# Лабораторна робота №6

## Кластеризація. Базові алгоритми кластеризації

***Мета роботи:*** Отримання навичок практичного застосування методів кластерного аналізу.

***Література***

<https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html><http://pypr.sourceforge.net/kmeans.html#k-means-example>

Scipy.org : [*https://docs.scipy.org/doc/scipy/reference/cluster.hierarchy.html*](https://docs.scipy.org/doc/scipy/reference/cluster.hierarchy.html)

Hierarchical clustering https://en.wikipedia.org/wiki/Hierarchical\_clustering

KMeans: [*https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html*](https://scikit-learn.org/stable/modules/generated/sklearn.cluster.KMeans.html)

## Зміст роботи

### ***Завдання 1.*** Ретельно опрацювати теоретичні відомості:

* ознайомитися з алгоритмом K- Means;
* ознайомитися з різними мірами відстаней;
* розібрати приклад;

### ***Завдання 2.*** За допомогою алгоритму K-means розбити заданий набір даних на кластери.

* Розібрати приклад 1;
* Згенерувати масив даних для проведення кластеризації;

|  |  |  |
| --- | --- | --- |
| Варіант | Кількість точок | Кількість кластерів |
| 1, 6, 11 | 2500 | 12 |
| 2, 7, 12 | 1300 | 10 |
| 3, 8, 13 | 2100 | 25 |
| 4, 9, 14 | 3500 | 17 |
| 5, 10, 15 | 3600 | 8 |

* Вибрати різне число кластерів випадково;
* Для визначення кількості кластерів застосувати метод «ліктя»

### ***Завдання 3.*** Провести кластерний аналіз заданого набору даних:

|  |  |
| --- | --- |
| Варіант | Вхідні дані |
| 1, 6, 11 | [7,8],[2,2],[7,9],[6,5],[3,7],[7,5],[7,5],[2,8],[3,6],[8,6] |
| 2, 7, 12 | [1,8],[2,8],[17,9],[6,15],[13,7],[7,15],[8,5],[7,28],[3,16],[18,6] |
| 3, 8, 13 | [7,25],[12,2],[7,19],[26,15],[13,17],[7,5],[27,15],[12,8],[13,26],[28,16] |
| 4, 9, 14 | [4,8],[23,22],[17,29],[16,5],[13,17],[27,35],[27,25],[12,28],[23,6],[18,6] |
| 5, 10,  15 | [7,14],[32,2],[17,29],[16,15],[13,27],[17,25],[11,15],[32,18],[23,6],[13,4] |

* Розібрати приклад 2;
* Побудуйте дендрограму;

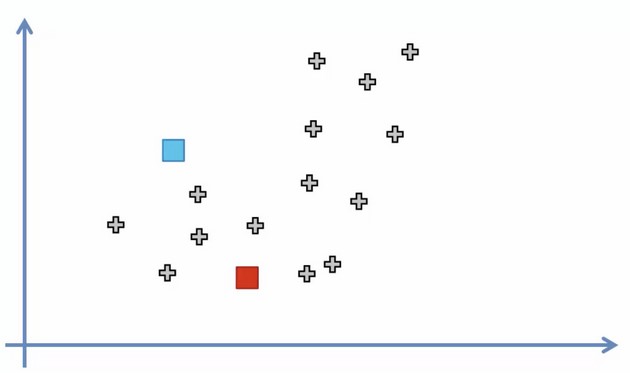
## Методичні рекомендації

*Кластеризація* (або кластерний аналіз) - це задача розбиття множини об'єктів на групи, які називаються кластерами. Усередині кожної групи повинні виявитися «схожі» об'єкти, а об'єкти різних групи повинні бути якомога більш відмінні. Головна відмінність кластеризації від класифікації полягає в тому, що перелік груп чітко не заданий і визначається в процесі роботи алгоритму.

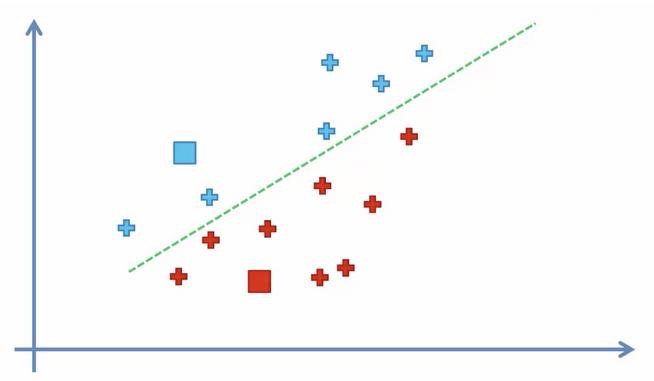
*K-Means (MacQueen, 1967)* - один з найпростіших алгоритмів навчання без контролю, який вирішує добре відому проблему кластеризації. Це простий спосіб класифікації заданого набору даних через певну кількість кластерів (наприклад, k). Основна ідея полягає в тому, щоб визначити k центроїдів, по одному для кожного кластера. Кращий вибір центроїда - розмістити їх якомога далі один від одного.

Алгоритм складається з наступних кроків:

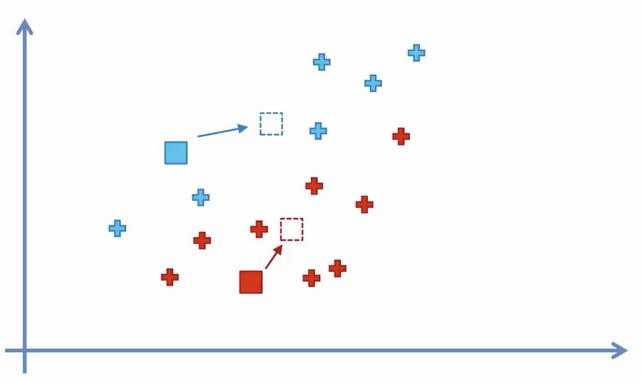
1. Виберіть K (на прикладі 2) випадкових точок як центри кластерів, що називаються *центроїдами*:



1. Згрупуйте точки даних до найближчого кластеру, обчисливши їх відстань відносно кожного центроїда:

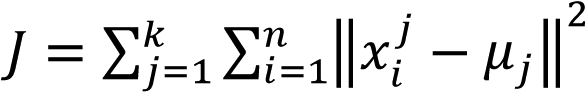


1. Визначте новий центр кластера, обчисливши середнє значення точок:



1. Повторюйте кроки 2 і 3 до того моменту, поки не досягнемо критерія збіжності:

Цей алгоритм спрямований на мінімізацію цільової функції, в даному випадку квадрат функції помилок. Цільова функція:

,

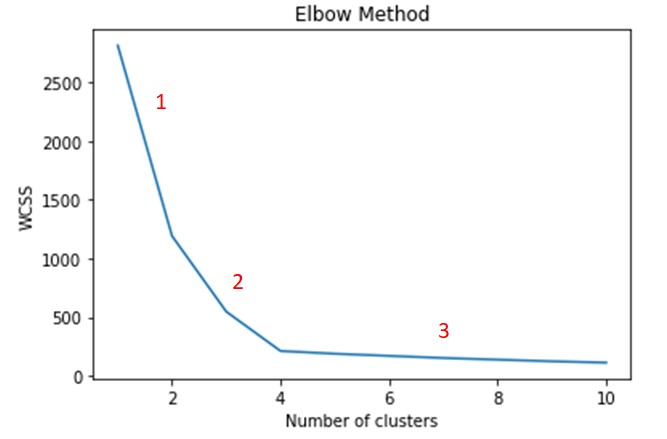
де ‖𝑥𝑖𝑗 − 𝜇𝑗‖2обрана міра відстані між точкою даних 𝑥𝑖𝑗 і центром кластера 𝜇𝑗, є показником відстані *n* точок даних від їх відповідних центрів кластера.

Алгоритм k-Means не завжди знаходить найбільш оптимальну конфігурацію, відповідну мінімуму глобальної цільової функції. Алгоритм також істотно чутливий до початкового випадково вибраного кластерного центра.

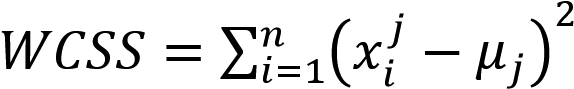
*Як вибрати потрібну кількість кластерів?*

Часто дані матимуть декілька вимірів, що ускладнює їх візуалізацію і як наслідок, кількість кластерів вже не очевидна. Є спосіб визначити це математично.

Необхідно побудувати графік залежності між кількістю кластерів і сумою квадратів в кластері (Within Cluster Sum of Squares,WCSS), а потім вибрати кількість кластерів, в яких зміна рівня починає вирівнюватися (*метод ліктя*).



WCSS визначається як сума квадратів відстані між кожним членом кластера і його центроїдом.

,

де 𝜇𝑗 - центр кластера, j – номер кластера

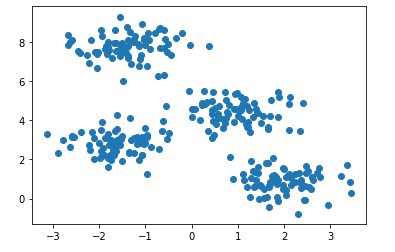
#### Приклад 1:

Як завжди, потрібно почати з імпорту потрібних бібліотек.

import numpy as np import pandas as pd from matplotlib import pyplot as plt from sklearn.datasets.samples\_generator import make\_blobs from sklearn.cluster import KMeans

Генеруємо дані за допомогою функції *make\_blobs* з модуля *sklearn.datasets*. Параметр *centers* вказує кількість кластерів.

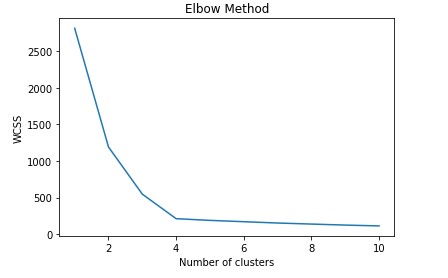
X,y=make\_blobs(n\_samples=300, centers=4, cluster\_std=0.60, random\_state=0) plt.scatter(X[:,0], X[:,1])



Незважаючи на те, що відома кількість кластерів, перевіримо визначення кількості кластерів методом «ліктя». wcss = []

for i in range(1, 11):

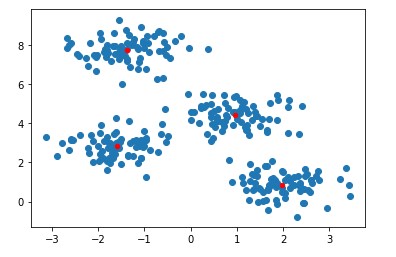
kmeans = KMeans(n\_clusters=i, init='kmeans++', max\_iter=300, n\_init=10, random\_state=0) kmeans.fit(X) wcss.append(kmeans.inertia\_) plt.plot(range(1, 11), wcss) plt.title('Elbow Method') plt.xlabel('Number of clusters') plt.ylabel('WCSS') plt.show()



Далі проводимо класифікацію, використовуючи оптимальну кількість кластерів (4), яку визначили на попередньому кроці.

kmeans = KMeans(n\_clusters=4, init='kmeans++', max\_iter=300, n\_init=10, random\_state=0) pred\_y = kmeans.fit\_predict(X)

plt.scatter(X[:,0], X[:,1]) plt.scatter(kmeans.cluster\_centers\_[:, 0], kmeans.cluster\_centers\_[:, 1], s=300, c='red') plt.show()



На даний час існує багато варіацій цього методу, що частково усувають недоліки, середніх: K‑Medoids, K‑Medians, K‑Modes, K‑means++, Intelligent K‑Means, Genetic K‑Means.

#### Приклад 2

Для відображення масиву даних знадобляться наступні бібліотеки:

import matplotlib.pyplot as plt

import numpy as np

Побудуємо на координатної площині задані точки:

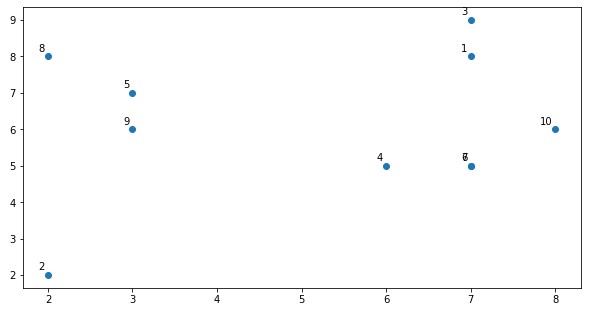
X=np.array([[7,8],[2,2],[7,9],[6,5],[3,7],[7,5],[7,5],[2,8],[3,6],[8,6],]) labels = range(1, 11) plt.figure(figsize = (10, 5)) plt.subplots\_adjust(bottom = 0.1) plt.scatter(X[:,0],X[:,1], label = 'True Position')

for label, x, y in zip(labels, X[:, 0], X[:, 1]):

plt.annotate( label,xy = (x, y), xytext = (-

3, 3),textcoords = 'offset points', ha = 'right', va = 'bottom') plt.show()

Результат:

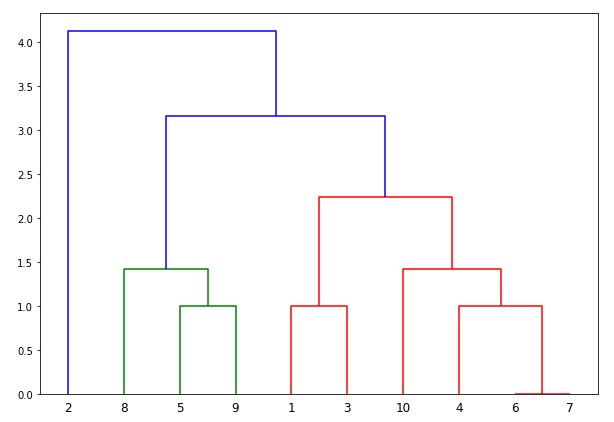


На діаграмі можна побачити 3 або 4 кластери, але у реальних даних може бути тисячі кластерів. Далі потрібно будувати дендрограми точок даних за допомогою бібліотеки Scipy

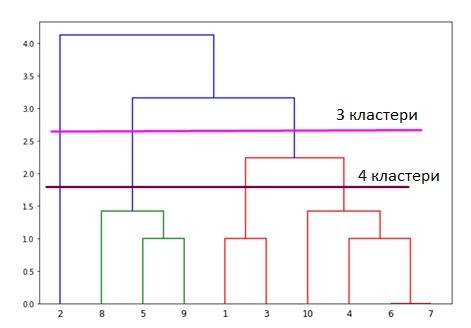
from scipy.cluster.hierarchy import dendrogram, linkage from matplotlib import pyplot as plt

linked = linkage(X, 'single') labelList = range(1, 11) plt.figure(figsize = (10, 7)) dendrogram(linked, orientation = 'top',labels = labelList, distance\_sort ='descending',show\_leaf\_counts = True) plt.show()

Результат:



Тепер, коли сформовано кластер, вибираємо найдовшу вертикальну відстань. Потім через неї проводимо лінію, як показано на наступному малюнку.



Оскільки горизонтальну лінію можна провести 2 способами, то розглянемо 2 варіанта рішення.

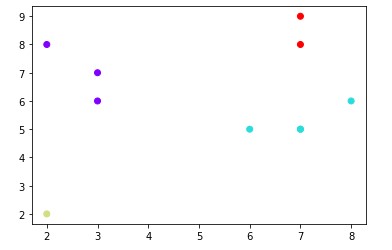
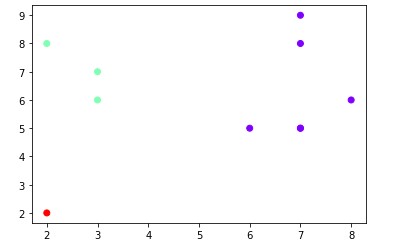
Далі потрібно імпортувати клас для кластеризації і викликати його метод fit\_predict для прогнозування кластера. Імпортуємо клас

AgglomerativeClustering бібліотеки sklearn.cluster –

from sklearn.cluster import AgglomerativeClustering cluster = AgglomerativeClustering(n\_clusters = 3, affinity = 'euclidean', linkage = 'ward') cluster.fit\_predict(X)

Потім побудуйте кластер за допомогою наступного коду -

plt.scatter(X[:,0],X[:,1], c = cluster.labels\_, cmap = 'rainbow') Результат:



На першому рисунку представлено 3 кластера на другому 4.

### Контрольні запитання

1. У чому полягає суть кластерного аналізу?
2. Як провести кластерний аналіз методом K-середніх?
3. Наведіть приклади функцій відстані між кластерами.
4. Опишіть алгоритм K-means - сформулюйте завдання оптимізації і опишіть алгоритм рішення.
5. Алгоритм K-means є точним або наближеним? Чому?
6. Як залежить результат кластеризації від початкових центрів? Як можна обійти цей недолік?
7. Як вибрати кількість кластерів?
8. Прокоментуйте переваги і недоліки методу кластеризації Kmeans.
9. Які ієрархічні методи кластерного аналізу вам відомі і як провести аналіз?
10. Чому необхідно унормувати дані при проведенні кластерного аналізу?
11. Наведіть приклади функцій відстані для об'єктів, які a) описані тільки числовими ознаками; b) описані тільки нечисловими ознаками - категоріальним, багатозначними. Прокоментуйте властивості наведених функцій.
12. Які є типи ієрархічної кластеризації? Чим вони відрізняються? У чому переваги і недоліки кожного?
13. Що таке дендрограма?
14. Як побудувати дендрограму?
15. Як за допомогою дендрограми вибрати число кластерів?
16. Опишіть метод ієрархічної (алгомератівной) кластеризації. Наведіть приклади функцій відстані між кластерами. Прокоментуйте властивості наведених функцій.
17. Опишіть алгоритм DBSCAN. Прокоментуйте його переваги і недоліки. Які параметри є у даного алгоритму? Як від них залежить розбиття об'єктів на кластери? Як підібрати оптимальні значення цих параметрів?