Министерство образования Республики Беларусь

Учреждение образования

«Брестский Государственный технический университет»

Кафедра ИИТ

Лабораторная работа №4

По дисциплине «МРЗИС»

Тема: “Кластеризация”

Выполнил:

Студент 3 курса

Группы ИИ-21

Романко Н.А.

Проверил:

Туз И.С.

Брест 2023

Цель:реализовать алгоритм кластеризации данных.

Вариант: DBSCAN

Код программы:

from sklearn.datasets import make\_blobs

from sklearn.preprocessing import StandardScaler

import matplotlib.pyplot as plt

import numpy

def DBSCAN(Data, eps, MinPts):

    labels = [0]\*len(Data)  #устанавливаем все ID на 0

    Curr\_ID = 0

    #выбор начальной точки для нового кластера

    for point in range(0, len(Data)):

        if not (labels[point] == 0):

           continue

        NeighborPoints = regionQuery(Data, point, eps)

        #если у точки меньше соседей чем MinPts, то это шум, иначе создаем кластер

        if len(NeighborPoints) < MinPts:

            labels[point] = -1

        else:

           Curr\_ID += 1

           growCluster(Data, labels, point, NeighborPoints, Curr\_ID, eps, MinPts)

    return labels

def growCluster(Data, labels, point, NeighborPoints, Cluster\_ID, eps, MinPts):

    #даем ID начальной точке

    labels[point] = Cluster\_ID

    i = 0

    while i < len(NeighborPoints):

        new\_point = NeighborPoints[i]

        if labels[new\_point] == -1:

           labels[new\_point] = Cluster\_ID

        elif labels[new\_point] == 0:

            labels[new\_point] = Cluster\_ID

            # находим всех соседей Pn

            PnNeighborPoints = regionQuery(Data, new\_point, eps)

            #если точка имеет больше MinPts соседей, добавяем в массив поиска

            if len(PnNeighborPoints) >= MinPts:

                NeighborPoints = NeighborPoints + PnNeighborPoints

        i += 1

def regionQuery(Data, point, eps):

    neighbors = []

    for Pn in range(0, len(Data)):

        if numpy.linalg.norm(Data[point] - Data[Pn]) < eps:

           neighbors.append(Pn)

    return neighbors

centers = [[1, 0], [-2, -1], [0, -2]]

X, labels\_true = make\_blobs(n\_samples=700, centers=centers, cluster\_std=0.35, random\_state=1)

X = StandardScaler().fit\_transform(X)

labels =  DBSCAN(X, eps=0.35, MinPts=10)

plt.scatter(X[:, 0], X[:, 1], c = labels)

plt.show()

Вывод программы:

Изображение выглядит как снимок экрана, текст

Автоматически созданное описание

Вывод: научился реализовывать алгоритм кластеризации DBSCAN и применил его на практике.