Білет №21 16.04.2022

Студент групи ФЕП-31 Сворень Я. В.

Львівський національний університет імені Івана Франка

Факультет електроніки та комп'ютерних технологій

Кафедра системного проектування

Викладач з курсу "Засобу машинного навчання": Рибак А. В.

Питання:

Опишіть процес кластерного аналізу даних з використанням бібліотеки Scikit-learn.

Відповідь на питання:

Кластерний аналіз – це процес поділу певної множини елементів на частки, або ж, кластери, які містять найбільш схожі за властивостями елементи множини.

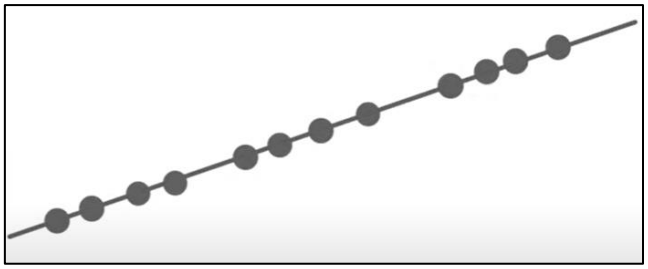
Бібліотека Scikit-learn для мови програмування Python надає можливість викликати аж цілих 10 методів кластеризації. Для прикладу буде докладно розглянуто метод кластерного аналізу даних KMeans().

KMeans() – це алгоритм кластеризації, головне завдання якого є згрупувати схожі елементи в кластери, окремі від решти елементів.

KMeans() належить до ексклюзивного типу кластеризації (Exclusive Clustering), значить кластери не мають ієрархічних зв'язків(Hierarchical Clustering) і елементи належать ексклюзивно одному кластеру, не перетинаючись з іншими(Overlapping Clustering).

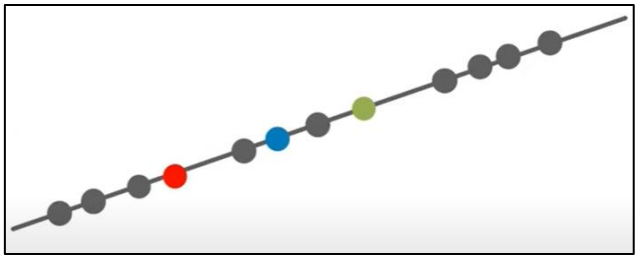
Буква K у назві цього алгоритму вказує на змінну кількість кластерів, який буде генерувати алгоритм, тобто що цей метод спроможний на розсуд користувача знаходити саме потрібну йому кількість. Саме за це відповідає параметр функції KMeans() під назвою n_clusters, який за замовчуванням рівний 8.

Розглянемо детально алгоритм KMeans(). Нехай у нас є пряма, на якій задано 8 точок. Наше завдання – провести кластерний аналіз цих точок та поділити їх на 3 кластери.

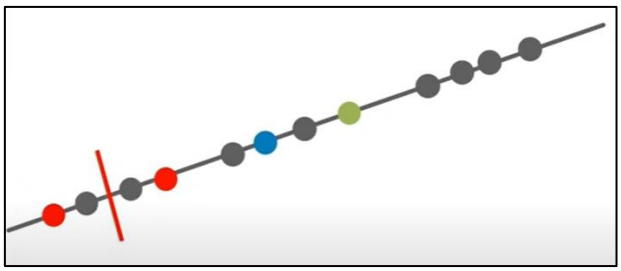


Крок 1: Визначитись над кількістю кластерів, які ми плануємо згенерувати. Згідно з умовою, K=3.

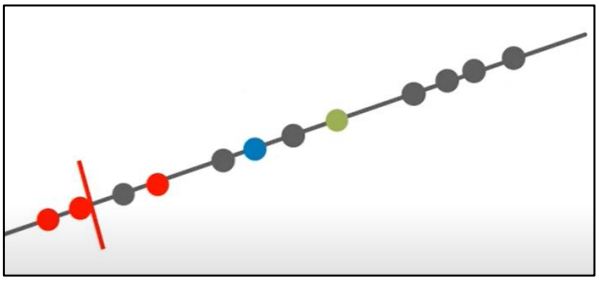
Крок 2: Випадковим чином обрати 3 точки з малюнку. З цього моменту вважаємо ці 3 точки трьома центрами наших майбутніх кластерів.



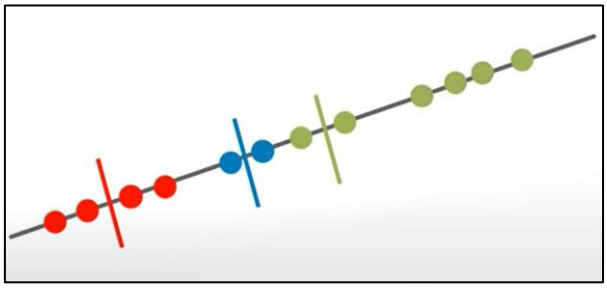
Крок 3: Порівнюємо відстань від першої незайнятої точки до кожного з центрів кластерів, після чого приєднуємо її до кластеру, центр якого є найближчим цій точці. Після цього обчислюємо нову середину кластеру. З цього моменту центром кластеру ми вважаємо точку на прямій, яка знаходиться посередині елементів кластеру.



Повторюємо алгоритм третього кроку до того моменту, поки кожна з незайнятих точок не буде присвоєна одному з кластерів. Таким чином наступною була друга точка на графіку, яка теж приєднується до червоного кластеру і центр кластеру знову змінюється.



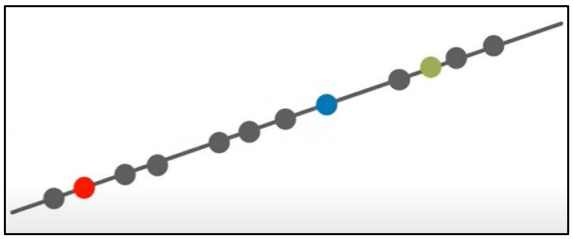
В результаті ми отримуємо таку картину. Неоднакова кількість елементів вказує на те, що такий вибір 3 початкових точок-центрів не дає нам бажаного результату.



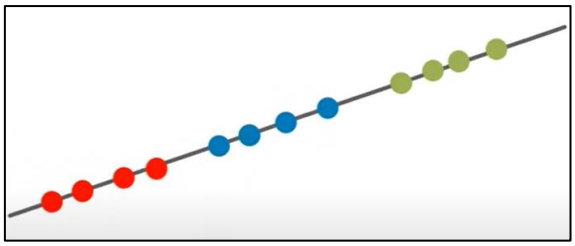
Крок 4: У циклі повторюємо виконання другого та третього кроку до того моменту, поки не отримаємо необхідну кількість кластерів з якомога ближчими до однакових кількостями елементів. У кожній наступній ітерації під час другого кроку обираємо таку пару трьох точок, яка не була обрана у попередніх ітераціях.

Кінець алгоритму

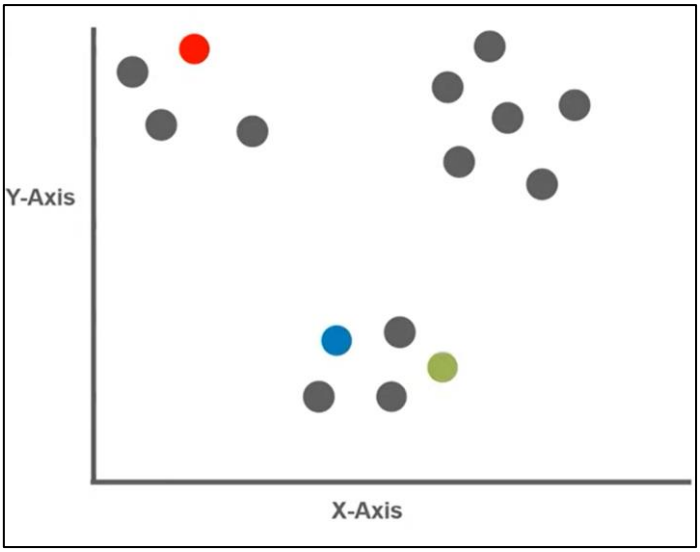
Найкращий результат ми можемо отримати, обравши ці 3 випадкові точки.



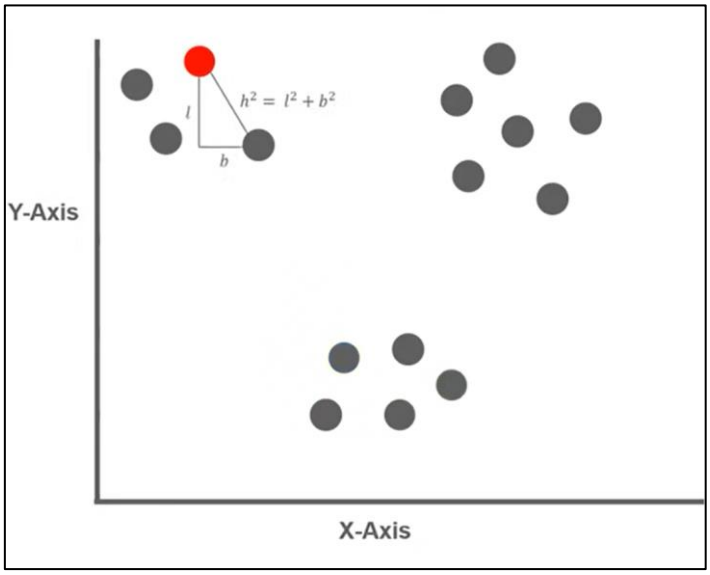
При виконанні алгоритму KMeans() з ними ми отримаємо бажаний результат з трьох кластерів з рівною кількістю елементів кожен.



Розглянемо ще схоже завдання у двовимірному просторі. Перший та другий крок алгоритму не міняється, визначаємось з кількістю кластерів та обираємо 3 випадкові точки.



Єдиною різницею від попередньо описаного алгоритму є зміна формули підрахунку евклідової відстані між точками. У двовимірному просторі для цього згодиться формула Піфагора.



Джерела інформації:

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

https://www.youtube.com/watch?v=1XqG0kaJVHY