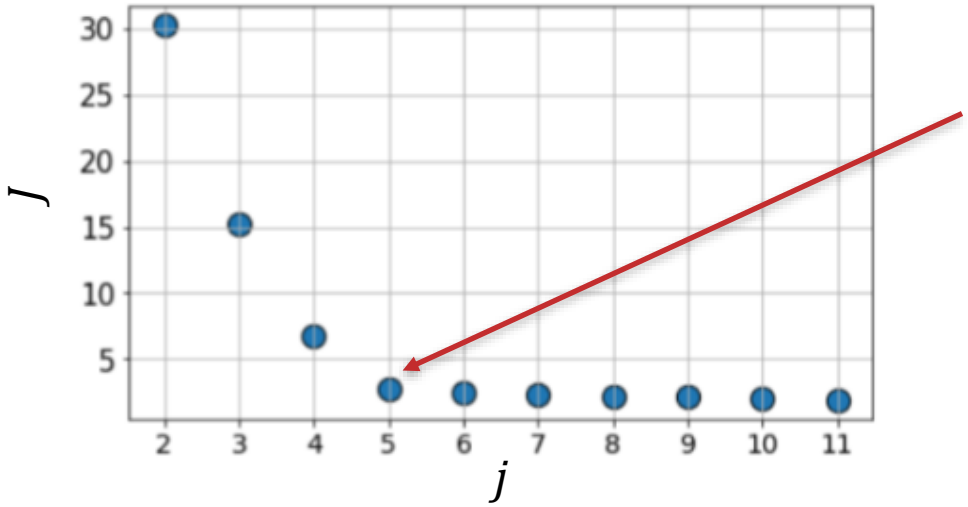
Вибір оптимальної кількості кластерів

Метод локтя (Elbow method)

Сума квадратів у кластері (Within-Cluster-Sum-of-Squares (WCSS) – сума квадратів відстаней кожної точки даних у всіх кластерах до відповідних центроїдів

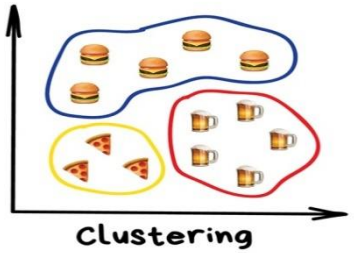
$$J = \sum_{j=1}^{k} \sum_{i=1}^{\ell} \left\| x_i^j - c_j \right\|^2$$

k – кількість кластерів; ℓ – розмір вибірки



Оптимальна кількість кластерів

Кластеризація:
метод k-means



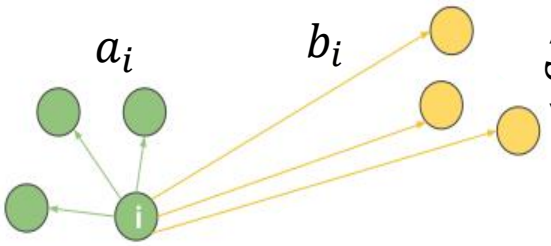
Вибір оптимальної кількості кластерів

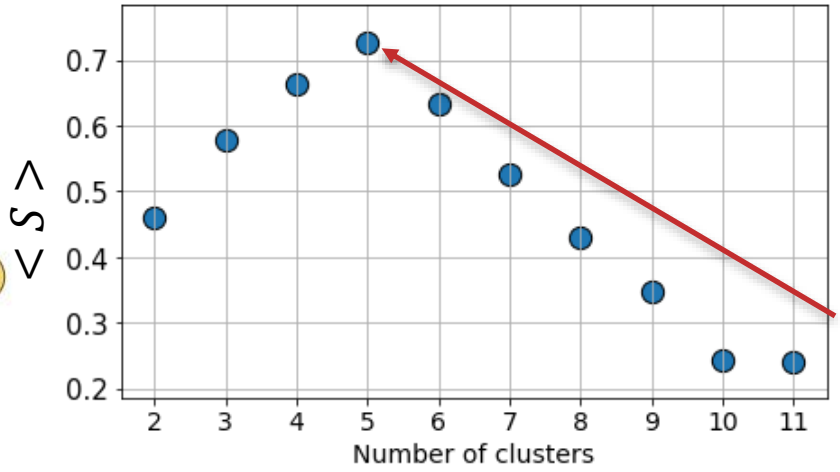
Метод «силуету» (Silhouette Score)

Коефіцієнт «силует» обчислюється за допомогою середньої внутрішньо-кластерної відстані (a) та середньої відстані до найближчого кластера (b) за кожним зразком

$$S_i = \frac{b_i - a_i}{\max(a_i, b_i)}$$

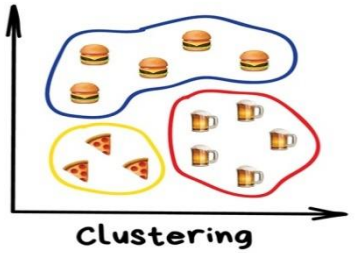
a_i – середня відстань від об'єкта i до об'єктів свого кластеру
b_i – середня відстань від об'єкта i до об'єктів іншого кластеру





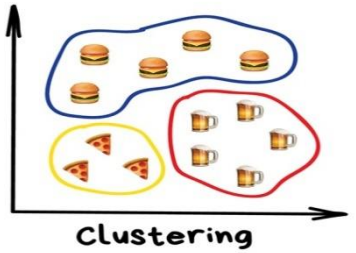
Оптимальна кількість кластерів

Кластеризація:
метод mean-shift



- **Зсув середнього значення** — це непараметрична техніка аналізу простору ознак визначення місця розташування максимуму щільності ймовірності, так званий алгоритм пошуку моди. Область застосування техніки - кластерний аналіз у комп'ютерному зору та обробці зображень.
- Суть методу полягає в тому, що задається функція оцінки густини розподілу точок у просторі ознак. Далі, для цих точок будується градієнт ядра оцінки щільності розподілу та обчислюється вектор середнього зсуву, що задається як вектор найбільшого збільшення градієнта.
- Алгоритм середнього зсуву призначає точки даних кластерам ітеративно, зміщуючи точки у напрямку найвищої густини точок даних, тобто центроїду кластера.
- Різниця між алгоритмом K-Means та Mean-Shift полягає в тому, що тут не потрібно заздалегідь вказувати кількість кластерів, оскільки кількість кластерів визначатиметься алгоритмом за даними.

Кластеризація:
метод mean-shift



Побудова непараметричної оцінки густини
Густина оцінюється як сумарний вплив елементів вибірки:

$$F(x) = \frac{1}{\ell h^d} \sum_{i=1}^{\ell} K\left(\frac{x - x_i}{h}\right)$$

Вклад кожного елемента описується за допомогою функції (ядра) $K(x)$, яка залежить від відстані до цього елементу. Ця функція визначає вагу найближчих точок для переоцінки середнього. Зазвичай використовують:

- пласке ядро $K(x) = \begin{cases} 1, x \le \lambda \\ 0, x > \lambda \end{cases}$
- Гаусове ядро $K(x_i - x) = \exp\left(-\frac{\|x_i - x\|^2}{2\sigma^2}\right)$

Зважене середнє густини у вікні, що визначається функцією K дорівнює

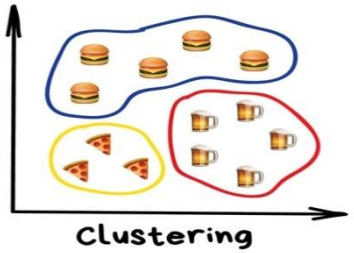
$$m(x) = \frac{\sum_{x_i \in N(x)} K(x_i - x) x_i}{\sum_{x_i \in N(x)} K(x_i - x)}$$

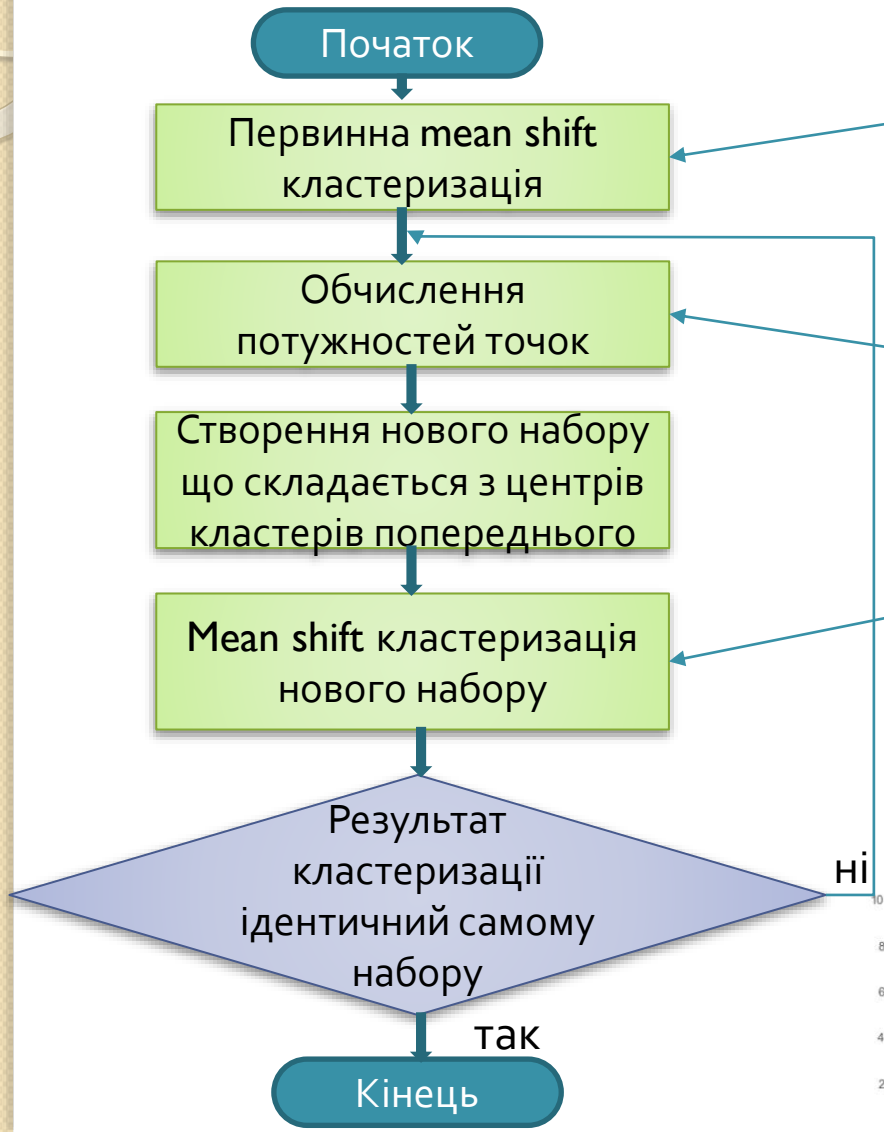
де $N(x)$ представляє окіл точки x, тобто є набором точок, для яких $K(x_i) \neq 0$

Різниця $(m(x) - x)$ називається *зміщенням середнього значення*.

Алгоритм *зміщення середнього значення* назначає $x \leftarrow m(x)$ та повторює оцінку, доки $m(x)$ не зійдеться.

Кластеризація:
метод mean-shift

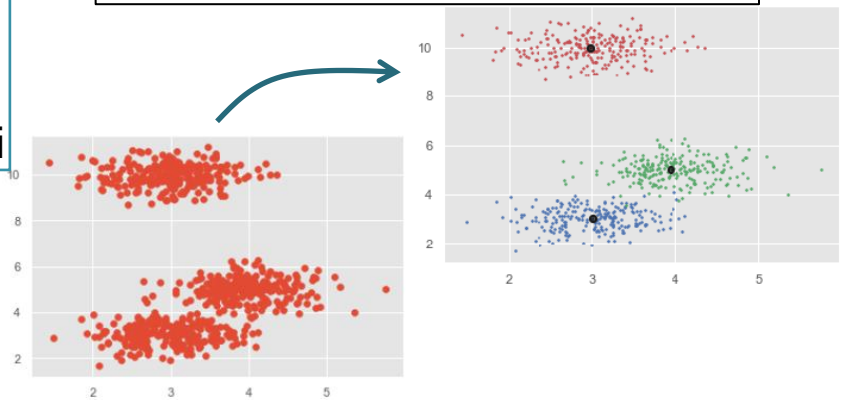




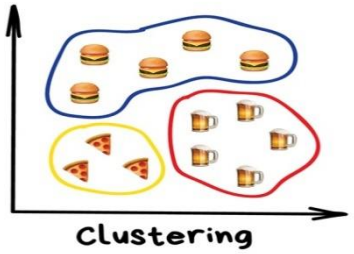
Визначення розмірів початкового вікна та безпосередня кластеризація зміщенням середнього

Присвоєння кожному кластеру ваги, що дорівнює кількості точок в ньому

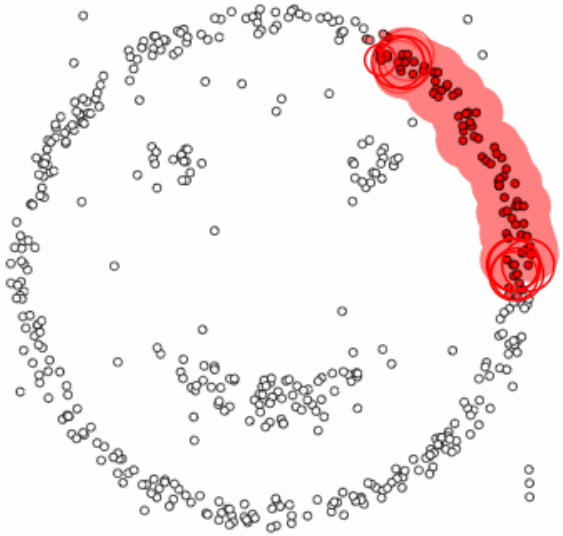
Визначення розмірів нового вікна та безпосередня кластеризація зміщенням середнього нового набору



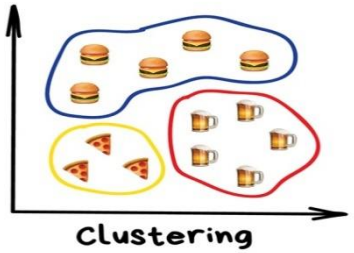
Кластеризація:
метод DBSCAN
(Density-Based Spatial Clustering of Applications with Noise)



- Шукати центроїди зручно та просто, але в реальних завданнях кластери можуть бути зовсім не круглою формою. Ось ви геолог, якому потрібно знайти на карті схожі за структурою гірські породи - ваші кластери не тільки будуть вкладені одна в одну, але ви ще й не знаєте, скільки їх взагалі вийде.
- Хитрим завданням – хитрі методи. DBSCAN сам знаходить скупчення точок і будує навколо кластери. Його легко зрозуміти, якщо уявити, що крапки – це люди на площі. Знаходимо (наприклад трьох) будь-яких близьких людей і говоримо їм взятися за руки.
- Потім вони починають брати за руку тих, кого можуть дістати. Так по ланцюжку, доки ніхто більше не зможе взяти когось за руку — це і буде перший кластер. Повторюємо, доки не поділимо всіх. Ті, кому взагалі нема кого брати за руку — це викиди, аномалії.



Кластеризація:
метод DBSCAN



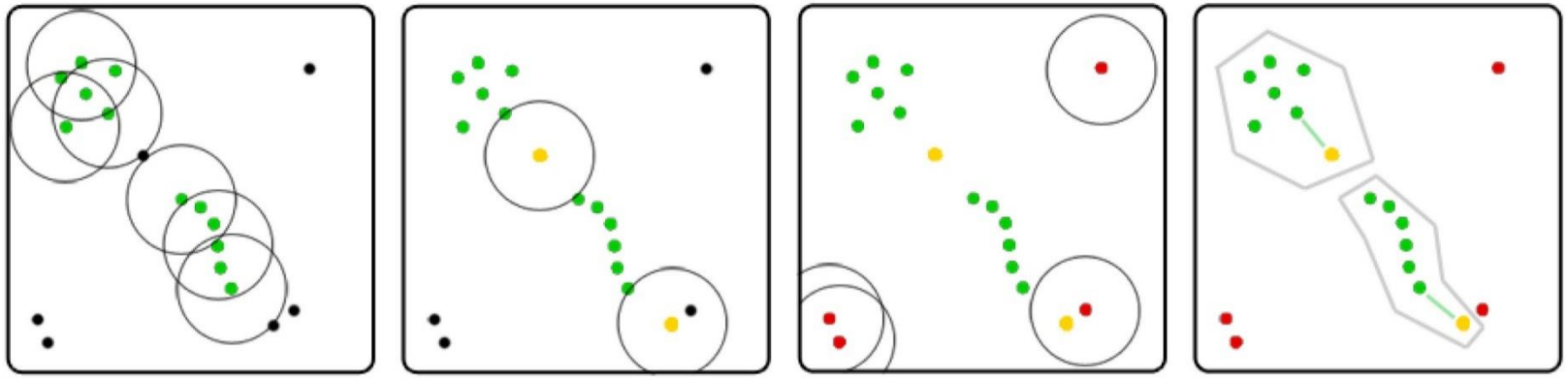
Є об'єкт $x \in U$ та його ε – окіл $U_\varepsilon(x) = \{u \in U: \ \rho(x,u) \le \varepsilon\}$

Кожен об'єкт може бути одного з трьох типів:

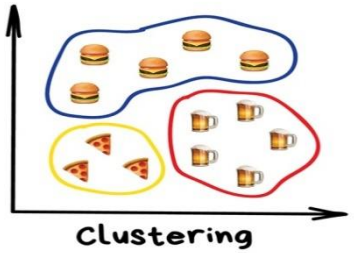
- кореневий: той що має щільний окіл $|U_\varepsilon(x)| \ge m$ (m – задана кількість об'єктів);
- прикордонний: не кореневий, але в околі кореневого;
- шумовий (викид): не кореневий і не прикордонний

Не розмічені дані

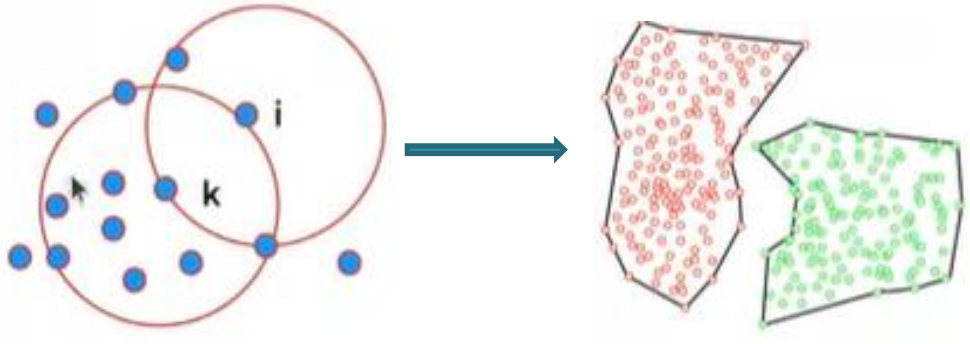
Покрокова реалізація алгоритму DBSCAN



Кластеризація:
метод DBSCAN



вхід: вибірка $X^\ell = \{x_1, \ldots, x_\ell\}$, параметри ε та m;
вихід: розбиття вибірки на кластери та шумові викиди;
фіксуємо $U := X^\ell$ – нерозмічені; $a := 0$;
повторюємо поки у вибірці є нопомічені точки, $U \neq \emptyset$:
    взяти випадкову точку $x \in U$;
    якщо $|U_\varepsilon(x)| < m$ то
        помітити об'єкт x як можливо шумовий;
    інакше
        створити новий кластер: $K := U_\varepsilon(x)$; $a := a + 1$;
        для всіх $x' \in K$, непомічених або шумових
            якщо $|U_\varepsilon(x')| \geq m$ то $K := K \cup U_\varepsilon(x')$;
            інакше помітити x' як прикордонний кластера K;
        $a_i := a$ для всіх $x_i \in K$;
        $U := U/K$;



Дякую за увагу