**Тема №1 Ознайомлення з методами кластерного аналізу на прикладі методу K-means його програмна реалізація та тестова перевірка.**

Кластеризація – це процедура (алгоритм), що реалізує поділ деякої множини об’єктів на окремі групи (кластери), що не перетинаються між собою. Об’єкти, що входять до одного кластеру мають спільні властивості, які відрізняють їх від об’єктів іншого кластеру. Саме поняття кластера є слабоформалізованим оскільки спирається на поняття “спільні властивості”, що потребує додаткової деталізації. Ця деталізація здебільшого визначається змістом прикладної задачі і реалізується особливостями самого алгоритму. Саме тому існує декілька десятків різних алгоритмів кластеризації та їх модифікацій.

Ми розглянемо один з найбільш поширених алгоритмів, який з’явився ще на початку 60-років **K-means** і набув значної популярності. Ідея алгоритму полягає в припущенні, що кожний кластер має деякий центроїд (типовий елемент) і елемент вихідної множини відносять до кластеру з мінімальною відстанню до центроїда. Тобто для алгоритму **K-means** спільною властивість елементів кластеру є властивість бути найближчими до свого центроїду.

Це визначає форму кластерів, який здатен ефективно знаходити метод **K-means**, а саме метод знаходить кластери сферичної форми, або форми близької до сферичної. Метод потребує задання кількості кластерів. Сама процедура кластеризації є ітераційною, тому вимагає задання критерія зупинки, наприклад, кількості ітерацій MaxIter, або Eps - відносної зміни положення центроїдів на двох сусідніх ітераціях.

**Формалізуємо опис алгоритму**

Нехай  множина об’єктів що підлягає кластеризації.

 мітки майбутніх кластерів. Як правило мітками виступають натуральні числа. -кількість кластерів. Задача кластеризації полягає в побудові відображення , тобто кожному елементу  ставиться у відповідність мітка  за правилом .

Метод базується на знаходженні для кожного кластера його центроїда  та подальшого віднесення кожного з об’єктів  до кластера за принципом мінімальної відстані до відповідного центроїда :

, де  - відстань між двома векторами в просторі . Процес обчислення положення центроїдів є ітераційним і тому передбачає задання деяких початкових наближень положень центроїдів. Найпростіший варіант вибору початкових наближень центроїдів, вибрати  довільних елементів множини . Але цей метод дуже часто призводить до повільної збіжності , або збіжності до неприйнятного результату.

***Структура алгоритму методу K-means***

1. *Вибір початкових центроїдів* 
   1. З множини випадковим чином обираємо 5-10% векторів .



* 1. Формуємо центроїди двох перших кластерів , обираючи їх серед точок множини . Обираємо точки відстань між якими є максимальною. .
  2. Формуємо центроїди усіх інших кластерів починаючи з третього. Нехай ми вже знайшли центроїди , тоді центроїд  знаходимо серед точок множини  за правилом:  
     . Тобто, для кожного елементу  обчислюється відстань до кожного з відомих центроїдів і серед них обирається мінімальне значення. Ці мінімальні значення утворюють послідовність, той елемент множини  для якого послідовність набуває максимального значення і обирається в якості наступного центроїда. Таким чином будуємо множину точок з максимальною середньою відстанню.

1. Розподіляємо точки множини  по кластерам за принципом мінімальної відстані до центроїда. . Утворюємо кластери .
2. Обчислюємо нові значення центроїдів для кожного кластера. , де  - кількість елементів кластера .
3. Перевіряємо виконання критерія зупинки ітераційного процесу., де  - матриці рядками яких є вектори центроїдів .

Якщо умова не виконана, кроки 2-4 повторюються до виконання умови. Додатковою умовою закінчення ітераційного процесу є обмеження кількості ітерацій.

**Структура програми**

1. Модуль формування вхідної множини об’єктів *X=Matrix(1..N,1..M)*, де *N*- кількість об’єктів, *M*- кількість ознак.
2. Модуль графічного відображення вхідної множини.
3. Модуль мінімаксної нормалізації вхідної множини об’єктів. Приводить зміну усіх значень вхідної множини до відрізку [0,1].
4. Модуль формування випадкової вибірки *X0* з матриці *X.*
5. Процедура вибору початкового наближення двох перших центроїдів.
6. Процедура вибору початкових наближень усіх інших центроїдів.
7. Процедура визначення кластерів.
8. Процедура визначення нових значень центроїдів.
9. Процедура перевірки умови збіжності ітераційного процесу.
10. Модуль послідовного виклика процедур п.п. 7-9.
11. Модуль відображення результатів кластеризації.

**Список пакетів Maple необхідних для реалізації.**

*LinearAlgebra, RandomTools, ListTools, Statistics*