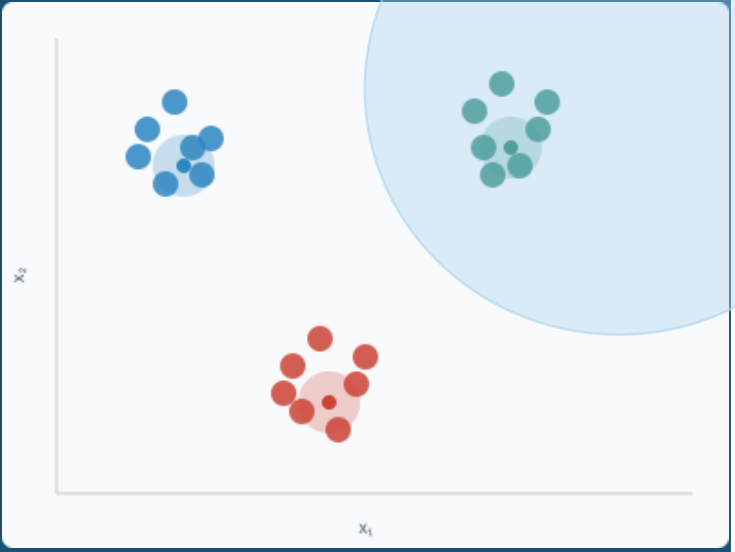


ЖИТОМИРСЬКА ПОЛІТЕХНІКА
ДЕРЖАВНИЙ УНІВЕРСИТЕТ

Кластерний аналіз

Алгоритми, метрики та практичне застосування у задачах Data Mining

Лекція 10

- Означення та задачі кластеризації
- K-Means та ієрархічна кластеризація
- DBSCAN та щільнісні методи
- Метрики якості кластерів
- Python: sklearn, scipy

План

01 **Означення та таксономія**

Що таке кластеризація, формальна постановка, застосування

02 **Відстані та подібності**

Евклідова, манхеттенська, косинусна, відстань Мінковського

03 **Ієрархічна кластеризація**

Agglomerative, Divisive, дендрограма, linkage

04 **K-Means та варіанти**

Алгоритм, K-Means++, K-Medoids, Mini-Batch

05 **Щільнісні методи**

DBSCAN, HDBSCAN: core/border/noise points

06 **Гауссові суміші (GMM)**

Probabilistic clustering, EM-алгоритм

07 **Метрики якості**

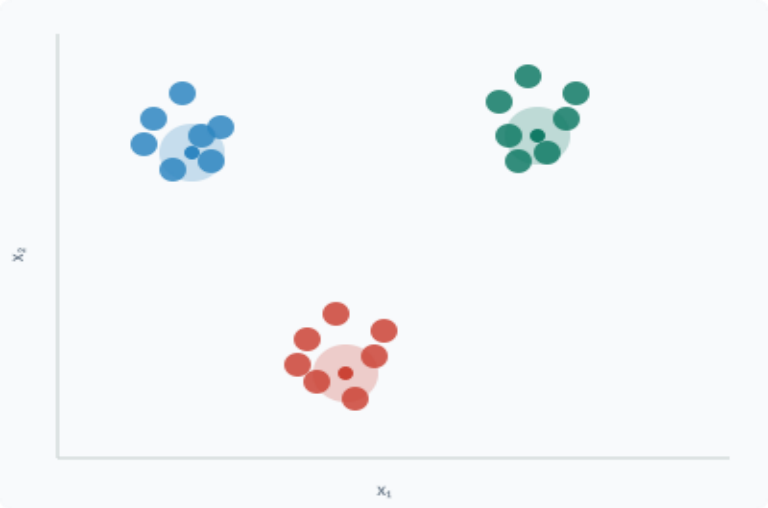
Silhouette, Davies-Bouldin, Calinski-Harabasz, Elbow

08 **Практика: Python sklearn**

Pipeline, GridSearchCV, візуалізація результатів

Що таке кластеризація?

Кластеризація (clustering) — задача розбиття множини об'єктів $X = \{x_1, x_2, \ldots, x_n\}$ на k непересічних груп (кластерів) $C_1, C_2, \ldots, C_k$ таким чином, щоб об'єкти всередині кластера були максимально подібними між собою, а між кластерами — максимально різними.



Без учителя (Unsupervised)

Мітки класів заздалегідь невідомі

Внутрішньокластерна схожість

$dist(x_i, x_j)$ мала при $x_i, x_j \in C_k$

Міжкластерна відмінність

$dist(C_k, C_l)$ велика при $k \neq l$

Кількість кластерів k

може бути задана або визначена автоматично

Застосування кластерного аналізу

Галузі та типові задачі

Маркетинг

Сегментація клієнтів за поведінкою, RFM-аналіз, персоналізація пропозицій

Біоінформатика

Кластеризація генів, виявлення типів клітин, аналіз транскриптому

Обробка тексту

Тематичне моделювання, виявлення схожих документів, NLP-задачі

Аномалії

Виявлення викидів у фінансових транзакціях, мережева безпека

Соціальні мережі

Виявлення спільнот, аналіз графів, рекомендаційні системи

Медицина

Субтипізація захворювань, аналіз медичних знімків, прогноз терапії

Метрики відстані та подібності

Основа для порівняння об'єктів у просторі ознак

Евклідова

$$d(x,y) = \sqrt{\sum_i (x_i - y_i)^2}$$

Геометрична відстань. Найпоширеніша. Чутлива до масштабу ознак.

Манхеттенська (L_1)

$$d(x,y) = \sum_i |x_i - y_i|$$

Сума абсолютних різниць. Стійкіша до викидів.

Мінковського (L_p)

$$d(x,y) = \left(\sum_i |x_i - y_i|^p\right)^{1/p}$$

Узагальнення: $p=1 \to L_1$, $p=2 \to L_2$, $p \to \infty \to$ Чебишев

Косинусна подібність

$$sim(x,y) = \frac{x \cdot y}{\|x\| \cdot \|y\|}$$

Кут між векторами. Для текстів, незалежно від норми.

Відстань Махаланобіса

$$d(x,y) = \sqrt{(x-y)^T S^{-1} (x-y)}$$

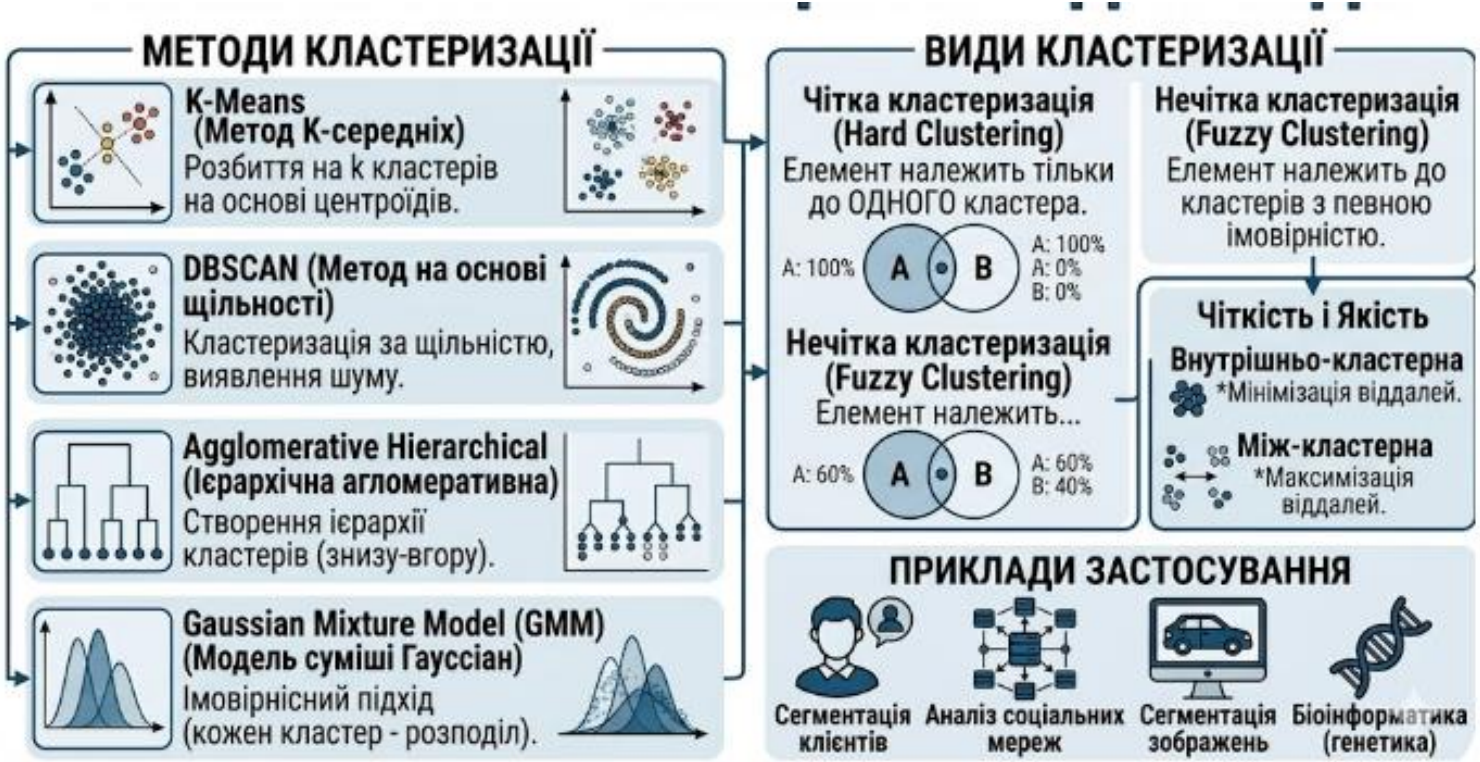
Враховує кореляцію ознак та масштаб. S — коваріаційна матриця.

Хемінгова відстань

$$d(x,y) = \sum_i \mathbb{1}[x_i \neq y_i]$$

Для категоріальних та бінарних даних. Кількість позицій, де відрізняються.

Методи і види кластерізації



Ієрархічна кластеризація

Agglomerative (знизу вгору) та Divisive (зверху вниз)

Агломеративні

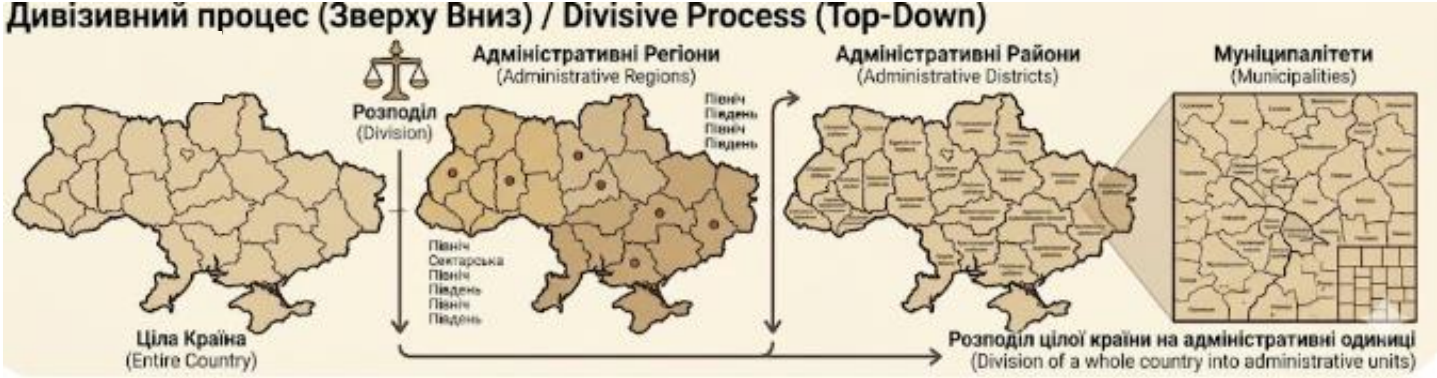
- AGNES (Agglomerative Nesting):
- CURE (Clustering Using REpresentatives);
- ROCK;
- CHAMELEON;
- тощо.

Ієрархічна кластеризація

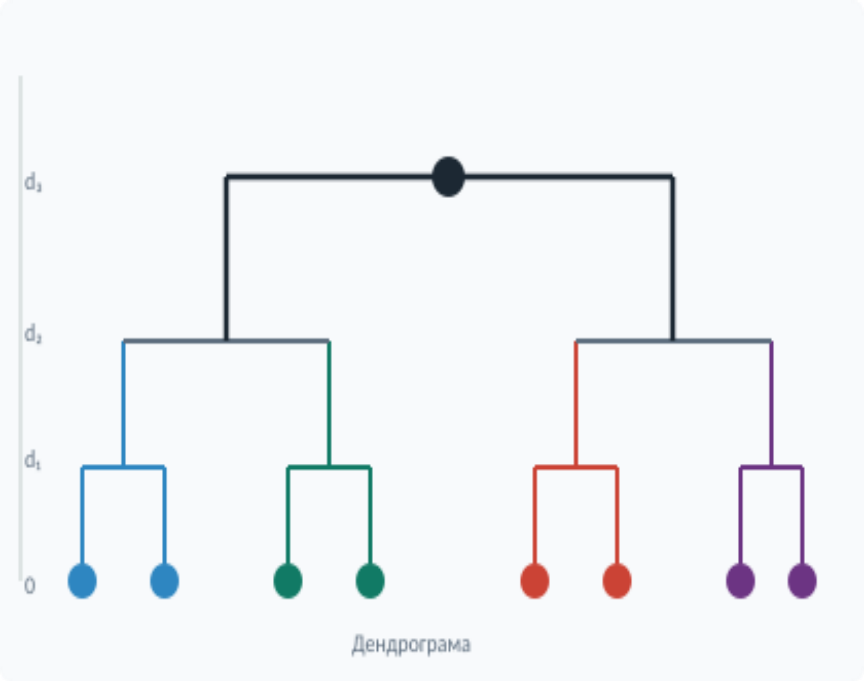
Agglomerative (знизу вгору) та Divisive (зверху вниз)

Дівізімні

- DIANA (Divisive Analysis):
- BIRCH (Balanced Iterative Reducing and Clustering using Hierarchies);
- MST (Algorithm based on Minimum Spanning Trees);
- тощо.



Ієрархічна кластеризація



Дендрограма

Алгоритм Agglomerative:

1. Кожен об'єкт — окремий кластер (n кластерів)
2. Знайти найближчу пару кластерів за обраним linkage
3. Об'єднати їх в один кластер
4. Повторювати кроки 2–3 до одного кластера
5. Обрати кількість кластерів, «розрізавши» дендрограму

Методи зв'язку (Linkage):

Single	Complete	Average	Ward
`min dist(a,b)`	`max dist(a,b)`	`mean dist(a,b)`	`min ΔWSS`

Алгоритм K-Means

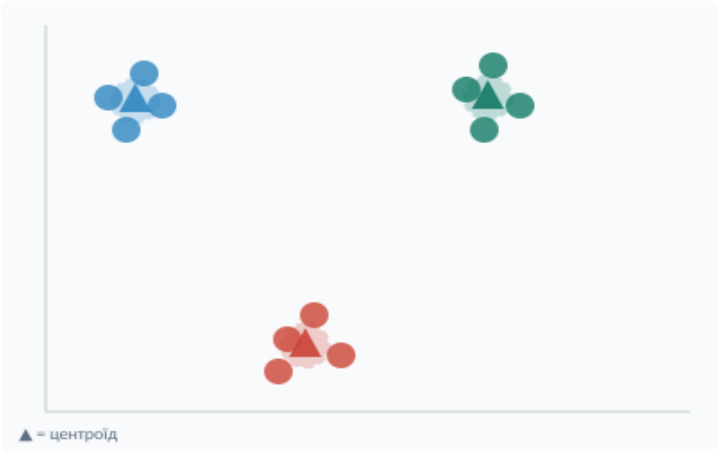
Найпопулярніший ітераційний метод кластеризації

Цільова функція:

$$J(C) = \sum_k \sum_{x_i \in C_k} \|x_i - \mu_k\|^2 \rightarrow \min$$

Кроки алгоритму:

1. **Ініціалізація**
   Обрати k центроїдів $\mu_1,\ldots,\mu_k$ (випадково або K-Means++)
2. **Призначення**
   Кожен x_i → найближчий кластер: $c_i = \text{argmin}_k \|x_i - \mu_k\|^2$
3. **Оновлення**
   $\mu_k = (1/|C_k|) \sum_{x_i \in C_k} x_i$
4. **Збіжність**
   Повторювати 2–3 до стабілізації центроїдів (або max ітерацій)



K-Means++ — краща ініціалізація: обирати нові центроїди з ймовірністю $\propto dist^2$

Варіанти K-Means та обмеження

K-Medoids, Mini-Batch K-Means, K-Means++

K-Means++

Покращена ініціалізація центроїдів. Ймовірність вибору нового центроїда пропорційна $dist^2(x,$ найближчий центроїд). Зменшує ймовірність локального мінімуму.

K-Medoids (PAM)

Центроїд — реальний об'єкт із датасету (medoid). Стійкіший до викидів. Складність $O(k(n-k)^2)$ — дорожчий.

Mini-Batch K-M.

Навчання на випадкових міні-батчах замість усього датасету. Значно швидший для великих даних. Незначна втрата якості.

Обмеження K-Means:

Задати k наперед

Потрібно знати кількість кластерів

Опуклі кластери

Погано для незвичних форм (rings, crescents)

Чутливість до викидів

Центроїд зміщується від викидів

Локальний мінімум

Результат залежить від ініціалізації

DBSCAN — Density-Based Spatial Clustering

Виявлення кластерів довільної форми та шуму

Гіперпараметри:

ε — радіус околу точки
MinPts — мінімальна кількість точок у ε-околі

Core point

У ε-колі ≥ MinPts точок. Ядро кластера.

Border point

У ε-колі < MinPts, але досяжна з Core.

Noise point

Не Core і не Border. Аномалія/викид.

Переваги: не потребує k, виявляє довільні форми, стійкий до шуму
Недоліки: складно підбирати ε і MinPts для різної щільності

Gaussian Mixture Models (GMM)

Ймовірнісний підхід до кластеризації через суміш розподілів

Модель суміші:

$$p(x) = \sum_{k=1}^{K} \pi_k \cdot \mathcal{N}(x \mid \mu_k, \Sigma_k) \qquad \text{де} \quad \sum_k \pi_k = 1, \quad \pi_k \geq 0$$

EM-алгоритм для навчання GMM:

E-крок

$$r_{nk} = \pi_k \mathcal{N}(x_n \mid \mu_k, \Sigma_k) \;/\; \sum_j \pi_j \mathcal{N}(x_n \mid \mu_j, \Sigma_j)$$

«Відповідальність» компоненти k за об'єкт n

M-крок

$$\mu_k = \sum_n r_{nk}\, x_n \;/\; \sum_n r_{nk}$$

Оновлення параметрів: μ_k, Σ_k, π_k через зважені суми

Порівняно з K-Means: GMM — «м'яке» призначення (soft assignment), враховує форму та розмір кластерів через коваріаційну матрицю Σ_k

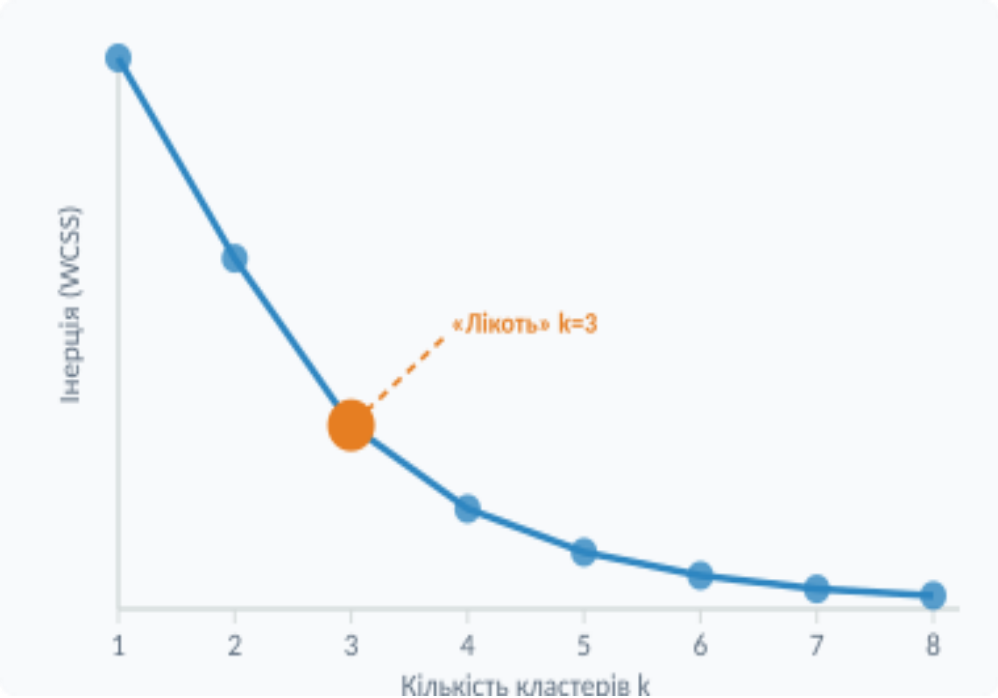
Огляд алгоритмів кластеризації

Порівняльна таблиця методів

Алгоритм	Тип	Задає k?	Форма кластерів	Складність	Викиди
K-Means	Партиційний	Так	Опуклі (кулі)	O(nkd·iter)	Погано
K-Medoids	Партиційний	Так	Опуклі	$O(k(n-k)^2)$	**Добре**
Agglomerative	Ієрархічний	Ні*	Довільна	$O(n^2 \log n)$	Нейтрально
DBSCAN	Щільнісний	Ні	Довільна	O(n logn)	**Відмінно**
HDBSCAN	Щільнісний	Ні	Довільна	O(n logn)	**Відмінно**
GMM / EM	Ймовірнісний	Так	Еліпсоїди	$O(nkd^2 \cdot iter)$	Помірно
Spectral	Графовий	Так	Довільна	$O(n^3)$	Погано

Метод ліктя (Elbow Method)

Визначення оптимальної кількості кластерів k



Як інтерпретувати:

WCSS

Within-Cluster Sum of Squares — сума квадратів відстаней від точок до центроїдів їхнього кластера

Лікоть

Точка, де зменшення WCSS при додаванні ще одного кластера стає незначним

Вибір k

Оптимальний k - у «лікті» кривої. На прикладі: k = 3

Формула: $WCSS(k) = \Sigma_k \Sigma_{x_i \in C_k} \|x_i - \mu_k\|^2$

Силуетний коефіцієнт (Silhouette Score)

Внутрішня метрика якості кластеризації без еталонних міток

Для об'єкта x_i:

$$s(i) = (b(i) - a(i)) / \max(a(i), b(i))$$

a(i) — середня відстань до об'єктів свого кластера
b(i) — середня відстань до об'єктів найближчого сусіднього кластера

Інтерпретація s(i):

0.7 – 1.0
Відмінна кластеризація

0.5 – 0.7
Прийнятна структура

0.25 – 0.5
Слабка структура

< 0.25
Немає структури

Середнє значення S̄ по всіх точках — загальна оцінка якості кластеризації

Метрики Davies-Bouldin та Calinski-Harabasz

Davies-Bouldin Index (DBI)

$$DB = (1/k)\ \Sigma_i\ \max_{j\neq i}\ [\ (\sigma_i+\sigma_j)\ /\ d(\mu_i,\mu_j)\ ]$$

σ_i — середня відстань у кластері i
$d(\mu_i,\mu_j)$ — відстань між центрами кластерів

Менше — краще (↓ DBI = кращі кластери)

Перевага: не потребує еталонних міток, легко інтерпретувати. Недолік: чутливий до опуклих кластерів.

Застосування: порівняння різних k або різних алгоритмів на одному датасеті.

Calinski-Harabasz (CH)

$$CH = [SS\_B/(k-1)]\ /\ [SS\_W/(n-k)]$$

SS_B — Between-cluster SS (дисперсія між кластерами)
SS_W — Within-cluster SS (дисперсія всередині)

Більше — краще (↑ CH = кращі кластери)

Перевага: швидкий, добре для опуклих кластерів. Недолік: занижує складні форми.

Застосування: вибір оптимального k через порівняння значень CH для різних конфігурацій.

Зовнішні метрики якості кластеризації

Коли відомі еталонні мітки класів

Rand Index (RI)

$$RI = (TP+TN)/(TP+TN+FP+FN)$$

Частка правильних попарних рішень (об'єднати / не об'єднати). Значення 0–1, де 1 — ідеальний збіг.

Adjusted Rand (ARI)

$$ARI = (RI - E[RI]) / (\max(RI) - E[RI])$$

Коригує RI на випадкове угадування. Очікуване значення для випадкового розбиття = 0.

Normalized Mutual Info (NMI)

$$NMI = I(Y;C) / \sqrt{H(Y) \cdot H(C)}$$

Взаємна інформація між мітками Y та кластерами C, нормована ентропіями. Від 0 до 1.

Homogeneity / Completeness

$$h = 1 - H(C|Y)/H(C) \quad / \quad c = 1 - H(Y|C)/H(Y)$$

Homogeneity: кожен кластер містить лише один клас.
Completeness: всі об'єкти класу в одному кластері.

V-measure

$$V = 2 \cdot h \cdot c / (h+c)$$

Гармонічне середнє Homogeneity і Completeness. Аналог F_1-score для кластеризації.

Fowlkes-Mallows (FMI)

$$FMI = TP / \sqrt{(TP+FP) \cdot (TP+FN)}$$

Геометричне середнє Precision та Recall за парами. Від 0 до 1, де 1 — ідеально.

Попередня обробка для кластеризації

Від сирих даних до якісних кластерів

01 Очищення даних

- Видалення дублікатів
- Обробка пропущених значень (imputation)
- Виявлення та обробка викидів

02 Масштабування

- StandardScaler: $z = (x - \mu)/\sigma$
- MinMaxScaler: $x' = (x - \min)/(\max - \min)$
- RobustScaler: медіана та IQR

03 Кодування ознак

- One-Hot Encoding для категорій
- Label Encoding
- Target Encoding

04 Зниження розмірності

- PCA для лінійного ущільнення
- t-SNE / UMAP для візуалізації
- Відбір ознак (Feature Selection)

Масштабування обов'язкове для K-Means, GMM, ієрархічної кластеризації. DBSCAN також чутливий до масштабу при підборі ε.

Реалізація в Python: scikit-learn

Pipeline кластеризації від завантаження до оцінки

cluster_analysis.py

```python
import numpy as np
import pandas as pd
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import KMeans, DBSCAN, AgglomerativeClustering
from sklearn.mixture import GaussianMixture
from sklearn.metrics import silhouette_score, davies_bouldin_score
from sklearn.decomposition import PCA

# 1. Завантаження та масштабування
X = pd.read_csv('data.csv').values
scaler = StandardScaler()
X_scaled = scaler.fit_transform(X)

# 2. K-Means з K-Means++
km = KMeans(n_clusters=3, init='k-means++', n_init=10, random_state=42)
labels_km = km.fit_predict(X_scaled)

# 3. DBSCAN
db = DBSCAN(eps=0.5, min_samples=5)
labels_db = db.fit_predict(X_scaled)

# 4. Оцінка якості
sil = silhouette_score(X_scaled, labels_km)
dbi = davies_bouldin_score(X_scaled, labels_km)
print(f'Silhouette: {sil:.3f}  |  DBI: {dbi:.3f}')
```

Ключові параметри:

`n_clusters`

Кількість кластерів k для KMeans/GMM

`init='k-means++'`

Покращена ініціалізація

`eps, min_samples`

Гіперпараметри DBSCAN

`n_init=10`

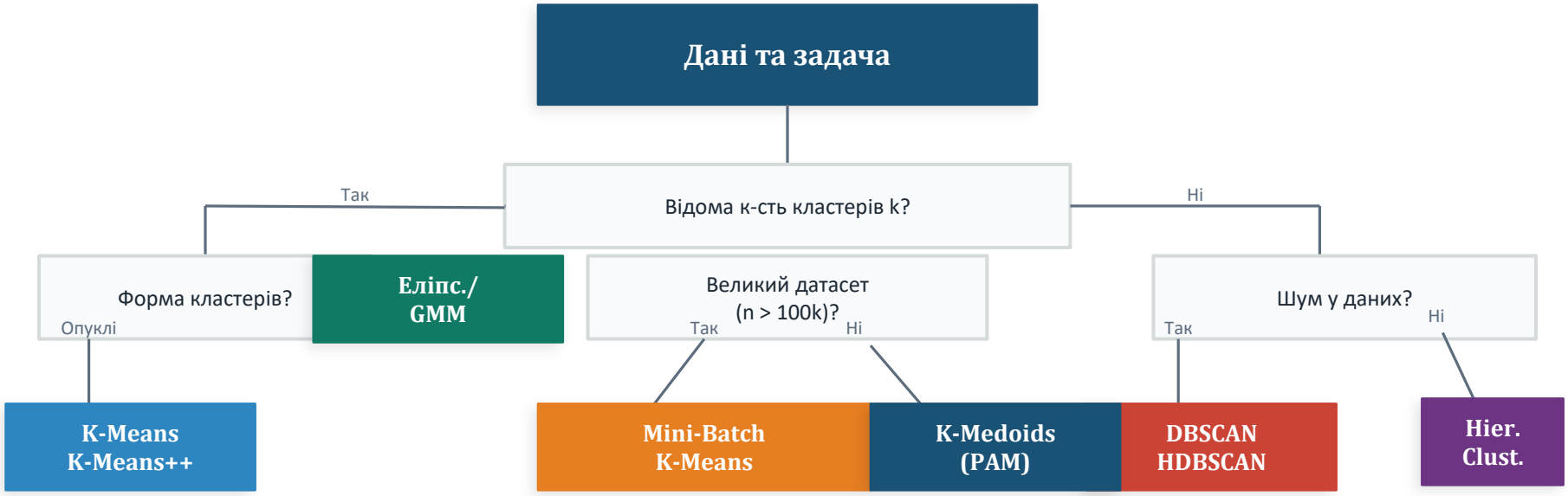
10 запусків, обрати найкращий

`random_state=42`

Відтворюваність результатів

Як обрати алгоритм кластеризації?

Покроковий гайд



Практична порада:
Завжди перевіряйте кілька алгоритмів та порівнюйте за Silhouette, DBI, CH. Використовуйте Elbow для K-Means і UMAP для візуалізації результатів.

Сегментація клієнтів (RFM)

Реальне застосування K-Means у маркетинговому аналізі

R — Recency
Скільки днів тому останній купив

F — Frequency
Кількість покупок за період

M — Monetary
Загальна сума витрат клієнта

Pipeline аналізу:

1. **Збір даних** — Транзакційні БД, CRM
2. **RFM-матриця** — Обрахунок R, F, M для кожного клієнта
3. **Нормалізація** — StandardScaler по R, F, M
4. **Elbow + K-Means** — Визначення k=4, навчання моделі
5. **Профілювання** — Характеристика кожного сегменту

Типові сегменти:

- **Champions** — Куплено нещодавно, часто, великі суми
- **Loyal Customers** — Регулярно купують, середня сума
- **At Risk** — Раніше активні, давно не купували
- **Lost Customers** — Давно, рідко, мала сума — відтік

Висновки

01 **Задача**

Кластеризація — unsupervised задача розбиття об'єктів на однорідні групи без апріорних міток.

02 **Методи**

K-Means (швидкий, опуклі), Hierarchical (дендрограма), DBSCAN (шум, довільні форми), GMM (ймовірнісний).

03 **Вибір k**

Elbow Method (WCSS), Silhouette Score, Davies-Bouldin, Calinski-Harabasz — комплексний аналіз.

04 **Попередня обробка**

Нормалізація обов'язкова. PCA/UMAP для зниження розмірності та візуалізації кластерів.

05 **Python Pipeline**

sklearn: KMeans, DBSCAN, AgglomerativeClustering, GaussianMixture + метрики + візуалізація.

06 **Практика**

Сегментація клієнтів (RFM), біоінформатика, аномалії, NLP — широкий спектр застосувань.