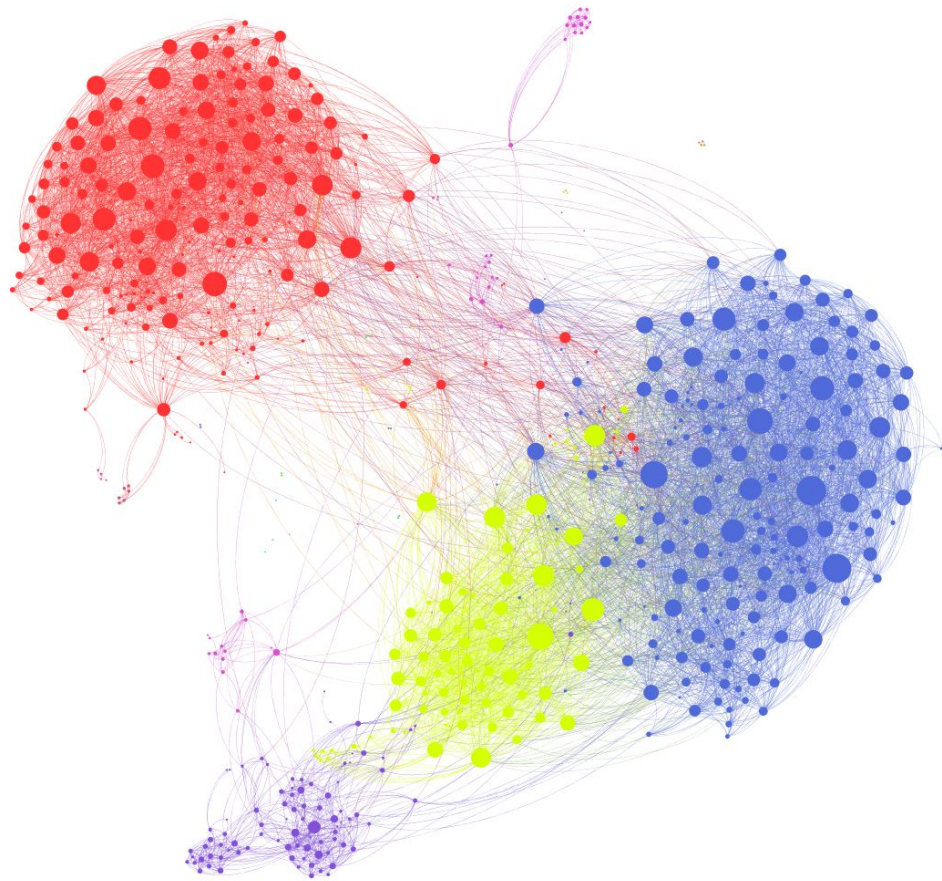


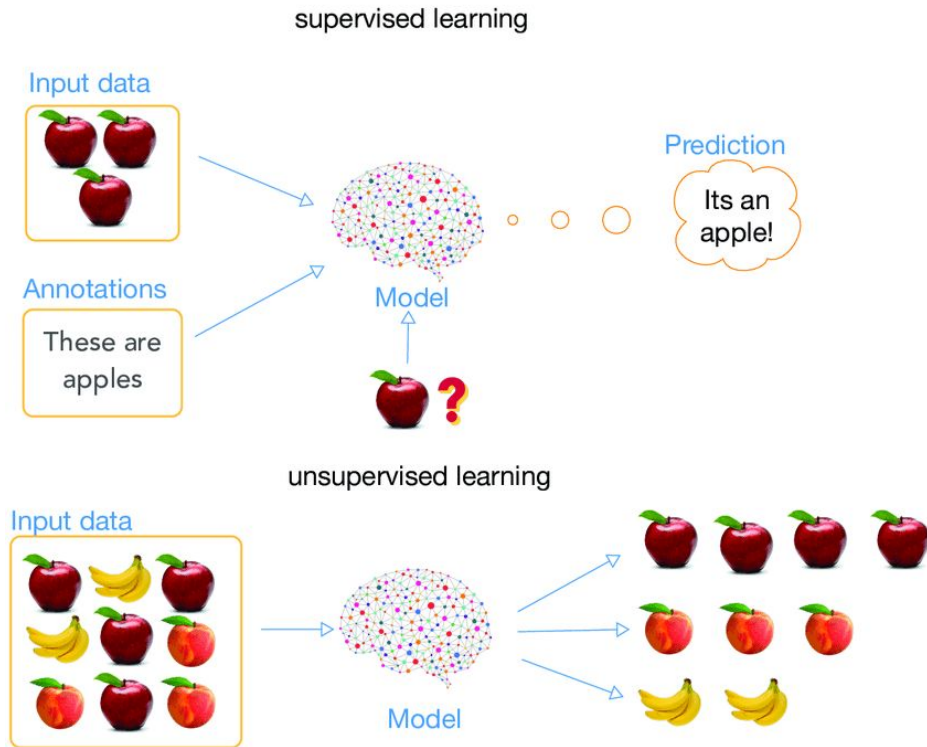
Алгоритми кластеризації



План заняття

- + Модель кластеризації як приклад задачі навчання без вчителя
- + Приклади задач з використанням кластеризації
- + Методи кластеризації, їх принцип роботи, переваги та недоліки
- + K-Means та Elbow method (метод ліктя) для підбору кількості кластерів
- + Hierarchical Clustering
- + Mean-Shift
- + DBSCAN
- + Silhouette метрика для оцінки якості алгоритму кластеризації

Навчання без вчителя



Кластеризація

Кластеризація – це метод **unsupervised** навчання, який дозволяє знаходити **групи схожих об'єктів** у наборі даних.

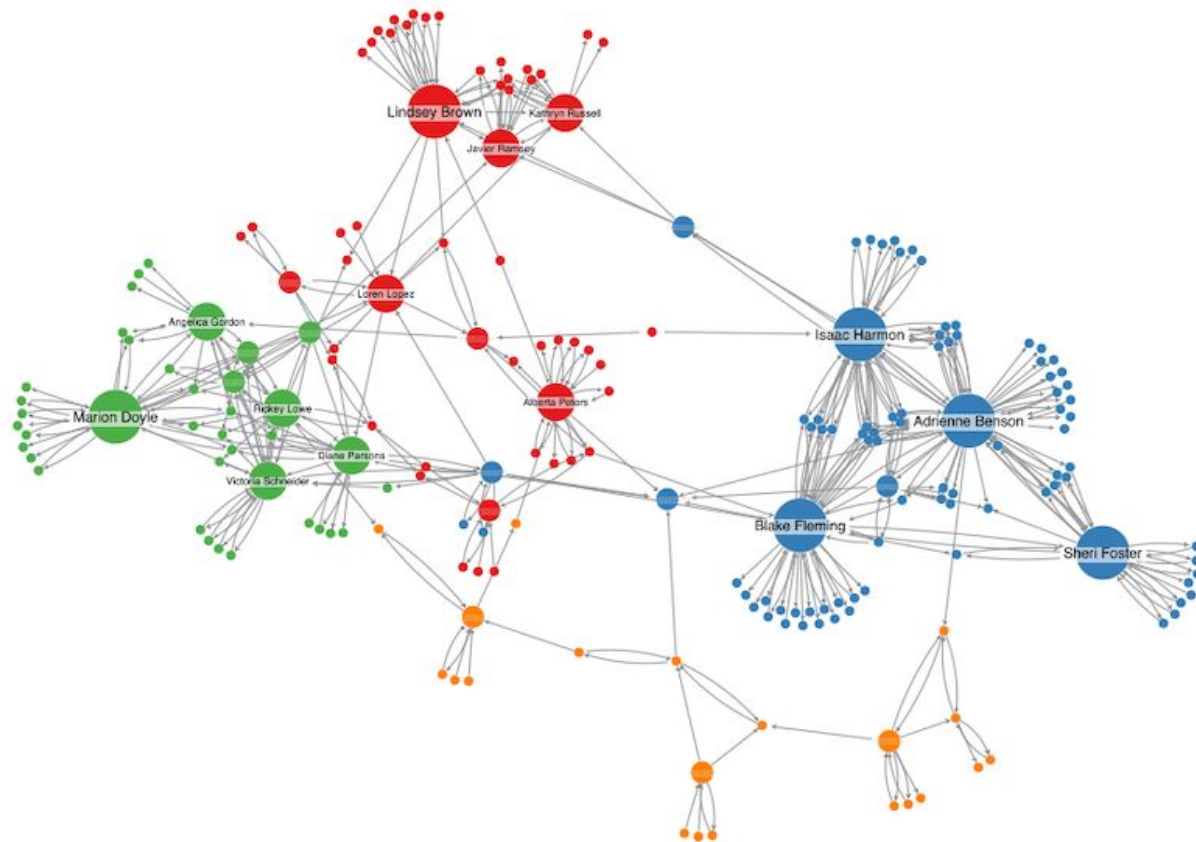
На відміну від supervised навчання, кластеризація не потребує попереднього **маркування** даних.

Цей метод є особливо корисним **для організації** великих наборів даних у значущі кластери, що дозволяє приймати подальші рішення на основі цих груп.

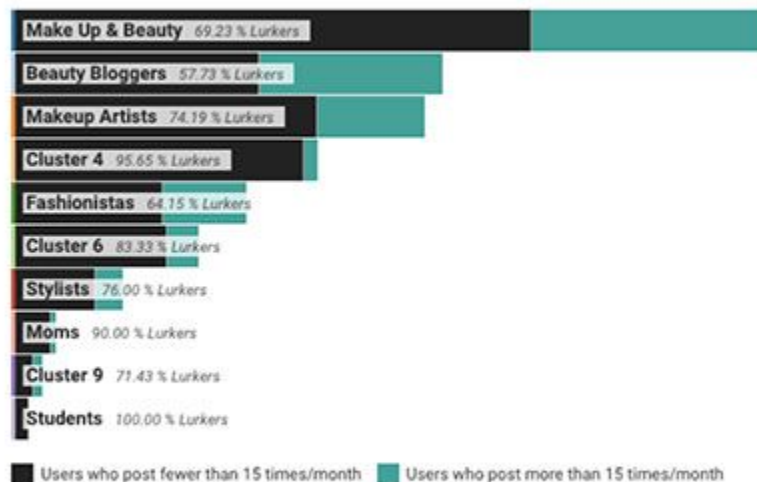
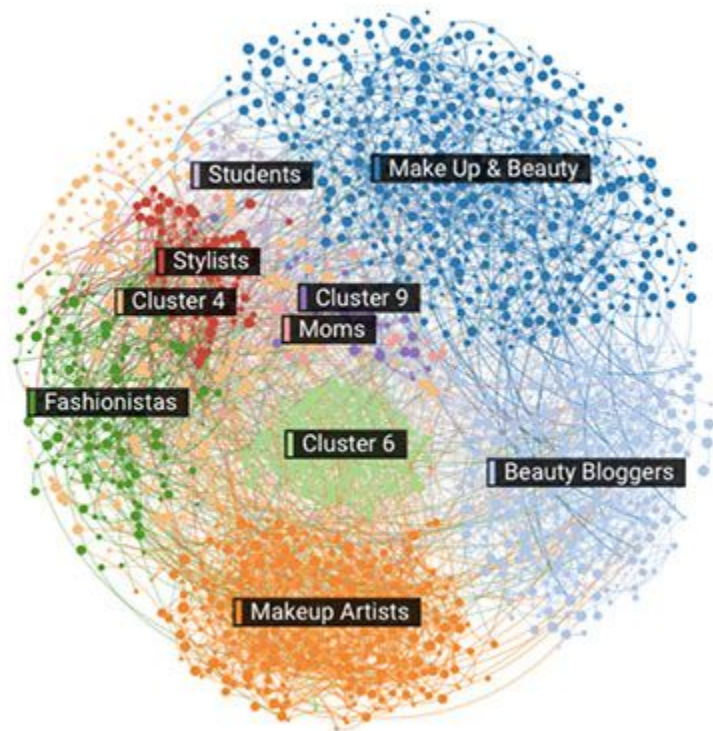
Популярні застосування кластеризації

1. Створення систем рекомендацій
2. Сегментація товарів і користувачів
3. Аналіз соціальних мереж
4. Групування результатів пошуку
5. Сегментація зображень
6. Виявлення аномалій

Кластеризація соц. мереж



Кластеризація соц. мереж



На які питання зазвичай відповідає кластеризація?

- Які типи веб-сторінок існують в Інтернеті?
- Які типи клієнтів є в нашому магазині?
- Які типи людей є в соціальній мережі?
- Які типи електронних листів є в моїй поштовій скриньці?
- Які типи генів містить людський геном?

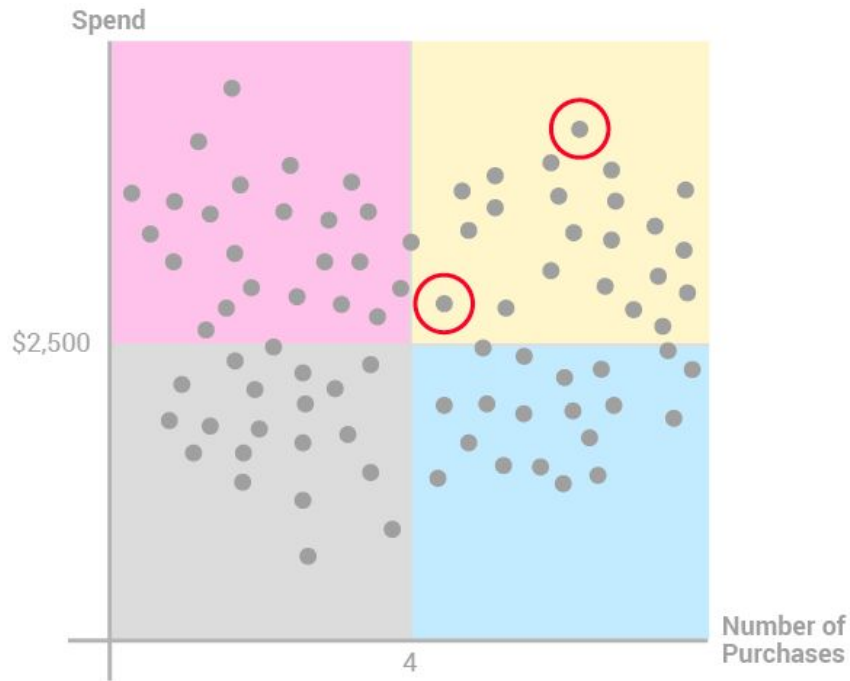
Приклади застосування кластеризації

Сегментація користувачів

Припустимо, у нас є кількість витрачених користувачами грошей і кількість здійснених покупок і ми хочемо знайти групи схожостей клієнтів.

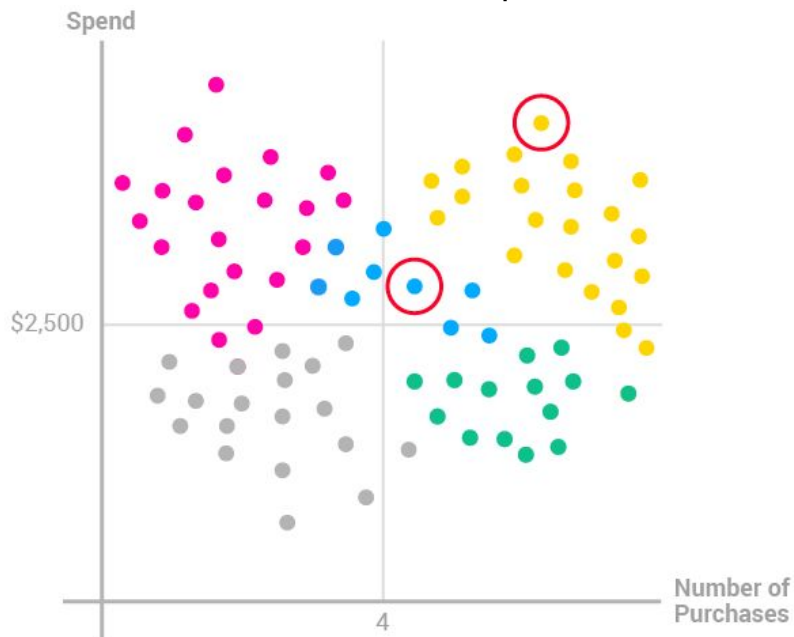
Ми могли б створити просту модель, зазначивши пороги значень, але тоді в одну групу могли б потрапити дуже різні користувачі.

А якщо уявити, що ОХ/ОУ мають експоненційні шкали, то метод ще більше неточний.



Сегментація користувачів

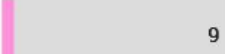














Замість цього можемо застосувати кластеризацію. Цей метод машинного навчання самостійно виявить для нас кластери.



Сегментація користувачів: RFM аналіз

RFM (Recency Frequency Monetary) аналіз — популярний метод кластеризації користувачів.

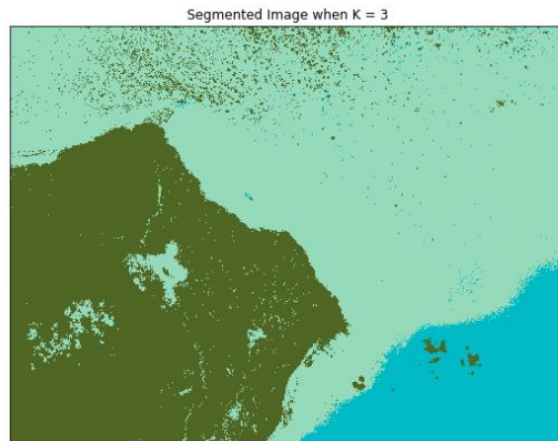
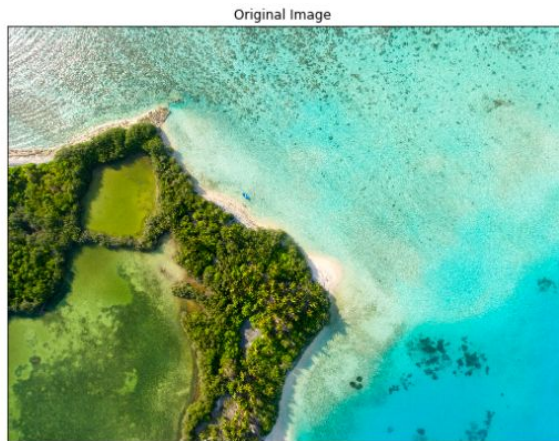
Часто робиться просто аналітично, використовуючи аналітичні пороги чи перцентилі, але можемо робити і з використанням ML-методів.

		R		F		M	
Customers		Days Since Last Purchase		Number of Purchases (Past 12 Months)		Net Revenue (Past 12 Months)	
High Spenders	922						
Mid Spenders	581						
Risk of Churn	807						
Low Spenders	1,361						
	3,671						
		447		3		\$87	

Сегментація зображення

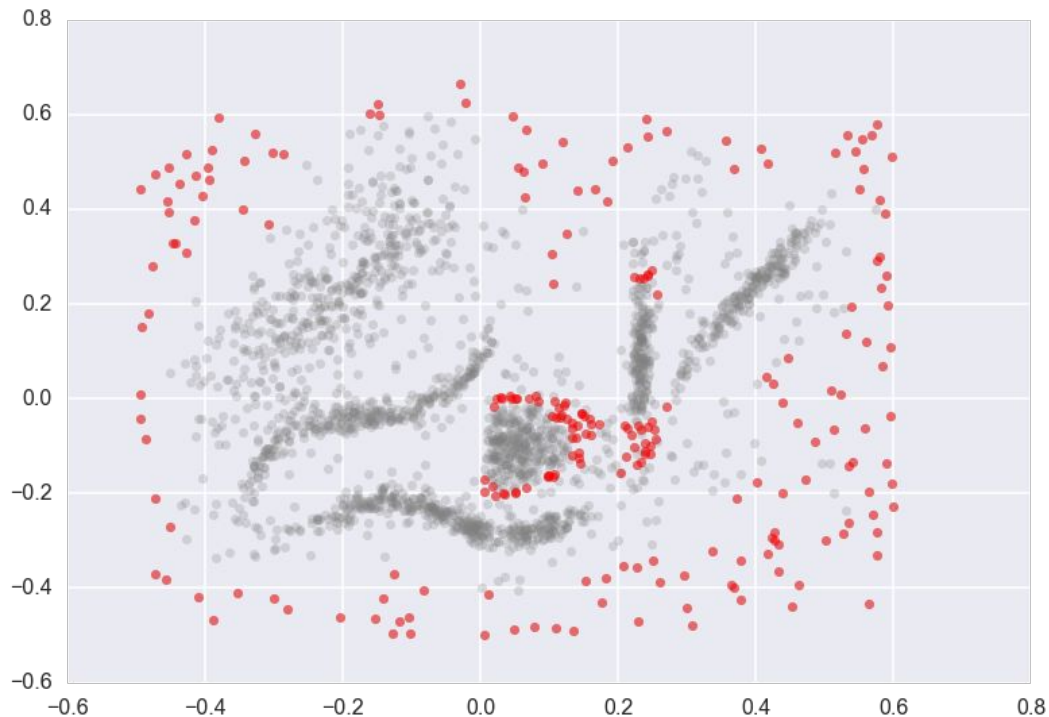
Цілі сегментації зображення:

- + Виявлення різних областей на зображенні та подальша обробка вибраної області
- + Знаходження схожості в деяких частинах зображення



Виявлення аномалій

Виявлення аномалій також часто вирішують з використанням алгоритмів кластеризації. Ми можемо визначити групи об'єктів з “нормальною” поведінкою, і всі об'єкти які не входять в ці групи визначити як аномальні.



Алгоритми кластеризації

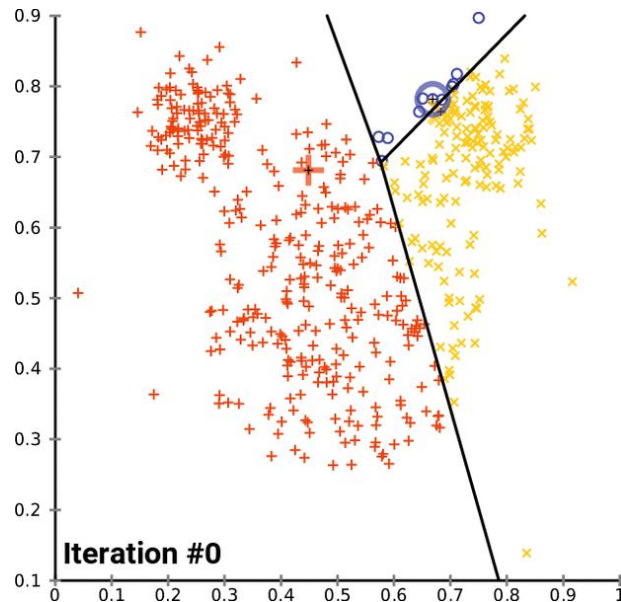
Алгоритми кластеризації

До популярних алгоритмів відносять

- + K-Means
- + Hierarchical Clustering
- + DBSCAN

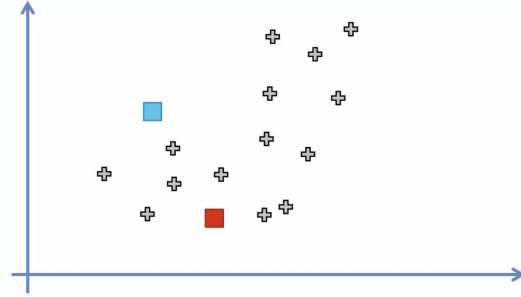
KMeans

Це метод кластеризації, який розподіляє дані на k кількість кластерів. Він працює шляхом повторного визначення центрів кластерів (центроїдів) і перерозподілу даних до найближчого центроїда, доки центроїди не перестануть змінюватися. Метод простий та ефективний для великих наборів даних, але потребує попереднього вибору кількості кластерів k .



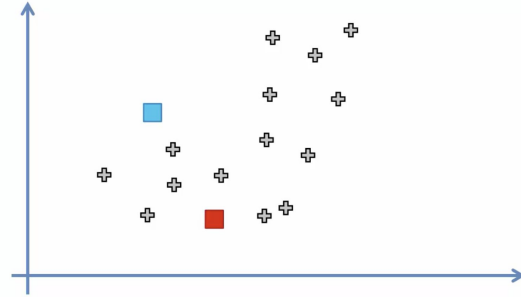
K-Means: принцип роботи

1. Вибираємо K (наприклад, 2) випадкові точки в якості центрів кластерів. Їх називають центроїдами

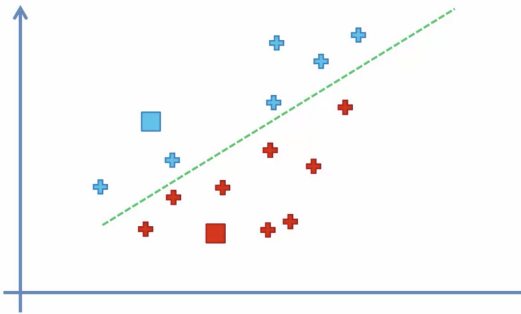


K-Means: принцип роботи

1. Вибираємо K (наприклад, 2) випадкові точки в якості центрів кластерів. Їх називають центроїдами

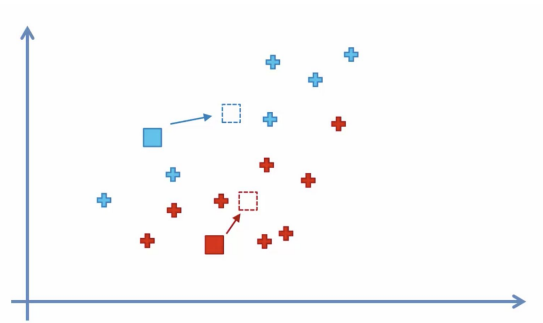


2. Кожну точку даних відносимо до найближчого кластеру, обчисливши відстань до кожного центроїда.



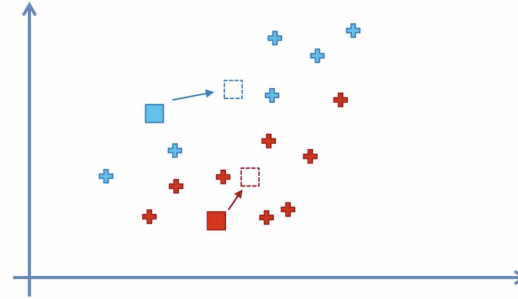
K-Means: принцип роботи

3. Визначаємо новий центр кластера, обчисливши середнє значення точок кожного кластера.

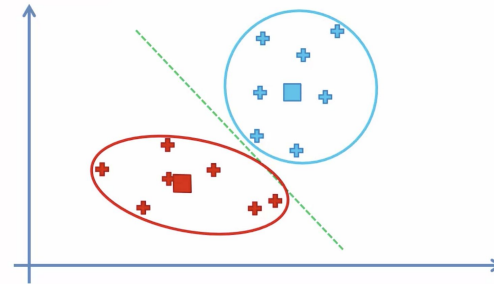


K-Means: принцип роботи

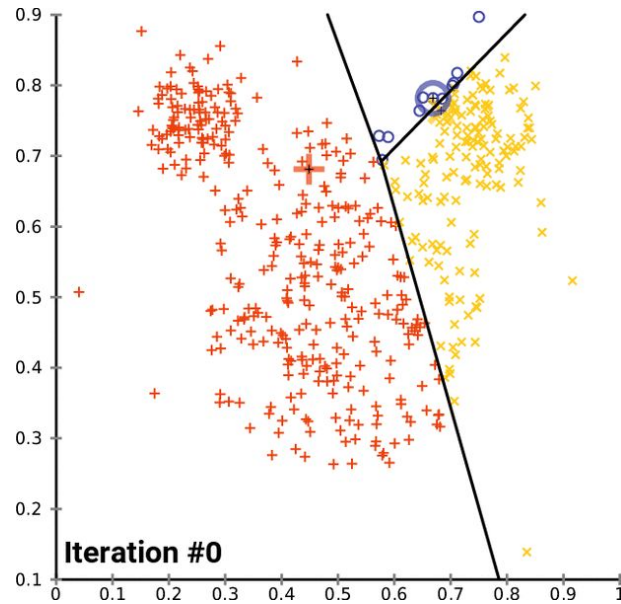
3. Визначаємо новий центр кластера, обчисливши середнє значення точок кожного кластера.



4. Повторюємо кроки 2 і 3 до збіжності, поки не дійдемо до ситуації, де жодне з призначень кластера не зміниться.



K-Means: Демонстрація роботи



K-Means:

Переваги та недоліки

Переваги:

- Простота для розуміння
- Швидкість: у алгоритма лінійна обчислювальна складність $O(n)$

Недоліки:

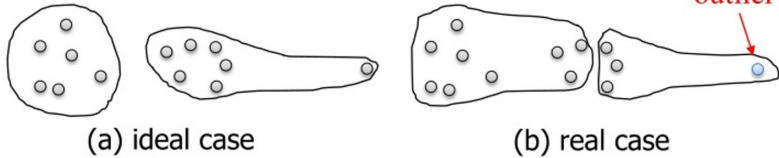
- Необхідність вибирати оптимальну кількість кластерів (іноді ми хочемо, щоб алгоритм кластеризації зробив це за нас)
- Різні номери кластерів після кожного перезапуску кластеризації на тих же даних (оскільки вибирає на початку випадкові точки)
- Працюють тільки для пошуку кластерів сферичної форми або випуклих кластерів. Іншими словами, підходить тільки для компактних і добре розділених кластерів.
- Чутливий до викидів у даних.

K-Medians

K-Means: commonly used clustering method

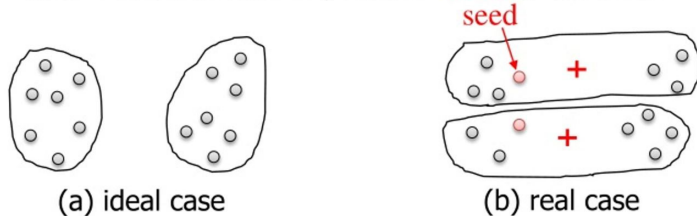
↔ Problem 1: sensitive to **outlier**

- New centroid is the **average** point of the cluster



↔ Problem 2: convergence to local optimum

- New centroid is found only from the **inside** of the cluster



ST

Можемо замість **середнього** використовувати **медіану**, щоб алгоритм був менш чутливим до викидів.

Даний алгоритм значно повільніший на великих наборах, оскільки нам потрібно постійно сортувати дані, щоб знайти медіану.

Як обрати оптимальну кількість кластерів?



Метрика для вимірювання якості кластеризації

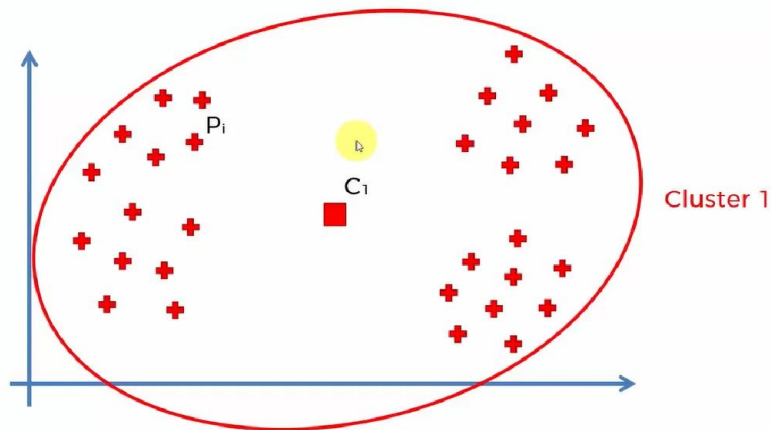
Як поміряти якість кластеризації? Для цього існує кілька метрик, наприклад, WSS.

WSS (within-cluster sum of square) — загальна сума квадратів або внутрішньокластерних варіацій всередині кластера.

В K-Means ми фактично мінімізуємо WSS.

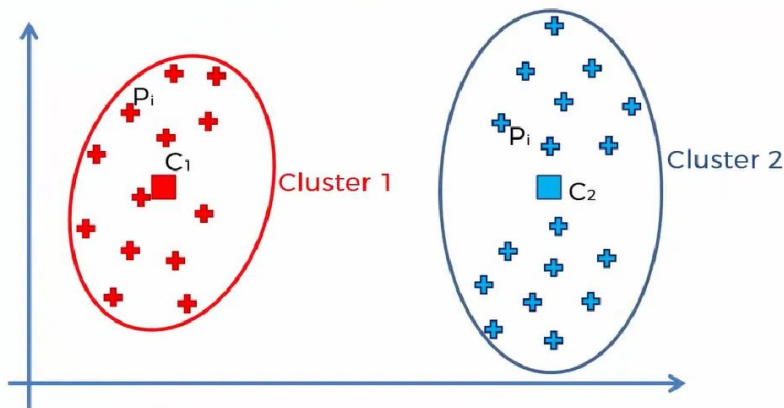
$$WCSS = \sum_{P_i \text{ in Cluster 1}} \text{distance}(P_i, C_1)^2 + \sum_{P_i \text{ in Cluster 2}} \text{distance}(P_i, C_2)^2 + \sum_{P_i \text{ in Cluster 3}} \text{distance}(P_i, C_3)^2$$

Порівняйте WSS для двох випадків



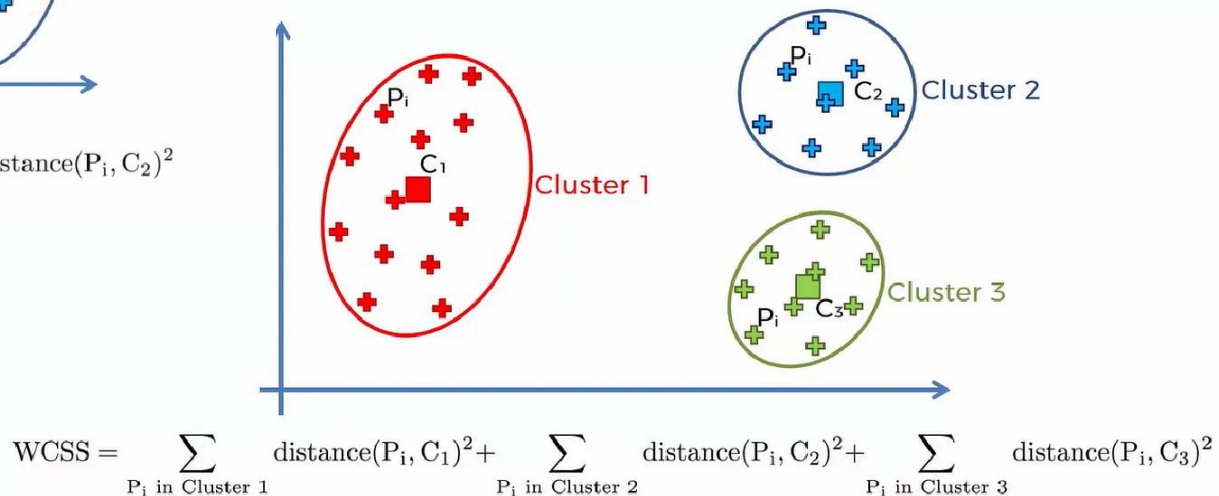
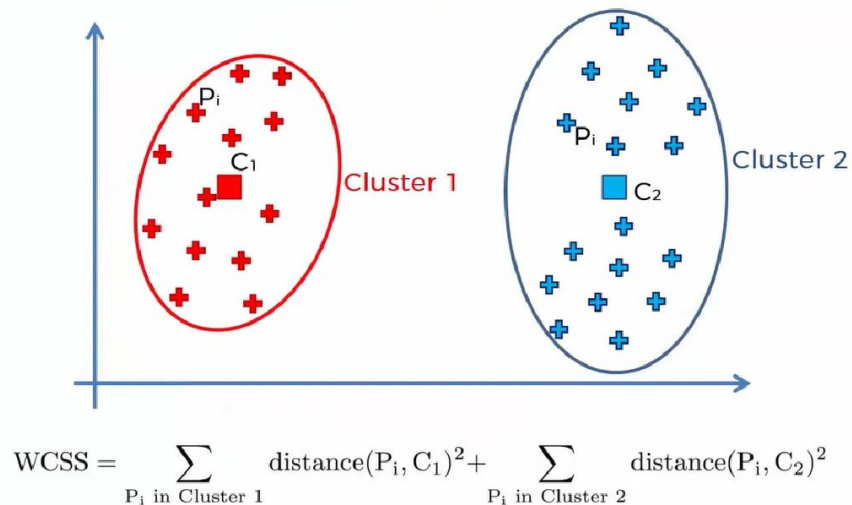
$$WCSS = \sum_{P_i \text{ in Cluster 1}} \text{distance}(P_i, C_1)^2$$

В першому випадку (на малюнку зліва) WSS буде явно більше, ніж у другому.



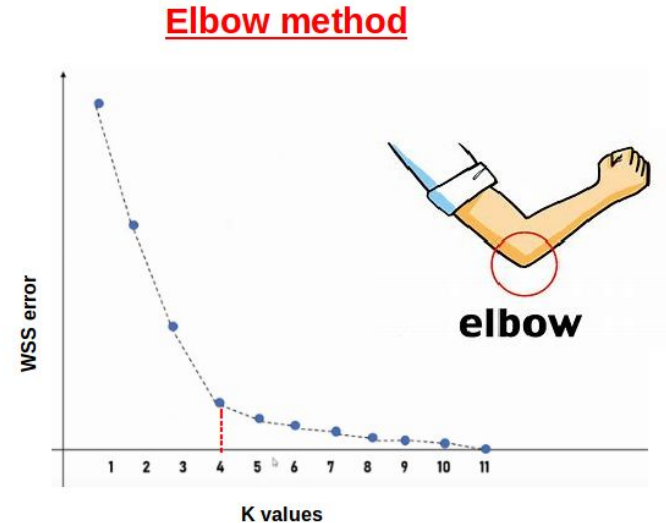
$$WCSS = \sum_{P_i \text{ in Cluster 1}} \text{distance}(P_i, C_1)^2 + \sum_{P_i \text{ in Cluster 2}} \text{distance}(P_i, C_2)^2$$

Порівняйте WSS для різних випадків



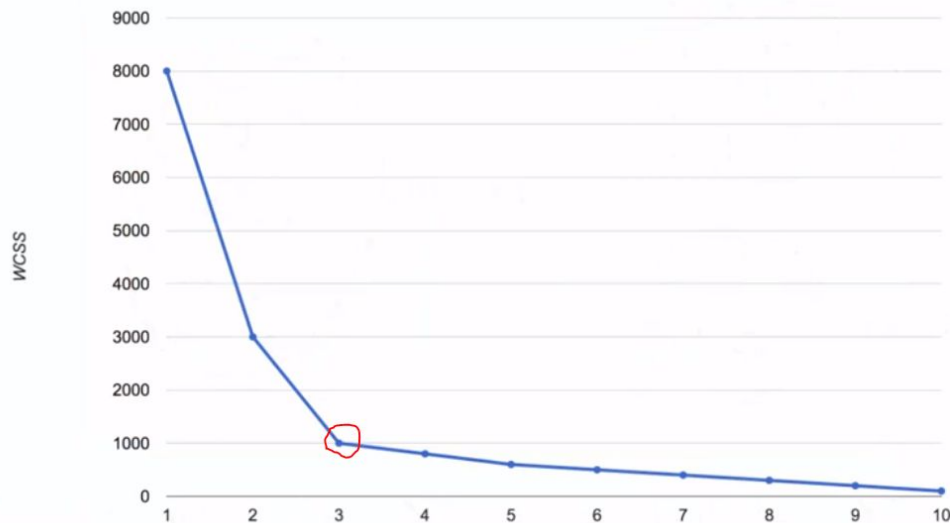
Метод ліктя / “Elbow method”

Аби обрати оптимальну кількість кластерів ми можемо просто порахувати WSS для різної кількості кластерів і це називається - **“Метод ліктя”**. Там, де WSS перестає різко спадати — буде оптимальна кількість кластерів.



Метод ліктя крок за кроком

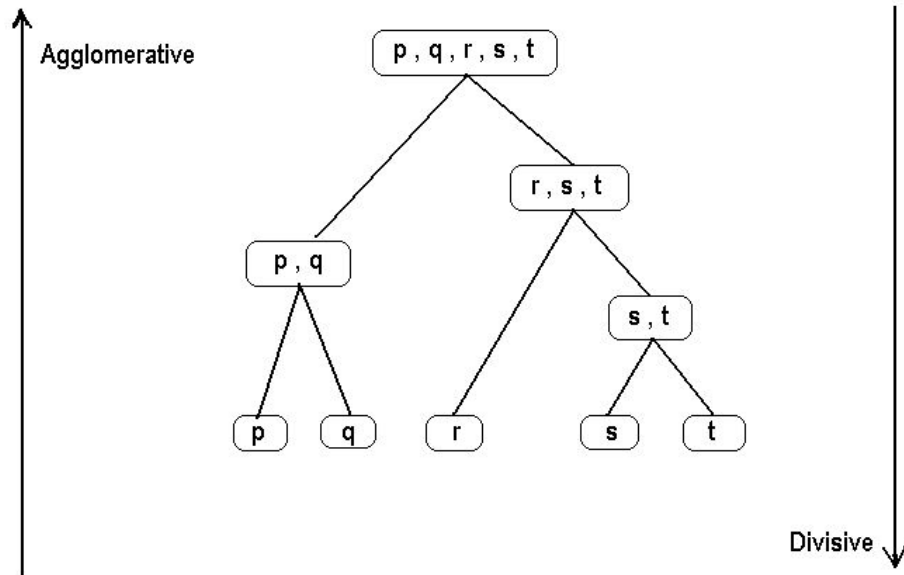
1. Обираємо алгоритм кластеризації (кластеризацію kMeans або kMedians) і навчаємо алгоритм для різних значень кількості кластерів k . Наприклад, змінюючи k від 1 до 10 кластерів.
2. Для кожного k обчислюємо загальну суму квадратів усередині кластера (WSS).
3. Будуємо криву WSS залежно від кількості кластерів k .
4. Розташування вигину ("ліктя") на графіку зазвичай вважається показником оптимальної кількості кластерів.



Hierarchical Clustering

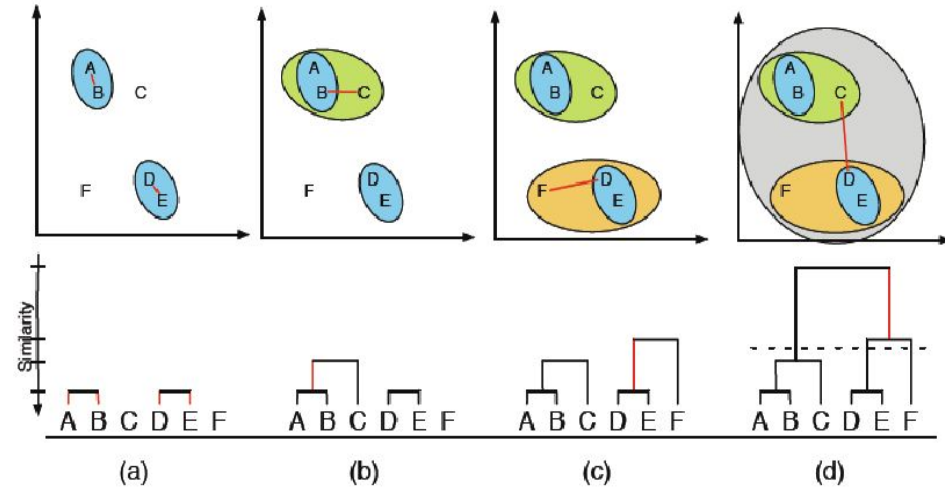
Ієрархічна кластеризація

Це метод, який створює ієрархію кластерів, поступово об'єднуючи або розділяючи існуючі кластери. Існує два підходи: агломеративний (знизу вгору, де кластери поступово об'єднуються) і дивізивний (зверху вниз, де кластери поступово діляться). Перевага цього методу в тому, що не потрібно задавати кількість кластерів наперед.



Агломеративна ієрархічна кластеризація / АНС

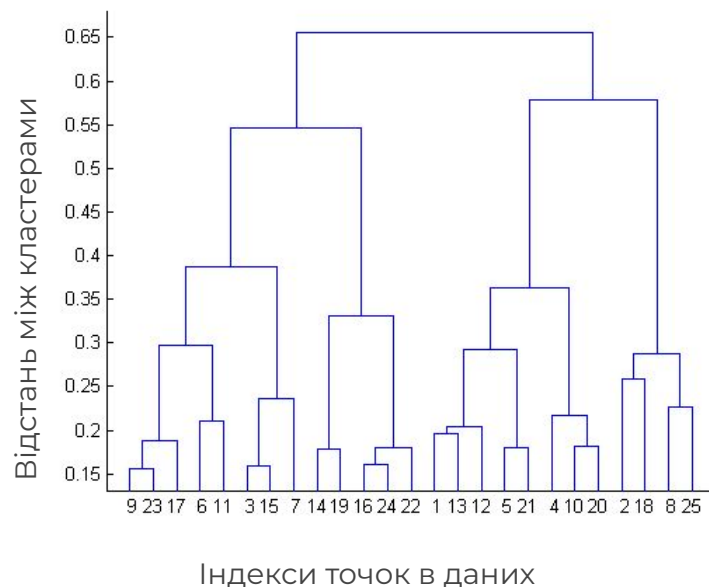
Цей алгоритм починається з усіх точок даних, всі вони належать до кластера самого себе. Потім два найближчих кластери об'єднуються в один кластер. Врешті-решт, цей алгоритм закінчується, коли залишається лише один кластер.



Агломеративна ієрархічна кластеризація: приклад

Приклад: Унизу ми починаємо з 25 точок даних, кожна з яких призначена окремим кластером. Потім два найближчих кластери об'єднуються, поки не залишиться тільки один кластер нагорі.

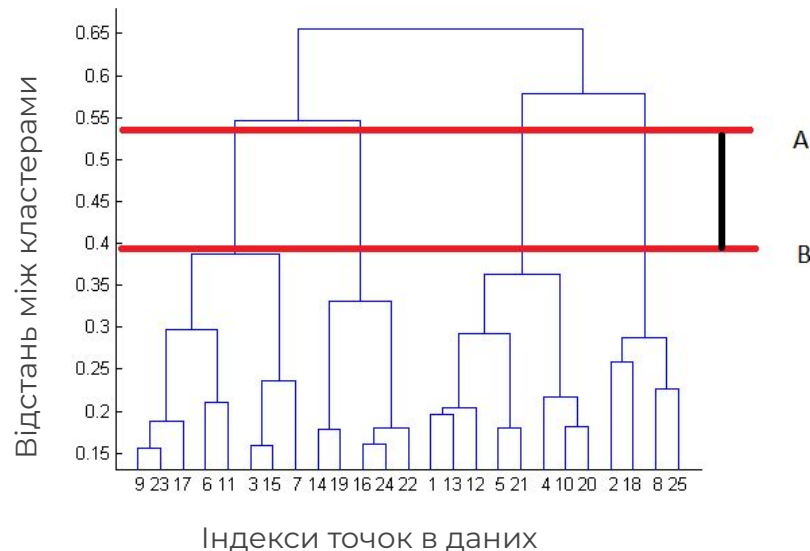
Висота на дендрограмі, на якій два кластери об'єднуються, представляє собою відстань між двома кластерами в просторі даних.



Вибір оптимальної кількості кластерів

Оптимальна кількість кластерів, які можуть найкраще відображати різні групи в даних, обирається виходячи з дендрограми.

Найкращий вибір з числа кластерів — це кількість вертикальних ліній на дендрограмі, які перетинаються горизонтальною лінією, що може відсікати максимальну відстань по вертикалі, не перетинаючи кластер (див. приклад на малюнку).

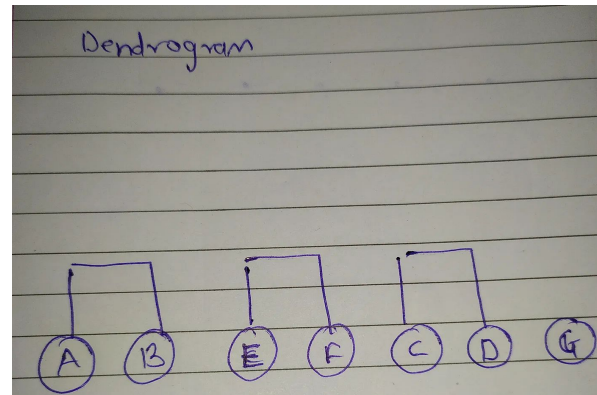


Ключовий параметр в АНС

Ключовим гіперпараметром у агломеративній кластеризації є так зване **зв'язування** (linkage). Цей параметр визначає, як ми будемо рахувати відстань між кластерами, коли там більше, ніж 1 точка.

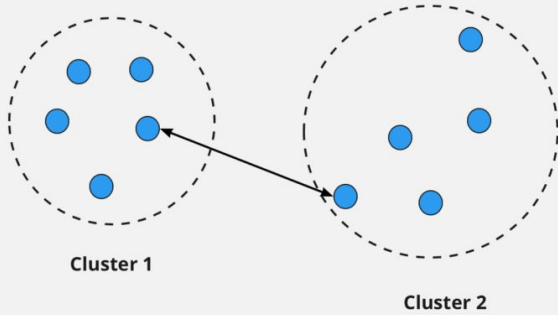
Linkage схоже по сенсу на середнє значення кластера M, яке використовується в методі k-Means кластеризації. Зв'язування може бути представлене кількома способами:

- **Одиничне зв'язування (Single linkage):** Відстань між двома найближчими точками між двома кластерами.
- **Повне зв'язування (Complete linkage):** Відстань між двома найвіддаленішими точками між двома кластерами.
- **Середнє зв'язування (Average linkage):** Це щось середнє між одиничним і повним зв'язуванням. В цьому випадку береться середнє значення між усіма парами точок. Цей метод стійкий до шуму.

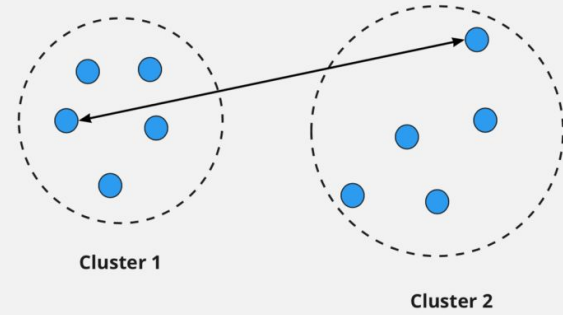


Як обчислити відстань між кластерами?

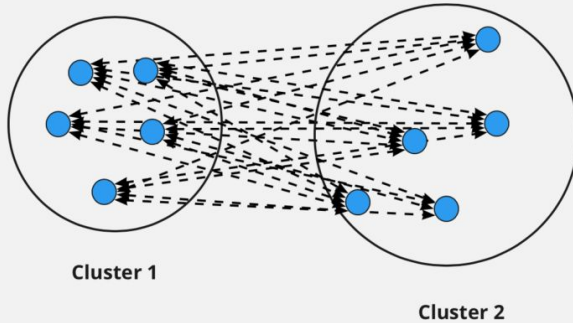
Simple Linkage Method



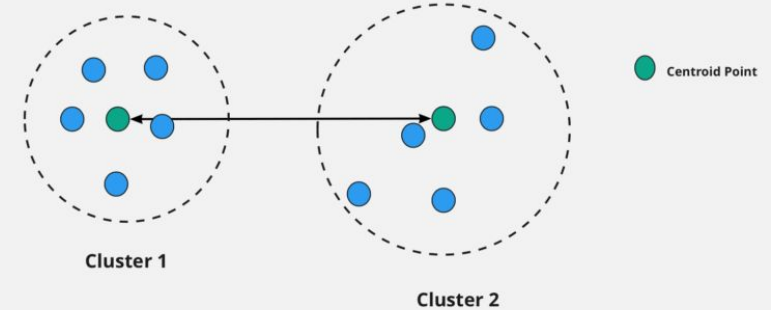
Complete Linkage Method



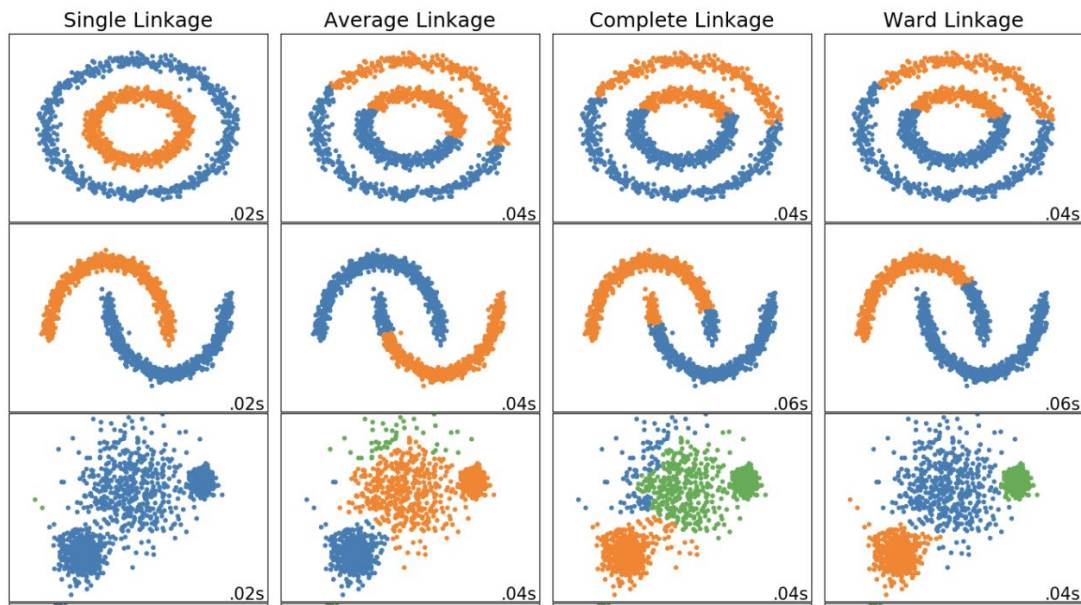
Average Linkage Method



Centroid Linkage Method



Від значення зв'язності залежить результат кластеризації



[Пояснення зв'язності на прикладах](#)

АНС: Переваги та недоліки

Переваги

- Є можливість вибрати кількість кластерів за допомогою алгоритму

Недоліки:

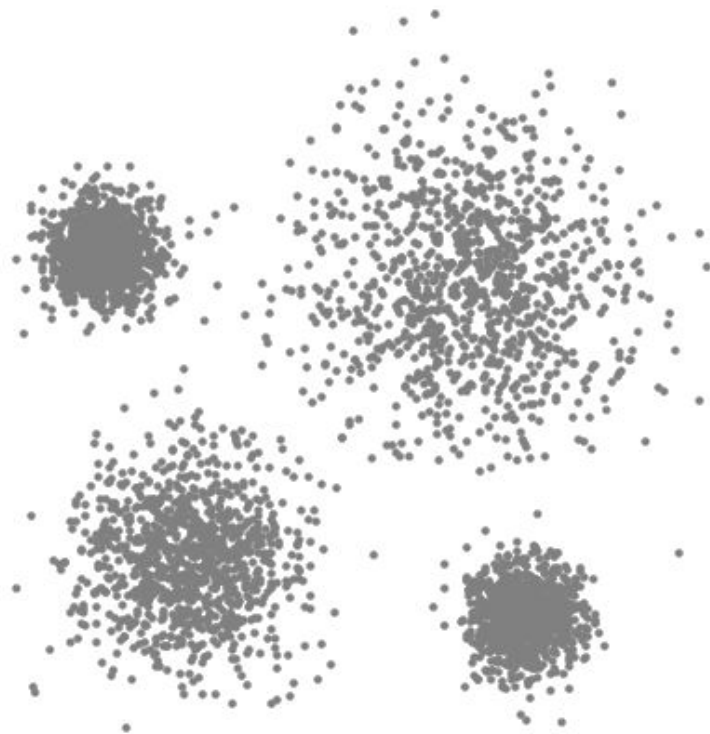
- Також, як K-Means, працює добре на опуклих, добре розділених кластерах
- З великими даними може довго працювати
- Чутливий до шуму і викидів
- Погано працює з кластерами різного розміру, розбиває великі кластери.
- У цьому методі порядок даних впливає на кінцеві результати.

Більше про алгоритм: <https://dataaspirant.com/hierarchical-clustering-algorithm/>

Mean-Shift

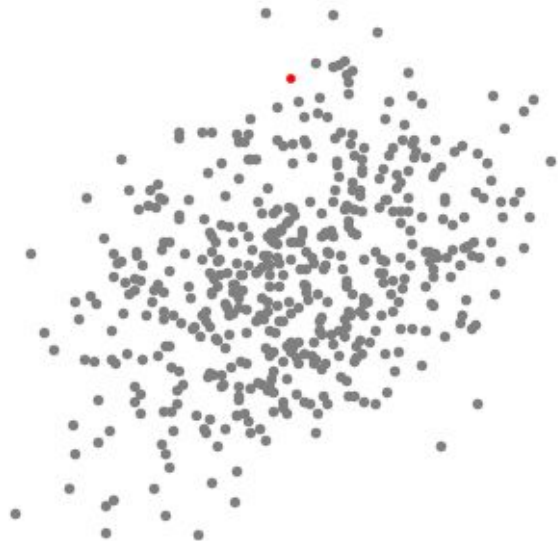
Це метод кластеризації, який **не вимагає знання кількості кластерів заздалегідь**. Він працює шляхом пошуку густих областей у просторі даних та переміщення центрів мас до регіонів з більшою густиною.

Метод добре працює з кластерами нерівномірної форми, але може бути обчислювально важким.

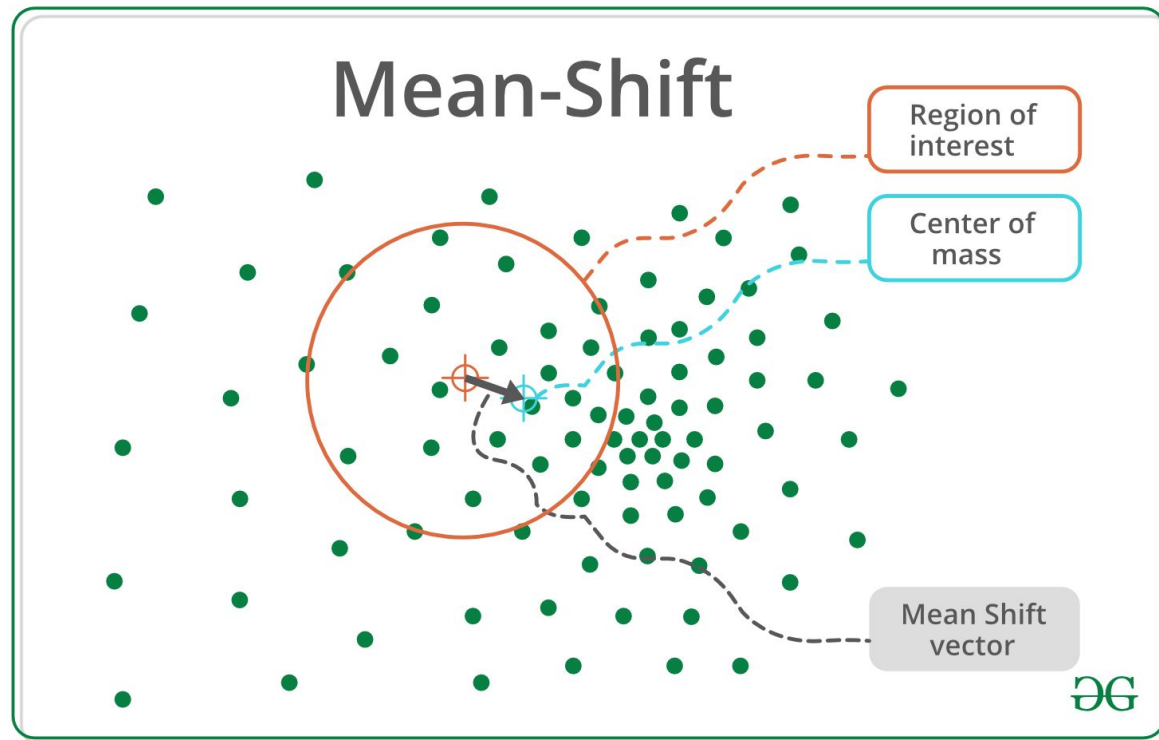


Принцип роботи Mean-Shift 1/2

1. Розглянемо набір точок у двовірному просторі.
2. Ми починаємо з круглого ковзного вікна з центром у точці C (вибраній випадковим чином) і з радіусом r як ядром.
Mean-shift — це алгоритм підйому вгору, який передбачає ітераційне зсування ядра в область з більшою щільністю на кожному кроці до збіжності.
3. На кожній ітерації ковзне вікно (ядро) зсувається в бік областей з більшою щільністю, зсовуючи центральну точку до середнього значення точок всередині вікна (звідси й назва “Mean-Shift”). Щільність у ковзному вікні пропорційна кількості точок всередині нього. Природним чином, при переході до середнього значення точок у вікні воно буде поступово переміщуватися до області з більшою щільністю точок.
4. Ми продовжуємо зсувати ковзне вікно згідно середнього значення, поки не залишаться напрямки, в яких зсування могло б вмістити більше точок всередині ядра.



Візуалізація ковзаючого вікна

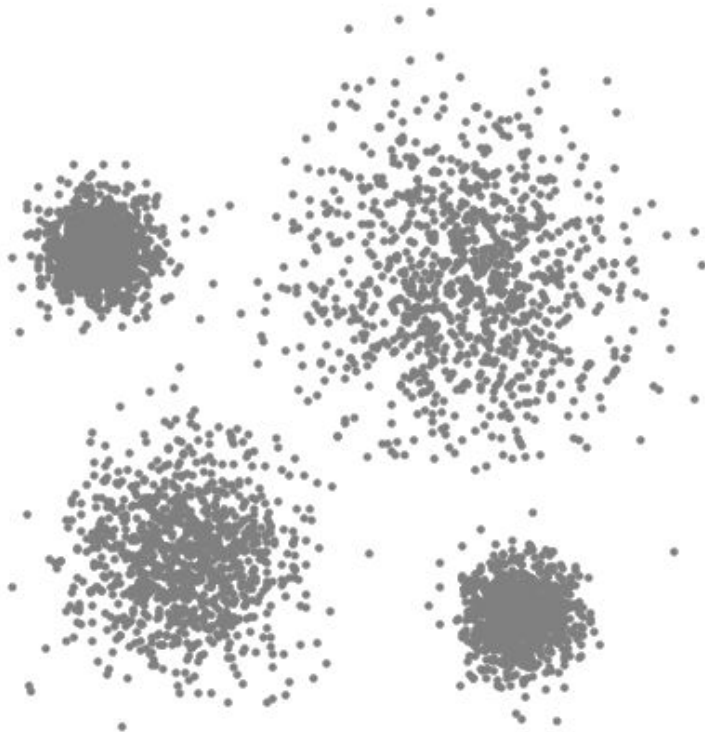


Принцип роботи Mean-Shift 2/2

Процес кроків з 1 по 3 виконується з безліччю ковзаючих вікон, поки всі точки не опиняться всередині одного вікна.

Коли кілька ковзаючих вікон перекриваються, вікно, що містить найбільшу кількість точок, зберігається.

Потім точки даних групуються відповідно до ковзаючого вікна, в якому вони знаходяться. Кожна точка потім належить кластеру найближчого центроїда.



Mean-Shift:

Переваги та недоліки

Переваги:

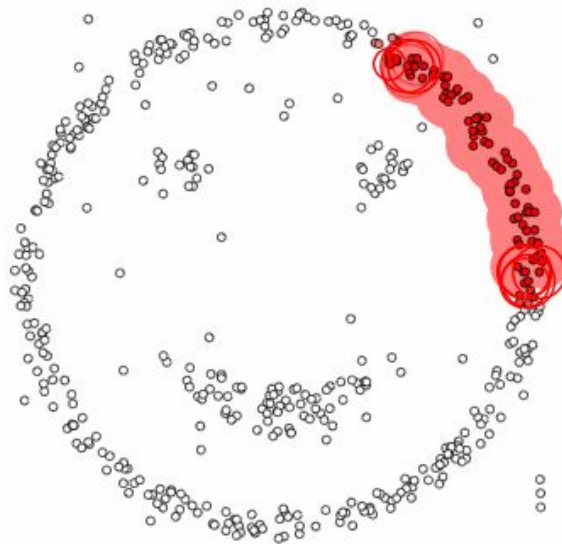
- немає необхідності задавати кількість кластерів, алгоритм робить це за нас
- всього один параметр — радіус
- стійкий до викидів

Недоліки:

- необхідно підбирати параметр
- обчислювальна складність $O(n^2)$ — на великому наборі даних вимагає значно більше часу, ніж K-Means

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

Це метод кластеризації на основі щільності, який групує точки, що знаходяться близько одна до одної, у кластери і визначає окремі точки як шум (що не належить жодному кластеру). DBSCAN не потребує попереднього вибору кількості кластерів і добре працює з кластерами різної форми та наявністю шуму в даних.



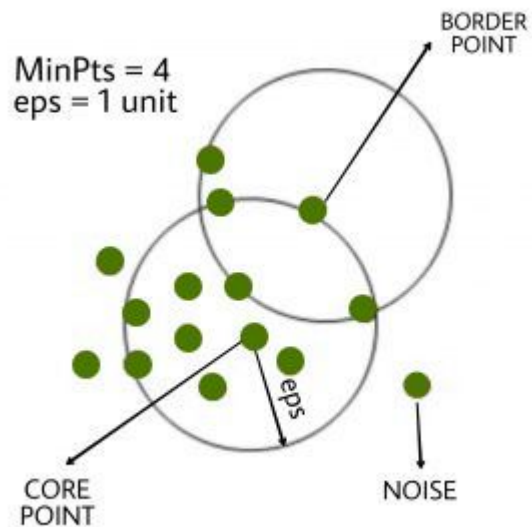
epsilon = 1.00
minPoints = 4

Restart

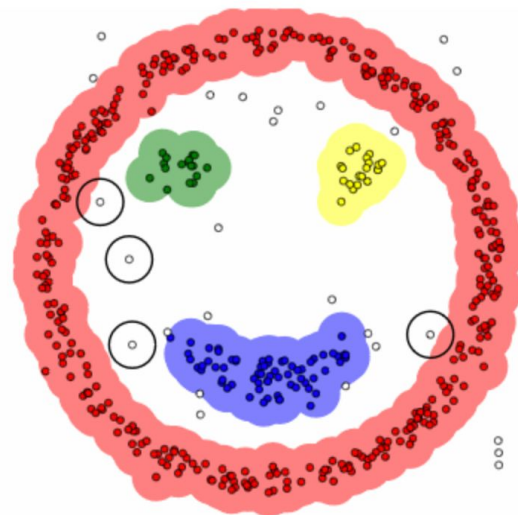


Pause

Принцип роботи DBSCAN



epsilon = 1.00
minPoints = 4



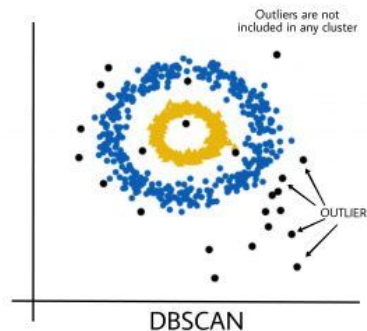
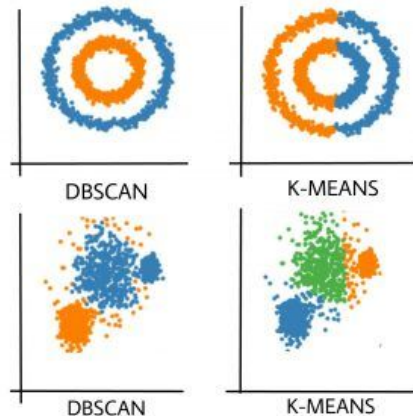
DBSCAN: Переваги та недоліки

Переваги:

- немає необхідності задавати кількість кластерів, алгоритм це робить за нас
- викиди визначає як шум (окрема категорія точок), а Mean-Shift поміщає викиди в кластер, навіть якщо точка дуже відрізняється
- виявляє кластери будь-якої форми та розміру

Недоліки:

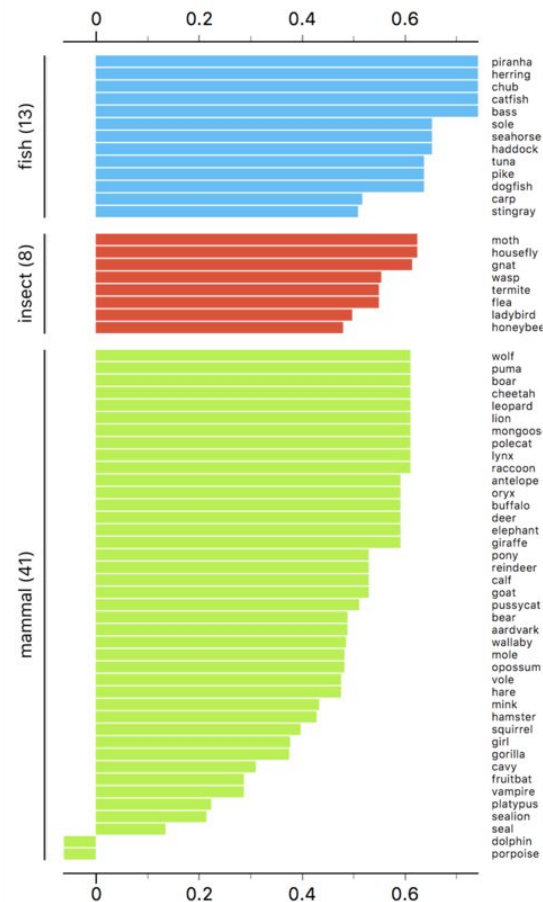
- погано працює, коли кластери різної щільності
- на даних високої розмірності складно підібрати хороший параметр радіусу вікна, щоб він задовольняв різну щільність в кластерах



Оцінка якості кластеризації

Метод середнього силуету (Average Silhouette Method)

Аналіз силуету вимірює, наскільки добре спостереження згруповані, і оцінює середню відстань між кластерами. Графік силуету відображає міру того, наскільки близько кожна точка в одному кластері знаходиться до точок у сусідніх кластерах.



Середнє значення силуету

Значення метрики коливається від -1 до 1 .

1 : означає, що групи добре відокремлені одна від одної і чітко розрізняються.

0 : означає, що кластери байдужі, або ми можемо сказати, що відстань між кластерами не має значення.

-1 : означає, що кластери призначені неправильно.

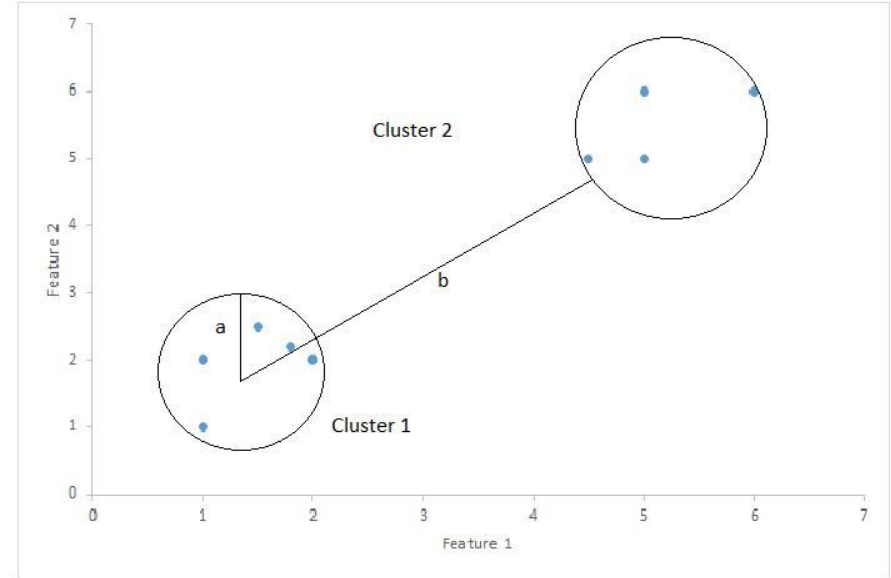
Average Silhouette Method: формула

Оцінка силуету = $(b - a) / \max(a, b)$

де

a — середнє відстань всередині кластера, тобто середнє відстань між кожною точкою в кластері.

b — середнє відстань між кластерами, тобто середнє відстань між усіма кластерами.



Average Silhouette Method: приклад

Припустимо, ми розділили набір точок на два кластери {1, 3, 5} і {8, 9, 10}. Розглянемо їх метрики силуету.

Cluster 1	Distance within	a	Distance outside	b	Si	Cluster
1	2,4	3	7,8,9	8	0.63	0.51
3	2,2	2	5,6,7	6	0.67	
5	2,4	3	3,4,5	4	0.25	
8	1,2	1.5	7,5,3	5	0.70	0.78
9	1,1	1	8,6,4	6	0.83	
10	1,2	1.5	9,8,7	8	0.81	

Висновки:

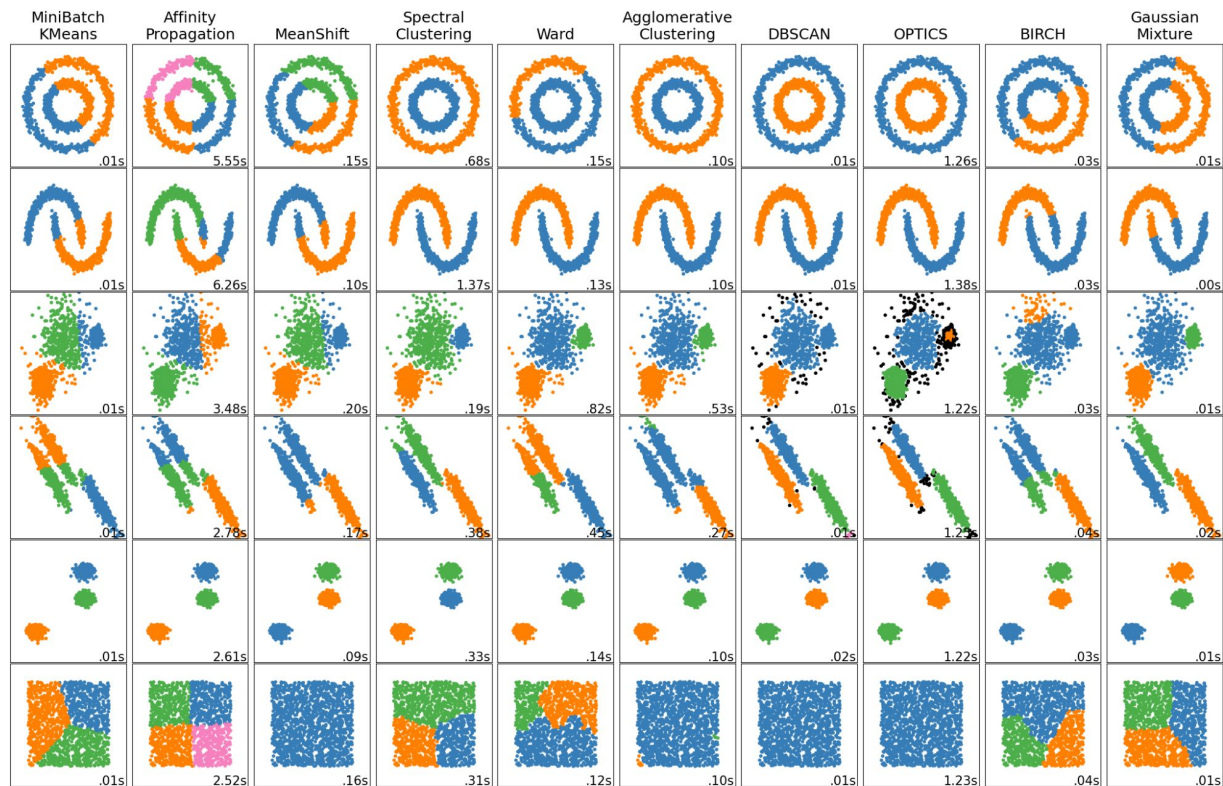
Другий кластер має кращий коефіцієнт силуету, оскільки він більш компактний
Центральні точки мають кращий коефіцієнт силуету в порівнянні з іншими
Граничні точки мають найнижчий коефіцієнт силуету

Інші метрики для виміру якості кластеризації

Метрик для оцінювання якості кластеризації існує більше. Можна ознайомитись з тими, які представлені в sklearn тут

<https://scikit-learn.org/stable/modules/clustering.html#clustering-performance-evaluation>

Порівняння роботи різних алгоритмів кластеризації



<https://scikit-learn.org/stable/modules/clustering.html>