Виконав:

студент групи КМ-32

Друзенко Є.Р.

**Звіт по виконанню завдання**

**Мета завдання:**  
Розробити програмний засіб для знаходження центрів ваги зв'язаних областей множини точок, заданих своїми координатами, та побудови діаграми Вороного для цих центрів ваги.

**Опис виконання:**

1. **Зчитування даних:** Спочатку була реалізована функція для зчитування даних з текстового файлу "DS5.txt". Кожен рядок файлу містить пару цілих чисел, які є координатами точок. Для цього використовувалася стандартна функція Python open() для відкриття файлу, та метод split() для розділення рядків на компоненти, які потім конвертуються в числа типу int.
2. **Кластеризація точок:** Для поділу датасету на зв'язані області була застосована кластеризація методом **DBSCAN** (Density-Based Spatial Clustering of Applications with Noise) з бібліотеки sklearn. Цей алгоритм кластеризує точки на основі їх щільності, де:
   * eps визначає максимальну відстань між точками, щоб вони потрапили в один кластер.
   * min\_samples вказує мінімальну кількість точок, необхідних для створення кластера.

Алгоритм DBSCAN добре підходить для знаходження зв'язаних областей, де не потрібно заздалегідь задавати кількість кластерів, а також здатен виявляти шуми (точки, які не належать жодному кластеру).

1. **Обчислення центрів ваги:** Для кожного кластеру було обчислено центр ваги, тобто середнє арифметичне координат усіх точок цього кластера. Це дозволяє знайти точку, яка найбільш "характеризує" кожну з областей. Для цього використовувалась функція np.mean(), яка обчислює середнє значення координат точок у кожному кластері.
2. **Побудова діаграми Вороного:** Після обчислення центрів ваги для кожної області була побудована **діаграма Вороного** для знайдених центрів ваги. Для цього використано функціонал з бібліотеки scipy.spatial:
   * Клас Voronoi для створення об'єкта діаграми Вороного.
   * Функція voronoi\_plot\_2d() для відображення діаграми Вороного на площині.

Діаграма Вороного дозволяє розділити площину на області, кожна з яких відповідає одному з центрів ваги, причому кожна точка в області ближче до свого центру ваги, ніж до інших.

1. **Відображення результату:** Для побудови графіків та відображення результатів використано бібліотеку **matplotlib**:
   * plt.scatter() для відображення точок датасету та центрів ваги.
   * plt.xlim() і plt.ylim() для налаштування меж осей.
   * plt.savefig() для збереження результату в графічному файлі у форматі PNG.
2. **Збереження результату:** Результат зображення (діаграма Вороного з центрами ваги) було збережено в файл "voronoi\_with\_centroids.png". Всі точки датасету були відображені чорними точками з низькою насиченістю (10%) для зменшення їх візуального впливу.

**Використані бібліотеки та методи:**

1. **sklearn.cluster.DBSCAN**:
   * DBSCAN(): для кластеризації точок на основі щільності.
2. **numpy**:
   * np.mean(): для обчислення середнього арифметичного (центру ваги).
3. **scipy.spatial.Voronoi**:
   * Voronoi(): для побудови діаграми Вороного.
   * voronoi\_plot\_2d(): для відображення діаграми Вороного.
4. **matplotlib.pyplot**:
   * plt.scatter(): для побудови точок на графіку.
   * plt.xlim() і plt.ylim(): для налаштування меж осей.
   * plt.savefig(): для збереження графіків у файл.

**Результат виконання:**

* Програма успішно зчитала дані з файлу, кластеризувала точки, обчислила центри ваги для кожного кластеру та побудувала діаграму Вороного для цих центрів.
* Результати були відображені на графіку, де:
  + Центри ваги позначені червоними колами.
  + Точки датасету відображені чорними точками з насиченістю 10%.
  + Діаграма Вороного показує розподіл простору на основі центрів ваги.
* Файл з результатами зберігається у графічному форматі PNG ("voronoi\_with\_centroids.png").

**Висновок:**

Завдання успішно виконано, і результат був збережений у графічному файлі. Програма продемонструвала ефективне використання методів кластеризації та візуалізації для аналізу зв'язаних областей та побудови діаграми Вороного.

