Информатика. Кластеризация. DBSCAN.

Содержание

1 Разбор задач с вебинара

1.1 Пример 1

Учёный решил провести кластеризацию некоторого множества звёзд по их расположению на карте звёздного неба. Кластер звёзд – это набор звёзд (точек) на графике, каждая из которых находится от хотя бы одной другой звезды на расстоянии не более R условных единиц. Каждая звезда обязательно принадлежит только одному из кластеров. Истинный центр кластера, или центроид, – это одна из звёзд на графике, сумма расстояний от которой до всех остальных звёзд кластера минимальна. Под расстоянием понимается расстояние Евклида между двумя точками $A(x1, y1)$ и $B(x2, y2)$ на плоскости, которое вычисляется по формуле:

$$d(A, B) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

Аномалиями назовём точки, находящиеся на расстоянии более одной условной единицы от точек кластеров. При расчётах аномалии учитывать не нужно.

В файле А хранятся данные о звёздах двух кластеров, где $R = 0,5$ для каждого кластера. В каждой строке записана информация о расположении на карте одной звезды: сначала координата x, затем координата y. Значения даны в условных единицах, которые представлены вещественными числами. Известно, что количество звёзд не превышает 2500.

В файле Б хранятся данные о звёздах четырех кластеров, где $R = 0,2$ для каждого кластера. Известно, что количество звёзд не превышает 10 000. Структура хранения информации о звездах в файле Б аналогична файлу А.

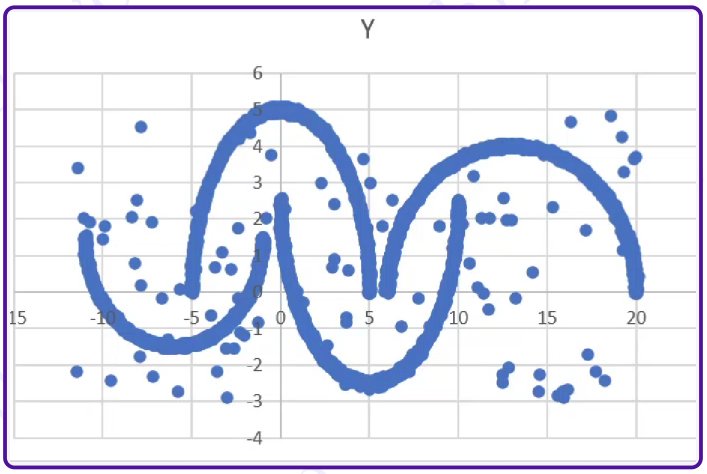
Для каждого файла определите координаты центра каждого кластера, затем вычислите два числа: P_x — среднее арифметическое абсцисс центров кластеров, и P_y – среднее арифметическое ординат центров кластеров.

В ответе запишите четыре числа через пробел: сначала целую часть произведения $P_x \cdot 100$ для файла А, затем $P_y \cdot 100$ для файла А, далее целую часть $P_x \cdot 100$ для файла Б и $P_y \cdot 100$ для файла Б.

Решение (файл Б):

Для начала визуально оценим данные в условии кластеры. Для этого откроем предложенные файлы в *Excel*, перейдем в раздел **«Вставка → Диаграммы → Точечная»**.

Диаграмма для файла Б имеет вид:




```python
from math import dist # Функция для вычисления расстояния Евклида

# Функция для кластеризации
def dbscan(a, r):
    clusters = [] # Создаем массив для кластеров
    while a: # Пока в списке 'a' есть звезды
        clusters.append([a.pop(0)]) # Создаем кластер, добавляем 1ую звезду
        for i in clusters[-1]: # Для каждой звезды в последнем кластере
            for j in a[:]: # Для каждой звезды в списке 'a'
                if dist(i[:2], j[:2]) < r: # Если рядом есть звезда
                    clusters[-1].append(j) # Добавляем звезду в кластер
                    a.remove(j) # Удаляем ее из исходного списка
    return clusters # Возвращаем список кластеров
```

```
f = open('1.txt') # Открываем файл
s = f.readline() # Считываем первую строку с названием столбцов
a = [list(map(float, i.replace(',', '.').split())) for i in f]
r = 0.2 # Радиус кластеризации
c = dbscan(a, r) # Кластеризуем звезды
c_res = [] # Кластеры без аномалий
for i in c: # Проход по кластерам
    if len(i) > 10: # Если элементов больше, чем 10 (аномалии отбрасываем)
        c_res. append(i) # Оставляем звезду в кластере
cx = cy = 0 # Суммы координат центров кластеров
centers = [] # Список координат центров кластеров
for i in c_res: # Проход по кластерам без аномалий
    ms = 1000050000 # Минимальное расстояние
    for j in i: # Перебираем каждую звезду как центр
        star = j
        s = 0 # Суммы расстояний между текущей и другими звездами
        for k in i: # Проходим по всем звездам в кластере
            s += dist(star, k) # Суммируем расстояния до всех других звёзд
        if s < ms: # Если эта сумма меньше минимальной
            ms = s
            star_center = star # Обновляем лучший центр
    cx += star_center[0] # Сумма абсцисс
    cy += star_center[1] # Сумма ординат
    centers.append(star_center) # Записываем центр кластера
print(int(cx/4 * 100), int(cy/4 * 100))  # Ответ
for i in range(4): # Ищем радиус каждого кластера
    mx = 0 # Максимальное расстояние от центра до звезды
    for j in c_res[i]:
        if dist(j, centers[i]) > mx: mx = dist(j, centers[i])
    print(mx)
```

Ответ: 315 127

1.2 Пример 2

Учёный решил провести кластеризацию некоторого множества звёзд по их расположению на карте звёздного неба. Кластер звёзд – это набор звёзд (точек) на графике, образующий звезду на небесном полотне. Каждая звезда обязательно принадлежит только одному из кластеров.

Также на звёздном небе присутствует пучок звёзд, образующий синусоиду. Кластер звёзд может считаться принадлежащим синусоиде, если график синусоиды проходит сквозь кластер

Истинный центр кластера, или **центроид**, – это одна из звёзд на графике, сумма расстояний от которой до всех остальных звёзд кластера минимальна.

Под расстоянием понимается расстояние Евклида между двумя точками $A(x_1, y_1)$ и $B(x_2, y_2)$ на плоскости, которое вычисляется по формуле:

$$d(A, B) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

В файле А хранятся данные о звёздах **четырёх** кластеров. В каждой строке записана информация о расположении на карте одной звезды: сначала координата x, затем координата y. Значения даны в условных единицах, которые представлены вещественными числами. Известно, что количество звёзд не превышает 5500.

В файле Б хранятся данные о звёздах **восьми** кластеров. Известно, что количество звёзд не превышает 35000. Структура хранения информации о звёздах в файле Б аналогична файлу А.

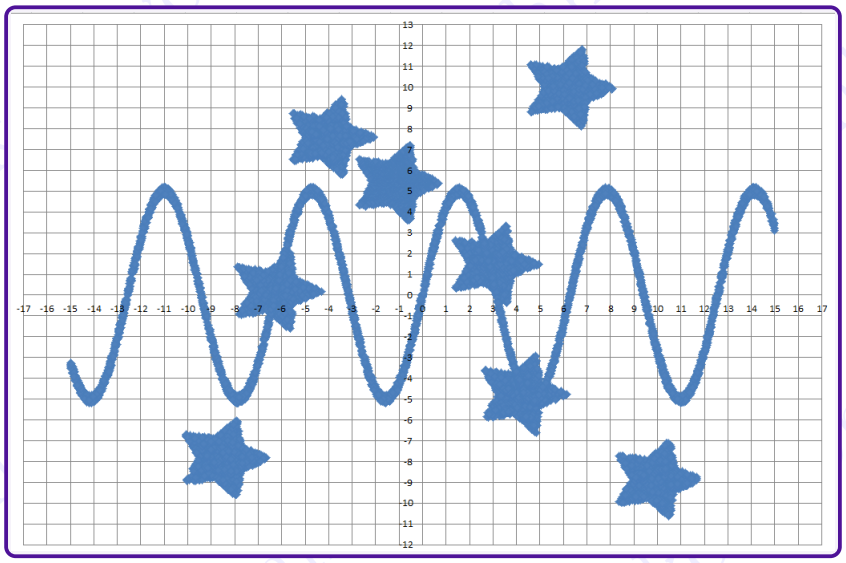
Для каждого файла определите координаты центра каждого кластера, затем вычислите два числа: P_x – среднее арифметическое абсцисс центров кластеров, и P_y – среднее арифметическое ординат центров кластеров. Кластеры, принадлежащие синусоиде, в вычислениях не учитывать.

В ответе запишите четыре числа через пробел: сначала целую часть произведений $P_x \cdot 100$ и $P_y \cdot 100$ для файла А, далее целую часть произведения $P_x \cdot 1000$ и $P_y \cdot 1000$ для файла Б.

Решение (файл Б):

Для начала визуально оценим данные в условии кластеры. Для этого откроем предложенные файлы в *Excel*, перейдем в раздел **«Вставка → Диаграммы → Точечная»**.

Диаграмма для файла Б имеет вид:



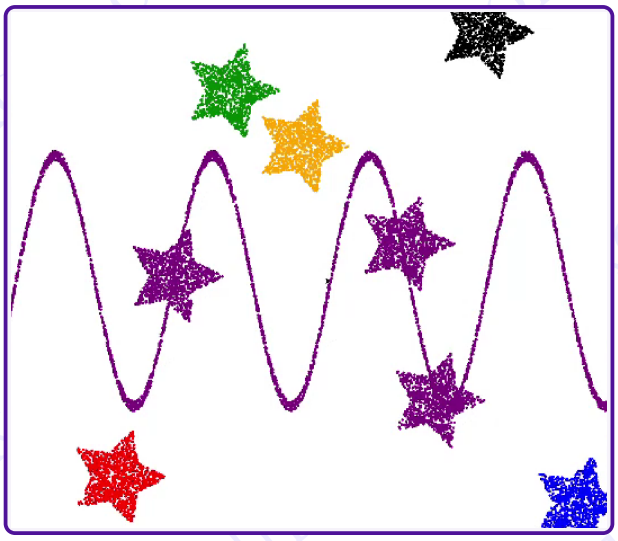
```python
from math import dist # Функция для вычисления расстояния Евклида

# Функция для кластеризации
def dbscan(a, r):
    clusters = [] # Создаем массив для кластеров
    while a: # Пока в списке 'a' есть звезды
        clusters.append([a.pop(0)]) # Создаем кластер, добавляем 1ую звезду
        for i in clusters[-1]: # Для каждой звезды в последнем кластере
            for j in a[:]: # Для каждой звезды в списке 'a'
                if dist(i[:2], j[:2]) < r: # Если рядом есть звезда
                    clusters[-1].append(j) # Добавляем звезду в кластер
                    a.remove(j) # Удаляем ее из исходного списка
    return clusters # Возвращаем список кластеров
```

Изобразим кластеры с помощью модуля turtle.

```
from turtle import *
colors = ['red', 'black', 'purple', 'green', 'blue', 'orange', 'pink']
pu() # Поднимаем хвост черепахи
tracer(0) # Отключаем анимацию
m = 50 # Масштаб
for i in range(len(c_res)): # Проходим по каждому кластеру
    if len(c_res[i]) > 10: # Если кластер не аномальный
        for j in c_res[i]: # Проходим по каждой звезде в кластере
            x,y = j  # Получаем координаты звезды
            goto(x*m, y*m) # Перемещаем черепаху
            dot(5, colors[i]) # Рисуем точку
done() # Завершаем рисование
```

Получили изображение:



Фиолетовый цвет, цвет синусоиды, в списке находится под индексом 2, следовательно пучок звёзд, образующий синусоиду, в списке кластеров имеет индекс 2.

Найдем центроиды каждого кластера и средние арифметические их координат:

```
cx = cy = 0 # Суммы координат центров кластеров
for i in range (len(c_res)): # Для каждого кластера
    if i != 2: # Если кластер не синусоида
        ms = 1000050000 # Минимальное расстояние
        for j in c_res[i]: # Перебираем звезды внутри кластера
            star = j # Предполагаемый центр
            s = 0 # Суммы расстояний между текущей и другими звездами
            for k in c_res[i]: # Проходим по всем звездам в кластере
                s += dist(star, k) # Суммируем расстояния
            if s < ms: # Если эта сумма меньше минимальной
                ms = s
                star_center = star # Обновляем лучший центр
        cx += star_center[0] # Сумма абсцисс
        cy += star_center[1] # Сумма ординат

print(int(cx/(len(c_res)-1)*1000), int(cy/(len(c_res)-1)*1000))
```

Ответ: 532 1177

1.3 Пример 3

Учёный решил провести кластеризацию некоторого множества звёзд по их расположению на карте звёздного неба. Кластер звёзд – это набор звёзд (точек) на графике, образующий цветы на небесном полотне. Каждая звезда обязательно принадлежит только одному из кластеров.

Истинная периферия кластера, или перифероид, – это одна из звёзд на графике, сумма расстояний от которой до всех остальных звёзд кластера максимальна.

Под расстоянием понимается расстояние Евклида между двумя точками $A(x_1, y_1)$ и $B(x_2, y_2)$ на плоскости, которое вычисляется по формуле:

$$d(A, B) = \sqrt{(x_2 - x_1)^2 + (y_2 - y_1)^2}$$

В файле А хранятся данные о звёздах **трёх** цветков. В каждой строке записана информация о расположении на карте одной звезды: сначала координата x, затем координата y. Значения даны в условных единицах, которые представлены вещественными числами. Известно, что количество звёзд не превышает 10000.

В файле Б хранятся данные о звёздах **пяти** цветков. Известно, что количество звёзд не превышает 25000. Структура хранения информации о звёздах в файле Б аналогична файлу А.

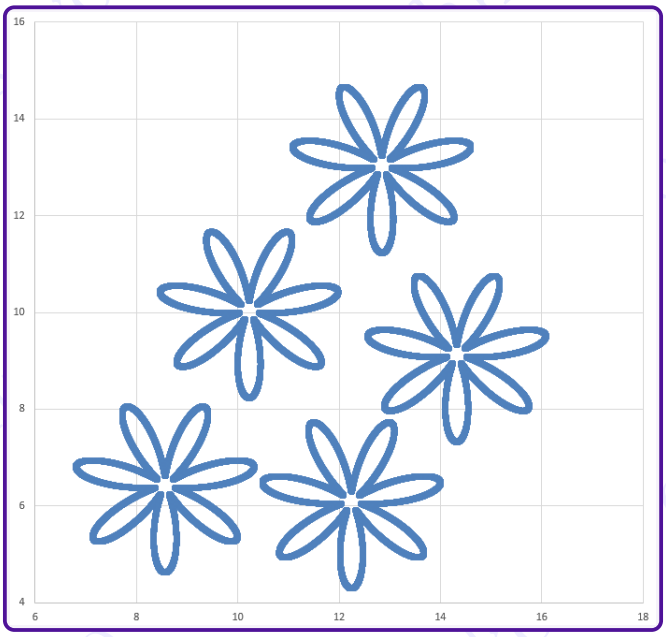
Для каждого файла определите координаты периферии каждого кластера, затем вычислите два числа: P_x — среднее арифметическое абсцисс периферий кластеров, и P_y – среднее арифметическое ординат периферий кластеров.

В ответе запишите четыре числа через пробел: сначала целую часть произведений $P_x \cdot 100$ и $P_y \cdot 100$ для файла А, далее целую часть произведения $P_x \cdot 100$ и $P_y \cdot 100$ для файла Б.

Решение (файл Б):

Для начала визуально оценим данные в условии кластеры. Для этого откроем предложенные файлы в *Excel*, перейдем в раздел **«Вставка → Диаграммы → Точечная»**.

Диаграмма для файла Б имеет вид:




```python
from math import dist # Функция для вычисления расстояния Евклида

# Функция для кластеризации
def dbscan(a, r):
    clusters = [] # Создаем массив для кластеров
    while a: # Пока в списке 'a' есть звезды
        clusters.append([a.pop(0)]) # Создаем кластер, добавляем 1ую звезду
        for i in clusters[-1]: # Для каждой звезды в последнем кластере
            for j in a[:]: # Для каждой звезды в списке 'a'
                if dist(i[:2], j[:2]) < r: # Если рядом есть звезда
                    clusters[-1].append(j) # Добавляем звезду в кластер
                    a.remove(j) # Удаляем ее из исходного списка
    return clusters # Возвращаем список кластеров
```

```python
f = open('3.txt') # Открываем файл
s = f.readline() # Считываем первую строку с названием столбцов
# Записываем координаты звезд в список 'a'
a = [list(map(float, i.replace(',', '.').split())) for i in f]
r = 0.2 # Радиус кластеризации
c = dbscan(a, r) # Кластеризуем звезды
c_res = [] # Кластеры без аномалий
for i in c: # Проход по кластерам
    if len(i) > 10: # Если элементов больше, чем 10 (аномалии отбрасываем)
        c_res. append(i) # Оставляем звезду в кластере

cx = cy = 0 # Суммы координат центров кластеров
for i in range (len(c_res)): # Для каждого кластера
    ms = 0 # Максимальное расстояние
    for j in c_res[i]: # Перебираем звезды внутри кластера
        star = j # Предполагаемый центр
        s = 0 # Суммы расстояний между текущей и другими звездами
        for k in c_res[i]: # Проходим по всем звездам в кластере
            s += dist(star, k) # Суммируем расстояния до всех других звёзд
        if s > ms: # Если эта сумма больше максимальной
            ms = s
            star_center = star # Обновляем лучший центр
    cx += star_center[0] # Сумма абсцисс
    cy += star_center[1] # Сумма ординат

print(int(cx/5*100), int(cy/5*100))
```

Ответ: 1179 946