Федеральное государственное автономное образовательное учреждение высшего образования

«Пермский государственный национальный исследовательский университет»

Институт Компьютерных Наук и Технологий

Лабораторная работа № 8

по дисциплине

«Введение в анализ данных»

Отчет

Студент Панькова Светлана

Группа ИТ-13-2023

Пермь 2025

Задание

В этом задании дан датасет с синтетическими (специально сгенерированными) данными.

Ученый решил провести кластеризацию некоторого множества звёзд по их расположению на карте звёздного неба. Кластер звезд – это набор звёзд (точек) на карте, лежащий внутри круга радиусом R. Каждая звезда принадлежит ровно одному кластеру. Центр кластера, или центроид, – это одна из звёзд на карте, сумма расстояний от которой до всех остальных звёзд кластера минимальна. Под расстоянием понимается расстояние Евклида.

В файле хранятся данные о звёздах трёх кластеров, R=3 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.

Определите координаты центра (центроида) каждого кластера.

```
import numpy as np
import pandas as pd
from sklearn.cluster import KMeans
import matplotlib.pyplot as plt
from scipy.spatial.distance import cdist

# Загружаем данные
df = pd.read_csv("27_B_17834.csv", sep=';')

# Преобразуем строки с координатами X и Y из
строкового формата с запятой в вещественные числа
df['X'] = df['X'].str.replace(",",
".").astype(float)
df['Y'] = df['Y'].str.replace(",",
".").astype(float)

# Кластеризация KMeans на 3 кластера
model = KMeans(n_clusters=3, random_state=0)
df['Cluster'] = model.fit_predict(df[['X', 'Y']])
```

```
# Находим реальные центроиды по определению задачи
real_centroids = []

for cluster in sorted(df['Cluster'].unique()):
    # Берём все точки кластера
    cluster_points = df[df['Cluster'] ==
cluster][['X', 'Y']].values

    # Считаем попарные расстояния внутри кластера
    distances = cdist(cluster_points, cluster_points,
'euclidean')

    # Суммируем расстояния от каждой точки до
остальных
    sums = distances.sum(axis=1)

    # Точка с минимальной суммой расстояний —
реальный центроид
    centroid = cluster_points[np.argmin(sums)]
    real_centroids.append(centroid)

real_centroids = np.array(real_centroids)

plt.figure(figsize=(6, 6))
plt.scatter(df['X'], df['Y'], c=df['Cluster'],
alpha=0.3, marker='.')
plt.scatter(real_centroids[:, 0], real_centroids[:,
1], c='r', s=150, marker='*', label='Центроиды')
plt.title('Центроиды кластеров на данных')
plt.xlabel('X')
plt.ylabel('Y')
plt.legend()
plt.grid(True)
plt.show()

for i, c in enumerate(real_centroids):
```

```
    print(f"Кластер {i + 1}: центроид ({c[0]:.3f},
{c[1]:.3f})")
```