Практична робота №7

Просторова кластеризація. Алгоритми DBSCAN та K-Means

Мета роботи: набути практичних навичок побудови та аналізу просторової кластеризації з використанням алгоритмів DBSCAN і K-Means засобами Python. Опанувати повний аналітичний цикл: завантаження та очищення геопросторових даних, вибір та налаштування алгоритмів кластеризації, оцінювання якості кластерів внутрішніми метриками, картографічна візуалізація результатів та порівняльний аналіз методів.

Стек технологій:

scikit-learn 1.x - *алгоритми KMeans, DBSCAN, AgglomerativeClustering; метрики якості кластеризації, scikit-learn.org*

pandas / NumPy - *маніпуляції з табличними та числовими даними*

Matplotlib / Seaborn - *побудова графіків, теплових карт, гістограм*

Basemap / Folium - *картографічна візуалізація геопросторових даних*

Google Colaboratory - *хмарне середовище виконання, безкоштовний доступ до GPU*

SciPy — *ієрархічна кластеризація, дендрограми (linkage, dendrogram)*

Теоретичний матеріал

Кластеризація - це задача розбиття множини об'єктів $X = \{x_1, x_2, \ldots, x_n\}$ на k груп (кластерів) $C_1, C_2, \ldots, C_k$ таким чином, що об'єкти всередині кластера є максимально подібними між собою, а об'єкти з різних кластерів - максимально відмінними. На відміну від класифікації, в задачі кластеризації мітки класів наперед невідомі та визначаються алгоритмом у процесі навчання.

Формально задача кластеризації зводиться до знаходження відображення $\varphi : X \to \{1, 2, \ldots, k\}$, що мінімізує деякий функціонал якості. Для K-Means - це функціонал суми квадратів внутрішньокластерних відстаней (WCSS):

$$J = \sum_{j=1}^{k} \sum_{x_i \in C_j} \|x_i - \mu_j\|^2$$

де μ_j - центроїд j-го кластера. Мінімізація J є NP-складною задачею в загальному випадку, тому K-Means використовує ітеративну евристику Ллойда.

DBSCAN (Density-Based Spatial Clustering of Applications with Noise, Ester et al., 1996) - алгоритм кластеризації, що ґрунтується на щільності розташування точок у просторі ознак. На відміну від K-Means, DBSCAN не потребує задання кількості кластерів наперед та здатний виявляти кластери довільної форми і шуму.

Алгоритм задається двома основними параметрами: ε - радіус уявного кола навколо точки та MinPts — мінімальна кількість точок, які повинні потрапити в це коло.

Залежно від цих параметрів кожна точка відноситься до одного з типів:

– *Точка ядро (core point)* - має достатню кількість сусідів (не менше MinPts) у межах ε, тому формує кластер.

– *Гранична точка (border point)* - знаходиться поруч із ядром, але сама не має достатньої кількості сусідів.

– *Шумова точка (noise point)* - не належить до жодного кластера та розглядається як окрема (позначається −1).

Алгоритм DBSCAN складається з таких кроків:

- Обираємо випадкову точку.
- Перевіряємо ε.
- Якщо це core → створюємо кластер.
- Розширюємо кластер:
  - додаємо сусідів
  - перевіряємо їх теж
- Повторюємо для всіх точок.

Складність DBSCAN становить O(nlogn) при використанні просторового індексу (kd-tree або ball-tree), що робить його ефективним для великих геопросторових датасетів.

Таблиця 1. *Порівняння ключових характеристик алгоритмів кластеризації*

Характеристика	**K-Means**	**K-Medoids**	**DBSCAN**	**HDBSCAN**	**Агломерат.**
Задати k наперед	Так	Так	Ні	Ні	Ні*
Форма кластерів	Опуклі	Опуклі	Довільна	Довільна	Довільна
Стійкість до шуму	Низька	Середня	Висока	Висока	Середня
Складність	O(nkd·iter)	O(k(n-k)²)	O(n log n)	O(n log n)	O(n²)
Стійкість до викидів	Низька	Висока	Висока	Висока	Середня

K-Means (MacQueen, 1967) - один із найвідоміших алгоритмів навчання без учителя. Ітеративний алгоритм Ллойда складається з чотирьох кроків:

1. *Ініціалізація***.** Обрати k початкових центроїдів μ_1, …, μ_k (випадково або за K-Means++).

2. *Призначення***.** Кожна точка x_i відноситься до найближчого кластера:

$$c_i = \arg\min_{j \in \{1,\ldots,k\}} \|x_i - \mu_j\|^2$$

3. *Оновлення***.** Перерахунок центроїдів як середнє значень кластера:

$$\mu_j = \frac{1}{|C_j|} \sum_{i \in C_j} x_i$$

4. *Збіжність***.** Повторювати кроки 2–3 до стабілізації центроїдів або досягнення maxiter.

Метод ліктя (Elbow Method) є стандартним підходом для вибору k. Будується залежність WCSS від k, і оптимальним вважається значення, після якого приріст WCSS різко сповільнюється:

$$\mathrm{WCSS}(k) = \sum_{i=1}^{n} \|x_i - \mu_{c_i}\|^2$$

K-Means++ - вдосконалена стратегія ініціалізації (Arthur & Vassilvitskii, 2007): перший центроїд обирається випадково, кожен наступний - з ймовірністю пропорційною квадрату відстані до найближчого вже обраного центроїда. Це зменшує кількість ітерацій та підвищує якість фінальної кластеризації.

Метрики якості кластеризації. Оскільки в задачі кластеризації відсутні еталонні мітки, для оцінки якості використовуються внутрішні метрики:

Таблиця 2. *Внутрішні метрики якості кластеризації*

Метрика	**Формула / Визначення**	**Інтерпретація**	**Діапазон**
Silhouette Score	$s(i) = \frac{b(i) - a(i)}{\max(a(i), b(i))}$	Близько до 1 - ідеальна кластеризація	[−1, 1]
Davies-Bouldin	$DB = \frac{1}{k}\sum_{i=1}^{k} \max_{j \neq i} \left(\frac{\sigma_i + \sigma_j}{d(\mu_i, \mu_j)} \right)$	Менше - краще	[0, ∞)
Calinski-Harabasz	$CH = \frac{SS_B}{SS_W} \cdot \frac{n-k}{k-1}$	Більше - краще	[0, ∞)
Inertia (WCSS)	$\sum_{i=1}^{n} \|x_i - \mu_{c_i}\|^2$	Менше - компактніші кластери	[0, ∞)

Зміст роботи

Завдання 1. *Для набору даних eng-climate-summaries-All-2_2015.csv провести просторову кластеризацію з використанням алгоритму DBSCAN. Послідовність виконання:*

5. *Завантаження та первинний аналіз.* **З**авантажте CSV-файл, виведіть перші 10 рядків, визначте типи стовпців та кількість пропущених значень.

6. *Очищення даних.* Вилучіть рядки з пропущеними значеннями у полях Lat, Long, Tm. Виведіть кількість видалених рядків та фінальний розмір датасету.

7. *Просторова візуалізація***.** Нанесіть усі метеостанції на карту Канади (Basemap або Folium). Оцініть просторовий розподіл точок.

8. *Нормалізація координат*. Перетворіть географічні координати (Lat, Long) у декартові (xm, ym) через проекцію Меркатора та стандартизуйте StandardScaler.

9. *Застосуйте DBSCAN(eps=0.15, min_samples=10).* Підрахуйте кількість кластерів та відсоток шумових точок.

10. Відобразіть кластери різними кольорами на карті. Для кожного кластера вивести середню температуру Tm та координати центру.

11. *Дослідження гіперпараметрів*. Змініть eps від 0.05 до 0.50 (крок 0.05) та min_samples від 3 до 20. Побудуйте теплову карту (*heatmap*) Silhouette Score для всіх комбінацій. Визначте оптимальні значення.

12. *Аналіз результатів.* Проінтерпретуйте отримані кластери з географічної та кліматичної точок зору. Чи відповідають кластери реальним кліматичним зонам Канади?

Завдання 2. *Для eng-climate-summaries-All-2_2015.csv реалізуйте алгоритм кластеризації K-Means. Результати відобразити на карті та порівняти з результатами DBSCAN.*

1. *Метод ліктя (Elbow Method)***.** Побудуйте графік WCSS для k від 2 до 12. Визначте оптимальне k. Обґрунтуйте вибір.

2. Застосуйте *KMeans*(n_clusters=k_opt, init='k-means++', n_init=10). Виведіть мітки та координати центроїдів.

3. *Картографічна візуалізація*. відобразіть кластери *K-Means* на карті. Порівняйте візуально з результатами *DBSCAN*.

4. Реалізуйте один з методів (таблиця 3). Порівняйте з базовим *K-Means* за метриками якості.

5. Обчисліть для всіх методів: *Silhouette Score, Davies-Bouldin Index, Calinski-Harabasz Index*.

6. Сформуйте підсумкову таблицю (таблиця 4) для всіх реалізованих методів: кількість кластерів, шум, три метрики якості, час виконання.

Таблиця 3. *Варіанти методів кластеризації*

Варіант	Метод	Ключова відмінність від K-Means
1, 7, 13, 19	K-Medoids (PAM)	Центроїд — реальний об'єкт датасету; стійкий до викидів
2, 8, 14, 20	K-Medians	Медіана замість середнього; стійкіший до аномалій
3, 9, 15, 21	K-Means++	Покращена ініціалізація центроїдів; менше ітерацій
4, 10, 16, 22	K-Modes	Для категоріальних даних; мода замість середнього
5, 11, 17, 23	Mini-Batch K-Means	Навчання на міні-батчах; масштабується на великі дані
6, 12, 18, 24	Bisecting K-Means	Ділення на 2 на кожному кроці; ієрархічний підхід

Завдання 3. Кластеризація за кліматичними ознаками.

1. Сформуйте матрицю ознак X із стовпців: *Lat*, *Long*, *Tm* (середня температура), *P* (опади), *S* (снігопад), *BS* (сонячне сяйво). Вилучіть рядки з пропусками.

2. Застосуйте *StandardScaler* до всіх ознак (включаючи координати). Проаналізуйте вплив масштабування на результат.

3. *K-Means по кліматичних ознаках*. Підберіть оптимальне k методом ліктя + *Silhouette*. Навчіть модель та додайте мітки до *DataFrame*.

4. Для кожного кластера обчисліть середнє значення всіх ознак. Побудуйте *radar chart (polar plot)* для візуального порівняння профілів кластерів.

5. Застосуйте DBSCAN до тих самих ознак. Підберіть оптимальні *eps* та *min_samples* через *GridSearch*. Порівняйте з *K-Means*.

6. Нанесіть кліматичні кластери на карту. Оцініть географічну зв'язність кластерів (чи перебувають вони в суміжних регіонах).

Завдання 4. *Оберіть одне із трьох завдань:*

– *Ієрархічна кластеризація.* Реалізуйте *AgglomerativeClustering* з методами зв'язку: *single, complete, average, Ward*. Побудуйте дендрограму для підвибірки (n≤100). Порівняйте з *K-Means* та *DBSCAN* за *Silhouette Score* та *Davies-Bouldin Index*. Визначте, який метод зв'язку краще відповідає географічній структурі даних.

– *Аналіз часових змін.* Завантажте дані за кілька місяців/років (якщо доступно). Застосуйте кластеризацію для кожного часового зрізу. Побудуйте анімовану карту (*FuncAnimation* або *GIF*) зміни кластерів. Проаналізуйте сезонні патерни кліматичних зон.

– *PCA + кластеризація.* Застосуйте *PCA* для зниження розмірності простору ознак до 2D та 3D. Побудуйте *scatter plot* у зниженому просторі з кольором за кластерами *K-Means* та *DBSCAN*. Проаналізуйте, яку частку дисперсії пояснюють перші компоненти. Порівняйте результати кластеризації в оригінальному та зниженому просторі.

Методичні рекомендації

Крок 1. Завантаження та первинна перевірка даних. Читаємо CSV-файл з кліматичними зведеннями метеостанцій Канади за 2015 рік з кодуванням latin-1 (нестандартні символи у назвах). Виводимо розмір таблиці, типи стовпців і топ-10 стовпців за кількістю пропущених значень, щоб зрозуміти якість даних перед будь-яким аналізом. Бібліотеки, які знадобляться для роботи:

```
import numpy as np
import pandas as pd
import seaborn as sns
import matplotlib.pyplot as plt
from sklearn.preprocessing import StandardScaler
from sklearn.cluster import DBSCAN, KMeans
from sklearn.metrics import silhouette_score, davies_bouldin_score
```

Крок 2. Очищення та аналіз розподілу. Видаляємо рядки, де відсутні координати станції (Lat, Long) або середня температура (Tm) - без цих полів запис непридатний для просторового аналізу. Виводимо базову описову статистику (mean, std, min/max, quartiles) для числових ознак.

```
print(df_clean[['Lat','Long','Tm','P','S']].describe())
```

де P - опади (мм), S - сніговий покрив (см). *describe()* одразу виявляє аномальні значення (наприклад, координати поза Канадою).

Крок 3. Просторова візуалізація. Фільтруємо станції за географічним вікном Канади (довгота −140…−50°, широта 40…65°). За допомогою *Basemap (обов'язково виконати установку)* будуємо карту у проєкції Меркатора з береговими лініями, кордонами та рельєфним тлом. Координати перетворюємо з градусів у пікселі карти та наносимо кожну станцію червоною точкою.

```
plt.figure(figsize=(14, 10))
Long_range = [-140, -50]
Lat_range  = [40, 65]
```

Фільтрація за діапазоном координат

```
df = df_clean[(df_clean['Long'] > Long_range[0]) & (df_clean['Long'] <
Long_range[1]) & (df_clean['Lat']  > Lat_range[0]) & (df_clean['Lat']  <
Lat_range[1])]
```

Побудова базової карти (Basemap)

```
from mpl_toolkits.basemap import Basemap
my_map = Basemap(projection='merc', resolution='l', area_thresh=1000.0,
    llcrnrlon=Long_range[0], urcrnrlon=Long_range[1],
    llcrnrlat=Lat_range[0],  urcrnrlat=Lat_range[1])
my_map.drawcoastlines()
my_map.drawcountries()
my_map.fillcontinents(color='grey', alpha=0.3)
my_map.shadedrelief()
```

Перетворення координат та нанесення точок

```
X_map, Y_map = my_map(df.Long.values, df.Lat.values)
df['xm'] = X_map
df['ym'] = Y_map
for (x, y) in zip(X_map, Y_map):
    my_map.plot(x, y, markerfacecolor='red', marker='o',
                markersize=5, alpha=0.75)
plt.title('Метеостанції Канади (2015)', fontsize=14)
plt.tight_layout(); plt.show()
```

Крок 4. Кластеризація DBSCAN. Стандартизуємо координати (нуль - середнє, одиниця - стандартне відхилення), щоб DBSCAN працював з рівномірним простором відстаней. Запускаємо DBSCAN(eps=0.15, min_samples=10): мітка −1 означає шум, всі інші - номер кластера. Зсуваємо мітки на +1, щоб шум мав значення 0 замість −1. Обчислюємо Silhouette Score та Davies-Bouldin Index лише на не-шумових точках.

```
from sklearn.utils import check_random_state
check_random_state(1000)
```

Стандартизація координат

```
loc = np.column_stack([X_map, Y_map])
loc = np.nan_to_num(loc)
scaler = StandardScaler()
loc_scaled = scaler.fit_transform(loc)
```

Застосування DBSCAN

```
db = DBSCAN(eps=0.15, min_samples=10)
db.fit(loc_scaled)
labels = db.labels_ + 1
df['DBSCAN_Clusters'] = labels

n_clusters = len(set(labels)) - (1 if 0 in labels else 0)
n_noise    = (labels == 0).sum()
print(f'Кластерів: {n_clusters} | Шумових точок: {n_noise}
({100*n_noise/len(labels):.1f}%)')
```

Silhouette Score (лише для не-шумових точок)

```
mask_valid = labels > 0
if mask_valid.sum() > 1 and n_clusters > 1:
    sil = silhouette_score(loc_scaled[mask_valid], labels[mask_valid])
    dbi = davies_bouldin_score(loc_scaled[mask_valid], labels[mask_valid])
    print(f'Silhouette: {sil:.4f} | Davies-Bouldin: {dbi:.4f}')
```

Крок 5. Дослідження гіперпараметрів DBSCAN. Перебираємо всі комбінації eps (0.05–0.50 із кроком 0.05) та min_samples (3–20) і записуємо Silhouette Score у матрицю. Будуємо теплову карту, це дозволить знайти оптимальні гіперпараметри без ручного підбору.

```
eps_range       = np.arange(0.05, 0.55, 0.05)
min_samples_range = range(3, 21)
sil_matrix = np.zeros((len(eps_range), len(min_samples_range)))

for i, eps in enumerate(eps_range):
    for j, ms in enumerate(min_samples_range):
        db_tmp = DBSCAN(eps=eps, min_samples=ms).fit(loc_scaled)
        lbl = db_tmp.labels_
        n_cl = len(set(lbl)) - (1 if -1 in lbl else 0)
        if n_cl > 1 and (lbl != -1).sum() > 1:
            mask = lbl != -1
            sil_matrix[i, j] = silhouette_score(loc_scaled[mask],
lbl[mask])

plt.figure(figsize=(12, 8))
sns.heatmap(sil_matrix, xticklabels=list(min_samples_range),
```

```
            yticklabels=[f'{e:.2f}' for e in eps_range],
            cmap='YlGnBu', annot=True, fmt='.2f')
plt.title('Silhouette Score: eps × min_samples', fontsize=13)
plt.xlabel('min_samples'); plt.ylabel('eps')
plt.tight_layout(); plt.show()
```

Повний перебір на сітці (grid search) - стандартний підхід для DBSCAN, оскільки немає аналітичного способу вибору параметрів. Чим більша матриця, тим точніший вибір, але довше виконання.

Крок 6. Метод ліктя для K-Means. Для кожного k від 2 до 12 навчаємо KMeans і записуємо WCSS (Within-Cluster Sum of Squares) - суму квадратів відстаней від кожної точки до її центроїда. Будуємо графік WCSS до k.

```
wcss = []
K_range = range(2, 13)
for k in K_range:
    km = KMeans(n_clusters=k, init='k-means++', n_init=10,
                random_state=42, max_iter=300)
    km.fit(loc_scaled)
    wcss.append(km.inertia_)

plt.figure(figsize=(10, 5))
plt.plot(K_range, wcss, 'bo-', linewidth=2, markersize=8)
plt.title('Метод ліктя (Elbow Method)', fontsize=13)
plt.xlabel('Кількість кластерів k')
plt.ylabel('WCSS (Inertia)')
plt.xticks(K_range); plt.grid(alpha=0.4)
plt.tight_layout(); plt.show()
```

init='k-means++' - розумна ініціалізація центроїді, де кожен наступний вибирається пропорційно квадрату відстані до найближчого вже обраного. n_init=10 - 10 запусків з різними стартами, повертається найкращий.

Крок 7. K-Means кластеризація та візуалізація. Використовуємо k, знайдене на кроці 6 (наприклад, 7), для фінального навчання K-Means. Зберігаємо мітки кластерів у DataFrame. Будуємо карту, де кожна станція пофарбована відповідно до свого кластера палітрою tab10. Виводимо Silhouette Score та DBI для оцінки якості.

```
k_opt = 7   # змінити відповідно до графіка ліктя
km_opt = KMeans(n_clusters=k_opt, init='k-means++', n_init=10,
random_state=42)
km_opt.fit(loc_scaled)
```

```
df['KMeans_Clusters'] = km_opt.labels_
```

Метрики якості K-Means

```
sil_km = silhouette_score(loc_scaled, km_opt.labels_)
dbi_km = davies_bouldin_score(loc_scaled, km_opt.labels_)
print(f'K-Means | k={k_opt} | Silhouette: {sil_km:.4f} | Davies-Bouldin:
{dbi_km:.4f}')
```

Візуалізація на карті

```
plt.figure(figsize=(14, 10))
my_map2 = Basemap(projection='merc', resolution='l', area_thresh=1000.0,
    llcrnrlon=Long_range[0], urcrnrlon=Long_range[1],
    llcrnrlat=Lat_range[0],  urcrnrlat=Lat_range[1])
my_map2.drawcoastlines(); my_map2.drawcountries()
my_map2.fillcontinents(color='grey', alpha=0.3)
colors_km = plt.get_cmap('tab10')(np.linspace(0, 1, k_opt))
for _, row in df.iterrows():
    my_map2.plot(row.xm, row.ym,
                 markerfacecolor=colors_km[row.KMeans_Clusters],
                 marker='o', markersize=5, alpha=0.8)
plt.title(f'K-Means кластеризація (k={k_opt})', fontsize=14)
plt.tight_layout(); plt.show()
```

Крок 8. Порівняльна таблиця методів. Запускаємо DBSCAN і K-Means ще раз, вимірюючи час виконання через time(). Формуємо таблицю з шістьма стовпцями (Таблиця 4): *назва методу, кількість кластерів, кількість шумових точок, Silhouette Score, Davies-Bouldin Index і час*.

Таблиця 4 *Порівняльна таблиця результатів кластеризації (заповнюється за результатами виконання)*

Метод	k/ε	Шум, %	Silhouette ↑	Davies-Bouldin ↓	Calinski-H. ↑	Час, с
DBSCAN	ε=0.15, min=10	—	—	—	—	—
K-Means (k=opt)	—	0	—	—	—	—
Варіант (з таблиці 3)	—	—	—	—	—	—

Контрольні запитання

1. У чому принципова відмінність задачі кластеризації від задачі класифікації? Наведіть приклади практичних задач кожного типу.

2. Дайте формальне означення кластера. Які критерії якості кластеризації можна запропонувати?

3. Охарактеризуйте алгоритм DBSCAN: ядрові, граничні та шумові точки. Яка обчислювальна складність алгоритму?

4. Які переваги та обмеження DBSCAN у порівнянні з K-Means? За яких умов DBSCAN є кращим вибором?

5. Опишіть алгоритм K-Means. Яку цільову функцію він мінімізує?

6. Чи є алгоритм K-Means точним або евристичним? Чому K-Means++ є кращою ініціалізацією?

7. Наведіть та поясніть щонайменше три функції відстані. Яку функцію доцільно використовувати для геопросторових даних?

8. Як вибрати оптимальну кількість кластерів? Поясніть метод ліктя (Elbow) та метод силуету. Яка різниця?

9. Що таке Silhouette Score? Які значення свідчать про якісну кластеризацію?

10. Порівняйте метрики Davies-Bouldin та Calinski-Harabasz. Яка з них чутливіша до форми кластерів?

11. Чому результат K-Means залежить від початкових центрів? Як мінімізувати цей вплив?

12. Опишіть обраний вами варіантний метод (з таблиці 3). У чому його ключова відмінність від базового K-Means?

13. Що таке стандартизація (StandardScaler) і чому вона є необхідною перед кластеризацією?

14. Поясніть, що означають кліматичні кластери, отримані у Завданні 3. Чи збігаються вони з відомими кліматичними зонами Канади?

15. Що таке HDBSCAN? У чому його переваги перед класичним DBSCAN при роботі з даними різної щільності?

Структура датасету

Датасет eng-climate-summaries-All-2_2015.csv — кліматичні зведення канадських метеостанцій за 2015 рік. Ключові поля:

Таблиця 6 *Опис полів датасету*

Поле	Тип	Одиниці	Опис
Stn_Name	str	—	Назва метеостанції
Lat	float	градуси	Широта (північ +, градуси)
Long	float	градуси	Довгота (захід −, градуси)
Prov	str	—	Провінція Канади
Tm	float	°C	Середня місячна температура
D	float	°C	Відхилення від норми 1981–2010
Tx	float	°C	Найвища максимальна температура за місяць
Tn	float	°C	Найнижча мінімальна температура за місяць
S	float	см	Снігопад (загальний за місяць)
P	float	мм	Загальна кількість опадів
S%N	float	%	Відсоток норми за снігопадами (1981–2010)
P%N	float	%	Відсоток норми за опадами (1981–2010)
BS	float	год	Яскраве сонячне сяйво (годин на місяць)
HDD	float	°C·день	Градусо-дні нижче 18 °C (опалення)
CDD	float	°C·день	Градусо-дні вище 18 °C (охолодження)
Stn_No	str	—	Ідентифікатор станції (перші 3 цифри = водозбір)

Список рекомендованих джерел

[1] M. Ester, H.-P. Kriegel, J. Sander, X. Xu, «A Density-Based Algorithm for Discovering Clusters in Large Spatial Databases with Noise,» KDD, 1996, pp. 226–231.

[2] J. MacQueen, «Some methods for classification and analysis of multivariate observations,» Proc. 5th Berkeley Symp., vol. 1, pp. 281–297, 1967.

[3] D. Arthur, S. Vassilvitskii, «k-means++: the advantages of careful seeding,» Proc. SODA, 2007.

[4] P. J. Rousseeuw, «Silhouettes: a graphical aid to the interpretation and validation of cluster analysis,» J. Comput. Appl. Math., vol. 20, pp. 53–65, 1987.

[5] scikit-learn: DBSCAN, KMeans документація. [Електронний ресурс]. Режим доступу: https://scikit-learn.org/stable/modules/clustering.html

[6] Real Python, «K-Means Clustering in Python: A Practical Guide». [Електронний ресурс]. Режим доступу: https://realpython.com/k-means-clustering-python/